HAL
open science

Abduction inELvia Translation to FOL

Fajar Haifani, Patrick Koopmann, Sophie Tourret

To cite this version:

Fajar Haifani, Patrick Koopmann, Sophie Tourret. Abduction inELvia Translation to FOL. SOQE 2021 - 2nd Workshop on Second-Order Quantifier Elimination and Related Topics, Nov 2021, Hanoï (online), Vietnam. pp.46-58. hal-03516691

HAL Id: hal-03516691

https://inria.hal.science/hal-03516691v1

Submitted on 1 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Abduction in $\mathcal{EL}$ via Translation to FOL *

Fajar Haifani[1], Patrick Koopmann[2], and Sophie Tourret[1,3]

[1] Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken Germany ({fhaifani,stourret}@mpi-inf.mpg.de)
[2] Institute for Theoretical Computer Science, Technische Universität Dresden, Germany (patrick.koopmann@tu-dresden.de)
[3] Université de Lorraine, CNRS, Inria, LORIA, Nancy, France

Abstract. We present a technique for performing TBox abduction in the description logic $\mathcal{EL}$. The input problem is converted into first-order formulas on which a prime implicate generation technique is applied, then $\mathcal{EL}$ hypotheses are reconstructed by combining the generated positive and negative implicates.

Keywords: Abduction · Description Logic $\mathcal{EL}$· Prime Implicates · First-order Logic.

1 Introduction

Description logic (DL) ontologies are used to formalize terminological knowledge in diverse areas such as medicine, biology, or the semantic web. Common reasoning tasks for ontologies are subsumption checking, which is to decide whether one description of a concept generalizes another, and classification, which is to compute the entire subsumption hierarchy of an ontology, that is, the set of all subsumption relationships that hold between atomic concepts in the ontology. Both problems can nowadays be computed rather efficiently using highly optimized description logic reasoners [24]. Modern ontologies can become very large and complex: for instance, the medical ontology SNOMED CT[4] contains over 350,000 statements, while the Gene Ontology GO defines over 50,000 concepts. But even for less complex ontologies, it is easy to introduce bugs, and possible inferences may not always be transparent to ontology users and engineers. Correspondingly, services for explaining those inferences can be very helpful. Typical techniques for explaining why subsumption can be inferred include justifications [4, 16, 27] and proofs [1, 2], both of which have been included in the popular ontology development tool Protégé [17, 18].

In order to explain why a subsumption relationship *cannot* be inferred, one can use abduction [21]. The general setup in abduction is that we have an

* Funded by the DFG grant 389792660 as part of TRR 248 – CPEC, see https://perspicuous-computing.science
Copyright © 2021 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[4] https://www.snomed.org

CEUR-WS.org/Vol-3009/paper2.pdf

input consisting of a knowledge base K and some logical statement α—the observation—that is not logically entailed by K. The aim is then to compute a hypothesis, a set of axioms whose addition to K would result in the entailment α. This way, the hypothesis not only provides for a possible explanation, it also offers a fix to repair the entailment in case it is supposed to hold. One way to avoid trivial hypotheses (such as the axiom α itself) is to restrict solutions syntactically, for instance by specifying a signature of predicates that can be used in the solution [20, 22].

In the context of DLs, depending on the shape of the observation to be explained, abduction can be classified into different categories:

- In *concept abduction* observation and hypothesis are DL concepts. That means we are looking for a concept that is subsumed by a user-given concept (with respect to an ontology) [5].
- In *ABox abduction*, observations and hypotheses have the form of ABoxes, that is, they describe facts, which is the more typical scenario for diagnosis [6–9, 11, 15, 19, 20, 25, 26].
- In *TBox abduction*, observation and hypothesis consist of TBox axioms, which can be used to find fixes for *missing subsumption relationships* [10, 28].
- Finally, in *knowledge base abduction*, the most general form, both hypotheses and observations can be arbitrarily composed of TBox and ABox axioms [13, 22].

We are interested in explaining missing subsumption relationships, which is what is done by TBox abduction, in which background knowledge, observation, and hypothesis all consist of TBox axioms, that is, DL axioms expressing subsumption relationships between concepts. Optionally, a set of abducible concepts can be specified as part of the input, of which every hypothesis needs to be composed. We work in the lightweight DL $\mathcal{EL}$, used in many large scale ontologies [3].

TBox abduction has been considered before, albeit in different settings. In [10], instead of a set of abducibles, a set of *justification patterns* is given, which is a set of syntactical patterns that hypotheses have to fit in. In contrast, [28] generalises both this work and the one in [10], in that it uses an arbitary oracle function to decide whether a solution is admissible or not (which may use abducibles, justification patterns, or something else). The work in [28] also considers $\mathcal{EL}$, and abduction under various minimality notions such as subset minimality and size minimality. It is there shown that deciding the existence of hypotheses is tractable, while for certain minimality criteria, it can become coNP-complete. In addition to the theoretical results, the authors of [28] present practical algorithms, and an evaluation of an implementation. Different to our approach, they use an external DL reasoner to decide entailment relationships. In contrast, we present an approach that directly exploits first-order reasoning, and thus has the potential to be generalisable to more expressive DLs.

