Uniform Interpolation and Forgetting for $\mathcal{ALC}$ Ontologies with ABoxes
Long Version

Patrick Koopmann and Renate A. Schmidt
School of Computer Science, The University of Manchester,
Oxford Road, Manchester M13 9PL, United Kingdom

Abstract

Uniform interpolation and the dual task of forgetting restrict the ontology to a specified subset of concept and role names. This makes them useful tools for ontology analysis, ontology evolution and information hiding. Most previous research focused on uniform interpolation of TBoxes. However, especially for applications in privacy and information hiding, it is essential that uniform interpolation methods can deal with ABoxes as well. We present the first method that can compute uniform interpolants of any $\mathcal{ALC}$ ontology with ABoxes. ABoxes bring their own challenges when computing uniform interpolants, possibly requiring disjunctive statements or nominals in the resulting ABox. Our method can compute representations of uniform interpolants in $\mathcal{ALCO}$. An evaluation on realistic ontologies shows that these uniform interpolants can be practically computed, and can often even be presented in pure $\mathcal{ALC}$.

Introduction

Ontologies are knowledge bases that are used in diverse applications ranging from medicine, bio-informatics and software development to the semantic web. They are used to define and store conceptual information about a domain of interest, and are usually represented using a description logic. Often, an ontology consists of a TBox and an ABox. The TBox, containing terminological information, defines concepts and relations. The ABox, containing factual information, uses these defined concepts and relations to make assertions about individuals.

Uniform interpolation allows for information that is irrelevant for a new context, or that should be hidden from certain users, to be removed from an ontology. This is done by restricting the set of concept and relation symbols that occur in the ontology, preserving all entailments of the original ontology that are expressible in the restricted signature. An alternative view of uniform interpolation is forgetting. The aim of forgetting is to eliminate a set of concept and role symbols from an ontology in such a way that all entailments in the remaining signature are preserved.

Uniform interpolation has numerous applications, of which we give some examples; more examples can be found in Lutz and Wolter (2011) and Ludwig and Konev (2014). **Ontology Reuse.** Often, only a subset of the vocabulary of an existing ontology is relevant for a particular application. Uniform interpolants can be used to extract small subsets of existing ontologies to reuse them in specialized contexts. **Ontology Analysis.** By computing a restricted view that uses only a limited set of symbols of interest, hidden relations between concepts and individuals are made explicit (Konev, Walther, and Wolter 2009). **Logical Difference.** Applying changes to an ontology can lead to undesired new entailments regarding already defined concepts. The new entailments in the common signature of two ontologies are referred to as logical difference (Konev, Walther, and Wolter 2008). The logical difference can easily be computed by checking for entailment of the axioms of the respective uniform interpolants (Ludwig and Konev 2014). **Information Hiding.** In applications where ontologies are accessed by multiple users, it is critical that confidential information is sufficiently protected (Grau 2010). This can be solved by sharing a uniform interpolant of the original ontology, where confidential concepts and relations have been eliminated.

Given its importance for all these applications, uniform interpolation has recently gained a lot of attention in the literature. First methods for simpler description logics such as DL-Lite (Wang et al. 2010) and $\mathcal{EL}$ (Konev, Walther, and Wolter 2009; Lutz, Seylan, and Wolter 2012; Nikitina and Rudolph 2014), as well as for more expressive ones such as $\mathcal{ALC}$ (Wang et al. 2014; Ludwig and Konev 2014; Koopmann and Schmidt 2013c), $\mathcal{ALCH}$ (Koopmann and Schmidt 2013a) and even $\mathcal{SHQ}$ (Koopmann and Schmidt 2014a) have been devised. However, none of the methods for the more expressive description logics works without limitations if ABoxes are involved.

Especially for privacy and information hiding applications, we believe support for ABoxes is important if uniform interpolation is to be used effectively. In a lot of cases, confidential information will be stored as facts in ABoxes or databases used in connection with ontologies. For instance, Grau (2010) mentions shared patient data records, for which hiding information is indispensable in order to preserve the privacy of the patients. So far no method is able to deal with these situations properly, if the data are to be shared.

The first method for uniform interpolation in $\mathcal{ALC}$, presented in Wang et al. (2009), already considers ontologies with ABoxes. This method is based on computing the disjunctive normal form of its input, which makes it unpractical for large ontologies. Later methods for expressive description logics presented in Ludwig and Konev (2014) and Koopmann and Schmidt (2013c; 2013a; 2014a) employed saturation based reasoning techniques to achieve practicality, but only apply to TBoxes. Moreover, it turns out that the original approach by Wang et al. (2009) cannot compute the right uniform interpolant for all ABoxes. In order to preserve all entailments in the desired signature of an ontology with ABox, it is necessary to use a more expressive description logic than $\mathcal{ALC}$ for the uniform interpolant.

Uniform interpolation and forgetting are tasks much more difficult than standard reasoning tasks. Not all uniform interpolants can be finitely represented using standard description logics, and it has been shown that the size of a uniform interpolant of a TBox can be triple exponential with respect to the input TBox, if represented in $\mathcal{ALC}$ (Lutz and Wolter 2011). For $\mathcal{ALC}$ TBoxes, finite representations can be obtained by extending the underlying description logic with fixpoint expressions (using $\mathcal{ALC}\nu$), or by allowing additional symbols in the uniform interpolant (Koopmann and Schmidt 2013c). This is however not sufficient if ABoxes are involved, as our results show. If we want to preserve all entailments in the desired signature, we have to represent the uniform interpolant in an extended language such as $\mathcal{ALC}\nu$ with disjunctive ABoxes or $\mathcal{ALCO}\nu$.

The method presented in this paper is based on a method for $\mathcal{ALCH}$ TBoxes from Koopmann and Schmidt (2013a), which is extended in non-trivial ways. First, we extend the calculus with unification based reasoning. As a by-product, we develop a decision procedure for $\mathcal{ALC}\nu$ ontologies with disjunctive ABoxes. Second, in order to represent the result in $\mathcal{ALCO}$ with classical ABoxes, we devise a method to approximate disjunctive $\mathcal{ALC}$ ABoxes into classical $\mathcal{ALCO}$ ABoxes in the same signature that preserve all classical $\mathcal{ALC}$ entailments. This method is also of interest for applications other than uniform interpolation.

To summarize, the contributions of this work are the following: (1) We define a new resolution-based decision procedure for $\mathcal{ALC}\nu$ ontologies with disjunctive ABoxes. (2) Based on this procedure, we define a method to compute uniform interpolants of $\mathcal{ALC}\nu$ ontologies with disjunctive ABoxes. This method is both able to forget concept and role symbols. By using helper concepts, these uniform interpolants can be represented in pure $\mathcal{ALC}$ with disjunctive ABoxes. (3) We define a method for efficiently transforming $\mathcal{ALC}$ ontologies with disjunctive ABoxes into classical $\mathcal{ALCO}$ ontologies that preserve all entailments in $\mathcal{ALC}$. (4) Based on these methods, we develop the first method that is able to compute uniform interpolants of all $\mathcal{ALC}$ ontologies with ABoxes, and represent them as $\mathcal{ALCO}$ ontologies. (5) We evaluated the method on realistic ontologies, showing that it is indeed practical, and that in most cases even $\mathcal{ALC}$ is sufficient to represent uniform interpolants of $\mathcal{ALC}$ ontologies with ABoxes.

Detailed proofs of all theorems can be found in the appendix. A preliminary version was presented at the 2014 Description Logic Workshop (Koopmann and Schmidt 2014b).

Description Logics

In this section, we recall the description logics $\mathcal{ALC}$ and $\mathcal{ALCO}$, and introduce $\mathcal{ALC}\nu$ with disjunctive ABoxes. Let N_c, N_r, N_i and N_v be pairwise disjoint sets of *concept symbols*, *role symbols*, *individuals* and *concept variables*. An $\mathcal{ALC}$ *concept* is an expression of the form A, $\neg C$, $C \sqcup D$, $C \sqcap D$, $\exists r.C$, $\forall r.C$, where $A \in N_c$, $r \in N_r$ and C, D are $\mathcal{ALC}$ concepts. An $\mathcal{ALC}$ *TBox* is a finite set of *general concept inclusion axioms* (GCIs) of the form $C \sqsubseteq D$, where C, D are $\mathcal{ALC}$ concepts. A *classical* $\mathcal{ALC}$ *ABox* is a set of *concept assertions* of the form $C(a)$, and *role assertions* of the form $r(a, b)$, where C is any $\mathcal{ALC}$ concept, $r \in N_r$ and $a, b \in N_i$. We refer to GCIs, concept assertions and role assertions collectively as *axioms*. A *classical* $\mathcal{ALC}$ *ontology* is a tuple $\langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is an $\mathcal{ALC}$ TBox and $\mathcal{A}$ an $\mathcal{ALC}$ ABox. The semantics of $\mathcal{ALC}$ is defined as usual (see, e.g., Baader and Nutt (2007)). We write $\mathcal{O} \models \alpha$, where $\mathcal{O}$ is an ontology and α any GCI, concept assertion or role assertion, to denote that α is true in every model of $\mathcal{O}$.

