Fitting Description Logic Ontologies to ABox and Query Examples

Maurice Funk, **Marvin Grosser**, **Carsten Lutz**
Leipzig University
ScaDS.AI Center Dresden/Leipzig
{mfunk, grosser, clu}@informatik.uni-leipzig.com

Abstract

We study a fitting problem inspired by ontology-mediated querying: given a collection of positive and negative examples of the form $(\mathcal{A}, q)$ with $\mathcal{A}$ an ABox and q a Boolean query, we seek an ontology $\mathcal{O}$ that satisfies $\mathcal{A} \cup \mathcal{O} \models q$ for all positive examples and $\mathcal{A} \cup \mathcal{O} \not\models q$ for all negative examples. We consider the description logics $\mathcal{ALC}$ and $\mathcal{ALCI}$ as ontology languages and a range of query languages that includes atomic queries (AQs), conjunctive queries (CQs), and unions thereof (UCQs). For all of the resulting fitting problems, we provide effective characterizations and determine the computational complexity of deciding whether a fitting ontology exists. This problem turns out to be coNP-complete for AQs and full CQs and 2ExpTime-complete for CQs and UCQs. These results hold for both $\mathcal{ALC}$ and $\mathcal{ALCI}$.

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 (Jacindha, Abishek, and Vasuki 2022). In machine learning, fitting a model to a given set of examples is closely linked to PAC-style generalization guarantees (Shalev-Shwartz and Ben-David 2014). And in database research, the traditional query-by-example paradigm asks to find a query that fits a given set of data examples (Li, Chan, and Maier 2015).

In this article, 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 (Bienvenu and Ortiz 2015; Xiao et al. 2018). Guided by this application, we use 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 and $\mathcal{A} \cup \mathcal{O} \not\models q$ for all negative examples. It is not a restriction that q is required to be Boolean since our queries may contain individuals from the ABox.

A main application of this ontology fitting problem 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 (Baader et al. 2007; Baader and Distel 2009; Kriegel 2024) and Angluin's framework of exact learning (Konev et al. 2017), see also the survey (Ozaki 2020). 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, see for instance (Lehmann and Hitzler 2010; Bühmann et al. 2018; Funk et al. 2019; Jung et al. 2021). 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. While it is probably unrealistic to assume that an ontology for an entire domain can be built in a single step from a given set of examples, we believe that small portions of the ontology can be constructed this way, thereby supporting a step-by-step development process by a human engineer. Moreover, in ontology-mediated querying there are applications where a more pragmatic view of an ontology seems appropriate: instead of providing a careful and detailed domain representation, one only wants the ontology to support more complete answers for some given query or a small set of queries (Calvanese et al. 2006; Kharlamov et al. 2017; Sequeda et al. 2019). In such a case, an ontology of rather small size may suffice and deriving it from a collection of examples seems natural, close in spirit to query-by-example.

As ontology languages, we concentrate on the expressive yet fundamental DLs $\mathcal{ALC}$ and $\mathcal{ALCI}$, and as query languages we consider atomic queries (AQs), conjunctive queries (CQs), full CQs (CQs without quantified variables), and unions of conjunctive queries (UCQs). In addition, we study a fitting problem in which the examples only consist of an ABox and where we seek an ontology that is consistent with the positive examples and inconsistent with the negative ones; this is related, but not identical to both AQ-based and full CQ-based fitting. For all of the resulting combinations, we provide effective characterizations and determine the precise complexity of deciding whether a fitting ontology exists. The algorithms that we use to prove the upper bounds are able to produce explicit fitting ontologies.

For consistency-based fitting and for AQs, our characterizations of fitting existence make use of the connection between ontology-mediated querying and constraint satisfaction problems (CSPs) established in (Bienvenu et al. 2014). While this connection does not extend to full CQs, the intuitions do and in all three cases our characterizations enable a CONP upper bound, both for $\mathcal{ALC}$- and $\mathcal{ALCI}$-ontologies. Corresponding lower bounds are easy to obtain by a reduction from the digraph homomorphism problem. We remark that the complexity is thus much lower than that of the associated query entailment problems, meaning to decide whether $\mathcal{A} \cup \mathcal{O} \models q$ for a given ABox $\mathcal{A}$, ontology $\mathcal{O}$, and query q. In fact, the complexity of query entailment is EXPTIME-complete for all cases discussed so far (Baader et al. 2017).

For CQs and UCQs, we give a characterization of fitting existence based on the existence of certain forest models $\mathcal{I}$. These models are potentially infinite, intuitively because the positive examples $(\mathcal{A}, q)$ act similarly to an existential rule: if we homomorphically find $\mathcal{A}$ in $\mathcal{I}$, then at the same place we must (in a certain, slightly unusual sense) also find q. Thus the existential quantifiers of q may enforce that every element of $\mathcal{I}$ has a successor, resulting in infinity. As a consequence of this effect, the computational complexity of fitting existence for CQs and UCQs turns out to be much higher than for AQs and full CQs: it is 2EXPTIME-complete both for CQs and UCQs, no matter whether we want to fit an $\mathcal{ALC}$- or an $\mathcal{ALCI}$-ontology. For $\mathcal{ALCI}$, the complexity thus coincides with that of query entailment, which is 2EXPTIME-complete both for CQs and UCQs (Lutz 2008). For $\mathcal{ALC}$, the complexity of the fitting problem is harder than that of the associated entailment problems, which are both EXPTIME-complete (Lutz 2008). Our upper bounds are obtained by a mosaic procedure. The lower bounds for $\mathcal{ALCI}$ are proved by reduction from query entailment and for $\mathcal{ALC}$ they are proved by reduction from the word problem of exponentially space-bounded alternating Turing machines.

Proofs are provided in the appendix of (Funk, Grosser, and Lutz 2025).

Related Work. To the best of our knowledge, the only other study of fitting problems for ontologies is a recent one by (Jung, Hosemann, and Lutz 2025). However, it uses interpretations as examples rather than ABox and queries. Vaguely related are fitting problems for DL concepts. These have been investigated from a practical angle by (Lehmann and Hitzler 2010; Bühmann et al. 2018; Rizzo, Fanizzi, and d'Amato 2020), and from a foundational perspective by (Funk et al. 2019; Jung et al. 2020; Jung et al. 2021; Jung et al. 2022). Other approaches that support the construction of an entire ontology include Angluin's framework of exact learning by (Konev et al. 2017) and formal concept analysis by (Baader et al. 2007; Baader and Distel 2009; Kriegel 2024). These and related approaches are surveyed by (Ozaki 2020).

2 Preliminaries

2.1 Description Logic

Let $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ be countably infinite sets of *concept names*, *role names*, and *individual names*. An *inverse role* takes the form r^- with r a role name, and a *role* is a role name or an inverse role. If $r = s^-$ is an inverse role, then we set $r^- = s$. An *$\mathcal{ALCI}$-concept* C is built according to

$$C, D ::= \top \mid A \mid \neg C \mid C \sqcap D \mid \exists r.D$$

where A ranges over concept names and r over roles. As usual, we write $\bot$ as abbreviation for $\neg\top$, $C \sqcup D$ for $\neg(\neg C \sqcap \neg D)$, and $\forall r.C$ for $\neg\exists r.\neg C$. An *$\mathcal{ALC}$-concept* is an $\mathcal{ALCI}$-concept that does not use inverse roles.

An *$\mathcal{ALCI}$-ontology* is a finite set of *concept inclusions (CIs)* $C \sqsubseteq D$, where C, D are $\mathcal{ALCI}$-concepts. *$\mathcal{ALC}$-ontologies* are defined likewise. We may write $C \equiv D$ as shorthand for $C \sqsubseteq D$ and $D \sqsubseteq C$. An *ABox* is a finite set of *concept assertions* $A(a)$ and *role assertions* $r(a, b)$ where A is a concept name, r a role name, and a, b are individual names. We use $\mathsf{ind}(\mathcal{A})$ to denote the set of individual names used in $\mathcal{A}$.

The semantics of concepts is defined as usual in terms of interpretations $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ with $\Delta^\mathcal{I}$ the (non-empty) *domain* and $\cdot^\mathcal{I}$ the *interpretation function*, we refer to (Baader et al. 2017) for full details. An interpretation $\mathcal{I}$ *satisfies* a CI $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$, a concept assertion $A(a)$ if $a \in A^\mathcal{I}$, and a role assertion $r(a, b)$ if $(a, b) \in r^\mathcal{I}$; we thus make the *standard names assumption*. We say that $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$, written $\mathcal{I} \models \mathcal{O}$, if it satisfies all concept inclusions in it, and likewise for ABoxes. An ontology is *satisfiable* if it has a model and an ABox is *consistent* with an ontology $\mathcal{O}$ if $\mathcal{A}$ and $\mathcal{O}$ have a common model.