The TBox abduction technique we present proceeds in three steps. The first step is the translation of the abduction problem into a first-order formula Φ. Then, we compute the prime implicates of Φ, that is, a minimal set of logical consequences of Φ that subsume all other consequences of Φ. In the final step, we

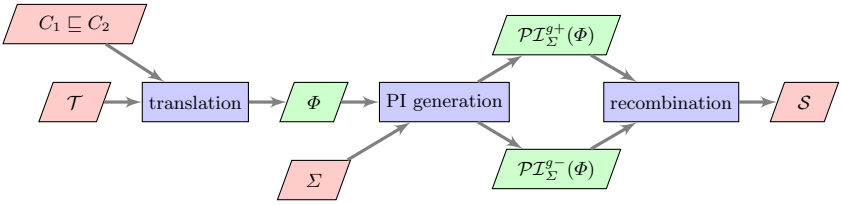

Fig. 1. $\mathcal{EL}$ abduction using prime implicate generation in FOL.

construct TBoxes that are hypotheses of the original abduction problem, from the prime implicates of Φ.

Figure 1 illustrates this process with some more details. The translation to first-order relies on the assumption that the TBox $\mathcal{T}$ is normalized and this results in a set of definite Horn clauses Φ. Prime implicate (PI) generation in first-order logic can be done with an off-the-shelf tool [12, 23] or, in our case, by slightly altering a resolution-based theorem prover, such as included in the SPASS theorem prover [29]. Since the input is Horn, $\mathcal{PI}_\Sigma^{g+}(\Phi)$, the set of ground positive prime implicates of Φ, contains only unit clauses. The recombination step looks at the negative ground prime implicates in $\mathcal{PI}_\Sigma^{g-}(\Phi)$ one after the other and attempts to match each literal in one such clause with clauses in $\mathcal{PI}_\Sigma^{g+}(\Phi)$. If this succeeds, the result is a solution to the original problem. We denote $\mathcal{S}$ the set of all solutions obtained in this way.

In this paper, we describe in details the translation and recombination steps and prove that the $\mathcal{S}$ obtained in that way contains only solutions to the original problem.

2 Preliminaries

We begin with an introduction of the relevant notions, first in $\mathcal{EL}$, then in first-order logic.

2.1 The Description Logic $\mathcal{EL}$

Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be pair-wise disjoint, countably infinite sets of respectively *atomic concepts* and *roles*. Generally, we use letters A, B, E, F,... for atomic concepts, and r for roles, possibly annotated, letters C, D, possibly annotated, denote $\mathcal{EL}$ *concepts*, built according to the syntax rule

$$C \ ::= \ \top \ | \ A \ | \ C \sqcap C \ | \ \exists r.C \ .$$

We use $\bigsqcap\{C_1, \ldots, C_m\}$ as abbreviation for $C_1 \sqcap \ldots \sqcap C_m$, and identify the empty conjunction (when $m = 0$) with $\top$. An $\mathcal{EL}$ *TBox* $\mathcal{T}$ is a finite set of *concept inclusions* (CIs) of the form $C \sqsubseteq D$. We use $C \equiv D$ as abbreviation for $\{C \sqsubseteq D, D \sqsubseteq C\}$. The semantics of $\mathcal{EL}$ is defined as usual using DL interpretations, which are tuples $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ made of a *domain* $\Delta^\mathcal{I}$ and an interpretation function

$\cdot^{\mathcal{I}}$ that maps atomic concepts $A \in \mathsf{N_C}$ to sets $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and roles $r \in \mathsf{N_R}$ to relations $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function $\cdot^{\mathcal{I}}$ is extended to complex concepts as follows:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad (C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$
$$(\exists r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \exists (d,e) \in r^{\mathcal{I}} \text{ s.t. } e \in C^{\mathcal{I}}\}$$

We say that $\mathcal{I}$ *satisfies* a CI $C \sqsubseteq D$, in symbols $\mathcal{I} \models C \sqsubseteq D$, if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and if $\mathcal{I}$ satisfies all axioms in a TBox $\mathcal{T}$, we write $\mathcal{I} \models \mathcal{T}$ and call $\mathcal{I}$ a *model* of $\mathcal{T}$. If a CI $C \sqsubseteq D$ is satisfied in every model of $\mathcal{T}$, we write $\mathcal{T} \models C \sqsubseteq D$ and say that $C \sqsubseteq D$ is *entailed* by $\mathcal{T}$. In this case, we say that C *is subsumed by* D and call C a *subsumee* of D and D a *subsumer* of C.

Note that, if we identify $\mathsf{N_C}$ with the set of unary predicates and $\mathsf{N_R}$ with the set of binary predicates, $\mathcal{I}$ is a classical first-order structure for those predicates. We can thus define satisfaction of first-order formulas in DL interpretations as usual, and also relate first-order formulas with CIs and TBoxes via the usual entailment relation.

2.2 Abduction in $\mathcal{EL}$

The abduction problem we are concerned with in this paper is the following.