A *greatest fixpoint* is a concept of the form $\nu X.C[X]$, where $X \in N_v$ and $C[X]$ is a concept in which X occurs as a concept symbol, but only positively, e.g., under an even number of negations. $\mathcal{ALC}\nu$ extends $\mathcal{ALC}$ with greatest fixpoints, which are only allowed to occur positively in concept assertions and on the right hand side of GCIs. Due to this condition, least fixpoints cannot be equivalently expressed in $\mathcal{ALC}\nu$. Intuitively, $\nu X.C[X]$ represents the most general concept C_ν, with respect to the concept inclusion relation, for which $C_\nu \equiv C[C_\nu]$ holds, where $C[C_\nu]$ is the result of replacing X in $C[X]$ by C_ν. For a formal definition of the semantics of fixpoint expressions, we refer to Calvanese, De Giacomo, and Lenzerini (1999).

A *disjunctive* $\mathcal{ALC}\nu$ *ABox* is a classical $\mathcal{ALC}\nu$ ABox that additionally contains *disjunctive concept assertions* of the form $C_1(a_1) \vee \ldots \vee C_n(a_n)$, where for $1 \leq i \leq n$, $a_i \in N_i$ and C_i is any $\mathcal{ALC}\nu$ concept. The semantics is as expected: $\mathcal{O} \models C_1(a_1) \vee \ldots \vee C_n(a_n)$ iff $\mathcal{O} \models C_i(a_i)$ for some $i \in \{1, \ldots, n\}$. A *general* $\mathcal{ALC}$ *(*$\mathcal{ALC}\nu$*) ontology* is a tuple $\langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{A}$ is a disjunctive ABox.

The description logics $\mathcal{ALCO}$ and $\mathcal{ALCO}\nu$ extend respectively $\mathcal{ALC}$ and $\mathcal{ALC}\nu$ with *nominal concepts* of the form $\{a\}$, $a \in N_i$. They allow to reference specific individuals in concepts and can be used in any combination with the other operators. For the semantics of $\mathcal{ALCO}$ and $\mathcal{ALCO}\nu$, we again refer to Baader and Nutt (2007) and Calvanese, De Giacomo, and Lenzerini (1999).

Uniform Interpolation

We now define $\mathcal{ALC}$ uniform interpolants formally. A *signature* is any subset $\mathcal{S}$ of $N_c \cup N_r$. The signature *sig*(E) denotes the concept and role symbols occurring in E, where E ranges over concepts, axioms and ontologies.

Definition 1. *Let $\mathcal{O}$ be a classical $\mathcal{ALC}$ ontology and $\mathcal{S}$ a signature. An ontology $\mathcal{O}^{\mathcal{S}}$ is an $\mathcal{ALC}$* uniform interpolant *of $\mathcal{O}$* for *$\mathcal{S}$ iff the following conditions hold:*

1. *sig*$(\mathcal{O}^{\mathcal{S}}) \subseteq \mathcal{S}$.
2. *For any $\mathcal{ALC}$ axiom α with sig$(\alpha) \subseteq \mathcal{S}$, $\mathcal{O}^{\mathcal{S}} \models \alpha$ iff $\mathcal{O} \models \alpha$.*

Note that we do not require an $\mathcal{ALC}$ uniform interpolant $\mathcal{O}^{\mathcal{S}}$ to be itself a classical $\mathcal{ALC}$ ontology. In particular, $\mathcal{O}^{\mathcal{S}}$ can also be an $\mathcal{ALCO}\nu$ ontology or an $\mathcal{ALC}\nu$ ontology with disjunctive concept assertions.

Before we describe our method for computing $\mathcal{ALC}$ uniform interpolants, we start in a generalized setting, namely, general $\mathcal{ALC}\nu$ uniform interpolants.

Definition 2. *Let $\mathcal{O}$ be a general $\mathcal{ALC}\nu$ ontology and $\mathcal{S}$ a signature. $\mathcal{O}^{\mathcal{S}}$ is a* general $\mathcal{ALC}\nu$ uniform interpolant *of $\mathcal{O}$ for $\mathcal{S}$, iff*

1. *sig$(\mathcal{O}^{\mathcal{S}}) \subseteq \mathcal{S}$ and*
2. *For any $\mathcal{ALC}\nu$ axiom or disjunctive concept assertion α with sig$(\alpha) \subseteq \mathcal{S}$, $\mathcal{O}^{\mathcal{S}} \models \alpha$ iff $\mathcal{O} \models \alpha$.*

It is easy to verify that the conditions in Definition 2 imply those in Definition 1. The converse does not hold, since a general $\mathcal{ALC}\nu$ uniform interpolant might entail disjunctive concept assertions, which are not necessarily preserved by classical $\mathcal{ALC}$ uniform interpolants. General $\mathcal{ALC}\nu$ uniform interpolants have the nice property that they can always be represented as general $\mathcal{ALC}\nu$ ontologies themselves:

Theorem 1. *Let $\mathcal{O}$ be any $\mathcal{ALC}\nu$ ontology, and $\mathcal{S}$ any signature. Then there exists a finite general $\mathcal{ALC}\nu$ ontology $\mathcal{O}^{\mathcal{S}}$, which is a uniform interpolant of $\mathcal{O}$ for $\mathcal{S}$.*

The validity of this theorem follows from the correctness of the method that we describe in the next sections.

Normalized Ontologies

Our approach is based on a method for computing uniform interpolants of $\mathcal{ALCH}$ TBoxes, introduced in Koopmann and Schmidt (2013a). A key ingredient of this method is that TBox axioms are represented in a certain normal form. We extend this presentation with variables and constants in order to incorporate ABox axioms.

Definition 3. *Let $N_d \subseteq N_c$ be a set of specific concept symbols called* definers*. A* concept literal *is a concept of the form A, $\neg A$, $\exists r.D$ or $\forall r.D$, where $A \in N_c$, $r \in N_r$ and $D \in N_d$. An ontology $\mathcal{O}$ is in* normal form *if every axiom is a role assertion, or a* clause *of one of these two forms, where L_i is a concept literal and $a_i \in N_i$ for $1 \leq i \leq n$.*

1. TBox clause: $L_1(x) \vee \ldots \vee L_n(x)$
2. ABox clause: $L_1(a_1) \vee \ldots \vee L_n(a_n)$

We view clauses as sets of literals, that is, they do not have duplicate literals and their order is not important. Furthermore, every clause is allowed to contain maximally one literal of the form $\neg D(x)$ and no literal of the form $\neg D(a)$, where $D \in N_d$ and $a \in N_i$.

A TBox clause $L_1(x) \vee \ldots \vee L_n(x)$ represents the equivalent GCI $\top \sqsubseteq L_1 \sqcup \ldots \sqcup L_n$. The symbol x occurring in TBox clauses is referred to as a *variable*. Observe that one variable x is sufficient in our representation. We call elements of the set $N_i \cup \{x\}$ *terms*.

Any general $\mathcal{ALC}\nu$ ontology can be transformed into normal form using the following rules, applied from left to right, where $\mathsf{Q} \in \{\forall, \exists\}$, $D \in N_d$ is fresh, $C[X]$ contains a concept variable X, and $C[D]$ denotes the result of replacing X in $C[X]$ by D.

1. $C_1 \vee \mathsf{Q}r.C_2(t_1) \quad \Leftrightarrow C_1 \vee \mathsf{Q}r.D(t_1), \neg D(x) \vee C_2(x)$
2. $C_1 \vee \mathsf{Q}r.\nu X.C_2[X](t_1) \Leftrightarrow C_1 \vee \mathsf{Q}r.D(t_1), \neg D(x) \vee C_2[D](x)$
3. $C_1 \vee \nu X.C_2[X](t_1) \Leftrightarrow C_1 \vee D(t_1), \neg D(x) \vee C_2[D](x)$

These rules are justified by Ackermann's Lemma (Ackermann 1935) and a generalization (Nonnengart and Szałas 1995), which show that the transformation preserves the same models modulo interpretation of the definer concepts. The transformation introduces only finitely many fresh definers.

Any ontology in normal form can be converted back into a general $\mathcal{ALC}\nu$ ontology without definers by applying the rules in the other direction. This is ensured by the last conditions in Definition 3. Whereas the transformations from left to right can just be applied to concepts as they are, the transformations from right to left require $\neg D(x)$ to be the only negative occurrence of the definer D in the ontology. This is achieved by grouping TBox clauses containing the same negative literal $\neg D(x)$ into one TBox axiom $\neg D(x) \vee C$. This is possible since negative definer literals only occur in TBox clauses, and since every TBox clause contains maximally one negative definer literal.

