Signature-Based ABox Abduction in $\mathcal{ALC}$ is Hard*

Patrick Koopmann

Institute for Theoretical Computer Science, Technische Universität Dresden, Germany

Abstract. In ABox abduction, we are given a knowledge base together with an ABox—the observation—that is not logically entailed by the knowledge base, and are looking for another ABox—the hypothesis—that, when added to the knowledge base, would make the observation entailed. This has applications in explaining negative entailments and missing query answers, as well as in diagnosis. To get useful hypotheses, in signature-based abduction, we additionally provide a signature of abducible names, and require the hypothesis to use only names from that signature. In the variant we are considering, the hypothesis may otherwise use fresh individual names, as well as complex concepts constructed in arbitrary way using the names in the signature. It was recently shown that this variant of abduction is in N2ExpTime$^{\text{NP}}$, and that hypotheses may require concepts that are of triple exponential size. We complement those results by showing a matching N2ExpTime$^{\text{NP}}$ lower bound, and show that in the worst case, hypotheses may also use a double exponential number of fresh individual names.

1 Introduction

Since inferences performed by description logic reasoners can be complex, and real ontologies often contain 10,000s of axioms, explanations of description logic reasoning has since longer been in the focus of research [26]. In particular for explaining positive entailments, that is, entailments that hold for a given knowledge base, there is a plethora of research, mostly focused on using justifications [33,4,30,17], but recently also on using proofs [1,2]. Explaining negative entailments, that is, entailments that do not hold for a given knowledge base, has been less in the focus of attention. Here, common approaches are showing the user a counter example in form of a description logic interpretation [5,3], or using abduction [29,23]. In abduction, one is given a knowledge base and an *observation*, a formula that is not logically entailed by the knowledge base, and is interested in finding a missing piece in the knowledge base that would make the observation logically entailed, called a *hypothesis* for the observation. Abduction can be used in different ways to explain negative entailments to ontology users [23,7,8] and to repair missing entailments [34]. Another application of abduction is diagnosis: here, the observation describes symptoms, for instance of a medical condition or a faulty machine [27,8], and the hypothesis is supposed to give a possible explanation for *how* the symptoms came into place.

* This work was supported by the DFG in grant 389792660 as part of TRR 248.

To illustrate, we reuse a simplified example from [21] from the geology domain. The observation is that the ground under a street has become unstable: $\{\mathsf{Unstable}(\mathsf{street})\}$, and the background knowledge involves the following axioms:

1. $\mathsf{EvaporiteFormation} \sqcap \exists \mathsf{borders}.(\mathsf{WaterWay} \sqcap \neg \exists \mathsf{lining}.\mathsf{WaterProof}) \sqsubseteq \exists \mathsf{affectedBy}.\mathsf{Dissolution}$
2. $\mathsf{EvaporiteFormation} \sqcap \exists \mathsf{affectedBy}.\mathsf{Dissolution} \sqsubseteq \forall \mathsf{above}.\mathsf{Unstable}$
3. $(\mathsf{Waterway} \sqcup \mathsf{Street}) \sqcap \mathsf{EvaporiteFormation} \sqsubseteq \bot$
4. $\mathsf{Waterway}(\mathsf{canal})$ 5. $\mathsf{Street}(\mathsf{street})$,

These state: 1. an evaporite formation bordering to a waterway without waterproof lining is affected by dissolution; 2. if an evaporite formation is affected by dissolution, then everything above it is unstable; 3. waterways and streets are different from evaporite formations; 4. canal is a waterway and 5. street is a street. A meaningful hypothesis could then be the following, which uses e as a fresh individual referring to a possible unknown formation under the street, as well as the complex concept $\forall \mathsf{lining}.\bot$:

$$\mathcal{H} = \{\ \mathsf{EvaporiteFormation}(e),\ \mathsf{above}(e, \mathsf{street}),\ \mathsf{borders}(e, \mathsf{canal}),\ \forall \mathsf{lining}.\bot(\mathsf{canal})\ \}$$

Depending on whether observation and hypothesis consist of concepts, facts, terminological axioms or knowledge bases, one distinguishes between concept abduction [6], ABox abduction [9,8,32,31,7,12,10,16,19], TBox abduction [11,34,15,14], and knowledge base abduction [24,13]. In this paper, we focus on ABox abduction, where, as in the above example, both the observation and the hypothesis consist of a set of facts about some set of individuals, a variant especially suited for diagnosis and for generating counter examples as in [23].

To avoid trivial answers (such as the observation itself), usually, the abduction problem is specified with additional constraints on the hypothesis. Those may be purely of logical nature [13], or based on syntactic properties. For instance, in [11], the user specifies syntactical patterns of axioms that the hypothesis must conform to. Another approach is to fix a finite set of abducible axioms—either directly as part of the input [31,8], or indirectly by posing strict conditions on the shape of axioms, such as being flat ABox assertions using only names and individuals from the input [32,10]. However, such a restriction might not be feasible in applications where we do not know the exact shape of the axioms we are looking for, for instance because they involve unknown individuals or complex concepts as in the example above. Another approach is to instead specify a set of abducible names or predicates, from which hypotheses can then be constructed in arbitrary fashion using the constructors of the respective description logic. In the context of diagnosis, this set would contain names that are connected to causes of symptoms rather than the symptoms themselves, specific terms rather than generic categories, and possibly describing aspects that can be verified to check the hypothesis. In the example above, examples for names that would not be included would be Unstable (because it is about the observation, not about the cause), Waterway and Street (because we know all waterways and streets in the area), affectedBy and Dissolution (because they are too unspecific, and we are not looking for the processes, but the objects that caused the street to become unstable.)