A *homomorphism* from an interpretation $\mathcal{I}_1$ to an interpretation $\mathcal{I}_2$ is a mapping $h \colon \Delta^{\mathcal{I}_1} \to \Delta^{\mathcal{I}_2}$ such that $d \in A^{\mathcal{I}_1}$ implies $h(d) \in A^{\mathcal{I}_2}$ and $(d, e) \in r^{\mathcal{I}_1}$ implies $(h(d), h(e)) \in r^{\mathcal{I}_2}$ for all concept names A, role names r, and $d, e \in \Delta^{\mathcal{I}_1}$. We write $\mathcal{I}_1 \to \mathcal{I}_2$ if there exists a homomorphism from $\mathcal{I}_1$ to $\mathcal{I}_2$ and $\mathcal{I}_1 \not\to \mathcal{I}_2$ otherwise. We will also use homomorphisms from ABoxes to ABoxes and from ABoxes to interpretations. These are defined as expected. In particular, homomorphisms from ABox to ABox need not map individual names to themselves, which would trivialize them.

2.2 Queries

A *conjunctive query (CQ)* takes the form $q = \exists \overline{x}\, \varphi(\overline{x})$ where $\overline{x}$ is a tuple of variables and φ a conjunction of *atoms* $A(t)$ and $r(t, t')$, with $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and t, t' variables from $\overline{x}$ or individuals from $\mathsf{N_I}$. With $\mathsf{var}(q)$, we denote the set of variables in $\overline{x}$. We take the liberty to view q as a set of atoms, writing e.g. $\alpha \in q$ to indicate that α is an atom in q. We may also write $r^-(x, y) \in q$ in place of $r(y, x) \in q$. An *atomic query (AQ)* is a CQ of the simple form $A(a)$, with A a concept name. A CQ is *full* if it does not contain any existentially quantified variables. A *union of conjunctive queries (UCQ)* q is a disjunction of CQs. We refer to each of these classes of queries as a *query language*.

A CQ q gives rise to an interpretation $\mathcal{I}_q$ with $\Delta^{\mathcal{I}_q}$ the set of all variables and individuals in q, $A^{\mathcal{I}_q} = \{t \mid A(t) \in q\}$, and $r^{\mathcal{I}_q} = \{(t, t') \mid r(t, t') \in q\}$ for all $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. With a homomorphism from a CQ q to an interpretation $\mathcal{I}$, we mean a homomorphism from $\mathcal{I}_q$ to $\mathcal{I}$ that is the identity on all individual names. If we want to emphasize the latter property, we may speak of a *strong* homomorphism. In contrast, a *weak* homomorphism from q to $\mathcal{I}$, as sometimes used in our proofs, need not be the identity on individual names. For an interpretation $\mathcal{I}$ and a UCQ q, we write $\mathcal{I} \models q$ if there is a (strong) homomorphism h from a CQ in q to $\mathcal{I}$. For an ABox $\mathcal{A}$ and ontology $\mathcal{O}$, we write $\mathcal{A} \cup \mathcal{O} \models q$ if $\mathcal{I} \models q$ for all models $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{O}$.

Note that all queries introduced above are Boolean, that is, they evaluate to true or false instead of producing answers. For the purposes of this paper, however, this is without loss of generality since we admit individual names in queries.

We use $||O||$ to denote the *size* of any syntactic object O such as a concept, an ontology, or a query. It is defined as the length of the encoding of O as a word over some suitable alphabet.

An $\mathcal{ALC}$*-forest model* $\mathcal{I}$ of an ABox $\mathcal{A}$ is a model of $\mathcal{A}$ such that

1. the directed graph $(\Delta^\mathcal{I}, \bigcup_r r^\mathcal{I} \setminus (\mathsf{ind}(\mathcal{A}) \times \mathsf{ind}(\mathcal{A}))$ is a forest (a disjoint union of trees) and
2. $r^\mathcal{I} \cap (\mathsf{ind}(\mathcal{A}) \times \mathsf{ind}(\mathcal{A})) = \{(a, b) \mid r(a, b) \in \mathcal{A}\}$.

$\mathcal{ALCI}$*-forest models* are defined likewise, but based on the undirected version of the graph in Point 1. In other words, in $\mathcal{ALC}$-forest models all edges must point away from the roots of the trees while this is not the case for $\mathcal{ALCI}$-forest models. With the *degree* of an interpretation, we mean the maximal number of neighbors of any element in its domain.

Lemma 1. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$, $\mathcal{O}$ be an $\mathcal{L}$-ontology, $\mathcal{A}$ an ABox, and q a UCQ. If $\mathcal{A} \cup \mathcal{O} \not\models q$, then there is an $\mathcal{L}$-forest model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{O}$ of degree at most $||\mathcal{O}||$ that satisfies $\mathcal{I} \not\models q$.*

The proof of Lemma 1, which can be found for instance in (Lutz 2008), relies on unraveling, which we shall also use in this article. Let $\mathcal{I}$ be an interpretation and $d \in \Delta^\mathcal{I}$. A *path* in $\mathcal{I}$ is a sequence $p = d_1 r_1 \cdots d_{n-1} r_{n-1} d_n$ of domain elements d_i from $\Delta^\mathcal{I}$ and role names r_i such that $(d_i, d_{i+1}) \in r_i^\mathcal{I}$ for $1 \leq i < n$. We say that the path *starts* at d_1 and use $\mathsf{tail}(p)$ to denote d_n.

The $\mathcal{ALC}$*-unraveling* of $\mathcal{I}$ at d is the interpretation $\mathcal{U}$ defined as follows:

$$\begin{aligned} \Delta^\mathcal{U} &= \text{set of all paths in } \mathcal{I} \text{ starting at } d \\ A^\mathcal{U} &= \{p \mid \mathsf{tail}(p) \in A^\mathcal{I}\} \\ r^\mathcal{U} &= \{(p, p') \mid p' = pre \text{ for some } e\}. \end{aligned}$$

The $\mathcal{ALCI}$-unraveling of $\mathcal{I}$ at d is defined likewise, with the modification that inverses of roles can also appear in paths and that (p, p') is also included in $r^\mathcal{U}$ if $p = p'r^-e$. Note that there is a homomorphism from $\mathcal{U}$ to $\mathcal{I}$ that maps every $p \in \Delta^\mathcal{U}$ to $\mathsf{tail}(p)$.

2.3 ABox Examples and the Fitting Problem

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 $\mathcal{A}$ an ABox and q a query from $\mathcal{Q}$ such that all individual names that appear in q are from $\mathsf{ind}(\mathcal{A})$.

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 $\mathcal{O}$ *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 $(\mathcal{A}, q) \in E^-$. The following example illustrates this central notion.

Example 1. *Consider the collection of labeled ABox-UCQ examples $E = (E^+, E^-)$, where*

$$\begin{aligned} E^+ = \{ & (\{\mathsf{authorOf}(a, b), \mathsf{Publication}(b)\}, \mathsf{Author}(a)), \\ & (\{\mathsf{Reviewer}(a)\}, \exists x\, \mathsf{reviews}(a, x) \land \mathsf{Publication}(x)), \\ & (\{\mathsf{Publication}(a)\}, \mathsf{Confpaper}(a) \lor \mathsf{Jarticle}(a)) \}, \end{aligned}$$

and $E^- = \emptyset$. An $\mathcal{ALC}$-ontology that fits (E^+, E^-) is

$$\begin{aligned} \mathcal{O} = \{ & \exists \mathsf{authorOf}.\mathsf{Publication} \sqsubseteq \mathsf{Author} \\ & \mathsf{Reviewer} \sqsubseteq \exists \mathsf{reviews}.\mathsf{Publication} \\ & \mathsf{Publication} \sqsubseteq \mathsf{Confpaper} \sqcup \mathsf{Jarticle} \}. \end{aligned}$$

There are, however, many other fitting $\mathcal{ALC}$-ontologies as well, including as an extreme $\mathcal{O}_\perp = \{\top \sqsubseteq \perp\}$ 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

$$(\{\mathsf{Author}(a)\}, \exists x\, \mathsf{authorOf}(a, x) \land \mathsf{Reviewer}(x)).$$

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. We generally assume that the ABoxes used in E have pairwise disjoint sets of individual names. It is not hard to verify that this is without loss of generality because consistently renaming individual names in a collection of examples has no impact on the existence of a fitting ontology.

There is a natural variation of $(\mathcal{L}, \mathcal{Q})$-ontology fitting where one additionally requires the fitting ontology to be consistent with all ABoxes that occur in positive examples.[1] We then speak of *consistent* $(\mathcal{L}, \mathcal{Q})$*-ontology fitting*. The following observation shows that it suffices to design algorithms for $(\mathcal{L}, \mathcal{Q})$-ontology fitting as originally introduced.