Example 1. *Let $\mathcal{O}_1 = \{A \sqsubseteq \forall r.(B \sqcap C)$, $r(a,b)$, $s(a,b)$, $\neg(A \sqcap B)(b)\}$. The normal form of $\mathcal{O}_1$ is $\mathcal{N}_1 = \{\neg A(x) \vee (\forall r.D_1)(x)$, $\neg D_1(x) \vee B(x)$, $\neg D_1(x) \vee C(x)$, $r(a,b)$, $s(a,b)$ $\neg A(b) \vee \neg B(b)\}$.*

The set of clauses $\mathcal{N}_2 = \{B(b) \vee (\forall r.D_1)(a)$, $\neg D_1(x) \vee A(x)$, $\neg D_1(x) \vee (\exists r.D_1)(x)\}$, $D_1 \in N_d$, is transformed into the general $\mathcal{ALC}\nu$ ontology $\mathcal{O}_2 = \{B(b) \vee (\forall r.\nu X.(A \sqcap \exists r.X))(a)\}$ without definers.

Our method for forgetting concept and role symbols works on the normal form representation of the input ontology. The definers are eliminated afterwards using the right to left transformations above.

The Calculus

Uniform interpolants are computed by saturating an ontology in normal form using the rules of the calculus shown in Figure 1. Before we describe the rules, a few notions have to be introduced.

Our normal form allows for a very simple form of unification. In our setting, given two terms t_1 and t_2, the *unifier* of t_1 and t_2 is a substitution that replaces t_1 by t_2, or vice versa. Two terms t_1 and t_2 only have a unifier if $t_1 = x$, $t_2 = x$ or $t_1 = t_2$. For example, the terms a and b do not have a unifier, and the unifier of a and x is $\sigma = [x \mapsto a]$. Applied to a clause $C = A(x) \vee B(x)$, this unifier produces the clause $C\sigma = A(a) \vee B(a)$.

To preserve the normal form, the calculus introduces new definers dynamically. More specifically, given two definers D_1 and D_2, a definer D_{12} representing $D_1 \sqcap D_2$ is introduced by adding the two clauses $\neg D_{12}(x) \vee D_1(x)$ and $\neg D_{12}(x) \vee D_2(x)$. These two clauses correspond to the TBox axiom $D_{12} \sqsubseteq D_1 \sqcap D_2$. New definers are only introduced if necessary. By reusing already introduced definers, we maximally introduce 2^n many new definers, where n

Resolution

$$\frac{C_1 \vee A(t_1) \quad C_2 \vee \neg A(t_2)}{(C_1 \vee C_2)\sigma}$$

Role Propagation

$$\frac{C_1 \vee (\forall r.D_1)(t_1) \quad C_2 \vee (\mathsf{Q}r.D_2)(t_2)}{(C_1 \vee C_2)\sigma \vee \mathsf{Q}r.D_{12}(t_1\sigma)}$$

Existential Role Restriction Elimination

$$\frac{C \vee (\exists r.D)(t) \quad \neg D(x)}{C}$$

Role Instantiation

$$\frac{C_1 \vee (\forall r.D)(t_1) \quad r(t_2, b)}{C_1\sigma \vee D(b)}$$

where $\mathsf{Q} \in \{\exists, \forall\}$, σ is the unifier of t_1 and t_2 if it exists, D_{12} is a possibly new definer representing $D_1 \sqcap D_2$ and $C_1 \vee C_2$ contains maximally one literal of the form $\neg D(x)$ and no literal of the form $\neg D(a)$.

Figure 1: The rules of the calculus.

is the number of definers in the normalized input. This results in a double exponential bound on the number of derived clauses, and guarantees termination of our method.

The first three rules in Figure 1 are generalizations of the rules used in Koopmann and Schmidt (2013a). Whereas the original calculus has only rules for TBox clauses, we extend them using unification to make them applicable for cases where the premises contain one or more ABox clauses. In addition, since an ABox can contain role assertions, we need a rule that propagates information for universal role restrictions $\forall r.D$ along role assertions. This is achieved by the role instantiation rule. Observe that, for a role assertion $r(a, b)$, the rule is only applicable to clauses of the form $C \vee (\forall r.D)(x)$ or $C \vee (\forall r.D)(a)$.

The side conditions of the rules ensure that only clauses in the normal form are derived. This way, we ensure that any derived set of clauses can be transformed back into an $\mathcal{ALC}\nu$ ontology without definers. Clauses of the form $\neg D(a) \vee C$ do not need to be derived, as is shown in the correctness proofs in the appendix.

Example 2. *Take the clause set $\mathcal{N}_1$ from Example 1. We can apply role instantiation on $\neg A(x) \vee (\forall r.D_1)(x)$ and $r(a, b)$, using the unifier $[x \mapsto a]$, and infer the clause $\neg A(a) \vee D_1(b)$. With the same unifier, we can apply resolution on $D_1(b)$ in this clause and derive $\neg A(a) \vee B(b)$ and $\neg A(a) \vee C(b)$. Resolution on $\neg A(a) \vee B(b)$ and $\neg A(b) \vee \neg B(b)$ derives $\neg A(a) \vee \neg A(b)$, where the unifier is $[b \mapsto b]$.*

Theorem 2. *For any general $\mathcal{ALC}\nu$ ontology $\mathcal{O}$, $\mathcal{O}$ is unsatisfiable iff the empty clause can be derived in finitely many steps from its normal form representation using the rules of the calculus.*

Proof (sketch). Termination follows from the fact that, starting from a set $\mathcal{N}$ of n $\mathcal{ALC}$ clauses, maximally $O(2^{2^n})$ many clauses can be derived. A derivation of the empty clause embodies a direct contradiction. If the empty clause cannot be derived, we can adapt the model construction used in Koopmann and Schmidt (2013c) to build a model based on the saturated set of clauses. Consequently, $\mathcal{N}$ has a model and is satisfiable. $\square$

Uniform Interpolation in Normal Form

Let $\mathcal{N}$ be any ontology in normal form, $\mathcal{S}$ any signature, and $\mathcal{N}^*$ the saturation of $\mathcal{N}$ using the rules of the calculus. The *clausal representation $\mathcal{N}^{\mathcal{S}}$ of the uniform interpolant of $\mathcal{N}$ for $\mathcal{S}$* is the smallest set of clauses $C \in \mathcal{N}^*$ with $sig(C) \subseteq \mathcal{S} \cup N_d$ that satisfy at least one of the following conditions:

1. $C \in \mathcal{N}$.
2. C is the result of applying a rule on a literal that is not in $\mathcal{S} \cup N_d$.
3. C contains a definer D that occurs in another clause in $\mathcal{N}^{\mathcal{S}}$.

It can be shown that every entailment α of $\mathcal{N}$ with $sig(\alpha) \subseteq \mathcal{S}$ is also entailed by $\mathcal{N}^{\mathcal{S}}$. Condition 1 ensures that we keep all clauses that were in the desired signature from the beginning. Condition 2 ensures that we preserve all possible inferences in $\mathcal{S}$ that involve symbols outside $\mathcal{S}$. It is possible that these inferences are made possible by applications of the role propagation rule. If this is the case, $\mathcal{N}^{\mathcal{S}}$ contains introduced definers. This is taken care of by Condition 3, which ensures that $\mathcal{N}^{\mathcal{S}}$ is closed under introduced definers. To be more specific, any existential or universal restrictions that refer to these introduced definers, and any clauses of the form $\neg D_{12}(x) \vee C(x)$ that are necessary to preserve the meaning of D_{12}, belong to $\mathcal{N}^{\mathcal{S}}$. This ensures that all entailments in $\mathcal{S}$ are still preserved after the elimination of all definers using the normal form transformation rules described in an earlier section.

Given any general $\mathcal{ALC}\nu$ ontology $\mathcal{O}$ and any signature $\mathcal{S}$, this is how a general $\mathcal{ALC}\nu$ uniform interpolant $\mathcal{O}^{\mathcal{S}}$ is computed: (1) $\mathcal{O}$ is transformed into a set $\mathcal{N}$ of clauses in normal form. (2) For $\mathcal{N}$, we compute the set $\mathcal{N}^{\mathcal{S}}$ defined above using the rules of the calculus. (3) Finally, we eliminate all definers by applying the normal form transformation rules from right to left. We have the following theorem.