This form of abduction, which we call signature-based abduction, has been theoretically investigated for DL-Lite in [7], and for more expressive DLs in [21]. There is also a practical implementation for signature-based ABox abduction [9] and one for signature-based KB abduction [24]. These two practical implementation reduce abduction to uniform interpolation via (extensions of) the tool LETHE [20], which can lead to triple exponentially large solutions [25]. As shown in [21], this is indeed in theory unavoidable. However, some crucial questions were left open in [21]: for the corresponding decision problem whether there exists a hypothesis for a given abduction problem, using $\mathcal{ALC}$ as description logic and allowing for both fresh individuals and complex concepts in the hypothesis, only an upper bound of N2EXPTIME$^{\text{NP}}$ is provided, while the lower bound is left open. Furthermore, for the number of fresh individuals one may need to introduce in the worst case, also only an upper bound was developed. In this paper, we show that those bounds are indeed tight, by proving that the general version of the signature-based ABox abduction problem for $\mathcal{ALC}$ is N2EXPTIME$^{\text{NP}}$-complete, and that hypotheses for those problems may require a number of fresh individuals that is double exponential in the size of the abduction problem.

2 Preliminaries

We recall the description logic $\mathcal{ALC}$ and define the signature-based ABox abduction problem formally.

Let $\mathsf{N_C}$, $\mathsf{N_R}$ and $\mathsf{N_I}$ be pair-wise disjoint, countably infinite sets of respectively *concept*, *role* and *individual names*. A *signature* is a finite set $\Sigma \subseteq \mathsf{N_C} \cup \mathsf{N_R}$. Concepts are then built according to the following syntax rule, where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$:

$$C := \top \mid \bot \mid A \mid \neg C \mid C \sqcup C \mid C \sqcap C \mid \exists r.C \mid \forall r.C$$

A *TBox* is a finite set of *general concept inclusion axioms* (GCIs) of the form $C \sqsubseteq D$ and *concept equivalence axioms* of the form $C \equiv D$, where C and D are concepts. An *ABox* is a finite set of *concept assertions* of the form $C(a)$ and *role assertions* of the form $r(a, b)$, where C is a concept, $r \in \mathsf{N_R}$ and $a, b \in \mathsf{N_I}$. A *knowledge base* (KB) $\mathcal{K}$ is a union of a TBox and an ABox, and thus generalises both notions. The contents of a KB are called *axioms*. For a concept/axiom/KB E, we denote *its signature*, that is, the concept and role names that occur in E, by $\mathsf{sig}(E)$.

The semantics of KBs is defined in terms of interpretations. An interpretation $\mathcal{I}$ is a tuple $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ of a countable but possibly infinite *domain* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$ that assigns to each individual name $a \in \mathsf{N_I}$ a domain element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, to each concept name $A \in \mathsf{N_C}$ a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and to each role $r \in \mathsf{N_R}$ a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is extended as follows to concepts:

$$\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \qquad \bot = \emptyset \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\exists r.C)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \text{ there is } \langle d, e \rangle \in r^{\mathcal{I}} \text{ s.t. } e \in C^{\mathcal{I}}\} \\
(\forall r.C)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \text{for every } \langle d, e \rangle \in r^{\mathcal{I}} \text{ we have } e \in C^{\mathcal{I}}\}.
\end{aligned}$$

Such an interpretation $\mathcal{I}$ satisfies an axiom α, in symbols $\mathcal{I} \models \alpha$ if $\alpha = C \sqsubseteq D$ and $C^\mathcal{I} \subseteq D^\mathcal{I}$, $\alpha = C \equiv D$ and $C^\mathcal{I} = D^\mathcal{I}$, $\alpha = C(a)$ and $a^\mathcal{I} \in C^\mathcal{I}$, or $\alpha = r(a,b)$ and $\langle a^\mathcal{I}, b^\mathcal{I}\rangle \in r^\mathcal{I}$. If $\mathcal{I}$ satisfies all axioms in a KB $\mathcal{K}$, we write $\mathcal{I} \models \mathcal{K}$ and call $\mathcal{I}$ a *model of* $\mathcal{K}$. If a KB $\mathcal{K}$ has no model, we say it is *inconsistent* and write $\mathcal{K} \models \bot$. If an axiom α is satisfied in every model of a KB $\mathcal{K}$, we write $\mathcal{K} \models \alpha$ and say α is *entailed by* $\mathcal{K}$.

We are now ready to define the signature-based ABox abduction problem for $\mathcal{ALC}$.

Definition 1. *A* signature based ABox abduction problem *is given by a tuple* $\mathfrak{A} = \langle \mathcal{K}, \Phi, \Sigma\rangle$ *of a KB* $\mathcal{K}$ *called* the background knowledge*, an ABox* Φ *called* the observation*, and a signature* Σ *of* abducible names*, for which we want to determine whether there exists a* hypothesis $\mathcal{H}$*, which is an ABox that satisfies the following conditions:*

A1 $\mathcal{K} \cup \mathcal{H} \not\models \bot$, ***A2*** $\mathcal{K} \cup \mathcal{H} \models \Phi$, *and* ***A3*** $\mathsf{sig}(\mathcal{H}) \subseteq \Sigma$.

Intuitively, the hypothesis should not contradict what we know (Condition **A1**), it should be sufficient to explain the observation (Condition **A2**), and it should only be constructed using the abducible names (Condition **A3**). Apart from that, we put no further restriction on $\mathcal{H}$: it may use individual names that occur neither in $\mathcal{K}$ nor in Φ (*fresh individuals*), and it may contain assertions $C(a)$ where C is a complex concept, arbitrarily composed using the constructors of $\mathcal{ALC}$ and the names in Σ.

Regarding the corresponding decision problem, the following was shown in [21]:

Theorem 1. *Existence of hypotheses for signature-based ABox abduction problems is in* $\text{N2ExpTime}^{\text{NP}}$.