Definition 1. *Given a TBox $\mathcal{T}$, a set of atomic concepts $\Sigma \subseteq \mathsf{N_C}$ and a concept inclusion $C_1 \sqsubseteq C_2$, called the* observation*, where C_1 and C_2 are atomic concepts and $\mathcal{T} \not\models C_1 \sqsubseteq C_2$, the corresponding* TBox abduction problem*, denoted by the tuple $\langle \mathcal{T}, \Sigma, C_1 \sqsubseteq C_2 \rangle$, is to find a TBox*

$$\mathcal{H} \subseteq \{A_{i_1} \sqcap \cdots \sqcap A_{i_n} \sqsubseteq B_{i_1} \sqcap \cdots \sqcap B_{i_m} \mid \{A_{i_1}, \ldots, A_{i_n}, B_{i_1}, \ldots, B_{i_m}\} \subseteq \Sigma\}$$

where $m > 0$, $n \geq 0$ and such that $\mathcal{T} \cup \mathcal{H} \models C_1 \sqsubseteq C_2$ and $\mathcal{T} \cap \mathcal{H} = \emptyset$. Such a solution $\mathcal{H}$ to the abduction problem is called a hypothesis*.*

As an example, suppose we have a TBox $\mathcal{T} = \{C_1 \sqsubseteq \exists r.A, \exists r.B \sqsubseteq C_2\}$. For the abduction problem $\langle \mathcal{T}, \{A, B\}, C_1 \sqsubseteq C_2 \rangle$, the set $\mathcal{H} = \{A \sqsubseteq B\}$ is a solution since $\mathcal{T} \uplus \mathcal{H} \models C_1 \sqsubseteq C_2$. Note that since $\mathcal{EL}$ TBoxes are always consistent, the consistency condition usually required on $\mathcal{T} \cup \mathcal{H}$ is not needed in our setting.

2.3 Prime Implicate Generation in First Order Logic

We consider first-order logic with unary and binary predicates, which we identify with the corresponding elements in $\mathsf{N_C}$ and $\mathsf{N_R}$. Let $\mathcal{V}$ be a set of *variables*, and Σ_{FOL} be a signature such that $\Sigma_{\text{FOL}} = \mathsf{N_C} \cup \mathsf{N_R}$, where $\mathsf{N_C}$ and $\mathsf{N_R}$ are the sets of $\mathcal{EL}$ atomic concepts and roles, used as unary and binary predicate symbols respectively. Furthermore, let $\mathcal{F}$ be a set of function symbols, which includes a set $\mathcal{C}$ of constants as nullary function symbols. A *term* t, possibly annotated, is either a variable $x \in \mathcal{V}$, a constant $c \in \mathcal{C}$, or of the form $f(t_1, \ldots, t_m)$, where f is an m-ary function symbol and t_1, …, t_m are terms. *Atoms* are either of

the form $A(t_1)$ or $r(t_1, t_2)$, where t_1 and t_2 are terms. Literals are either atoms κ (positive literals) or negated atoms $\neg\kappa$ (negative literals). Clauses, denoted φ with possible annotations, are disjunctions of literals implicitly universally quantified. Formulas, denoted Φ, are described using universal and existential quantification, conjunction, disjunction, and negation. For convenience, by abuse of notation, we may sometimes identify sets of formulas or clauses with the conjunction over those.

Skolemization removes existentially quantified variables from formulas in the usual way: for each existentially quantified variable, it adds a fresh function with variable arguments taken from the universally quantified variables outside the scope of the variable under consideration (if no such variables exist, we use a nullary function, ie. a constant). For example, $\forall x.\neg A(x) \vee \exists y.r(x, y)$ is Skolemized into $\forall x.\neg A(x) \vee r(x, f(x))$. Here, we assume that the formula is in *negation normal form*, that is, every negation symbol occurs only in front of an atom. Furthermore, we assume that such Skolem functions are taken from a set $\mathsf{N_S}$ not in the signature. In our context, we only require a single Skolem constant, namely sk_0, since Skolemized atoms, literals, and clauses range over Skolem terms either built over a given variable or sk_0 instead of ranging over variables. We let $\mathsf{T}_x(\mathsf{N_S})$ be the set of Skolem terms built on x, where x is either sk_0 or a variable in $\mathcal{V}$.

Since we only consider formulas over unary and binary predicates, they can be satisfied by DL interpretations as defined above, understood naturally as first-order structures. We call interpretations $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \mathsf{T}_{\mathrm{sk}_0}(\mathsf{N_S})$ *Herbrand interpretations*, which for convenience, by abuse of notation, we may treat as sets of ground atoms s.t. $A(t) \in \mathcal{I}$ if and only if $t \in A^{\mathcal{I}}$ and $r(t, t') \in \mathcal{I}$ if and only if $(t, t') \in r^{\mathcal{I}}$.