Theorem 3. *Let $\mathcal{O}$ be any general $\mathcal{ALC}\nu$ ontology and $\mathcal{S}$ any signature. The described method always terminates and the returned ontology, $\mathcal{O}^{\mathcal{S}}$, is a general $\mathcal{ALC}\nu$ uniform interpolant of $\mathcal{O}$ for $\mathcal{S}$. Furthermore, $\mathcal{O}^{\mathcal{S}}$ is in the worst case of size $O(2^{2^n})$, where n is the size of $\mathcal{O}$.*

Example 3. *Let $\mathcal{O}_1$ and $\mathcal{N}_1$ be is as in Example 1 and $\mathcal{S} = \{A, C, r, s\}$. We already computed all inferences for $\mathcal{N}_1$ in Example 2. Following the above conditions, we have that the clausal representation of the uniform interpolant of $\mathcal{O}_1$ for $\mathcal{S}$ is $\mathcal{N}_1^{\mathcal{S}} = \{\neg A(x) \vee (\forall r.D_1)(x), \neg D_1(x) \vee C(x), r(a, b), s(a, b), \neg A(a) \vee \neg A(b)\}$. After eliminating the only definer D_1, we obtain a uniform interpolant of $\mathcal{O}_1$ for $\mathcal{S}$, which is $\mathcal{O}_1^{\mathcal{S}} = \{A \sqsubseteq \forall r.C, r(a, b), s(a, b), \neg A(a) \vee \neg A(b)\}$.*

Representing the Result in $\mathcal{ALCO}$

Since all classical $\mathcal{ALC}$ ontologies are also $\mathcal{ALC}\nu$ ontologies, the method can be used for computing a finite uniform interpolant of any $\mathcal{ALC}$ ontology, but in an extended language, because these uniform interpolants may contain greatest fixpoint concepts and disjunctive concept assertions. These are not supported by standard description logic reasoners or the web ontology standard language OWL. For this reason, it is of interest to represent the uniform interpolant in a more common description logic.

If a definer can only be eliminated by introducing a fixpoint, we can omit this elimination and keep the corresponding cyclic definer concept. As a result, we obtain an ontology that is not completely in the desired signature, but preserves all entailments we are interested in. The cyclic definers that stay in the ontology can be seen as helper concepts that "simulate" the greatest fixpoints and make a finite representation possible, despite cycles in the TBox. For applications that require the uniform interpolant to be completely in the desired signature, e.g. logical difference, the fixpoint can be approximated by bounded iterative unfolding, as described in Koopmann and Schmidt (2013c).

Another problem is illustrated by the uniform interpolant $\mathcal{O}_1^{\mathcal{S}}$ computed in Example 3. Due to the clause $\neg A(a) \vee \neg A(b)$, and because a and b are connected by the two role assertions $r(a,b)$ and $s(a,b)$, one can verify that $\mathcal{O}_1^{\mathcal{S}} \models (\neg A \sqcup \exists r.(\neg A \sqcap E) \sqcup \exists s.(\neg A \sqcap \neg E))(a)$, for any $\mathcal{ALC}$ concept E. These cannot be captures by a finite classical $\mathcal{ALC}$ ontology. However, in $\mathcal{ALCO}$, we can capture them by the concept assertion $(\neg A \sqcup \exists r.(\neg A \sqcap \{b\}))(a)$, taking into account the role assertions $r(a,b)$ and $s(a,b)$ in the uniform interpolant. (Another solution, using a technique from Areces et al. (2003), involves the introduction of additional roles for each disjunction. But this approach unnecessarily extends the signature of the uniform interpolant.)

If all individuals occurring in an ABox clause C are connected to some root individual a via a chain of role assertions, C can be represented as a classical $\mathcal{ALCO}$ concept assertion on a in the same way as in the example. We call these concept assertions $\mathcal{ALCO}$ *convertible*. If an ABox clause C is not $\mathcal{ALCO}$ convertible, we cannot express C as a classical concept assertion, but C might still contribute to the entailment of other classical concept assertions. To compute a set of clauses that can be fully translated into $\mathcal{ALCO}$, and that preserves all entailments which are classical $\mathcal{ALC}$ axioms, we use our calculus in a similar way as for computing uniform interpolants. Let $\mathcal{N}$ be any set of clauses and $\mathcal{N}^*$ the saturation of $\mathcal{N}$. The set $\mathcal{N}^{conv}$ is the smallest set of clauses $C \in \mathcal{N}^*$ that are $\mathcal{ALCO}$ convertible and satisfy at least one of the following conditions:

1. $C \in \mathcal{N}$.
2. C is the conclusion of any rule application on a clause that is not $\mathcal{ALCO}$ convertible.

$\mathcal{N}^{conv}$ preserves all entailments of $\mathcal{N}$ that are representable as classical $\mathcal{ALC}$ axioms. $\mathcal{N}^{conv}$ can be transformed into an $\mathcal{ALCO}$ ontology without definers, and all remaining disjunctive concept assertions can be represented as classical concept assertions using nominals.

Theorem 4. *Let $\mathcal{N}$ be any ontology in normal form. Then, the described method for approximating disjunctive concept assertions computes an $\mathcal{ALC}$ or $\mathcal{ALCO}$ ontology $\mathcal{O}$ with classical ABox, and we have for any classical $\mathcal{ALC}$ axiom α without definers that $\mathcal{O} \models \alpha$ iff $\mathcal{N} \models \alpha$.*

By combining this technique with our method for computing general $\mathcal{ALC}\nu$ uniform interpolants, we can compute $\mathcal{ALC}$ uniform interpolants in $\mathcal{ALCO}\nu$ with classical ABoxes for any $\mathcal{ALC}$ input ontology. Furthermore, by keeping definers that can only be eliminated using fixpoints, we can represent all uniform interpolants as classical $\mathcal{ALCO}$ ontologies.

Evaluation

To investigate practicality of our approach, we have implemented a prototype using the OWL API[1] and some of the optimisations mentioned in Ludwig and Konev (2014) and Koopmann and Schmidt (2013b). Experiments were conducted on a set of ontologies taken from the NCBO BioPortal[2] and the Oxford Ontology[3] repositories, which we restricted to axioms fully expressible in $\mathcal{ALC}$, where domain and range restrictions were interpreted as corresponding $\mathcal{ALC}$ axioms. For a detailed description of these repositories, see Matentzoglu, Bail, and Parsia (2013). The experiments have been performed on a desktop PC with Intel Core i7 350GHz CPU and 8 GB RAM.

To obtain a set of ontologies that test the ABox processing of our method and can be evaluated in reasonable time, we selected ontologies according to the following criteria: They (1) could be downloaded and parsed by the OWL API without errors, (2) contain more ABox axioms than TBox axioms, (3) are consistent, (4) contain at least 100 TBox axioms, (5) contain at least 80% TBox axioms that are in $\mathcal{ALC}$, (6) contain at least one axiom that is in $\mathcal{ALC}$ but not in $\mathcal{EL}$, and (7) contain at most 40,000 axioms. The selected ontologies are listed in Table 1, which shows the number of TBox axioms, ABox axioms, concept symbols and role symbols for each ontology. CCON, CTX, ICPS, ICF and SSE were taken from the NCBO BioPortal repository, whereas 00104, 00596, 00597 and 00773 were taken from the Oxford Ontology repository.

For the experiments, our assumption has been that the targeted applications require either uniform interpolants for relatively big signatures (e.g., logical difference, information hiding) or for relatively small signatures (e.g., ontology analysis). We therefore generated for each ontology 350 signatures which included any concept or role symbol with a probability of 90% (forgetting about 10% of all symbols), and 350 signatures which included any concept or role symbol with a probability of 10% (forgetting about 90% of all symbols). We then computed uniform interpolants for these signatures represented as $\mathcal{ALCO}$ ontologies with classical ABoxes. For each experimental run, the timeout was

The results are shown in Table 2 and 3, where we show the number of timeouts, the average duration of each success-

[1] `http://owlapi.sourceforge.net/`

[2] `http://bioportal.bioontology.org/`

[3] `http://www.cs.ox.ac.uk/isg/ontologies/`

Ontology	TBox	ABox	Concepts	Roles
CCON	214	364	86	28
CTX	364	1,553	290	22
ICPS	953	4,254	432	135
ICF	1,991	17,223	1,596	41
SSE	267	2,323	243	18
00104	1,157	2,451	1,094	6
00596	2,257	2,658	2,023	19
00597	2,887	3,646	2,341	25
00773	581	2,334	244	83

Table 1: The input ontologies.

Ontology	Timeouts	Duration	TBox	ABox
CCON	1.1%	14.1 sec.	89.1%	92.0%
CTX	4.0%	72.6 sec.	87.5%	153.5%
ICPS	21.1%	11.1 sec.	243.2%	81.0%
ICF	0.0%	13.0 sec.	86.6%	38.1%
SSE	0.0%	1.9 sec.	85.3%	99.5%
00104	0.0%	8.2 sec.	87.9%	97.8%
00596	0.0%	10.6 sec.	85.3%	89.4%
00597	6.6%	108.8 sec.	167.6%	91.1%
00773	0.0%	3.9 sec.	109.2%	104.2%

Table 2: Uniform interpolants for symbols selected with 90% probability.

ful run, and the percentage of the average number of TBox and ABox axioms in the computed uniform interpolants in comparison to those in the input ontologies. For the uniform interpolants that included any symbol with a probability with 90%, each ABox axiom contained on average 2.6 symbols, whereas each TBox axiom contained on average 8.5 symbols (counting concept symbols, role symbols, individuals and operators). For the uniform interpolants that included any symbol with a probability of 10%, each ABox axiom contained on average 2.4 symbols and each TBox axiom 13.5 symbols. As the table shows, many uniform interpolants that included symbols with a probability of 10% only had small TBoxes. This can be explained by the small number of symbols present in these ontologies, which restricts the number of axioms that can be expressed.