However, a corresponding lower bound was not yet provided, which is the contribution of this paper. In addition, [21] provided the following bounds on the size of hypotheses.

Theorem 2. *There exists a family of signature-based ABox abduction problems for which every hypothesis is of size triple exponential in the size of the problem. Furthermore, for every signature-based ABox abduction problem, if there exists a hypothesis, then there exist one that is of size at most triple exponential in the size of the problem.*

The family of abduction problems used for the lower bound is such that only a single assertion is needed for the hypothesis. Therefore, an open problem remained how many fresh individuals may be required in a hypothesis. Inspection of the proof used for the upper bound of Theorem 2 gives at least an upper bound for this.

Theorem 3. *For every signature-based $\mathcal{ALC}$-ABox abduction problem, if there exists a hypothesis, then there exist one that uses a number of fresh individual names that is at most double exponential in the size of the abduction problem.*

Proof. This is a consequence of the following lemmas from [21]: 1. every hypothesis can be converted into a so-called *hypothesis abstraction* (Lemma 1), 2. every hypothesis abstraction can be restricted to refer to at most double exponentially many individual names (Lemma 2), and 3. those bounds are preserved when translating the hypothesis abstraction back into a hypothesis (Lemma 3). □

We will also show that this bound is tight: in our reduction, we create a class of signature-based abduction problems whose hypotheses always need a double exponential number of fresh individuals.

3 Overview of the Reduction

To prove hardness for N2ExpTime$^{\text{NP}}$, we provide a reduction from the word problem for non-deterministic, double exponential Turing machines with NP oracle. Specifically, given such a Turing machine M and a word w, we will construct an abduction problem $\mathfrak{A}_{M,w}$ in polynomial time s.t. M accepts w iff there exists a hypothesis for $\mathfrak{A}_{M,w}$. The individual names in this hypothesis will be organised into a $N \times N \times N$-cube, where $N = \{0, \dots, 2^n \cdot 2^{2^n}\}$, and n is polynomial in the size of w. That is, for every $\langle x, y, z\rangle \in N \times N \times N$, there will be a corresponding individual $a_{x,y,z}$. The individuals on one side of the cube, that this, those of the form $a_{x,y,0}$, will encode an accepting configuration history for the Turing machine. The remaining individuals supply the "guessing space" for the oracle via the models of the hypothesis. Specifically, for a fixed y, the individuals $a_{x,y,z}$ will be such that they cannot possibly encode a successful computation history for the oracle in any model of the hypothesis and the background knowledge.

Our abduction problem is composed of different components, that we define as separate abduction problems:

1. we reuse a construction from [21] to obtain three double exponential counters;
2. those are used to create the different individuals for all coordinates in $N \times N \times N$;
3. we then enforce that every hypothesis forms a cube in which every coordinate is represented by exactly one individual;
4. finally, we make sure that hypothesis corresponds to a computation history of the Turing machine as described above.

4 The Double Exponential Counters

We encode numbers with 2^n bits using chains of 2^n individuals, where each individual is an instance of the concept name Bit if it represents a bit with value 1, and otherwise an instance of $\neg\mathsf{Bit}$. An abduction problem that exactly provides this is constructed in the proof for Theorem 5 in [21,22] to show the triple exponential lower bound on the size of hypothesis with complex concepts. This abduction problem is of the form $\mathfrak{A}_c = \langle \mathcal{K}_c, \mathsf{Goal}(a), \Sigma_c\rangle$, where $\{r, \mathsf{Bit}, B\} \subseteq \Sigma_c$. $\mathcal{K}_c$ is such that, for any individual name b, the only way to entail $\mathsf{Goal}(b)$ using only names from Σ_c is by enforcing that every path of r-successors starting from b encodes a 2^n-bit counter as required, starting from a value of $2^{2^n} - 1$ and counting down to 0, ending at an individual satisfying B. (The construction also increments this counter along another role s, which is however not relevant for our purposes here.) This also means that no such path can contain any cycles or shortcuts before reaching the instance of B after $2^n \cdot 2^{2^n}$ r-successors.

We will need three such double exponential counters, for which purpose we rename all names in $\mathfrak{A}_c$ to obtain the three signature-disjoint variants $\mathfrak{A}_x = \langle \mathcal{K}_x, \mathsf{Goal}_x(a), \Sigma_x\rangle$, $\mathfrak{A}_y = \langle \mathcal{K}_y, \mathsf{Goal}_y(a), \Sigma_y\rangle$ and $\mathfrak{A}_z = \langle \mathcal{K}_z, \mathsf{Goal}_z(a), \Sigma_z\rangle$. Specifically, these use now roles r_x, r_y and r_z to connect the individuals in the counting sequence, the concept names Bit_x, Bit_y and Bit_z for the respective bit values, and B_x, B_y and B_z to mark the end points of the counters. All counters count backwards: that is, they start at the maximal value, and then step-wise decrease along the role-successors.

5 Creating the Coordinates

To generate all coordinates in our cube, we now need to connect those abduction problems appropriately. Specifically, we need the counter for a coordinate $d \in \{x, y, z\}$ to start not only at the individual name a (the root of the cube), but also at any other individual name along the corresponding side of the cube.