Definition 2 (Prime Implicate). *Given a set of clauses Φ, a clause φ is an* implicate *of Φ if $\Phi \models \varphi$. Additionally, φ is a* prime implicate *of Φ if for any other implicate φ' of Φ s.t. $\varphi' \models \varphi$, it also holds that $\varphi \models \varphi'$. Given a Skolemized Φ, the set $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ is the set of all positive ground prime implicates of Φ and the set $\mathcal{PI}^{g-}_{\Sigma}(\Phi)$ is the set of all negative ground prime implicates where all of the predicate symbols belong to Σ.*

Example 3. Given Φ as follows:

$$\Phi = \{A(\mathrm{sk}_0), \neg B(\mathrm{sk}_0), \neg A(x) \vee r(x, sk(x)), \neg A(x) \vee E(sk(x)), \\ \neg G(x) \vee \neg r(x, y) \vee \neg E(y) \vee B(\mathrm{sk}_0)\},$$

where sk_0 is a constant and sk is a function symbol, the ground prime implicates of Φ are $\mathcal{PI}^{g+}_{\Sigma}(\Phi) = \{A(\mathrm{sk}_0), E(sk(\mathrm{sk}_0))\}$ and $\mathcal{PI}^{g-}_{\Sigma}(\Phi) = \{\neg B(\mathrm{sk}_0), \neg G(\mathrm{sk}_0) \vee \neg E(sk(\mathrm{sk}_0))\}$. They are implicates because all of them are consequences of Φ (they are entailed by Φ). Moreover, these implicates are also prime implicates because it is not possible to obtain an implicate of Φ by removing a literal from one of them.

In some cases, the set of prime implicates can even be infinite.

Example 4. Consider Φ', a slight variation on Φ from the previous example, where $E(sk(x))$ is replaced by $A(sk(x))$, so that:

$$\Phi' = \{A(\mathrm{sk}_0), \neg B(\mathrm{sk}_0), \neg A(x) \vee r(x, sk(x)), \neg A(x) \vee A(sk(x)), \\ \neg G(x) \vee \neg r(x, y) \vee \neg E(y) \vee B(\mathrm{sk}_0)\},$$

Whereas the negative ground prime implicates are unchanged ($\mathcal{PI}^{g-}_{\Sigma}(\Phi') = \mathcal{PI}^{g-}_{\Sigma}(\Phi)$), there are now infinitely many positive ground prime implicates, because $\mathcal{PI}^{g+}_{\Sigma}(\Phi')$ contains $A(\mathrm{sk}_0)$, $A(sk(\mathrm{sk}_0))$, $A(sk(sk(\mathrm{sk}_0)))$ ...

3 From $\mathcal{EL}$ to FOL

We assume the $\mathcal{EL}$ TBox in the input to be in normal form as defined in [3], which means that every CI is of the form $A \sqsubseteq B$, $A_1 \sqcap A_2 \sqsubseteq B$, $\exists r.A \sqsubseteq B$ or $A \sqsubseteq \exists r.B$, where $r \in \mathsf{N_R}$ and $A, A_1, A_2, B \in \mathsf{N_C} \cup \{\top\}$. Every $\mathcal{EL}$ TBox can be transformed in polynomial time into this normal form through the introduction of fresh concept names. This transformation preserves all entailments in the signature of the original ontology, even after the addition of CIs (assuming the same CIs are added also to the input ontology, and they don't use any of the introduced concept names). Note that this transformation extends the signature, and thus potentially the solution space of the abduction problem: to avoid this, we may simply intersect the set of abducibles Σ with the signature of the original input ontology.

At the level of TBoxes, we use the standard, semantics-preserving translation from $\mathcal{EL}$ to FOL [3].

Definition 5 (TBox translation). *The function π translates concepts into first-order formulas:*

$$\begin{aligned} \pi(\top, x) &= \textbf{\textit{true}} \\ \pi(A, x) &= A(x) \\ \pi(\exists r.C, x) &= \exists y.(r(x, y) \wedge \pi(C, y)) \\ \pi(C \sqcap D, x) &= \pi(C, x) \wedge \pi(D, x) \end{aligned}$$

We lift π to concept inclusions with: $\pi(C \sqsubseteq D) = \forall x.\neg\pi(C, x) \vee \pi(D, x)$. Finally, we lift π to TBoxes with:

$$\pi(\mathcal{T}) = \{\pi(C \sqsubseteq D) \mid C \sqsubseteq D \in \mathcal{T}\}.$$

If $\mathcal{T}$ is in normal form, $\pi(\mathcal{T})$ contains only axioms of the following shapes:

$$\begin{aligned}
\text{shapes from CIs with } \top : \; & \forall x.\mathbf{true} \\
& \forall x.B(x) \\
& \forall x.\neg A(x) \vee \exists y.r(x,y), \\
& \forall x.\neg(\exists y.r(x,y)) \vee B(x) \\
\text{shapes from CIs without } \top : \; & \forall x.\neg A(x) \vee B(x), \\
& \forall x.\neg A_1(x) \vee \neg A_2(x) \vee B(x), \\
& \forall x.\neg(\exists y.r(x,y) \wedge A(y)) \vee B(x), \\
& \forall x.\neg A(x) \vee \exists y.(r(x,y) \wedge B(y)).
\end{aligned}$$

In practice, since the symbol $\top$ will be preprocessed, only the four last forms are relevant in our work.

The function π is semantics-preserving in the following sense:

Theorem 6 (First-Order Semantics Preservation [3]). *For any $\mathcal{EL}$ TBox $\mathcal{T}$ and possibly complex concepts C and D, $\mathcal{T} \models C \sqsubseteq D$ if and only if $\pi(\mathcal{T}) \models \pi(C \sqsubseteq D)$.*

The translation of a TBox abduction problem, built on top of the standard TBox translation, is less straightforward.