Proposition 1. *Let $\mathcal{L}$ be any ontology language and $\mathcal{Q} \in \{AQ, FullCQ, CQ, UCQ\}$. Then there is a polynomial time reduction from consistent $(\mathcal{L}, \mathcal{Q})$-ontology fitting to $(\mathcal{L}, \mathcal{Q})$-ontology fitting.*

Proof. We exemplarily treat the case $\mathcal{Q} = \text{AQ}$. The other cases are similar. Let E be a collection of labeled ABox-AQ examples. We extend E to a collection E' by adding, for each positive example $(\mathcal{A}, Q(a)) \in E^+$, a negative example $(\mathcal{A}, X(a))$ where X is a concept name that is not mentioned in E. Then E admits a fitting $\mathcal{L}$-ontology that is consistent with all ABoxes in positive examples if and only if E' admits a fitting $\mathcal{L}$-ontology. In fact, any $\mathcal{L}$-ontology that fits E, does not mention X, and is consistent with all ABoxes in positive examples is also a fitting of E'. Conversely, any ontology that fits E' must be consistent with all ABoxes that occur in positive examples as otherwise one of the additional negative examples would be violated. ❑

[1]Note that it is implicit already in the original formulation that the fitting ontology must be consistent with all ABoxes that occur in negative examples $(\mathcal{A}, q)$, as otherwise $\mathcal{A} \cup \mathcal{O} \models q$.

3 Consistency-Based Fitting

We start with a version of ontology fitting that is based on ABox consistency rather than on querying. An example is then simply an ABox, and an ontology $\mathcal{O}$ *fits* a collection of examples $E = (E^+, E^-)$ if $\mathcal{A}$ is consistent with $\mathcal{O}$ for all $\mathcal{A} \in E^+$ and inconsistent with $\mathcal{O}$ for all $\mathcal{A} \in E^-$. We refer to the induced decision problem as *consistent $\mathcal{L}$-ontology fitting*.[2] We believe that it is natural to consider this basic case as a warm-up.

Example 2. *Consider the collection of labeled ABox examples $E = (E^+, E^-)$, where*

- *E^+ contains the ABox $\mathcal{A}_1 = \{r(a_1, a_2)\}$ and*
- *E^- contains the ABox $\mathcal{A}_2 = \{r(b, b)\}$.*

Then $\mathcal{O} = \{\exists r.\exists r.\top \sqsubseteq \bot\}$ is an $\mathcal{ALC}$-ontology that fits (E^+, E^-). If we swap E^+ and E^-, then there is no fitting $\mathcal{ALC}$-ontology or $\mathcal{ALCI}$-ontology.

We start with a characterization of consistent $\mathcal{L}$-ontology fitting in terms of homomorphisms.

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. *$\mathcal{A} \not\rightarrow \mathcal{A}^+$ for all $\mathcal{A} \in E^-$.*

Note that the characterizations for $\mathcal{ALC}$ and $\mathcal{ALCI}$ are identical, and thus a collection of labeled ABox-consistency examples admits a fitting $\mathcal{ALC}$-ontology if and only if it admits a fitting $\mathcal{ALCI}$-ontology. It is clear from the proofs that further adding role inclusions, see (Baader et al. 2017), does not increase the separating power either. Adding number restrictions, however, has an impact; see Section 7.

The proof of Theorem 1 makes use of the connection between ontology-mediated querying and constraint satisfaction problems (CSPs) established in (Lutz and Wolter 2012). In particular, for the "2 $\Rightarrow$ 1" direction we use the fact that for every ABox $\mathcal{A}$, one can construct an ontology $\mathcal{O}$ such that for all ABoxes $\mathcal{B}$ that only use concept and role names from $\mathcal{A}$, the following holds: $\mathcal{B} \rightarrow \mathcal{A}$ if and only if $\mathcal{B}$ is consistent with $\mathcal{O}$. We apply this choosing $\mathcal{A} = \mathcal{A}^+$.

We obtain an upper bound for consistent ontology fitting by a straightforward implementation of Point 2 of Theorem 1 and a corresponding lower bound by an easy reduction of the homomorphism problem for directed graphs.

Theorem 2. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. Then consistent $\mathcal{L}$-ontology fitting is* coNP*-complete.*

It might be worthwhile to point out as a corollary of Theorem 1 that negative examples can be treated independently, in the following sense.

Corollary 1. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$ and E be a collection of labeled ABox examples, with $E^- = \{\mathcal{A}_1, \dots, \mathcal{A}_n\}$. Then E admits a fitting $\mathcal{L}$-ontology if and only if for $1 \leq i \leq n$, the collection of ABox examples $(E^+, \{\mathcal{A}_i\})$ admits a fitting $\mathcal{L}$-ontology.*

[2] Not to be confused with consistent $(\mathcal{L}, \mathcal{Q})$-ontology fitting as briefly considered in Proposition 1.

We note that, in related fitting settings such as the one studied in (Funk et al. 2019), statements of this form can often be shown in a very direct way rather than via a characterization. This does not appear to be the case here.

4 Atomic Queries

We consider atomic queries and again present a characterization in terms of homomorphisms. These are now in the other direction, from the positive examples to the negative examples, corresponding to the complementation involved in the well-known reductions from ABox consistency to AQ entailment and vice versa. What is more important, however, is that it does no longer suffice to work directly with the (negative) examples. In fact, the positive examples act like a form of implication on the negative examples, similarly to an existential rule (with atomic unary rule head), and as a result we must first suitably enrich the negative examples.

Let $E = (E^+, E^-)$ be a collection of labeled ABox-AQ examples. A *completion* for E is an ABox $\mathcal{C}$ that extends the ABox $\mathcal{A}^- := \biguplus_{(\mathcal{A}, Q(a)) \in E^-} \mathcal{A}$ by concept assertions $Q(b)$ where $b \in \mathsf{ind}(\mathcal{A}^-)$ and Q a concept name that occurs as an AQ in E^+.

Theorem 3. *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 $(\mathcal{A}, Q(a)) \in E^-$: $Q(a) \notin \mathcal{C}$.*

The announced behavior of positive examples as an implication is reflected by Point 2a. Note that, as in the consistency case, there is no difference between $\mathcal{ALC}$ and $\mathcal{ALCI}$. The proof of Theorem 3 is similar to that of Theorem 1. It might be worthwhile to note that Theorem 3 does not suggest a counterpart of Corollary 1. Such a counterpart would speak about single positive examples rather than single negative ones, because of the complementation mentioned above. The following example illustrates that it does not suffice to concentrate on a single positive example (nor a single negative one). Intuitively, this is due to the fact that the ABox $\mathcal{A}$ in Point 2a may be disconnected.

Example 3. *Consider the collection of labeled ABox-AQ examples $E = (E^+, E^-)$ with*

$$E^+ = \{\ (\{A_2(a)\}, A_1(a)),\ (\{A_3(b), A_4(b')\}, A_2(b))\ \}$$
$$E^- = \{\ (\{A_3(c)\}, A_1(c)),\ (\{A_4(d)\}, A_5(d))\ \}.$$

E does not admit a fitting $\mathcal{ALCI}$-ontology, which can be seen by applying Theorem 3: by definition, any completion $\mathcal{C}$ must satisfy $\mathcal{A}^- = \{A_3(c), A_4(d)\} \subseteq \mathcal{C}$. To satisfy Condition 2a of Theorem 3, it must then also satisfy $A_2(c) \in \mathcal{C}$ and $A_1(c) \in \mathcal{C}$. But then $\mathcal{C}$ violates Condition 2b of Theorem 3 for the first negative example. If we drop any of the positive examples, we find completions $\mathcal{C} = \mathcal{A}^-$ and $\mathcal{C} = \mathcal{A}^- \cup \{B(c)\}$, respectively, which satisfy Conditions 2a and 2b. Also dropping any negative example leads to a satisfying completion.

In contrast to the case of consistent $\mathcal{L}$-ontology fitting, a naive implementation of the characterization given in Theorem 3 only gives a Σ_2^p-upper bound: guess the completion $\mathcal{C}$ for E and then co-guess the homomorphisms in Point (a). In the following, we show how to improve this to coNP. The main observation is that we can do better than guessing $\mathcal{C}$ blindly, by treating positive examples as rules.