Most concept assertions used just one concept symbol, whereas a few used simple concept disjunctions or role restrictions. On the other hand, the structure of the TBox axioms was affected more, especially in uniform interpolants with small signatures. Whereas a lot of axioms had simple forms such as $A \sqsubseteq B$, $A \sqsubseteq \exists r.B$ or $A \sqcup B \sqcup C \sqsubseteq \bot$, some would involve deep nestings of role restrictions. We also noticed that more complex axioms sometimes contained redundant information that was not detected by our prototype. A typical example are patterns such as $A \sqcap (\neg A \sqcup C)$, that could have been simplified using further resolution steps. In general, the majority of axioms was still

Only uniform interpolants of the ontologies CCON, 00597 and 00773 contained cyclic definers. Of their uniform interpolants, only 7.2% contained cyclic definers. Interest-

Ontology	Timeouts	Duration	TBox	ABox
CCON	0.0%	4.0 sec.	3.3%	62.5%
CTX	91.7%	607.1 sec.	3.5%	28.2%
ICPS	19.1%	28.2 sec.	68.4%	465.7%
ICF	0.0%	7.4 sec.	4.5%	11.2%
SSE	0.0%	23.4 sec.	3.3%	28.3%
00104	0.0%	2.4 sec.	3.7%	48.0%
00596	0.0%	10.4 sec.	3.5%	10.5%
00597	51.1%	705.5 sec.	179.8%	11.4%
00773	0.0%	24.9 sec.	29.1%	286.3%

Table 3: Uniform interpolants for symbols selected with 10% probability.

ingly, no uniform interpolant made use of nominals. The reason is that the ontologies of our corpus only contained few role assertions, and imposed a relatively simple structure. However, in 25.3% of cases we had to use our method for eliminating disjunctive ABox statements, which always succeeded without introducing nominals. Hence, eliminating all non $\mathcal{ALCO}$ convertible clauses quickly resulted in clause sets fully representable in $\mathcal{ALC}$.

As the tables show, the results varied considerably depending on the structure of the ontology, and the size was not the only determining factor. For example, CTX has only 290 TBox axioms, but caused in 91.7% of the cases timeouts when computing small uniform interpolants. Analysis of the signatures that caused timeouts showed that in most cases just a small number of symbols was responsible, whereas the majority of concepts could be eliminated very quickly. In practical applications, one might therefore consider to extend the desired signature by one or two symbols, determined by some heuristics, if the computation turns out to be too expensive and the application allows it.

Conclusion and Future Work

We have presented a new method for computing uniform interpolants of $\mathcal{ALC}$ ontologies with ABoxes. Though the problem is complex and theoretical results show the target language needs to be slightly more expressive, our experiments show that in most cases uniform interpolants can be computed in reasonable time and do not fall outside the boundary of $\mathcal{ALC}$.

We are currently extending the approach to more expressive description logics, allowing for role hierarchies, transitive roles, inverse roles or cardinality restrictions. Even though our prototype already uses optimisations, further optimisations would strengthen the approach further, especially for ontologies with large ABoxes, but also for computing smaller axioms.

Acknowledgments

Patrick Koopmann is supported by an EPSRC doctoral training award.

References

Ackermann, W. 1935. Untersuchungen über das Eliminationsproblem der mathematischen Logik. *Mathematische Annalen* 110(1):390–413.

Areces, C.; Blackburn, P.; Hernández, B. M.; and Marx, M. 2003. Handling Boolean ABoxes. In *Proc. DL'03*, volume 81 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Baader, F., and Nutt, W. 2007. Basic description logics. In Baader, F.; Calvanese, D.; McGuiness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds., *The Description Logic Handbook*. Cambridge University Press, second edition. chapter 2.

Calvanese, D.; De Giacomo, G.; and Lenzerini, M. 1999. Reasoning in expressive description logics with fixpoints based on automata on infinite trees. In *Proc. IJCAI '99*, 84–89. Morgan Kaufmann.

Grau, B. C. 2010. Privacy in ontology-based information systems: A pending matter. *Semantic Web* 1(1-2):137–141.

Konev, B.; Walther, D.; and Wolter, F. 2008. The logical difference problem for description logic terminologies. In *Automated Reasoning*. Springer. 259–274.

Konev, B.; Walther, D.; and Wolter, F. 2009. Forgetting and uniform interpolation in large-scale description logic terminologies. In *Proc. IJCAI '09*, 830–835. AAAI Press.

Koopmann, P., and Schmidt, R. A. 2013a. Forgetting concept and role symbols in $\mathcal{ALCH}$-ontologies. In *Proc. LPAR'13*, volume 8312 of *LNCS*, 552–567. Springer.

Koopmann, P., and Schmidt, R. A. 2013b. Implementation and evaluation of forgetting in $\mathcal{ALC}$-ontologies. In *Proc. WoMO'13*, volume 1081 of *CEUR Workshop Proceedings*, 37–48. CEUR-WS.org.

Koopmann, P., and Schmidt, R. A. 2013c. Uniform interpolation of $\mathcal{ALC}$-ontologies using fixpoints. In *Proc. FroCoS'13*, volume 8152 of *LNCS*, 87–102. Springer.

Koopmann, P., and Schmidt, R. A. 2014a. Count and forget: Uniform interpolation of $\mathcal{SHQ}$-ontologies. In *Proc. IJCAR'14*, volume 8562 of *LNCS*, 434–448. Springer.

Koopmann, P., and Schmidt, R. A. 2014b. Forgetting and uniform interpolation for $\mathcal{ALC}$-ontologies with ABoxes. volume 1193 of *CEUR Workshop Proceedings*, 245–257. CEUR-WS.org. Proc. DL'14.

Ludwig, M., and Konev, B. 2014. Practical uniform interpolation and forgetting for $\mathcal{ALC}$ TBoxes with applications to logical difference. In *Proc. KR'14*. AAAI Press.

Lutz, C., and Wolter, F. 2011. Foundations for uniform interpolation and forgetting in expressive description logics. In *Proc. IJCAI '11*, 989–995. AAAI Press.

Lutz, C.; Seylan, I.; and Wolter, F. 2012. An automata-theoretic approach to uniform interpolation and approximation in the description logic $\mathcal{EL}$. In *Proc. KR'12*, 286–296. AAAI Press.

Matentzoglu, N.; Bail, S.; and Parsia, B. 2013. A snapshot of the OWL web. In *Proc. ISWC'13*. Springer. 331–346.

Nikitina, N., and Rudolph, S. 2014. (non-) succinctness of uniform interpolants of general terminologies in the description logic $\mathcal{EL}$. *Artificial Intelligence* 215:120–140.

Nonnengart, A., and Szałas, A. 1995. A fixpoint approach to second order quantifier elimination with applications to correspondence theory. Technical report.

Wang, K.; Wang, Z.; Topor, R.; Pan, J.; and Antoniou, G. 2009. Concept and role forgetting in $\mathcal{ALC}$ ontologies. 666–681.

Wang, Z.; Wang, K.; Topor, R. W.; and Pan, J. Z. 2010. Forgetting for knowledge bases in DL-Lite. *Ann. Math. Artif. Intell.* 58(1–2):117–151.

Wang, K.; Wang, Z.; Topor, R. W.; Pan, J. Z.; and Antoniou, G. 2014. Eliminating concepts and roles from ontologies in expressive descriptive logics. *Computational Intelligence* 30(2):205–232.

Proofs of Theorems

Completeness of the Calculus

We begin by proving the refutational completeness of the calculus for general $\mathcal{ALC}\nu$ ontologies. Using the normal form transformation rules shown in the paper, we can represent any general $\mathcal{ALC}\nu$ ontology $\mathcal{O}$ as an equi-satisfiable set $\mathcal{N}$ of clauses. In order to prove refutational completeness, we have to show that, whenever an ontology is inconsistent, we can infer the empty clause from its normal form presentation using the calculus. Or, if we cannot derive the empty clause from the normal form representation of a general $\mathcal{ALC}\nu$ ontology, then $\mathcal{O}$ is consistent, and we can find a model for it.