For $d \in \{x, y, z\}$, we use a concept name Max_d to mark individuals that are at the beginning of the sequence for the d-counter (specifically: at the last bit of the maximal counter value). This is established using the following axioms:

$$\mathsf{Max}_d(a) \qquad \mathsf{Max}_d \sqsubseteq \prod_{e \in \{x,y,z\} \setminus d} \forall r_e.\mathsf{Max}_d \sqcap \forall r_d.\neg\mathsf{Max}_d$$

$$\neg\mathsf{Max}_d \sqsubseteq \prod_{e \in \{x,y,z\}} \forall r_e.\neg\mathsf{Max}_d$$

Intuitively, the counter for the coordinate d should be initialised at all points where Max_d is satisfied. However, we cannot initialise the counters directly, but need to use the respective abduction problem. Here, we use that $\mathsf{Goal}_d(b)$ is only entailed for individual names b from which every path along r_d corresponds to a double exponential counter for d. To make sure all counters are initialised at the right positions, we use another concept name $\mathsf{AllStarted}$ that requires that all reachable individuals that satisfy Max_d also satisfy Goal_d. Recall that every counter for d reaches its end at a domain element satisfying B.

$$B_x \sqcup B_y \sqcup B_z \sqsubseteq \mathsf{AllStarted}$$

$$\prod_{d \in \{x,y,z\}} \prod_{e \in \{x,y,z\}} \exists r_d.(\mathsf{AllStarted} \sqcap (\neg\mathsf{Max}_e \sqcup \mathsf{Goal}_e)) \sqsubseteq \mathsf{AllStarted}$$

The concept name Coords now requires all counters to have started:

$$\mathsf{AllStarted} \sqcap \mathsf{Goal}_x \sqcap \mathsf{Goal}_y \sqcap \mathsf{Goal}_z \sqsubseteq \mathsf{Coords}.$$

Set $\mathcal{K}_{\text{coord}}$ to be the union of $\mathcal{K}_x$, $\mathcal{K}_y$, $\mathcal{K}_z$ and these new axioms. The abduction problem

$$\mathfrak{A}_{\text{coord}} = \langle \mathcal{K}_{\text{coord}}, \mathsf{Coords}(a), \Sigma_x \cup \Sigma_y \cup \Sigma_z \rangle$$

now produces the required counters in its hypotheses, reachable by the corresponding paths of r_x, r_y and r_z-successors.

6 Enforcing the Grid Shape

So far, our abduction problem would still allow for a solution with just a single individual, which would produce the counters using a complex concept. This changes with the next extension, which makes sure that hypotheses need to use a number of fresh individuals that is double exponential in the size of the input, and that those individuals are organised into a three-dimensional grid via the three roles r_x, r_y and r_z. Rather than

enforcing this grid shape directly, we define a concept to detect whether a hypothesis is not of the desired grid shape, and then require for the observation to be entailed that the root individual is not an instance of this concept.

For every two $d, e \in \{x, y, z\}$ st. $d \neq e$, we add the following axiom

$$\mathsf{NonSquare} \sqsubseteq \exists r_d.\exists r_e.D \sqcap \exists r_e.\exists r_d.\neg D,$$

NonSquare is only satisfiable by domain elements for which there is a path along r_d-r_e that ends in a different domain element than a path along r_e-r_d. As in an interpretation, there may be more r_d and r_e successors than specified in the ABox, we use the following axioms to make sure NonSquare cannot be satisfied by an individual in an ABox in which the next two r_d and r_e-successors form a square.

$$\exists r_d.D \equiv \forall r_d.D \qquad \exists r_e.D \equiv \forall r_e.D$$
$$\exists r_d.\exists r_e.D \equiv \forall r_d.\exists r_e.D \qquad \exists r_e.\exists r_d.D \equiv \forall r_e.\exists r_d.D$$

The only way to entail $\neg\mathsf{NonSquare}(b)$ for an individual b, and without using D or NonSquare, is now by providing an r_d-r_e-path and an r_e-r_d-path that both end on the same individual, and thus by creating one cell of a grid. Next, we define the concept NonGrid to detect appearances of NonSquare anywhere on a path between the root element and the end of our paths marked by the concept names B_x, B_y and B_z. Specifically, for all $d \in \{x, y, z\}$, we add:

$$\mathsf{NonGrid} \sqsubseteq \neg(B_x \sqcup B_y \sqcup B_z) \sqcap (\mathsf{NonSquare} \sqcup \exists r_d.\mathsf{NonGrid})$$
$$\exists r_d.\neg\mathsf{NonGrid} \sqsubseteq \forall r_d.\neg\mathsf{NonGrid}$$

The first axiom ensures that, if a domain element is in NonGrid, then there must be a path to some element in NonSquare that does not go through $B_x \sqcup B_y \sqcup B_z$. Together with the first axiom, the second axiom makes sure that, once one path fails to satisfy NonGrid at some point, all predecessors will fail to do so as well, unless NonSquare is already satisfied. Consequently, to satisfy $\neg$NonGrid, we have to ensure that on all paths, $\neg$NonSquare remains satisfied before the paths reach $B_x \sqcup B_y \sqcup B_z$. The only way to do so is by organising the role successors into a grid as required.

The final axiom to be added is the following:

$$\mathsf{Coords} \sqsubseteq \mathsf{NonGrid} \sqcup \mathsf{Grid}$$

Every instance of Coords is either in NonGrid or Grid. To entail $\mathsf{Grid}(a)$, we need to entail $\mathsf{Coords}(a)$ and $\neg\mathsf{NonGrid}(a)$, which is only possible by having all coordinates organised into a cube.

Denote the resulting KB with all axioms mentioned until now by $\mathcal{K}_{\mathsf{grid}}$. Our abduction problem for this section is defined as $\mathfrak{A}_{\mathsf{grid}} \sqsubseteq= \{\mathcal{K}_{\mathsf{grid}}, \mathsf{Grid}(a), \Sigma_x \cup \Sigma_y \cup \Sigma_z\}$. As now every hypothesis needs an individual for every coordinate in the cube, we obtain the following theorem.