Definition 1. *Let $E = (E^+, E^-)$ be a collection of labeled ABox-AQ examples, and let $\mathcal{A}^- = \biguplus_{(\mathcal{A},Q(a))\in E^-} \mathcal{A}$. A* refutation candidate *for E is an ABox $\mathcal{C}$ that can be obtained by starting with $\mathcal{A}^-$ and then applying the following rule zero or more times:*

(R) *if $(\mathcal{A}, Q(a)) \in E^+$ and h is a homomorphism from $\mathcal{A}$ to $\mathcal{C}$, then set $\mathcal{C} = \mathcal{C} \cup \{Q(h(a))\}$.*

Rule (R) can add at most $|E^+| \cdot |\mathsf{ind}(\mathcal{A}^-)|$ (and thus only polynomially many) assertions to $\mathcal{A}^-$.

Proposition 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 no fitting $\mathcal{L}$-ontology;*
2. *there is a refutation candidate $\mathcal{C}$ for E such that $Q(a) \in \mathcal{C}$ for some $(\mathcal{A}, Q(a)) \in E^-$.*

Note that Point 1 of Proposition 2 is the complement of Point 1 of Theorem 3, and thus $\mathcal{C}$ has a different role: we may co-guess it, in contrast to the $\mathcal{C}$ in Theorem 3 which needs to be guessed. A close look reveals that we indeed obtain a coNP upper bound.

Theorem 4. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. Then $(\mathcal{L}$,AQ$)$-ontology fitting is* coNP*-complete.*

Proof. coNP-hardness can be proved as in the consistency case. It thus remains to argue that the complement of the $(\mathcal{L}$, AQ$)$-ontology fitting problem is in NP. By Proposition 2, it suffices to guess an ABox $\mathcal{C}$ with the same individuals as $\mathcal{A}^-$ and to verify that (i) $\mathcal{C}$ is a refutation candidate and (ii) $Q(a) \in \mathcal{C}$ for some $(\mathcal{A}, Q(a)) \in E^-$. To verify (i) we may guess, along with $\mathcal{C}$, a sequence of positive examples $(\mathcal{A}, Q(a)) \in E^+$ with associated homomorphisms from $\mathcal{A}$ that demonstrate the construction of $\mathcal{C}$ from $\mathcal{A}^-$ by repeated applications of Rule (R). The maximum length of the sequence is $|E^+| \cdot |\mathsf{ind}(\mathcal{A}^-)|$. With the sequence at hand, it is then easy to verify deterministically in polynomial time that $\mathcal{C}$ is a refutation candidate. ❑

In view of the close connection between ABox consistency and AQ entailment, one may wonder whether the two fitting problems studied in this and the preceding section are, in some reasonable sense, identical. We may ask whether for every instance E of consistent $\mathcal{L}$-ontology fitting, there is an instance E' of $(\mathcal{L}, AQ)$-ontology fitting with the same set of fitting ontologies and vice versa. It turns out that neither is the case. For better readability, in the following we refer to ABox examples as ABox-consistency examples. For $\mathcal{L}$ an ontology language and E a collection of labeled examples, let $O_{E,\mathcal{L}}$ be the set of all $\mathcal{L}$-ontologies that fit E.

Proposition 3. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. Then*

1. *there exists a collection of ABox-AQ examples E, such that there is no collection of ABox-consistency examples E' with $O_{E,\mathcal{L}} = O_{E',\mathcal{L}}$;*
2. *there exists a collection of ABox-consistency examples, such that there is no collection of ABox-AQ examples E' with $O_{E,\mathcal{L}} = O_{E',\mathcal{L}}$.*

Proof. For Point 1, consider the collection of ABox-AQ examples $E = (E^+, E^-)$ with $E^+ = \{(\{A(a)\}, B_1(a))\}$ and $E^- = \{(\{A(a)\}, B_2(a))\}$. Note that we use $A(a)$ only to ensure that a occurs in the ABoxes. It is easy to see that for $\mathcal{O}_1 = \{\top \sqsubseteq B_1\}$ and $\mathcal{O}_2 = \{\top \sqsubseteq B_2\}$, $\mathcal{O}_1$ fits E and $\mathcal{O}_2$ does not. Furthermore, every ABox is consistent with both $\mathcal{O}_1$ and $\mathcal{O}_2$. Therefore, for every collection of ABox-consistency example E', either $\{\mathcal{O}_1, \mathcal{O}_2\} \subseteq O_{E',\mathcal{L}}$ or $\mathcal{O}_1 \notin O_{E',\mathcal{L}}$ and $\mathcal{O}_2 \notin O_{E',\mathcal{L}}$.

For Point 2, consider the collection of ABox-consistency examples $E = (E^+, \emptyset)$ with $E^+ = \{\{s(a, b)\}\}$ and let E' be any collection of ABox-AQ examples. If there is a negative example $(\mathcal{A}, A(a))$ in E' for some concept name A, then for $\mathcal{O} = \{\top \sqsubseteq A\}$, $\mathcal{O}$ fits E but $\mathcal{O}$ does not fit E'. If there is no negative example in E', then for $\mathcal{O}' = \{\top \sqsubseteq \bot\}$, $\mathcal{O}'$ fits E', but $\mathcal{O}'$ does not fit E. ❑

5 Full Conjunctive Queries

We next study the case of full conjunctive queries. Technically, it is closely related to both the AQ-based case and the ABox consistency-based case. However, the potential presence of role atoms in queries brings some technical complications.

We call an example $(\mathcal{A}, q)$ *inconsistent* if any $\mathcal{ALCI}$-ontology $\mathcal{O}$ that satisfies $\mathcal{A} \cup \mathcal{O} \models q$ is inconsistent with $\mathcal{A}$, and *consistent* otherwise. It is easy to see that any example $(\mathcal{A}, q)$ such that q contains a role atom $r(a, b) \notin \mathcal{A}$ must be inconsistent. In fact, this follows from Lemma 1. Conversely, any example $(\mathcal{A}, q)$ such that q does not contain such a role atom is consistent. This is witnessed by the ontology

$$\mathcal{O} = \{\top \sqsubseteq A \mid A(a) \in q\}.$$

Note that an inconsistent positive example $(\mathcal{A}, q)$ expresses the constraint that $\mathcal{A}$ must be inconsistent with the fitting ontology $\mathcal{O}$ and an inconsistent negative example $(\mathcal{A}, q)$ expresses that $\mathcal{A}$ must be consistent with the fitting ontology $\mathcal{O}$. In view of this, it is clear that, up to swapping positive and negative examples, FullCQ-based fitting generalizes consistency-based fitting. Moreover, it trivially generalizes AQ-based fitting since every AQ is a full CQ.

Proposition 4. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. Then for every collection of ABox-consistency or ABox-AQ examples E, there is a collection of ABox-FullCQ examples such that $O_{E,\mathcal{L}} = O_{E',\mathcal{L}}$.*

For the following development, we would ideally like to get rid of inconsistent examples to achieve simpler characterizations.

There is, however, no obvious way to achieve this for inconsistent positive examples. We can get rid of inconsistent negative examples based on the following observation. Assume that a collection of ABox-FullCQ examples E contains an inconsistent negative example $(\mathcal{A}, q)$. We replace it with the negative example $(\mathcal{A}, X(a))$ where X is a fresh concept name and $a \in \mathsf{ind}(\mathcal{A})$ is chosen arbitrarily. The set of fitting $\mathcal{ALCI}$-ontologies for the resulting set of examples E' remains essentially the same.

Lemma 2. *For $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$, there is an $\mathcal{L}$-ontology that fits E if and only if there is one that fits E'.*

Completions for collections of ABox-FullCQ examples are defined in exact analogy with completions for collections of ABox-AQ examples.

Theorem 5. *Let $E = (E^+, E^-)$ be a collection of labeled ABox-FullCQ 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 consistent $(\mathcal{A}, q) \in E^+$: if h is a homomorphism from $\mathcal{A}$ to $\mathcal{C}$ and $Q(a) \in q$, then $Q(h(a)) \in \mathcal{C}$;*
   (b) *for all $(\mathcal{A}, q) \in E^-$: there is a $Q(a) \in q$ such that $Q(a) \notin \mathcal{C}$;*
   (c) *for all inconsistent $(\mathcal{A}, q) \in E^+$: there is no homomorphism from $\mathcal{A}$ to $\mathcal{C}$.*

The proof of Theorem 5 uses Theorem 3. Note that, once more, there is no difference between $\mathcal{ALC}$ and $\mathcal{ALCI}$. As in the case of AQs, our characterization suggests only a Σ^p_2 upper bound. However, we can get down to coNP in the same way as for AQs. The following is in exact analogy with Definition 1.

Definition 2. *Let $E = (E^+, E^-)$ be a collection of labeled ABox-FullCQ examples, and let $\mathcal{A}^- = \biguplus_{(\mathcal{A},q)\in E^-} \mathcal{A}$. A* refutation candidate *for E is an ABox $\mathcal{C}$ that can be obtained by starting with $\mathcal{A}^-$ and then applying the following rule zero or more times:*

(R) *if $(\mathcal{A}, q) \in E^+$ is consistent and h is a homomorphism from $\mathcal{A}$ to $\mathcal{C}$ and $Q(a) \in q$, then set $\mathcal{C} = \mathcal{C} \cup \{Q(h(a))\}$.*

The proof of the following is then analogous to that of Proposition 2. Details are omitted.