Given a set of clauses $\mathcal{N}$, we denote by $\mathcal{N}^*$ the result of saturating this clause set using the calculus. Soundness and refutational completeness of the TBox cases of the rules has already been shown in (Koopmann and Schmidt 2013c). We extend this proof by building on their results and adapting the construction to include ABox clauses.

Koopmann and Schmidt describe a method to build a model based on a saturated set of TBox clauses. Using this construction, we can build a model of the set of TBox clauses in $\mathcal{N}^*$. Denote this model by $\mathcal{I}^\mathcal{T} = \langle \Delta^{\mathcal{I}^\mathcal{T}}, \cdot^{\mathcal{I}^\mathcal{T}} \rangle$. The model construction ensures that $\mathcal{I}^\mathcal{T}$ has the following properties:

Property 1. *Every TBox clause is satisfied: $\mathcal{I}^\mathcal{T} \models C$ iff $C \in \mathcal{N}^*$ and C is a TBox clause.*

Property 2. *Every satisfiable definer is represented using a domain element, that is, for every definer D occurring in $\mathcal{N}^*$ for which $\neg D(x) \notin \mathcal{N}^*$, there is some $x_D \in \Delta^{\mathcal{I}^\mathcal{T}}$ such that $x_D \in D^{\mathcal{I}^\mathcal{T}}$.*

We show how we can extend this model $\mathcal{I}^\mathcal{T}$ to a model $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ of the complete set $\mathcal{N}^*$, if $\bot \notin \mathcal{N}^*$. Our construction is only based on ABox clauses. To ensure that TBox clauses are satisfied as well, we extend the initial clause set by instantiating all TBox clauses for every individual.

$$\mathcal{N}^{*2} = \mathcal{N}^* \cup \{C[x \mapsto a] \mid a \in N_i, C \in \mathcal{N}^* \text{ and } C \text{ is a TBox clause}\}$$

One can verify that $\mathcal{N}^{*2}$ is still saturated. Note that we only unify two individual names if they are the same. The same holds for variables, since there is only one variable in our normal form. Accordingly, for every new pair of clauses $C_1[x \mapsto a], C_2[x \mapsto a] \in (\mathcal{N}^{2*} \setminus \mathcal{N}^*)$ on which a rule is applicable with the unifier $\sigma = [a \mapsto a]$, the same rule is applicable to C_1 and C_2 with $\sigma = [x \mapsto x]$. Accordingly, $\mathcal{N}^*$ contains the conclusion C_3. $C_3[x \mapsto a]$ is the corresponding conclusion on $C_1[x \mapsto a]$ and $C_2[x \mapsto a]$, and this clause is included in $\mathcal{N}^{*2}$ by construction.

Due to the substitution, the clauses in $\mathcal{N}^{*2}$ may contain negative definer literals of the form $\neg D(a)$, and do therefore not conform to the conditions imposed on the normal form in Definition 3. Note that resolvents with these clauses on $\neg D(a)$ are still present in $\mathcal{N}^{2*}$ as resolvents of the corresponding TBox clause, using either a unifier $\sigma = [x \mapsto a]$ or $\sigma = [x \mapsto x]$. This is the main reason why it is not necessary to derive clauses containing $\neg D(a)$ directly by the calculus.

If we extend $\mathcal{I}^\mathcal{T}$ to an interpretation $\mathcal{I}$ that satisfies all ABox clauses in $\mathcal{N}^{*2}$, $\mathcal{I}$ also satisfies all TBox clauses in $\mathcal{N}^*$, due to the following property of interpretations.

Property 3. *Let C be any TBox clause. If for all domain elements $x \in \Delta^\mathcal{T}$ we have $x \in C^\mathcal{I}$, then $\mathcal{I} \models C$.*

In order to extend the interpretation function $\cdot^{\mathcal{I}^\mathcal{T}}$ to satisfy all ABox clauses in $\mathcal{N}^{*2}$, we first introduce an ordering on ABox literals. The ordering $\prec_l$ is defined as any total ordering on literals satisfying the following constraints.

- $D \prec_l \neg D \prec_l A \prec_l \neg A \prec_l \exists r.D' \prec_l \forall r.D''$ for any $D, D', D'' \in N_d$, $A \in N_c$ and $r \in N_r$.
- If $\neg D_1(x) \vee D_2(x) \in \mathcal{N}^{*2}$, $D_1 \prec_l D_2$, $\exists r.D_1 \prec_l \exists r.D_2$ and $\forall r.D_2 \prec_l \forall r.D_1$.

Let $\prec_i$ be any ordering between individual symbols. We extend $\prec_l$ to an ordering between ABox literals by $L_1(a) \prec_l L_2(b)$, if $a \prec_i b$, and $L_1(a) \prec_l L_2(a)$, if $L_1 \prec L_2$. We say a ABox literal L is *maximal* in an ABox clause C if for all ABox literals $L' \in C \setminus \{L\}$ we have $L' \prec_l L$. $\prec_l$ is still total, which means that every clause has a unique maximal literal.

We define an ordering $\prec_c$ between ABox clauses as the multiset extension $(\prec_l)_{mul}$ of $\prec_l$, that is, $C_1 \prec_c C_2$ holds iff for every ABox literal $L_1 \in C_1$ we have $L_1 \prec_l L_2$ for some ABox literal $L_2 \in C_2$. The ordering allows to enumerate the ABox clauses in $\mathcal{N}^{*2}$. In the following, C_i denotes the ith ABox clause in $\mathcal{N}^{*2}$ following this ordering. This means, $i < k$ implies $C_i \prec_c C_k$.

We now complete the model $\mathcal{I}^\mathcal{T} = \langle \Delta^\mathcal{T}, \cdot^\mathcal{T} \rangle$ of the TBox clauses in $\mathcal{N}^{2*}$ to a model $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ of all clauses in $\mathcal{N}^{*2}$. For the domain, we add one element x_a for every individual: $\Delta^\mathcal{I} = \Delta^{\mathcal{I}^\mathcal{T}} \cup \{x_a \mid a \in N_i\}$. The interpretation function $\cdot^\mathcal{I}$ is defined incrementally as follows, where $\mathcal{I}_i = \langle \Delta^\mathcal{I}, \cdot^{\mathcal{I}_i} \rangle$.

1. $\cdot^{\mathcal{I}_0}$ is equal to $\cdot^{\mathcal{I}^\mathcal{T}}$, except that for every role $r \in N_r$: $r^{\mathcal{I}_0} = r^{\mathcal{I}^\mathcal{T}} \cup \{(x_a, x_b) \mid r(a,b) \in \mathcal{N}\}$ and $a^{\mathcal{I}_0} = x_a$ for all individuals $a \in N_i$.
2. If $\mathcal{I}_i \models C_i$, then $\cdot^{\mathcal{I}_{i+1}} = \cdot^{\mathcal{I}_i}$. Otherwise:
   (a) If the maximal literal in C_i is $A(a)$, $A \in N_c$, $\cdot^{\mathcal{I}_{i+1}}$ is equal to $\cdot^{\mathcal{I}_i}$, except that $A^{\mathcal{I}_{i+1}} = A^{\mathcal{I}_i} \cup \{x_a\}$.
   (b) If the maximal literal in C_i is $(\exists r.D)(a)$ and $x_D \in \Delta^\mathcal{I}$, $\cdot^{\mathcal{I}_{i+1}}$ is equal to $\cdot^{\mathcal{I}_i}$, except that $r^{\mathcal{I}_{i+1}} = A^{\mathcal{I}_i} \cup \{(x_a, x_D)\}$.
3. $\cdot^\mathcal{I} = \cdot^{\mathcal{I}_n}$, where n is the number of ABox clauses in $\mathcal{N}^{*2}$.

For Step 2b, observe that every definer D occurring in $\mathcal{N}^{*2}$ has a corresponding domain element $x_D \in \Delta^\mathcal{I}$, except if there is a clause $\neg D(x) \in \mathcal{N}^{*2}$ (Property 2). We first prove the monotonicity of our model construction.