Theorem 4. *There exists an unbounded family of $\mathcal{ALC}$ ABox abduction problems such that every hypothesis uses a number of fresh individuals that is at least double exponential in the size of the abduction problem.*

7 Encoding the Turing machine

It remains to encode the Turing machine to finish the reduction. Let

$$M_o = \langle Q_o, \Sigma_o, \Gamma_o, \Delta_o, q_{o0}, \not\flat, F_o \rangle$$

be a non-deterministic Turing machine with states Q_o, input alphabet Σ_o, tape alphabet Γ_o, transition relation $\Delta_o \subseteq (Q_o \times \Gamma_o) \times (Q_o \times \Gamma_o \times \{-1, +1\})$, initial state q_{o0}, blank symbol $\not\flat$ and accepting states F_o. We assume that there exists a polynomial p such that for input words w, M_o stops after at most $p(|w|)$ steps. Furthermore, let

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_?, \not\flat, F, M_o \rangle$$

be a non-deterministic Turing machine that uses M_o as oracle, where Q is the set of states, Σ the input alphabet, Γ the tape alphabet,

$$\Delta \subseteq (Q \times \Gamma \times \Gamma_o) \to (Q \times \Gamma \times \Gamma_o \times \{-1, +1\})$$

the transition relation that now operates on two tapes (the work tape and the oracle tape), q_0 the starting state, $q_?$ the oracle query state, q_- the oracle answer state, and F the set of accepting states. The definition of runs and accepting runs is as usual, however, for convenience, we define it with two differences: 1) our tape head is always on the same position for the work tape and the oracle tape, and 2) when in the oracle state, the Turing machine M goes into state q_- if the oracle rejects the current content of the oracle tape, and otherwise M rejects the input (instead of going into a special state reserved for positive query answers). That 1) is without loss of generality can be seen using a standard trick for reducing multi tape machines to single tape machines: specifically, to simulate a different tape head for the oracle tape, we introduce a "marked" tape symbol γ_T for every tape symbol γ in the original TM, and use those marked symbols to mark the virtual position of the head on the oracle tape, and similarly for the work tape. In order to move the two tapes in different positions, we just move back and forth between the cells on the tapes that are marked in this way. This is possible with only quadratic time overhead.

For Difference 2), we use the following argument: assume we have a Turing machine that, based on the result of the oracle, would go either into a state q_+ (if the oracle accepts) or into a state q_- (if the oracle rejects). We can simulate this behaviour with our type of Turing machine by first guessing the result of the oracle. If we guess for a positive answer, we verify it without the oracle (since the Turing machine is non-deterministic), if we guess for a negative answer, we use the oracle to verify this.

We assume that there is a polynomial q s.t. for words w on the input, M stops after at most $2^{2^{q(|w|)}}$ steps, that is, it accepts a language in $\mathrm{N2ExpTime}^{\mathrm{NP}}$. Fix one such input word w. We choose our n used in the previous sections s.t. $n \geq p(q(|w|))$, this way ensuring that n bounds the length of accepting runs, as well as the maximal length of the tape content, both for M and the oracle machine M_o. Our reduction will now proceed in such a way that for the resulting abduction problem, every hypothesis will encode a successful run on the side of the cube for which the z-coordinate is 0: specifically, we will represent the content of the two tapes of M along the x-axis, and succeeding configurations along the y-axis.

For convenience, we assume Q, Γ, Q_o and Γ_o to be pairwise disjoint, and $Q \cup \Gamma \cup Q_o \cup \Gamma_o \subseteq \mathsf{N_C}$, so that we can identify states and tape symbols directly with concept names. We however need to make sure that every domain element can represent at most one state and tape symbol at a time:

$$q \sqcap q' \sqsubseteq \bot \qquad \text{for every } q, q' \in Q, q \neq q' \tag{1}$$
$$\gamma \sqcap \gamma' \sqsubseteq \bot \qquad \text{for every } \gamma, \gamma' \in \Gamma, \gamma \neq \gamma' \tag{2}$$
$$\gamma_o \sqcap \gamma'_o \sqsubseteq \bot \qquad \text{for every } \gamma_o, \gamma'_o \in \Gamma_o, \gamma_o \neq \gamma'_o \tag{3}$$

The input word $w = w_0 \dots w_m$ is encoded as follows:

$$w_0(a), r_x(a, a_1), w_1(a_1), r_x(a_1, a_2), \dots, w_m(a_m), r_x(a_m, a_{m+1}), \not\flat(a_{m+1}) \tag{4}$$

To make sure the blank symbol is propagated along the available space on the tape, we use the following TBox axiom. Recall that Max_y and Max_z are satisfied on all individuals where the y and z-counters are initialised, and B_x at the end of the counting sequence along r_x.

$$\not\flat \sqcap \mathsf{Max}_y \sqcap \mathsf{Max}_z \sqcap \neg B_x \sqsubseteq \forall r_x.\not\flat \tag{5}$$

We represent the current state on every node in the grid, and mark the position of the tape head with a special concept name Head:

$$\mathsf{Head}(a) \tag{6}$$
$$\mathsf{Max}_y \sqcap \mathsf{Max}_z \sqsubseteq q_0 \tag{7}$$

This completes the encoding of the initial configuration.