The translation $\Pi(\mathcal{T}, C_1 \sqsubseteq C_2)$ of $\langle \mathcal{T}, \Sigma, C_1 \sqsubseteq C_2 \rangle$ to first-order logic is the Skolemization of

$$\pi(\mathcal{T} \uplus \mathcal{T}') \wedge \neg\pi(C_1 \sqsubseteq C_2')$$

where sk_0 is used as the unique fresh Skolem constant such that the Skolemization of $\neg\pi(C_1 \sqsubseteq C_2')$ results in $\{C_1(\mathrm{sk}_0), \neg C_2'(\mathrm{sk}_0)\}$, and where several other things happen that are explained in the following paragraphs.

Before Skolemization, $\mathcal{T}$ is first preprocessed by replacing every occurrence of the concept $\top$ in $\mathcal{T}$ with that of the fresh atomic concept $A_\top$ and we add $\exists r.A_\top \sqsubseteq A_\top$ and $B \sqsubseteq A_\top$ to $\mathcal{T}$ for every role r and atomic concept B. This simulates for $A_\top$ the implicit property that $C \sqsubseteq \top$ holds for any C no matter what the TBox is. In particular, this ensures that whenever there is a positive prime implicate $B(t)$ or $r(t, t_1)$, $A_\top(t)$ also becomes a prime implicate.

Finally, we define $\mathcal{T}'$ by renaming all atomic concepts A from $\mathcal{T}$ using fresh symbols A'. Notice that this is also done for C_2, renamed to C_2' in $\Pi(\mathcal{T}, C_1 \sqsubseteq C_2)$. Thanks to this encoding trick, the inferences can be traced back from $C_1(\mathrm{sk}_0)$ or from $\neg C_2'(\mathrm{sk}_0)$ and do not interfere with each other, making it easier to relate ground prime implicates and concept inclusions and capturing interesting solutions that would otherwise not be found.

We now look at a complete encoding example in details.

Example 7. Suppose given $\mathcal{T}$ over the signature where $\mathsf{N_C} = \{A, B, C_1, C_2, E, F, G, H\}$ and $\mathsf{N_R} = \{r_1, r_2\}$ as follows.

$$\begin{aligned}
\mathcal{T} = \{ & C_1 \sqsubseteq \exists r_1.A,\ C_1 \sqsubseteq B,\ C_1 \sqsubseteq \exists r_2.G, \\
& \exists r_2.E \sqsubseteq B,\ \exists r_1.F \sqsubseteq H,\ B \sqcap H \sqsubseteq C_2\}
\end{aligned}$$

Consider the abduction problem $\langle \mathcal{T}, \mathsf{N_C}, C_1 \sqsubseteq C_2 \rangle$. Preprocessing the $\top$ symbol turns $\mathcal{T}$ into:

$$\begin{aligned}\mathcal{T} = \{&C_1 \sqsubseteq \exists r_1.A,\ C_1 \sqsubseteq B,\ C_1 \sqsubseteq \exists r_2.G,\\ &\exists r_2.E \sqsubseteq B,\ \exists r_1.F \sqsubseteq H,\ B \sqcap H \sqsubseteq C_2\}\\ &\uplus\{\exists r_1.A_\top \sqsubseteq A_\top,\ \exists r_2.A_\top \sqsubseteq A_\top\}\\ &\uplus\{J \sqsubseteq A_\top \mid J \in \{A, B, C_1, C_2, E, F, G, H\}\}\end{aligned}$$

and the renaming of atomic concepts produces:

$$\begin{aligned}\mathcal{T}' = \{&C_1' \sqsubseteq \exists r_1.A',\ C_1' \sqsubseteq B',\ C_1' \sqsubseteq \exists r_2.G',\\ &\exists r_2.E' \sqsubseteq B',\ \exists r_1.F' \sqsubseteq H',\ B' \sqcap H' \sqsubseteq C_2'\}\\ &\uplus\{\exists r_1.A_\top' \sqsubseteq A_\top',\ \exists r_2.A_\top' \sqsubseteq A_\top'\}\\ &\uplus\{J \sqsubseteq A_\top' \mid J \in \{A', B', C_1', C_2', E', F', G', H'\}\}\end{aligned}$$

The translation of $\mathcal{T}$ results in $\Phi_{\mathcal{T}}$, the Skolemization of $\pi(\mathcal{T})$:

$$\begin{aligned}\Phi_{\mathcal{T}} = \{&\neg C_1(x) \vee r_1(x, sk_1(x)),\ \neg C_1(x) \vee A(sk_1(x)),\\ &\neg C_1(x) \vee B(x),\\ &\neg C_1(x) \vee r_2(x, sk_2(x)),\ \neg C_1(x) \vee G(sk_2(x)),\\ &\neg r_2(x, y) \vee \neg E(y) \vee B(x),\\ &\neg r_1(x, y) \vee \neg F(y) \vee H(x),\\ &\neg B(x) \vee \neg H(x) \vee C_2(x),\\ &\uplus\{\neg r_1(x, y) \vee A_\top(y) \vee A_\top(x),\ \neg r_2(x, y) \vee A_\top(y) \vee A_\top(x)\}\\ &\uplus\{\neg J(x) \vee A_\top(x) \mid J \in \{A, B, C_1, C_2, E, F, G, H\}\}\end{aligned}$$