Lemma 1. *For every ABox clause C_i, $\mathcal{I}_{i+1} \models C_i$ implies $\mathcal{I} \models C_i$.*

Proof. If $\mathcal{I}_{i+1} \models C_i$, there must be a literal $L \in C_i$ such that $\mathcal{I}_{i+1} \models L$. We distinguish the cases for L, and show that we always have $\mathcal{I} \models L$. (i) L is a positive literal of the form $A(a)$ or $(\exists r.D)(a)$. Since the model construction only adds values to the interpretation function, but does not remove any, $\mathcal{I} \models L$ holds. (ii) L is of the form $\neg A(a)$. $A(a)$ cannot be a positive and maximal literal in any clause C_j larger than C_i. Therefore, it is impossible that x_a is added to $A_j^{\mathcal{I}}$ for any index j larger than i, and we have $\mathcal{I} \models L$. (iii) L is of the form $(\forall r.D)(a)$. In every clause C_j with $C_i \prec C_j$, the maximal literal is also of the form $(\forall r'.D')(a')$. Therefore, nothing is added to the interpretation function for any clause C_j with $j > i$, and we have $\mathcal{I} \models L$. □

Lemma 2. *For every ABox clause C_i, $\mathcal{I}_{i+1} \not\models C_i$ implies $\mathcal{I} \not\models C_i$.*

Proof. $\mathcal{I}_{i+1} \not\models C_i$ can only be the case if the maximal literal L in C_i is of the form $\neg A(a)$ or $(\forall r.D)(a)$, since the other cases are taken care of by the model construction. We distinguish both cases. (i) L is of the form $\neg A(a)$. Then, due to the ordering, all literals in C_i are of the form $A'(a')$ or $\neg A'(a')$. The negation of none of these literals can be maximal in any clause C_j with $C_i \prec_c C_j$. Therefore, we have $\mathcal{I} \not\models L$ for all $L \in C_i$, and also $\mathcal{I} \not\models C_i$. (ii) L is of the form $(\forall r.D)(a)$. Then, due to the ordering, the maximal literal in every clause C_j with $j > i$ is also a universal restriction. Clauses in which the maximal literal is a universal restriction have no effect on the constructed model. Therefore, we have $\mathcal{I}_{i+1} = \mathcal{I}$, and since $\mathcal{I}_{i+1} \not\models C_i$, we also have $\mathcal{I} \not\models C_i$. □

We can now prove that $\mathcal{I}$ satisfies all ABox clauses C_i, which establishes that $\mathcal{I}$ is a model of $\mathcal{N}$.

Lemma 3. *For every ABox clause C_i, we have $\mathcal{I} \models C_i$.*

Proof. The proof is by contradiction. Let C_i be the smallest ABox clause according to $\prec_c$ such that $\mathcal{I} \not\models C_i$. We distinguish the different cases for the maximal literal L in C_i, where $C_i = L \vee C_i'$. Remember that, due to our ordering and the fact that clauses are sets, the maximal literal is always unique, and L is strictly maximal in C_i. Since $\mathcal{I} \not\models C_i$, we also have $\mathcal{I} \not\models L$.

1. $L = A(a)$. Then $\mathcal{I}_{i+1} \models C_i$ holds due to Step 2a of the model construction, and $\mathcal{I} \models C_i$ due to Lemma 1, which contradicts our assumption.
2. $L = \neg A(a)$. Since $\mathcal{I} \not\models L$, we then have $x_a \in A^{\mathcal{I}}$. $x_a \in A^{\mathcal{I}}$, can only be the case due to Step 2a of the model construction. This means there is a smaller clause $C_j = A(a) \vee C_j'$, where $A(a)$ is maximal in C_j and $\mathcal{I}_j \not\models C_j'$. But then, due to the resolution rule, there is also the clause $C_k = (C_j' \cup C_i')$. We have $k < j$, since $A(a)$ is maximal in C_j, $\neg A(a)$ is maximal in C_i, and, by definition, there can be no literal between $A(a)$ and $\neg A(a)$ in the ordering. Since $\mathcal{I}_i \not\models C_i$ and $\mathcal{I}_j \not\models C_j$, we have that $\mathcal{I}_{k+1} \not\models C_k$, and due Lemma 2 also $\mathcal{I} \not\models C_k$. But this contradicts our initial assumption that C_i is the smallest clause with $\mathcal{I}_i \not\models C_i$.
3. $L = (\exists r.D)(a)$. If $x_D \in \Delta^{\mathcal{I}}$, Step 2b is applied. Consequently, $(x_a, x_D) \in r^{\mathcal{I}_{i+1}}$. $x_D \in D^{\mathcal{I}}$ already holds for the original model $\mathcal{I}^{\mathcal{T}}$, which we only extended (Property 2). Therefore, $\mathcal{I} \models L$, which contradicts our assumption. Assume $x_D \notin \Delta^{\mathcal{I}}$. $x_D \notin \Delta^{\mathcal{I}}$ only holds if $\neg D(x) \in \mathcal{N}^{*2}$ (Property 2). But then, due to the existential role restriction elimination rule, there is also the clause $C_j = C_i' \in \mathcal{N}^{*2}$. $j < i$, since $(\exists r.D)(a)$ is maximal in C_i, and hence larger than all literals in C_j, and $\mathcal{I}_{i+1} \not\models C_j$, since $\mathcal{I}_{i+1} \not\models C_i'$. Due to Lemma 2, $\mathcal{I} \not\models C_j$ holds. This contradicts our initial assumption that C_i is the smallest clause with $\mathcal{I} \not\models C_j$.
4. $L = (\forall r.D)(a)$. This means, we have an edge $(x_a, x') \in r^{\mathcal{I}}$ such that $x' \notin D^{\mathcal{I}}$. $(x_a, x') \in r^{\mathcal{I}}$ can be the case for to two reasons.
   (a) There is a role assertion $r(a,b) \in \mathcal{N}^{*2}$ and $x' = x_b$. But then the role assertion instantiation rule applies between $r(a,b)$ and C_i, and the clause $C_j = C_i' \vee D(b)$ is inferred. Since $\mathcal{I} \not\models C_i$, we also have $\mathcal{I} \not\models C_i'$, and since $x_b \notin D^{\mathcal{I}}$, we have $\mathcal{I} \not\models C_j$. But in the ordering $D(b)$ is smaller than $(\forall r.D)(a)$, and therefore $C_j \prec_c C_i$ holds. This contradicts our initial assumption that C_i is the smallest clause that is not satisfied by the model.
   (b) There is a clause $C_j = (\exists r.D')(a) \vee C_j'$, where $(\exists r.D')(a)$ is maximal, $\mathcal{I}_j \not\models C_j$, and $\neg D'(x) \vee D(x) \notin \mathcal{N}^{*2}$. ($\neg D'(x) \vee D(x) \in \mathcal{N}^{*2}$ would imply $x_{D'} \in D^{\mathcal{I}}$ due to Properties 1 and 2.) Due to the role propagation rule, we also have the clause $C_k = C_i' \vee C_j' \vee (\exists r.D'')(a)$, together with the two clauses $\neg D''(x) \vee D(x)$ and $\neg D''(x) \vee D'(x)$. Since $(\exists r.D'')(a) \prec_l (\forall r.D)(a)$ and $C_j' \prec C_i$, C_k is smaller than C_i. We also have that $\mathcal{I} \not\models C_i'$ and $\mathcal{I} \not\models C_j'$. There are two possibilities.
      i. If $k < j$, $(\exists r.D'')(a)$ must be maximal in C_k, since $\mathcal{I}_j \not\models C_j'$ and $\mathcal{I}_{k+1} \models C_k$. Due to Step 2b of the model construction, we then have $\mathcal{I}_j \models (\exists r.D'')(a)$, and also $\mathcal{I}_j \models (\exists r.D')(a)$. But then $\mathcal{I}_j \models (\exists r.D')(a) \vee C_j'$, which contradicts that $\mathcal{I}_j \not\models C_j$.
      ii. $j < k$. This means there are literals in C_k that are larger than all literals in C_j. This implies, since $(\exists r.D'')(a)$ is smaller than all literals in C_j, that $(\exists r.D'')(a)$ is not maximal in C_k. If $(\exists r.D'')(a)$ is not maximal in C_k, Step 2b of the model construction does not lead to $\mathcal{I}_{k+1} \models (\exists r.D'')(a)$, and since also $\mathcal{I} \not\models C_j'$ and $\mathcal{I} \not\models C_i'$, we also have that $\mathcal{I}_{k+1} \not\models (C_j' \cup C_i')$. But then, due to Lemma 2, no literal in C_k can be satisfied in $\mathcal{I}$, which contradicts our initial assumption that C_i is the smallest clause not satisfied by $\mathcal{I}$.

□

This establishes that $\mathcal{I}$ is a model of the ABox part of $\mathcal{N}^{*2}$. Due to Property 3, we also have $\mathcal{I} \models C$ for all TBox clauses $C \in \mathcal{N}^{*2}$, and therefore $\mathcal{I}$ is a model of $\mathcal{N}^{*2}$. This implies $\mathcal{I} \models \mathcal{N}^*$, because $\mathcal{N}^* \subseteq \mathcal{N}^{*2}$. Since our calculus is sound, $\mathcal{N}^*$ is a conservative extension of $\mathcal{N}$, so we can restrict $\mathcal{I}$ to a model of $\mathcal{N}$ by removing the introduced definers. This brings us the following lemma, and enables us to establish Theorem 2.