We further make sure that also on the other configurations, every individual along the x-axis gets assigned the same state $q \in Q$:

$$q \equiv \forall r_x.q \tag{8}$$

and that at most one individual satisfies Head. For the latter, we use the concept $\mathsf{LeftOfHead}$ to mark individuals left of the head. $q \in Q$:

$$q \equiv \forall r_x.q \tag{9}$$
$$(\mathsf{Max}_x \sqcap \neg \mathsf{Head}) \sqsubseteq \mathsf{LeftOfHead} \tag{10}$$
$$\mathsf{LeftOfHead} \sqsubseteq \forall r_x(\mathsf{Head} \sqcup \mathsf{LeaftOfHead}) \tag{11}$$
$$\mathsf{Head} \sqsubseteq \neg \mathsf{LeftOfHead} \sqcap \forall r_x.(\neg \mathsf{Head} \sqcap \neg \mathsf{LeftOfHead}) \tag{12}$$
$$\neg \mathsf{LeftOfHead} \sqcap \neg \mathsf{Head} \sqsubseteq \forall r_x.(\neg \mathsf{Head} \sqcap \neg \mathsf{LeftOfHead}) \tag{13}$$

The transitions that do not use the oracle are encoded using the following TBox axioms. For every $q \in Q \setminus (F \cup \{q_?\})$, $\gamma \in \Gamma$, $\gamma_o \in \Gamma_o$:

$$\exists r_x.(q \sqcap \mathsf{Head} \sqcap \gamma \sqcap \gamma_o) \tag{14}$$
$$\sqsubseteq \bigsqcup\nolimits_{\langle\langle q,\gamma,\gamma_o\rangle,\langle q',\gamma',\gamma'_o,-1\rangle\rangle \in \Delta} \forall r_y.(\mathsf{Head} \sqcap q' \sqcap \forall r_x.(\gamma' \sqcap \gamma'_o)) \tag{15}$$
$$\sqcup \bigsqcup\nolimits_{\langle\langle q,\gamma,\gamma_o\rangle,\langle q',\gamma',\gamma'_o,+1\rangle\rangle \in \Delta} \forall r_y.\forall r_x.(\gamma' \sqcap \gamma'_o \sqcap q' \sqcap \forall r_x.\mathsf{Head}) \tag{16}$$
$$\gamma \sqcap \neg \mathsf{Head} \sqsubseteq \forall r_y.\gamma \tag{17}$$
$$\gamma_o \sqcap \neg \mathsf{Head} \sqsubseteq \forall r_y.\gamma_o \tag{18}$$

The following axioms now propagate the acceptance back to the root, so that we can require it for the observation to be entailed. Here, we use a fresh concept name Accept.

$$\bigsqcup_{q \in F} q \sqsubseteq \mathsf{Accept} \tag{19}$$

$$\exists r_y.\mathsf{Accept} \sqsubseteq \mathsf{Accept} \tag{20}$$

$$\exists r_x.\mathsf{Accept} \sqsubseteq \mathsf{Accept} \tag{21}$$

Our signature $\Sigma_{T,w}$ for the final abduction problem is defined as

$$\Sigma_{T,w} = \Sigma_x \cup \Sigma_y \cup \Sigma_z \cup Q \cup \Gamma \cup \Gamma_o \cup \mathsf{Head} \tag{22}$$

It remains to encode the oracle. While we want the run of M to be represented in the hypothesis, we cannot do the same for the oracle, as we need to quantify over all possible runs (and ensure that none of them is accepting). We thus use a fresh concept name γ_o^* for every $\gamma_o \in \Gamma_o$, since Γ_o is already used in $\Sigma_{T,w}$. The following axioms, for every $\gamma_o \in \Gamma_o$, copy the information from the oracle tape to the input tape of the oracle machine, when the oracle is queried:

$$q_? \sqcap \gamma_o \sqsubseteq \gamma_o^*.$$

Similarly, we use a concept name Tape^* outside of the signature $\Sigma_{T,w}$ for the tape positions of the oracle machine. The following axiom ensures that the head of the oracle tape is always at the left-most position at the beginning:

$$\mathsf{Max}_x \sqcap \mathsf{Max}_z \sqsubseteq \mathsf{Tape}^*$$

Except for axioms 14–16, the rest is encoded using the same axioms as in 7 – 18, however with the role r_y replaced by r_z, Tape replaced by Tape^*, the tape alphabet now using the names γ_o^* instead of γ_o, and of course all states and transitions now referring to the Turing machine M_o.

To detect whether the oracle is rejecting, instead of the axioms 14–16, we represent the transitions as follows for every $q \in Q_o \setminus F_o$ and $\gamma \in \Gamma_o$.

$$\begin{aligned}
\exists r_x.(q \sqcap \mathsf{Head}^* \sqcap \gamma^*) \sqsubseteq & \bigsqcup_{\langle\langle q,\gamma\rangle,\langle q',\gamma',-1\rangle\rangle \in \Delta_o} \forall r_z.(\mathsf{Head}^* \sqcap q' \sqcap \forall r_x.(\gamma')^*) \\
& \sqcup \bigsqcup_{\langle\langle q,\gamma\rangle,\langle q',\gamma',+1\rangle\rangle \in \Delta} \forall r_z.\forall r_x.((\gamma')^* \sqcap q' \sqcap \forall r_x.\mathsf{Head}^*) \\
& \sqcup \mathsf{Reject} \\
\exists x.\mathsf{Reject} \sqsubseteq & \ \mathsf{Reject} \\
\exists z.\mathsf{Reject} \sqsubseteq & \ \mathsf{Reject}
\end{aligned}$$

Those state that, if every computation path in M_o ends in a dead end that is not an accepting state, then in every model, Reject will be satisfied by the domain elements that represent the beginning of the computation.

Finally, we connect both components, the encoding of the Turing machine M and the encoding of the Turing machine M_o, by introducing a concept name $\mathsf{OracleChecked}$ that is satisfied if on every configuration between the accepting configuration and the

initial configuration, either there is no oracle call ($\neg q_?$), or there is an oracle call ($q_?$), the oracle rejects, and the next state is q_-.

$$\bigsqcup_{q \in F} q \sqsubseteq \mathsf{OracleChecked}$$
$$\neg q_? \sqcap \exists r_y.\mathsf{OracleChecked} \sqsubseteq \mathsf{OracleChecked}$$
$$q_? \sqcap \mathsf{Reject} \sqcap \exists r_y.(\mathsf{OracleChecked} \sqcap q_-) \sqsubseteq \mathsf{OracleChecked}$$

To complete the construction, we add the following GCI:

$$\mathsf{Accept} \sqcap \mathsf{OracleChecked} \sqcap \mathsf{Grid} \sqsubseteq \mathsf{Observation}$$

Denote the knowledge base consisting of all axioms shown so far by $\mathcal{K}_{T,w}$. The final abduction problem is now the tuple $\mathfrak{A}_{T,w} = \langle \mathcal{K}_{T,w}, \mathsf{Observation}(a), \Sigma_{T,w} \rangle$.