Proposition 5. *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 no fitting $\mathcal{L}$-ontology;*
2. *there is a refutation candidate $\mathcal{C}$ for E such that one of the following conditions is satisfied:*
   (a) *there is an $(\mathcal{A}, q) \in E^-$ such that $Q(a) \in \mathcal{C}$ for all $Q(a) \in q$;*
   (b) *there is an inconsistent $(\mathcal{A}, q) \in E^+$ and a homomorphism from $\mathcal{A}$ to $\mathcal{C}$.*

And finally, the proof of the following is similar to that of Theorem 4.

Theorem 6. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. Then $(\mathcal{L}$, FullCQ$)$-ontology fitting is* coNP*-complete.*

6 CQs and UCQs

We now turn to conjunctive queries and UCQs, which constitute the most challenging case. This is due to the fact that, since positive examples act as implications, the presence of existentially quantified variables in the query effectively turns these examples into a form of existential rule. Thus, completions as used for AQs and full CQs are no longer finite.

Throughout this section, we assume that ABoxes in positive examples are never empty. This is mainly to avoid dealing with too many special cases in the technical development. We conjecture that admitting empty ABoxes does not change the obtained results.

6.1 Characterization for $\mathcal{ALC}$ and $\mathcal{ALCI}$

We start with a characterization for the case of UCQs (and thus also CQs) that is similar in spirit to the one for full CQs given in Theorem 5. The characterization applies to both $\mathcal{ALC}$ and $\mathcal{ALCI}$ in a uniform, though not identical way. As already mentioned, finite completions no longer suffice and we replace them with potentially infinite interpretations. There is another interesting view on this: the fitting ontologies constructed (as part of the proofs) in Sections 3 and 4 do not make existential statements, that is, their sets of models are closed under taking induced subinterpretations. This, however, cannot be achieved for CQs and UCQs. We illustrate this by the following example which also shows that, unlike for AQs and full CQs, there is a difference between fitting $\mathcal{ALC}$-ontologies and fitting $\mathcal{ALCI}$-ontologies. Induced subinterpretations are defined in exact analogy with induced substructures in model theory.

Example 4. *Consider the collection of ABox-CQ examples $E = (E^+, E^-)$ where*

$$\begin{aligned} E^+ &= \{(\{A_1(a)\}, \exists x\, r(x,a) \wedge A_2(x)), \\ &\qquad (\{A_2(a)\}, \exists x\, r(x,a) \wedge A_1(x))\ \} \\ E^- &= \{(\{A_1(a)\}, B(a)),\ (\{A_2(a)\}, B(a))\}. \end{aligned}$$

Then there is a fitting $\mathcal{ALCI}$-ontology:

$$\mathcal{O} = \{\ A_1 \sqsubseteq \exists r^-.A_2,\ A_2 \sqsubseteq \exists r^-.A_1\ \}.$$

The set of models of $\mathcal{O}$ is clearly not closed under taking induced subinterpretations. In fact, this is true for every $\mathcal{ALCI}$-ontology $\mathcal{O}'$ that fits E since any such $\mathcal{O}'$ must (i) logically imply $\mathcal{O}$ and (ii) be consistent with the ABoxes $\{A(a)\}$ and $\{B(a)\}$, due to the negative examples.

Moreover, it is easy to see that there is no $\mathcal{ALC}$-ontology $\mathcal{O}$ that fits E. This is due to Lemma 1 and the negative examples, ensuring that any such $\mathcal{O}$ would have to be consistent with the ABoxes in E^+.

We start with a preliminary. Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$, let $\mathcal{A}$ be an ABox, $\mathcal{I}$ an interpretation, and h a homomorphism from $\mathcal{A}$ to $\mathcal{I}$. We define an interpretation $\mathcal{I}_{\mathcal{A},h,\mathcal{L}}$ as follows. Start with interpretation $\mathcal{I}_0$:

$$\begin{aligned} \Delta^{\mathcal{I}_0} &= \mathsf{ind}(\mathcal{A}) \\ A^{\mathcal{I}_0} &= \{a \mid h(a) \in A^{\mathcal{I}}\} \quad \text{for all } A \in \mathsf{N_C} \\ r^{\mathcal{I}_0} &= \{(a,b) \mid r(a,b) \in \mathcal{A}\} \quad \text{for all } r \in \mathsf{N_R}. \end{aligned}$$

Then $\mathcal{I}_{\mathcal{A},h,\mathcal{L}}$ is obtained by taking, for every $a \in \mathsf{ind}(\mathcal{A})$, the $\mathcal{L}$-unraveling of $\mathcal{I}$ at $h(a)$ and disjointly adding it to $\mathcal{I}_0$, identifying the root with a. It can be shown that if $\mathcal{I}$ is a model of some $\mathcal{L}$-ontology $\mathcal{O}$, then $\mathcal{I}_{\mathcal{A},h,\mathcal{L}}$ is also a model of $\mathcal{O}$. Informally, we use $\mathcal{I}_{\mathcal{A},h,\mathcal{L}}$ to 'undo' the potential identification of individual names by h, in this way obtaining a forest model of $\mathcal{A}$.

Theorem 7. *Let $E = (E^+, E^-)$ be a collection of labeled ABox-UCQ examples with $E^- \neq \emptyset$ and let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. Then the following are equivalent:*

1. *there is an $\mathcal{L}$-ontology $\mathcal{O}$ that fits E;*
2. *there is an interpretation $\mathcal{I}$ with degree at most $2^{\|E\|^2}$ such that*
   (a) *$\mathcal{I} = \biguplus_{e \in E^-} \mathcal{I}_e$ where, for each $e = (\mathcal{A}, q) \in E^-$, $\mathcal{I}_e$ is an $\mathcal{L}$-forest model of $\mathcal{A}$ with $\mathcal{I}_e \not\models q$;*
   (b) *for all $(\mathcal{A}, q) \in E^+$: if h is a homomorphism from $\mathcal{A}$ to $\mathcal{I}$, then $\mathcal{I}_{\mathcal{A},h,\mathcal{L}} \models q$.*

The proof of Theorem 7 follows the same intuitions as the proofs of our previous characterizations, but is more technical. One challenge is that, in the "2 $\Rightarrow$ 1" direction, we first need to construct from an interpretation $\mathcal{I}$ as in the theorem a suitable finite interpretation that we can then use to identify a fitting $\mathcal{L}$-ontology. For this we adopt the finite model construction for ontology-mediated querying from (Gogacz, Ibáñez-García, and Murlak 2018).

6.2 Upper Bounds

Our aim is to prove the following.

Theorem 8. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$ and $\mathcal{Q} \in \{CQ, UCQ\}$. Then $(\mathcal{L}, \mathcal{Q})$-ontology fitting is in* 2ExpTime.

It suffices to prove the theorem for $\mathcal{Q} = \text{UCQ}$. We prove it for $\mathcal{ALC}$ and $\mathcal{ALCI}$ simultaneously. We use the characterization provided by Theorem 7 combined with a mosaic procedure, that is, we attempt to assemble the interpretation $\mathcal{I}$ from Point 2 of Theorem 7 by combining small pieces.

Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$ and assume that we are given a set of ABox-UCQ examples $E_0 = (E_0^+, E_0^-)$. We will often consider maximally connected components of ABoxes and CQs which, for brevity, we simply call *components*. We wish to work with only connected queries in positive examples. This can be achieved as follows. If $(\mathcal{A}, q) \in E_0^+$ with $q = q_1 \vee \cdots \vee q_n$ and q_i has components $p_1, \ldots, p_k$, $k > 1$, then we replace $(\mathcal{A}, q)$ with positive examples $(\mathcal{A}, \widehat{q}_1), \ldots, (\mathcal{A}, \widehat{q}_k)$ where $\widehat{q}_j$ is obtained from q by replacing the disjunct q_i with p_j. This leads to an exponential blowup of the number of positive examples, which, however, does not compromise our upper bound because the size of the examples themselves does not increase.

Throughout this section, we shall be concerned with $\mathcal{L}$-forest models $\mathcal{I}$ of ABoxes $\mathcal{A}$. We generally assume the following naming convention in such models. All elements of $\Delta^\mathcal{I}$ must be of the form aw where $a \in \mathsf{ind}(\mathcal{A})$ and $w \in \mathbb{N}^*$, that is, w is a finite word over the infinite alphabet $\mathbb{N}$. Moreover, $(d, e) \in r^\mathcal{I}$ implies that $d, e \in \mathsf{ind}(\mathcal{A})$ or $e = dc$ or $d = ec$ (if $\mathcal{L} = \mathcal{ALCI}$) where $c \in \mathbb{N}$. If $d, e \in \Delta^\mathcal{I}$ and $e = dc$, then we call e a *successor* of d. Note that a successor may be connected to its predecessor via a role name, an inverse role, or not connected at all. The *depth* of aw is defined as the length of w .