$\Phi_{\mathcal{T}'}$, the Skolemization of $\mathcal{T}'$, is similar except that the atomic concepts are renamed. Finally, $\Pi(\mathcal{T}, C_1 \sqsubseteq C_2) = \Phi_{\mathcal{T}} \cup \Phi_{\mathcal{T}'} \cup \{C_1(\mathrm{sk}_0), \neg C_2'(\mathrm{sk}_0)\}$.

Let us now look in more detail at the reason for replacing $\top$ with $A_\top$. First, remember that $\top$ is not considered as a part of $\mathsf{N_C}$ and represents an empty conjunction. Since it is translated to **true** in first-order logic, it never appears in a translated formula, as can be seen in the text following Definition 5. However, our technique is only looking for the ground prime implicates. There are many other non-ground prime implicates that are generated and that we only use in the hope of finally grounding them. It would not be practical to have to sort out the non-ground PIs due to the $\top$ symbol from the rest of them. Consider the following example.

Example 8. Given $\Sigma = \{A, E, F, G\}$ and the TBox

$$\begin{aligned}\mathcal{T} = \{&\top \sqsubseteq \exists r_1.F,\ C_1 \sqsubseteq \exists r_2.A,\ A \sqsubseteq \exists r_3.G,\ \exists r_2.E \sqsubseteq J_1,\ \exists r_1.J_1 \sqsubseteq J_0,\\ &\exists r_2.\top \sqsubseteq J_3,\ J_0 \sqcap J_3 \sqsubseteq C_2\}\end{aligned}$$

Some of the ground PIs of $\Pi(\mathcal{T}, C_1 \sqsubseteq C_2)$ are as follows:

- $A(sk_2(\mathrm{sk}_0))$, $F(sk_1(sk_2(\mathrm{sk}_0)))$, $G(sk_3(sk_2(\mathrm{sk}_0)))$ are positive PIs,
- $\neg E'(sk_1(sk_2(\mathrm{sk}_0))) \vee \neg A'_\top(sk_3(sk_2(\mathrm{sk}_0)))$ is a negative PI,

where the Skolem functions sk_1, sk_2, and sk_3, are from the Skolemization of r_1, r_2, and r_3 respectively in the first three axioms.

Let us take a close look at the reason why $F(sk_1(sk_2(\mathrm{sk}_0)))$ is an implicate. We observe that $A(sk_2(\mathrm{sk}_0))$ is obtained as a consequence of the presence of $C_1 \sqsubseteq \exists r_2.A$ in $\mathcal{T}$. Thanks to the addition of $A \sqsubseteq A_\top$ to $\mathcal{T}$, $A_\top(sk_2(\mathrm{sk}_0))$ is also an implicate. Finally due to the presence of the renaming of $\top \sqsubseteq \exists r_1.F$ in $\mathcal{T}$, namely $A_\top \sqsubseteq \exists r_1.F$, the clause $F(sk_1(sk_2(\mathrm{sk}_0)))$ is also an implicate. With a direct translation of $\top$, we would only be able to derive $F(sk_1(x))$ and there would be no way to obtain a ground implicate from this clause.

Similarly, it is only thanks to the presence of $\exists r_3.A_\top \sqsubseteq A_\top$ in $\mathcal{T}$ that the prime implicate $\neg E'(sk_1(sk_2(\mathrm{sk}_0))) \vee \neg A'_\top(sk_3(sk_2(\mathrm{sk}_0)))$ can be deduced.

4 Building $\mathcal{EL}$ hypotheses from prime implicates

We use the ground prime implicates of $\Phi = \Pi(\mathcal{T}, C_1 \sqsubseteq C_2)$ to produce solutions to the abduction problem $\langle \mathcal{T}, \Sigma, C_1 \sqsubseteq C_2 \rangle$ in $\mathcal{EL}$. To do this, we identify subsumers of C_1 and subsumees of C_2 that can be generated from the ground prime implicates in Φ, and that can be matched with each other based on the Skolem terms they contain.

Definition 9 (Solution reconstruction). *Let $\langle \mathcal{T}, \Sigma, C_1 \sqsubseteq C_2 \rangle$ be an abduction problem and let $\Phi = \Pi(\mathcal{T}, C_1 \sqsubseteq C_2)$, and assume there are prime implicates*

$$\{A_1(t_1), \ldots, A_n(t_n)\} \subseteq \mathcal{PI}^{g+}_\Sigma(\Phi) \text{ and } \neg B'_1(t'_1) \vee \ldots \vee \neg B'_m(t'_m) \in \mathcal{PI}^{g-}_\Sigma(\Phi)$$

such that $\{t_i\}_{i\in[1,n]} = \{t'_i\}_{i\in[1,m]}$. We can then obtain a hypothesis $\mathcal{H}$ for the abduction problem

$$\mathcal{H} = \{l_1(t) \sqsubseteq l_2(t) \mid t \in L \text{ and } \mathcal{T} \not\models l_1(t) \sqsubseteq l_2(t)\},$$

such that

- $L = \{t_i\}_{i\in[1,n]} = \{t'_i\}_{i\in[1,m]}$,
- $l_1(t) = \bigsqcap\{A \mid A(t) \in \{A_1(t_1), \ldots, A_n(t_n)\}\}$, *and*
- $l_2(t) = \bigsqcap\{B \mid B'(t) \in \{B'_1(t'_1), \ldots, B'_m(t'_m)\}\}$.