Lemma 1. *There is a hypothesis for $\mathfrak{A}_{T,w}$ iff T accepts w.*

Together with the upper bound from [21], we thus obtain the following theorem.

Theorem 5. *Signature-based ABox abduction in $\mathcal{ALC}$ is* N2ExpTime$^{\text{NP}}$*-complete.*

8 Conclusion

One conclusion one could draw from the high complexity is that it may make sense to look into more restricted variants of ABox abduction, such as flat ABox abduction, where complex concepts are not allowed in the hypothesis: here hypotheses can get at most exponentially large, and the decision problem (for $\mathcal{ALC}$) is only coNExpTime-complete [21]. On the other hand, especially in the context of description logics, it is often observed that high theoretical complexity results do not mean that there cannot be implementations that work well in practice: for instance, reasoning in $\mathcal{SROIQ}$, the DL underlying the web ontology language standard OWL, is N2ExpTime-complete [18], while modern optimised DL reasoners process even large OWL ontologies in practice [28]. An example more related to signature-based abduction is uniform interpolation, which for $\mathcal{ALC}$ may also lead to triple exponentially large ontologies [25], while systems like Lethe [20] and Fame [35] can often successfully compute uniform interpolants of realistic ontologies in practice.

The current prototypes for signature-based abduction for $\mathcal{ALC}$ [9,24] performed reasonably well in an evaluation on real ontologies, however, they do not introduce fresh individual names. Instead, the more general approach in [24] uses additional DL constructors like inverse roles and nominals to express connections between named individuals. Arguably, the results of abduction would become more user friendly if they would use fresh individual names instead. An approach could be to first compute hypotheses with those systems, and afterwards simplify concepts by introduction of fresh individuals. The results in this paper indicate however that this could turn out challenging in practice, as the number of those individuals may become large in theory.

References

1. Alrabbaa, C., Baader, F., Borgwardt, S., Koopmann, P., Kovtunova, A.: Finding small proofs for description logic entailments: Theory and practice. In: Albert, E., Kovacs, L. (eds.) LPAR-23: 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning. EPiC Series in Computing, vol. 73, pp. 32–67. EasyChair (2020). https://doi.org/10.29007/nhpp
2. Alrabbaa, C., Baader, F., Borgwardt, S., Koopmann, P., Kovtunova, A.: Finding good proofs for description logic entailments using recursive quality measures. In: Platzer, A., Sutcliffe, G. (eds.) Proceedings of the 28th International Conference on Automated Deduction (CADE'21). LNCS (2021), to appear.
3. Alrabbaa, C., Hieke, W., Turhan, A.Y.: Counter model transformation for explaining non-subsumption in $\mathcal{EL}$. In: Beierle, C., Ragni, M., Stolzenburg, F., Thimm, M. (eds.) Proceedings of the 7th Workshop on Formal and Cognitive Reasoning (FCR-2021) co-located with the 44th German Conference on Artificial Intelligence (KI-2021). CEUR Workshop Proceedings, vol. 2961, pp. 9–22. CEUR-WS.org (2021)
4. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic $\mathcal{EL}^+$. In: KI 2007: Advances in Artificial Intelligence, 30th Annual German Conference on AI, KI 2007, Osnabrück, Germany, September 10-13, 2007, Proceedings. pp. 52–67 (2007)
5. Bauer, J., Sattler, U., Parsia, B.: Explaining by example: Model exploration for ontology comprehension. In: Grau, B.C., Horrocks, I., Motik, B., Sattler, U. (eds.) Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009. CEUR Workshop Proceedings, vol. 477. CEUR-WS.org (2009), http://ceur-ws.org/Vol-477/paper_37.pdf
6. Bienvenu, M.: Complexity of abduction in the $\mathcal{EL}$ family of lightweight description logics. In: Proceedings of KR 2008. pp. 220–230. AAAI Press (2008), http://www.aaai.org/Library/KR/2008/kr08-022.php
7. Calvanese, D., Ortiz, M., Simkus, M., Stefanoni, G.: Reasoning about explanations for negative query answers in DL-Lite. J. Artif. Intell. Res. **48**, 635–669 (2013). https://doi.org/10.1613/jair.3870
8. Ceylan, İ.İ., Lukasiewicz, T., Malizia, E., Molinaro, C., Vaicenavicius, A.: Explanations for negative query answers under existential rules. In: Calvanese, D., Erdem, E., Thielscher, M. (eds.) Proceedings of KR 2020. pp. 223–232. AAAI Press (2020). https://doi.org/10.24963/kr.2020/23
9. Del-Pinto, W., Schmidt, R.A.: ABox abduction via forgetting in $\mathcal{ALC}$. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019. pp. 2768–2775. AAAI Press (2019). https://doi.org/10.1609/aaai.v33i01.33012768
10. Du, J., Qi, G., Shen, Y., Pan, J.Z.: Towards practical ABox abduction in large description logic ontologies. Int. J. Semantic Web Inf. Syst. **8**(2), 1–33 (2012). https://doi.org/10.4018/jswis.2012040101
11. Du, J., Wan, H., Ma, H.: Practical TBox abduction based on justification patterns. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. pp. 1100–1106 (2017), http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14402
12. Du, J., Wang, K., Shen, Y.: A tractable approach to ABox abduction over description logic ontologies. In: Brodley, C.E., Stone, P. (eds.) Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. pp. 1034–1040. AAAI Press (2014), http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8191
13. Elsenbroich, C., Kutz, O., Sattler, U.: A case for abductive reasoning over ontologies. In: Proceedings of the OWLED'06 Workshop on OWL: Experiences and Directions (2006), http://ceur-ws.org/Vol-216/submission_25.pdf