Since mosaics represent 'local' pieces of an interpretation, disconnected ABoxes in examples pose a challenge: a homomorphism may map their components into different parts of a forest model that are far away from each other. We thus need some preparation to deal with disconnected ABoxes. For positive examples, one important ingredient is the following observation.

Lemma 3. *Let $\mathcal{I}$ be an interpretation and $(\mathcal{A}, q) \in E^+$ a positive example such that Condition (b) from Theorem 7 is satisfied and each CQ in q is connected. Then there exists a component $\mathcal{B}$ of $\mathcal{A}$ such that: if h is a homomorphism from $\mathcal{A}$ to $\mathcal{I}$, then $\mathcal{I}_{\mathcal{B},h,\mathcal{L}} \models q$*

Note that Lemma 3 requires the component $\mathcal{B}$ to be uniform across all homomorphisms h. For each $e = (\mathcal{A}, q) \in E^+$, we choose a component $\mathsf{ch}(e)$ of $\mathcal{A}$. Intuitively, $\mathsf{ch}(e)$ is the component $\mathcal{B}$ from Lemma 3 with $\mathcal{I}$ the interpretation from Point 2 of Theorem 7. Since, however, we do not know $\mathcal{I}$, ch acts like a guess and our algorithm shall iterate over all possible choice functions ch.

To deal with an example $e = (\mathcal{A}, q) \in E^+$, we shall focus on the component $\mathsf{ch}(e)$ of $\mathcal{A}$. The other components of $\mathcal{A}$, however, cannot be ignored. We need to know whether they have a homomorphism to $\mathcal{I}$, possibly some remote part of it. This is not easily possible from the local perspective of a mosaic, so we again resort to guessing. We choose a set $\mathfrak{A}$ of ABoxes that are a component of the ABox of some positive example. We will take care (locally!) that no ABox in $\mathfrak{A}$ admits a homomorphism to $\mathcal{I}$. All other components of ABoxes in positive examples may or may not have a homomorphism to $\mathcal{I}$, we shall simply treat them as if they do. We say that a positive example $(\mathcal{A}, q) \in E^+$ is $\mathfrak{A}$-*enabled* if no component of $\mathcal{A}$ is in $\mathfrak{A}$.

Note that the number of choices for $\mathfrak{A}$ and ch is double exponential (it is not single exponential since we have an exponential number of positive examples, see above).

The queries in negative examples need not be connected. To falsify a non-connected CQ, it clearly suffices to falsify one of its components. We use another choice function to choose these components: for each $e = (\mathcal{A}, p_1 \vee \cdots \vee p_k) \in E^-$ and $1 \leq i \leq k$, choose a component $\mathsf{ch}(e, p_i)$ of p_i. There are single exponentially many choices.

Our mosaic procedure tries to assemble the $\mathcal{L}$-forest model $\mathcal{I}$ starting from a large piece that contains the ABox part of $\mathcal{I}$ as well as the tree parts up to depth $3||E||$. The potentially infinite remainder of the trees is then assembled from smaller pieces. We start with defining the large pieces.

Definition 3. *A* base candidate *for* ch *and* $\mathfrak{A}$ *is an interpretation $\mathcal{J} = \biguplus_{e \in E^-} \mathcal{I}_e$ that satisfies the following conditions:*

1. *for each $e = (\mathcal{A}, p_1 \vee \cdots \vee p_k) \in E^-$, $\mathcal{I}_e$ is an $\mathcal{L}$-forest model of $\mathcal{A}$ such that $\mathcal{I}_e \not\models \mathsf{ch}(e, p_i)$ for $1 \leq i \leq k$;*
2. *no ABox from $\mathfrak{A}$ has a homomorphism to $\mathcal{J}$;*
3. *$\mathcal{J}$ has depth at most $3||E_0||$ and degree at most $2^{\|E_0\|^2}$;*

4. *for all $e = (\mathcal{A}, q) \in E^+$ that are $\mathfrak{A}$-enabled: if h is a homomorphism from* $\mathsf{ch}(e)$ *to $\mathcal{J}$ whose range contains only elements of depth at most $2||E_0||$, then $\mathcal{J}_{\mathsf{ch}(e),h,\mathcal{L}} \models q$.*

To make sure that there are only finitely many (in fact double exponentially many) base candidates, we assume that (i) $\mathcal{J}$ interprets only concept and role names that occur in E and (ii) if $w \in \Delta^{\mathcal{J}}$ has k successors $wc_1, \dots, wc_k$, then $\{c_1, \dots, c_k\} = \{1, \dots, k\}$.

We next define the small mosaics. An *$\mathcal{L}$-tree interpretation* $\mathcal{J}$ is defined exactly like an $\mathcal{L}$-forest model, except that all domain elements are of the form $w \in \mathbb{N}^*$, that is, there is no leading individual name. We additionally require the domain $\Delta^{\mathcal{J}}$ to be prefix-closed and call $\varepsilon \in \Delta^{\mathcal{J}}$ the *root* of $\mathcal{J}$.

We say that $\mathcal{J}'$ is a *subtree* of a tree interpretation $\mathcal{J}$ if, for some successor c of ε, $\mathcal{J}'$ is the restriction of $\mathcal{J}$ to all domain elements of the form cw, with $w \in \mathbb{N}^*$.

Definition 4. *A* mosaic for ch, $\mathfrak{A}$, *and $e = (\mathcal{A}, p_1 \vee \cdots \vee p_k) \in E^-$ is an $\mathcal{L}$-tree interpretation $\mathcal{M}$ that satisfies the following conditions:*

1. *$\mathcal{M} \not\models \mathsf{ch}(e, p_i)$ for $1 \leq i \leq k$;*
2. *no ABox from $\mathfrak{A}$ has a homomorphism to $\mathcal{M}$;*
3. *$\mathcal{M}$ has depth at most $3||E_0||$ and degree at most $2^{||E_0||^2}$;*
4. *for all $e = (\mathcal{A}, q) \in E^+$ that are $\mathfrak{A}$-enabled: if h is a homomorphism from* $\mathsf{ch}(e)$ *to $\mathcal{M}$ whose range contains only elements of depth at least $||E_0||$ and at most $2||E_0||$, then $\mathcal{M}_{\mathsf{ch}(e),h,\mathcal{L}} \models q$.*

Let $\mathcal{I}$ be an $\mathcal{L}$-forest model and $d \in \Delta^{\mathcal{I}}$. With $\mathcal{I}|_d^{\downarrow}$, we mean the restriction of $\mathcal{I}$ to all elements of the form dw, with $w \in \mathbb{N}^*$. We say that a mosaic $\mathcal{M}$ *glues to d in $\mathcal{I}$* if $\mathcal{I}|_d^{\downarrow}$ is identical to the interpretation obtained from $\mathcal{M}$ in the following way:

- remove all elements of depth exactly $3||E_0||$;
- prefix every domain element with d, that is, every $w \in \Delta^{\mathcal{M}}$ is renamed to dw.

Our algorithm now works as follows. In an outer loop, we iterate over all possible choices for ch and $\mathfrak{A}$. For each ch and $\mathfrak{A}$, as well as for each $e \in E^-$ we construct the set $S_{e,0}$ of all mosaics for ch, $\mathfrak{A}$, and e and then apply an elimination procedure, producing a sequence of sets

$$S_{e,0} \supseteq S_{e,1} \supseteq S_{e,2} \supseteq \cdots .$$

More precisely, $S_{e,i+1}$ is the subset of mosaics $\mathcal{M} \in S_{e,i}$ that satisfy the following condition:

($*$) for all successors c of ε, there is an $\mathcal{M}' \in S_{e,i}$ that glues to c in $\mathcal{M}$.

Let S_e be the set of mosaics obtained after stabilization.

We next iterate over all base candidates $\mathcal{J} = \bigcup_{e \in E^-} \mathcal{I}_e$ for ch and $\mathcal{A}$, for each of them checking whether there is, for every element $d \in \Delta^{\mathcal{I}_e}$ of depth 1, a mosaic $\mathcal{M} \in S_e$ that glues to d in $\Delta^{\mathcal{I}_e}$. If the check succeeds for some ch, $\mathfrak{A}$ and $\mathcal{J}$, we return 'fitting exists'. Otherwise, we return 'no fitting exists'.