We denote by $\mathcal{S}(\Phi, \Sigma)$ the set containing all such $\mathcal{H}$.

Example 10. From Example 7,

$$\begin{aligned}\mathcal{T} = \{\ & C_1 \sqsubseteq \exists r_1.A,\ C_1 \sqsubseteq B,\\ & C_1 \sqsubseteq \exists r_2.F,\ B \sqcap \exists r_1.E \sqsubseteq C_2,\\ & \exists r_1.A_\top \sqsubseteq A_\top,\ \exists r_2.A_\top \sqsubseteq A_\top\}\\ & \uplus\{J \sqsubseteq A_\top \mid J \in \{C_1, C_2, A, B, E, F, G\}\}.\end{aligned}$$

Consider the prime implicates from $\Pi(\mathcal{T}, C_1 \sqsubseteq C_2)$

– $\{B(\mathrm{sk}_0), A(sk_1(\mathrm{sk}_0))\} \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, and
– $\neg B'(\mathrm{sk}_0) \vee C'(sk_1(\mathrm{sk}_0)) \in \mathcal{PI}^{g-}_{\Sigma}(\Phi)$.

From the Skolem term sk_0 and $sk_1(\mathrm{sk}_0)$, we have $l_1(\mathrm{sk}_0) \sqsubseteq l_2(\mathrm{sk}_0) = B \sqsubseteq B$ and $l_1(sk_1(\mathrm{sk}_0)) \sqsubseteq l_2(sk_1(\mathrm{sk}_0)) = A \sqsubseteq C$ respectively. However, $B \sqsubseteq B$ is a tautology. Consequently, $\mathcal{H} = \{A \sqsubseteq C\}$ is one possible hypothesis.

Theorem 11. *Given a TBox abduction problem $\langle \mathcal{T}, \Sigma, C_1 \sqsubseteq C_2 \rangle$, and denoting Φ the set $\Pi(\mathcal{T}, C_1 \sqsubseteq C_2)$, the set $\mathcal{S}(\Phi, \Sigma)$ contains only hypotheses for the abduction problem.*

Proof. Let $\Phi_{\mathcal{T}}$ denote the Skolemization of $\pi(\mathcal{T})$. If $\neg B'_1(t'_1) \vee \ldots \vee \neg B'_m(t'_m) \in \mathcal{PI}^{g-}_{\Sigma}(\Phi)$, then $\Phi_{\mathcal{T}} \cup \{C_1(\mathrm{sk}_0), \neg C_2(\mathrm{sk}_0)\} \models \neg B_1(t_1) \vee \ldots \vee \neg B_m(t_m)$ because using, e.g., resolution, any inference to derive the primed disjunction can be imitated without primes. Moreover, $A(t) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ implies $A(t) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi_{\mathcal{T}} \cup \{C_1(\mathrm{sk}_0)\})$ since only non-primed clauses are relevant to entail positive literals. Thus $\Phi_{\mathcal{T}} \cup \{C_1(\mathrm{sk}_0)\} \models A_1(t_1) \wedge \ldots \wedge A_n(t_n)$.

By combining the two results for compatible positive and negative clauses, we obtain $\Phi_{\mathcal{T}} \cup \{C_1(\mathrm{sk}_0), \neg C_2(\mathrm{sk}_0)\} \models \pi(l_1(t), t) \wedge \neg\pi(l_2(t), t)$ for all $t \in L$ where L, $l_1(t)$ and $l_2(t)$ are as defined in Definition 9. In turn, this implies $\Phi_{\mathcal{T}} \cup \{C_1(\mathrm{sk}_0), \neg C_2(\mathrm{sk}_0)\} \models \exists x.(\pi(l_1(t), x) \wedge \neg\pi(l_2(t), x))$ for all $t \in L$. Thus $\Phi_{\mathcal{T}} \cup \{C_1(\mathrm{sk}_0), \neg C_2(\mathrm{sk}_0)\} \cup \{\forall x.(\neg\pi(l_1(t), x) \vee \pi(l_2(t), x))$ for all $t \in L\}$. Hence, for any hypothesis $\mathcal{H} \in \mathcal{S}(\Phi, \Sigma)$, we know that $\Phi_{\mathcal{T}} \cup \{C_1(\mathrm{sk}_0), \neg C_2(\mathrm{sk}_0)\} \cup \pi(\mathcal{H})$ is unsatisfiable. Consequently, because Skolemization is satisfiability-preserving, $\pi(\mathcal{T}) \cup \neg\pi(C_1 \sqsubseteq C_2) \cup \pi(\mathcal{H})$ is also unsatisfiable, hence $\pi(\mathcal{T}) \cup \pi(\mathcal{H}) \models \pi(C_1 \sqsubseteq C_2)$. By Theorem 6, this in turn means that $\mathcal{T} \cup \mathcal{H} \models C_1 \sqsubseteq C_2$. □

5 Conclusion

We have introduced a technique to compute hypotheses to solve TBox abduction problems in $\mathcal{EL}$. This technique turns the input TBox into a first-order formula, of which the ground prime implicates can be used to build hypotheses answering the problem.