14. Haifani, F., Koopmann, P., Tourret, S.: Abduction in $\mathcal{EL}$ via translation to FOL. In: Schmidt, R.A., Wernhard, C., Zhao, Y. (eds.) SOQE 2021: The 2nd Workshop on Second-Order Quantifier Elimination and Related Topics. CEUR Workshop Proceedings, CEUR-WS.org (2021)
15. Haifani, F., Koopmann, P., Tourret, S.: Introducing connection minimal abduction for $\mathcal{EL}$ ontologies. In: Explainable Logic-Based Knowledge Representation (XLoKR 2021) (2021)
16. Halland, K., Britz, K.: ABox abduction in $\mathcal{ALC}$ using a DL tableau. In: 2012 South African Institute of Computer Scientists and Information Technologists Conference, SAICSIT '12. pp. 51–58 (2012). https://doi.org/10.1145/2389836.2389843
17. Horridge, M.: Justification Based Explanation in Ontologies. Ph.D. thesis, University of Manchester, UK (2011), https://www.research.manchester.ac.uk/portal/files/54511395/FULL_TEXT.PDF
18. Kazakov, Y.: RIQ and SROIQ are harder than SHOIQ. In: Brewka, G., Lang, J. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008. pp. 274–284. AAAI Press (2008), http://www.aaai.org/Library/KR/2008/kr08-027.php
19. Klarman, S., Endriss, U., Schlobach, S.: ABox abduction in the description logic $\mathcal{ALC}$. Journal of Automated Reasoning **46**(1), 43–80 (2011). https://doi.org/10.1007/s10817-010-9168-z
20. Koopmann, P.: LETHE: forgetting and uniform interpolation for expressive description logics. Künstliche Intell. **34**(3), 381–387 (2020). https://doi.org/10.1007/s13218-020-00655-w
21. Koopmann, P.: Signature-based abduction with fresh individuals and complex concepts for description logics. In: Zhou, Z. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021. pp. 1929–1935. ijcai.org (2021), https://doi.org/10.24963/ijcai.2021/266
22. Koopmann, P.: Signature-based abduction with fresh individuals and complex concepts for description logics (extended version). CoRR **abs/2105.00274** (2021), https://arxiv.org/abs/2105.00274
23. Koopmann, P.: Two ways of explaining negative entailments in description logics using abduction. In: Explainable Logic-Based Knowledge Representation (XLoKR 2021) (2021)
24. Koopmann, P., Del-Pinto, W., Tourret, S., Schmidt, R.A.: Signature-based abduction for expressive description logics. In: Calvanese, D., Erdem, E., Thielscher, M. (eds.) Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020. pp. 592–602. AAAI Press (2020). https://doi.org/10.24963/kr.2020/59
25. Lutz, C., Wolter, F.: Foundations for uniform interpolation and forgetting in expressive description logics. In: Proceedings of IJCAI 2011. pp. 989–995. IJCAI/AAAI (2011). https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-170
26. McGuinness, D.L.: Explaining Reasoning in Description Logics. Ph.D. thesis, Rutgers University, NJ, USA (1996). https://doi.org/10.7282/t3-q0c6-5305
27. Obeid, M., Obeid, Z., Moubaiddin, A., Obeid, N.: Using description logic and Abox abduction to capture medical diagnosis. In: Wotawa, F., Friedrich, G., Pill, I., Koitz-Hristov, R., Ali, M. (eds.) Advances and Trends in Artificial Intelligence. From Theory to Practice. pp. 376–388. Springer International Publishing, Cham (2019)
28. Parsia, B., Matentzoglu, N., Gonçalves, R.S., Glimm, B., Steigmiller, A.: The OWL reasoner evaluation (ORE) 2015 competition report. J. Autom. Reason. **59**(4), 455–482 (2017). https://doi.org/10.1007/s10817-017-9406-8
29. Peirce, C.S.: Deduction, induction, and hypothesis. Popular science monthly **13**, 470–482 (1878)
30. Peñaloza, R.: Axiom-Pinpointing in Description Logics and Beyond. Ph.D. thesis, Technische Universität Dresden, Germany (2009)

31. Pukancová, J., Homola, M.: Tableau-based ABox abduction for the $\mathcal{ALCHO}$ description logic. In: Proceedings of the 30th International Workshop on Description Logics (2017), http://ceur-ws.org/Vol-1879/paper11.pdf
32. Pukancová, J., Homola, M.: The AAA ABox abduction solver. Künstliche Intell. **34**(4), 517–522 (2020). https://doi.org/10.1007/s13218-020-00685-4
33. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Gottlob, G., Walsh, T. (eds.) Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003). pp. 355–362. Morgan Kaufmann, Acapulco, Mexico (2003), http://ijcai.org/Proceedings/03/Papers/053.pdf
34. Wei-Kleiner, F., Dragisic, Z., Lambrix, P.: Abduction framework for repairing incomplete $\mathcal{EL}$ ontologies: Complexity results and algorithms. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. pp. 1120–1127. AAAI Press (2014), http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8239
35. Zhao, Y., Schmidt, R.A.: FAME(Q): an automated tool for forgetting in description logics with qualified number restrictions. In: Fontaine, P. (ed.) Automated Deduction - CADE 27 - 27th International Conference on Automated Deduction, Natal, Brazil, August 27-30, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11716, pp. 568–579. Springer (2019). https://doi.org/10.1007/978-3-030-29436-6_34