Lemma 4. *The algorithm returns 'fitting exists' if and only if there is an $\mathcal{L}$-ontology that fits E_0.*

It remains to verify that the algorithm runs in double exponential time. Most importantly, we need an effective way to check Condition 4 of Definitions 3 and 4. This is provided by the subsequent lemma.

Let $\mathcal{A}$ be an ABox and p a CQ. An *$\mathcal{A}$-variation* of p is a CQ p' that can be obtained from p by consistently replacing zero or more variables with individual names from $\mathsf{ind}(\mathcal{A})$ and possibly identifying variables. We say that p' is *proper* if the following conditions are satisfied:

1. if $r(a, b) \in p'$ with $a, b \in \mathsf{N_I}$, then $r(a, b) \in \mathcal{A}$;
2. $\mathcal{I}_{p'}$ is an $\mathcal{L}$-forest model of $\mathcal{A} \cap p'$.

Further, let $\mathcal{I}$ be an interpretation, h a homomorphism from $\mathcal{A}$ to $\mathcal{I}$, p' an $\mathcal{A}$-variation of p and g a weak homomorphism from p' to $\mathcal{I}$. We say that g is *compatible* with h if

1. $h(a) = g(a)$ for all individual names a in p';
2. for every variable x in p', there is an $a \in \mathsf{ind}(\mathcal{A})$ such that $g(x)$ is $\mathcal{L}$-reachable from $h(a)$ in $\mathcal{I}$.

Here, an element $e \in \Delta^{\mathcal{I}}$ is $\mathcal{ALCI}$-reachable from $d \in \Delta^{\mathcal{I}}$ if there are $d_0, \dots, d_n \in \Delta^{\mathcal{I}}$ such that $d = d_0$, $d_n = e$, and, for $0 \leq i < n$, $(d_i, d_{i+1}) \in r^{\mathcal{I}}$ for some role r. In $\mathcal{ALC}$-reachability, 'role' is replaced by 'role name'.

Lemma 5. *Let $(\mathcal{A}, q)$ be an example, $\mathcal{I}$ an interpretation, and h a homomorphism from $\mathcal{A}$ to $\mathcal{I}$. Then the following are equivalent*

1. *$\mathcal{I}_{\mathcal{A},h,\mathcal{L}} \models q$*
2. *there exists a proper $\mathcal{A}$-variation p' of a CQ p in q and a weak homomorphism from p' to $\mathcal{I}$ that is compatible with h.*

The Conditions in Point 2 of Lemma 5 can clearly be checked by brute force in single exponential time, and so can Conditions 1 to 3 of Definitions 3 and 4. Based on Conditions 3 in these definitions, it is thus easy to see that we can produce the set of all base candidates and of all mosaics by a straightforward enumeration in double exponential time. The elimination phase of the algorithm also clearly needs only double exponential time.

6.3 Lower Bounds

Theorem 9. *The $(\mathcal{ALCI}, \mathrm{CQ})$-ontology fitting problem is* 2ExpTime*-hard.*

To prove Theorem 9, we give a polynomial time reduction from the complement of Boolean CQ entailment in $\mathcal{ALCI}$, which is 2ExpTime-hard (Lutz 2007). Assume that we are given $\mathcal{A}$, $\mathcal{O}$, and q, and want to decide whether $\mathcal{A} \cup \mathcal{O} \models q$. We construct a collection of labeled ABox-CQ examples (E^+, E^-) such that $\mathcal{A} \cup \mathcal{O} \not\models q$ if and only if there is an $\mathcal{ALCI}$-ontology $\mathcal{O}'$ that fits (E^+, E^-).

To keep the reduction simple, we assume $\mathcal{O}$ to be in *normal form*, meaning that every concept inclusion in $\mathcal{O}$ has one of the following forms: $\top \sqsubseteq A$, $A_1 \sqcap A_2 \sqsubseteq A$, $A \sqsubseteq \exists r.B$, $\exists r.B \sqsubseteq A$, $A \sqsubseteq \neg B$, $\neg B \sqsubseteq A$. It is well-known and easy to see that any $\mathcal{ALCI}$-ontology $\mathcal{O}$ can be rewritten into an ontology $\mathcal{O}'$ of this form in polynomial time, introducing fresh concept names as needed, such that $\mathcal{A} \cup \mathcal{O} \models q'$ if and only if $\mathcal{A} \cup \mathcal{O}' \models q'$ for all Boolean CQs q' that do not use the fresh concept names.

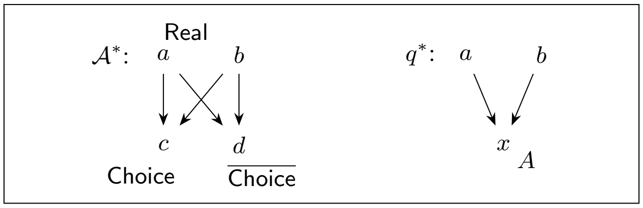

Figure 1: Arrows denote s-edges.

The reduction uses fresh concept names Real, Choice, $\overline{\text{Choice}}$, and F, a fresh concept name $\overline{A}$ for every concept name A in $\mathcal{O}$ and q, and a fresh role name s. It is helpful to have the characterization in Theorem 7 in mind when reading on.

We use a single negative example to ensure that the interpretation $\mathcal{I}$ from Point 2 of Theorem 7 is a model of $\mathcal{A}$ and makes the concept name F false everywhere:

$$(\mathcal{A} \cup \{\mathsf{Real}(a) \mid a \in \mathsf{ind}(\mathcal{A})\}, \exists x\, F(x)).$$

We will use a gadget that introduces auxiliary domain elements. To distinguish the domain elements of primary interest from the auxiliary ones, we label the former with the concept name Real.

The interesting part of the reduction is to guarantee that at each element labeled with Real and for each concept name A in $\mathcal{O}$ or q, exactly one of the concept names A and $\overline{A}$ is true. It is easy to express that both concept names cannot be true simultaneously, via the following positive example:

$$(\{A(a), \overline{A}(a)\}, \exists x\, F(x)).$$

To ensure that at least one of A and $\overline{A}$ is true, we use the announced gadget. We first introduce one successor that satisfies Choice and one that satisfies $\overline{\text{Choice}}$, via the following positive examples:

$$(\{\mathsf{Real}(a)\}, q) \text{ with } q = \exists x\, (s(a,x) \land \mathsf{Choice})$$
$$(\{\mathsf{Real}(a)\}, q) \text{ with } q = \exists x\, (s(a,x) \land \overline{\mathsf{Choice}}).$$

We then use a positive example $(\mathcal{A}^*, q^*)$ where $\mathcal{A}^*$ and q^* are displayed in Figure 1. To understand the gadget, recall Condition (b) of Theorem 7 and the fact that $\mathcal{I}_{\mathcal{A}^*,h,\mathcal{L}}$ is a forest model of $\mathcal{A}^*$, for any $\mathcal{I}$. The variable x in q^* has two distinct individual names as a predecessor, but the only elements in $\mathcal{I}_{\mathcal{A}^*,h,\mathcal{L}}$ with this property are from $\mathsf{ind}(\mathcal{A}^*)$. It follows that any homomorphism that witnesses $\mathcal{I}_{\mathcal{A}^*,h,\mathcal{L}} \models q^*$ as required by Condition (b) of Theorem 7 maps x to c or to d. We transfer the choice back to the original element:

$$(\mathcal{A}, A(a)) \text{ with } \mathcal{A} = \{\mathsf{Real}(a), s(a,b), \mathsf{Choice}(b), A(b)\}$$
$$(\mathcal{A}, \overline{A}(a)) \text{ with } \mathcal{A} = \{\mathsf{Real}(a), s(a,b), \overline{\mathsf{Choice}}(b), A(b)\}.$$

We next include positive examples that encode $\mathcal{O}$:

$$\begin{array}{ll}
(\{\mathsf{Real}(a)\}, A(a)) & \text{for every } \top \sqsubseteq A \in \mathcal{O} \\
(\{A_1(a), A_2(a)\}, A(a)) & \text{for every } A_1 \sqcap A_2 \sqsubseteq A \in \mathcal{O} \\
\multicolumn{2}{l}{(\{A(a)\}, q) \text{ with } q = \exists y\, (r(a,y) \land \mathsf{Real}(y) \land B(y)))} \\
 & \text{for every } A \sqsubseteq \exists r.B \in \mathcal{O} \\
(\{r(a,b), B(b)\}, A(a)) & \text{for every } \exists r.B \sqsubseteq A \in \mathcal{O} \\
(\{A(a)\}, \overline{B}(a)) & \text{for every } A \sqsubseteq \neg B \in \mathcal{O} \\
(\{\overline{B}(a)\}, A(a)) & \text{for every } \neg B \sqsubseteq A \in \mathcal{O}.
\end{array}$$

Finally, we add a positive example that ensures that F is non-empty if q is made true:

$$(\mathcal{A}_q, \exists x\, F(x)),$$

where $\mathcal{A}_q$ is q viewed as an ABox, that is, variables become individuals and atoms become assertions. It remains to show the following.