We have not shown the prime implicate computation step itself for two reasons. First, existing tools such as SOLAR [23] and GPiD [12] can compute prime implicates automatically in first-order logic. Second, there is an issue with the termination of such tools for problems with cyclic dependencies in the CIs, e.g., such that $\mathcal{T} \models A \sqsubseteq \exists r.A$, because in that case, the set of prime implicates may be unbounded. However, there exists only a finite number of hypotheses to be found and we conjecture that prime implicates with polynomial nesting depth of Skolem terms are sufficient for finding all relevant hypotheses. We plan to verify this conjecture and implement a prime implicate generation tool on top of SPASS [29] as the next step of this work.

An orthogonal concern that we are currently addressing is to relate the obtained hypotheses with connection-minimality [14], a notion we recently introduced to characterize solutions to the abduction problem in $\mathcal{EL}$ that we find

particularly interesting, because they connect the subsumers of C_1 to the subsumees of C_2 in a less contrived way than other solutions. We believe that the present technique is sound and complete with respect to finding connection-minimal hypotheses, but that remains to be formally proved.

Finally, as our approach works by generating ground prime implicates starting from a set of ground literals, it should also be possible to use it for a mix of ABox and TBox abduction, where our background knowledge consists of a DL knowledge base containing both a TBox and ground facts, the observation is another set of ground facts, and we are looking for hypotheses in the form of a TBox. This would have applications for TBox learning from data.

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.) Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12699, pp. 291–308. Springer (2021). https://doi.org/10.1007/978-3-030-79876-5_17
3. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press (2017). https://doi.org/10.1017/9781139025355
4. Baader, F., Suntisrivaraporn, B.: Debugging SNOMED CT using axiom pinpointing in the description logic $\mathcal{EL}^+$. In: Proc. of the 3rd Conference on Knowledge Representation in Medicine (KR-MED'08): Representing and Sharing Knowledge Using SNOMED. CEUR-WS, vol. 410 (2008), http://ceur-ws.org/Vol-410/Paper01.pdf
5. 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
6. 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
7. 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
8. 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
9. 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

10. 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
11. 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
12. Echenim, M., Peltier, N., Sellami, Y.: A generic framework for implicate generation modulo theories. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10900, pp. 279–294. Springer (2018). https://doi.org/10.1007/978-3-319-94205-6_19, https://doi.org/10.1007/978-3-319-94205-6_19
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.: Introducing connection-minimal abduction for $\mathcal{EL}$ ontologies. XLoKR workshop (2021)
15. 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
16. 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
17. Horridge, M., Parsia, B., Sattler, U.: Explanation of OWL entailments in protege 4. In: Bizer, C., Joshi, A. (eds.) Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008), Karlsruhe, Germany, October 28, 2008. CEUR Workshop Proceedings, vol. 401. CEUR-WS.org (2008), http://ceur-ws.org/Vol-401/iswc2008pd_submission_47.pdf
18. Kazakov, Y., Klinov, P., Stupnikov, A.: Towards reusable explanation services in protege. In: Artale, A., Glimm, B., Kontchakov, R. (eds.) Proceedings of the 30th International Workshop on Description Logics, Montpellier, France, July 18-21, 2017. CEUR Workshop Proceedings, vol. 1879. CEUR-WS.org (2017), http://ceur-ws.org/Vol-1879/paper31.pdf
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.: 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, Virtual Event / Montreal, Canada, 19-27 August 2021. pp. 1929–1935. ijcai.org (2021). https://doi.org/10.24963/ijcai.2021/266
21. Koopmann, P.: Two ways of explaining negative entailments in description logics using abduction. In: Explainable Logic-Based Knowledge Representation (XLoKR 2021) (2021)
22. 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

23. Nabeshima, H., Iwanuma, K., Inoue, K., Ray, O.: SOLAR: an automated deduction system for consequence finding. AI Commun. **23**(2-3), 183–203 (2010). https://doi.org/10.3233/AIC-2010-0465, https://doi.org/10.3233/AIC-2010-0465
24. 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
25. 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
26. 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
27. 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
28. 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
29. Weidenbach, C., Schmidt, R.A., Hillenbrand, T., Rusev, R., Topic, D.: System description: Spass Version 3.0. In: Pfenning, F. (ed.) Automated Deduction - CADE-21, 21st International Conference on Automated Deduction, Bremen, Germany, July 17-20, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4603, pp. 514–520. Springer (2007). https://doi.org/10.1007/978-3-540-73595-3_38, https://doi.org/10.1007/978-3-540-73595-3_38