Lemma 6. *$\mathcal{A} \cup \mathcal{O} \not\models q$ if and only if there is an $\mathcal{ALCI}$-ontology that fits (E^+, E^-).*

The reduction used in the proof of Theorem 9 also works for $\mathcal{ALC}$. But since CQ entailment in $\mathcal{ALC}$ is only ExpTime-complete, this does not deliver the desired lower bound. We thus resort to a reduction of the word problem for exponentially space-bounded alternating Turing machines (ATMs). Such reductions have been used oftentimes for DL query entailment problems, see e.g. (Lutz 2007; Eiter et al. 2009; Bednarczyk and Rudolph 2022).

Theorem 10. *The ($\mathcal{ALC}$, CQ)-ontology fitting problem is 2ExpTime-hard.*

The crucial step in an ATM reduction of this kind is to ensure that tape cells of two consecutive configurations are labeled in a matching way. This is typically achieved by copying the labeling of each configuration to all successor configurations so that the actual comparison can take place locally. We achieve this with a gadget that is based on the same basic idea as the gadget used in the proof of Theorem 9, but much more intricate.

7 Conclusion

We introduced ontology fitting problems based on ABox-query examples and presented algorithms and complexity results, concentrating on the ontology languages $\mathcal{ALC}$ and $\mathcal{ALCI}$. We believe that our results can be adapted to cover many common extensions of these. As an illustration, we show in the appendix the following result for the extension $\mathcal{ALCQ}$ of $\mathcal{ALC}$ with qualified number restrictions. A homomorphism h from an ABox $\mathcal{A}_1$ to an ABox $\mathcal{A}_2$ is *locally injective* if $h(b) \neq h(c)$ for all $r(a,b), r(a,c) \in \mathcal{A}_1$.

Theorem 11. *Let $E = (E^+, E^-)$ be a collection of labeled ABox examples and $\mathcal{A}^+ = \biguplus E^+$. Then the following are equivalent:*

1. *E admits a fitting $\mathcal{ALCQ}$-ontology;*
2. *there is no homomorphism from any $\mathcal{A} \in E^-$ to $\mathcal{A}^+$ that is locally injective.*

Apart from extensions of $\mathcal{ALC}$, there are many other natural ontology languages of interest that can be studied in future work, including Horn DLs such as $\mathcal{EL}$ and existential rules. One can also vary the framework in several natural ways and, for instance, consider the case where a signature for the fitting ontology is given as an additional input or where negative examples have a stronger semantics, namely $\mathcal{A} \cup \mathcal{O} \models \neg q$ in place of $\mathcal{A} \cup \mathcal{O} \not\models q$.

Acknowledgements

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

References

Baader, F., and Distel, F. 2009. Exploring finite models in the description logic. In Ferré, S., and Rudolph, S., eds., *Proc. of ICFCA 2009*, volume 5548 of *Lecture Notes in Computer Science*, 146–161. Springer.

Baader, F.; Ganter, B.; Sertkaya, B.; and Sattler, U. 2007. Completing description logic knowledge bases using formal concept analysis. In Veloso, M. M., ed., *Proc. of IJCAI 2007*, 230–235.

Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logic*. Cambridge University Press.

Bednarczyk, B., and Rudolph, S. 2022. The price of selfishness: Conjunctive query entailment for ALCSelf is 2EXPTIME-hard. In *36th Proc. of AAAI 2022*, 5495–5502. AAAI Press.

Bienvenu, M., and Ortiz, M. 2015. Ontology-mediated query answering with data-tractable description logics. In *Proc. of Reasoning Web*, volume 9203 of *LNCS*, 218–307. Springer.

Bienvenu, M.; ten Cate, B.; Lutz, C.; and Wolter, F. 2014. Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. *ACM Trans. Database Syst.* 39(4):33:1–33:44.

Bühmann, L.; Lehmann, J.; Westphal, P.; and Bin, S. 2018. DL-Learner structured machine learning on semantic web data. In *Proc. of WWW*, 467–471. ACM.

Calvanese, D.; Dragone, L.; Nardi, D.; Rosati, R.; and Trisolini, S. 2006. Enterprise modeling and data warehousing in telecom italia. *Inf. Syst.* 31(1):1–32.

Eiter, T.; Lutz, C.; Ortiz, M.; and Simkus, M. 2009. Query answering in description logics with transitive roles. In Boutilier, C., ed., *Proc. of IJCAI 2009*, 759–764.

Funk, M.; Jung, J. C.; Lutz, C.; Pulcini, H.; and Wolter, F. 2019. Learning description logic concepts: When can positive and negative examples be separated? In *Proc. of IJCAI 2019*, 1682–1688. ijcai.org.

Funk, M.; Grosser, M.; and Lutz, C. 2025. Fitting description logic ontologies to abox and query examples. arXiv. 2508.08007.

Gogacz, T.; Ibáñez-García, Y. A.; and Murlak, F. 2018. Finite query answering in expressive description logics with transitive roles. In *Proc. of KR 2018*, 369–378. AAAI Press.

Jacindha, S.; Abishek, G.; and Vasuki, P. 2022. Program synthesis—a survey. In *Computational Intelligence in Machine Learning*, 409–421. Singapore: Springer Nature Singapore.

Jung, J. C.; Lutz, C.; Pulcini, H.; and Wolter, F. 2020. Logical separability of incomplete data under ontologies. In *Proc. of KR*, 517–528.

Jung, J. C.; Lutz, C.; Pulcini, H.; and Wolter, F. 2021. Separating data examples by description logic concepts with restricted signatures. In *Proc. of KR*, 390–399.

Jung, J. C.; Lutz, C.; Pulcini, H.; and Wolter, F. 2022. Logical separability of labeled data examples under ontologies. *Artif. Intell.* 313:103785.

Jung, J. C.; Hosemann, S.; and Lutz, C. 2025. Fitting ontologies and constraints to relational structures. In *Proc. of KR*.

Kharlamov, E.; Hovland, D.; Skjæveland, M. G.; Bilidas, D.; Jiménez-Ruiz, E.; Xiao, G.; Soylu, A.; Lanti, D.; Rezk, M.; Zheleznyakov, D.; Giese, M.; Lie, H.; Ioannidis, Y. E.; Kotidis, Y.; Koubarakis, M.; and Waaler, A. 2017. Ontology based data access in Statoil. *J. Web Semant.* 44:3–36.

Konev, B.; Lutz, C.; Ozaki, A.; and Wolter, F. 2017. Exact learning of lightweight description logic ontologies. *J. Mach. Learn. Res.* 18:201:1–201:63.

Kriegel, F. 2024. Efficient axiomatization of OWL 2 EL ontologies from data by means of formal concept analysis. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Proc. of AAAI 2024*, 10597–10606. AAAI Press.

Lehmann, J., and Hitzler, P. 2010. Concept learning in description logics using refinement operators. *Mach. Learn.* 78(1-2):203–250.

Li, H.; Chan, C.-Y.; and Maier, D. 2015. Query from examples: An iterative, data-driven approach to query construction. *Proc. VLDB Endow.* 8(13):2158–2169.

Lutz, C., and Wolter, F. 2012. Non-uniform data complexity of query answering in description logics. In *Proc. KR 2012*. AAAI Press.

Lutz, C. 2007. Inverse roles make conjunctive queries hard. In *Proc. of DL 2007*, volume 250 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Lutz, C. 2008. Two upper bounds for conjunctive query answering in SHIQ. In *Proc. of DL 2008*, volume 353 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Ozaki, A. 2020. Learning description logic ontologies: Five approaches. where do they stand? *Künstliche Intell.* 34(3):317–327.

Rizzo, G.; Fanizzi, N.; and d'Amato, C. 2020. Class expression induction as concept space exploration: From DL-Foil to DL-Focl. *Future Gener. Comput. Syst.* 108:256–272.

Sequeda, J. F.; Briggs, W. J.; Miranker, D. P.; and Heideman, W. P. 2019. A pay-as-you-go methodology to design and build enterprise knowledge graphs from relational databases. In *Proc. of ISWC*, volume 11779 of *LNCS*, 526–545. Springer.

Shalev-Shwartz, S., and Ben-David, S. 2014. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press.

Xiao, G.; Calvanese, D.; Kontchakov, R.; Lembo, D.; Poggi, A.; Rosati, R.; and Zakharyaschev, M. 2018. Ontology-based data access: A survey. In *Proc. of IJCAI*, 5511–5519. ijcai.org.