Lemma 4. *Let $\mathcal{N}^*$ be any set of clauses saturated using the calculus. If $\mathcal{N}^*$ does not contain the empty clause, we can build a model for it.*

Theorem 2 (Decision Procedure for General $\mathcal{ALC}\nu$ Ontologies)**.** *For any general $\mathcal{ALC}\nu$ ontology $\mathcal{O}$, $\mathcal{O}$ is unsatisfiable iff the empty clause can be derived in finitely many steps from its normal form representation using the rules of the calculus.*

Proof. Since only finitely many definer symbols are introduced and clauses are represented as sets, the number of clauses that can be derived is bounded. This establishes termination of the calculus. The soundness of the rules can be argued in the same way as in (Koopmann and Schmidt 2013c). Hence, if $\bot \in \mathcal{N}^*$, $\mathcal{N}$ must be unsatisfiable. On the other hand, if $\bot \notin \mathcal{N}^*$, we can build a model for $\mathcal{N}$ (Lemma 4). Using our normal form transformation, we can transform any general $\mathcal{ALC}\nu$ ontology into an equisatisfiable set of clauses. Therefore, the calculus forms a sound and refutationally complete decision procedure for satisfiability of general $\mathcal{ALC}\nu$ ontologies. $\square$

In fact, we can even formulate a stronger theorem, which comes in handy for the next section. Observe that for the proof of Lemma 3, we only considered derivations on the maximal literal in each clause. For the refutational completeness of our calculus, it is therefore sufficient to apply rules only on the maximal literals in each clause. Denote the calculus obeying these restrictions by $Res_\prec$. We have the following theorem.

Theorem 5. *$Res_\prec$ is sound and refutationally complete, and provides a decision procedure for satisfiability of general $\mathcal{ALC}\nu$ ontologies.*

Correctness of the Uniform Interpolation Method

For Theorem 3, we have to show that for any axiom or disjunctive concept assertion α with $sig(\alpha) \subseteq \mathcal{S}$, we have $\mathcal{O}^\mathcal{S} \models \alpha$ iff $\mathcal{O} \models \alpha$, where $\mathcal{O}$ is any general $\mathcal{ALC}\nu$ ontology, $\mathcal{S}$ any signature, and $\mathcal{O}^\mathcal{S}$ is computed using our method to compute the general $\mathcal{ALC}\nu$ uniform interpolant. The corresponding clause sets are denoted by $\mathcal{N}$ and $\mathcal{N}^\mathcal{S}$. The normal form transformations preserve all entailments modulo definer concepts. We can therefore safely assume that each ontology and its corresponding normal form share the same entailments, given these entailments do not contain definers.

Since $\mathcal{N}^\mathcal{S}$ is a subset of the saturated set $\mathcal{N}^*$, we have for every axiom $\alpha \in \mathcal{O}^\mathcal{S}$: $\mathcal{O} \models \alpha$. Consequently, we have $\mathcal{O} \models \mathcal{O}^\mathcal{S}$, and this establishes the first implication we have to prove, namely that if $\mathcal{O}^\mathcal{S} \models \alpha$, then $\mathcal{O} \models \alpha$. What remains to show is the other direction, namely that if $\mathcal{O} \models \alpha$ and $sig(\alpha) \subseteq \mathcal{S}$, then $\mathcal{O}^\mathcal{S} \models \alpha$.

If α is a role assertion, this follows trivially, since in $\mathcal{ALC}\nu$, for any role assertion $r(a, b)$ and consistent ontology $\mathcal{O}$, we have $\mathcal{O} \models r(a, b)$ iff $r(a, b) \in \mathcal{O}$. Assume α is a GCI, a concept assertion or a disjunctive concept assertion. Observe that for any ontology $\mathcal{O}$, we have $\mathcal{O} \models C \sqsubseteq D$ iff $\mathcal{O} \cup \{(C \sqcap \neg D)(a^*)\} \models \bot$, where a^* is a new individual not occurring in $\mathcal{O}$, $\mathcal{O} \models C(a)$ iff $\mathcal{O} \cup \{\neg C(a)\} \models \bot$, and $\mathcal{O} \models C_1(a_1) \vee \ldots \vee C_n(a_n)$ iff $\mathcal{O} \cup \{\neg C_1(a_1), \ldots, \neg C_n(a_n)\} \models \bot$. It is therefore sufficient to show that, for any set $\mathcal{C}$ of concept assertions C with $sig(C) \subseteq \mathcal{S}$, we have that if $\mathcal{O} \cup \mathcal{C} \models \bot$, then $\mathcal{O}^\mathcal{S} \cup \mathcal{C} \models \bot$. Let $\mathcal{M}$ be the normal form of $\mathcal{C}$. We show that if $\mathcal{N} \cup \mathcal{M} \models \bot$, then $\mathcal{N}^\mathcal{S} \cup \mathcal{M} \models \bot$.

$\mathcal{M}$ is a set of clauses generated independently of $\mathcal{N}$ for a set of concept assertions $\mathcal{C}$ with $sig(\mathcal{C}) \subseteq \mathcal{S}$. Because of this, it has the following properties.

1. For every clause $C \in \mathcal{M}$, $sig(C) \subseteq \mathcal{S} \cup N_d$.
2. Every TBox clause in $\mathcal{M}$ is of the form $\neg D(x) \vee C$, where D does not occur in $\mathcal{N}$ or $\mathcal{N}^\mathcal{S}$.

Since $Res_\prec$, the calculus refined with an ordering, is refutationally sound and complete, we can use it to show that $\mathcal{N} \cup \mathcal{M} \models \bot$ implies $\mathcal{N}^\mathcal{S} \cup \mathcal{M} \models \bot$. Assume the ordering $\prec_l$ on literals used by $Res_\prec$ is refined so that the following conditions all hold:

- $A \prec_l \neg A \prec_l B \prec_l \neg B$, for all $A, B \in N_c$ with $A \in \mathcal{S}$ and $B \notin \mathcal{S}$.
- $\exists r.D \prec_l \exists s.D$, for all $r, s \in N_r$ with $r \in \mathcal{S}$ and $s \notin \mathcal{S}$.
- $\forall r.D \prec_l \forall s.D$, for all $r, s \in N_r$ with $r \in \mathcal{S}$ and $s \notin \mathcal{S}$.

Assume $\mathcal{N} \cup \mathcal{M} \models \bot$. Then there is a sequence π of inferences using $Res_\prec$ that derives the empty clause $\bot$ from $\mathcal{N} \cup \mathcal{M}$. On any clause C that contains a literal $(\forall r.D)(t)$ or a literal $(\exists r.D)(t)$ with $r \notin \mathcal{S}$, we can only apply either the role propagation or the role instantiation rule, and we can only apply it on a clause that also contains either a literal of the form $(\forall r'.D')(t)$ or a literal of the form $(\exists r'.D')(t)$, where $r' \notin \mathcal{S}$. No such clause exists in $\mathcal{M}$, and therefore we can only apply an inference between C and a clause that is derivable from $\mathcal{N}$. All these inferences are already included in $\mathcal{N}^\mathcal{S}$ by construction.

Now assume we have a clause C that contains a literal L of the form $A(t)$ or $\neg A(t)$ with $A \notin \mathcal{S}$. If resolution is applicable on C, then again only with clauses where the maximal literal is of the form $A'(t')$ or $\neg A'(t')$ with $A' \notin \mathcal{S}$. Assume the maximal literal in C is a role restriction $\mathsf{Q}r.D$ with $r \in \mathcal{S}$. Then, due to our ordering, resolution on L is not possible. We can divide the literals in C into two parts C_1 and C_2, where every literal in C_1 is of the form $A'(t')$ or $\neg A'(t')$, and every literal in C_2 is of the form $\mathsf{Q}r'.D$. Assume that if we ignore the ordering, resolution on C with $L(t')$ would be possible, where $L(t')$ is maximal in C_1. Denote the conclusion of such a rule application by $C_1' \vee C_2$. If C is involved in π, that is the derivation of the empty clause from $\mathcal{N} \cup \mathcal{M}$, then it must be possible to derive from C, possibly after a sequence of steps, a clause C' that does not contain any role restrictions. Since role restrictions have precedence in our ordering, C' is of the form $C_1 \vee C''$. We can now apply resolution on $L(t')$, and the conclusion is $C_1' \vee C''$. Hence, we obtain the same clause that we obtain if we would first resolve on $L(t')$.

Accordingly, we can rearrange any sequence π of inferences that ends with the empty clause in such a way, that we first apply the rules only on literals L with $\mathit{sig}(L) \notin \mathcal{S}$, and afterwards apply rules on literals L with $\mathit{sig}(L) \in \mathcal{S}$. Let π' be such a rearranged sequence of inference steps, and let π_1 be the inferences on literals L with $\mathit{sig}(L) \notin \mathcal{S}$, that have been performed first, and π_2 the remaining inferences. All inferences in π_1 only involve clauses in $\mathcal{N}$, since $\mathcal{M}$ does not contain any clauses that have a literal L with $\mathit{sig}(L) \notin \mathcal{S}$. These inferences have already been performed when computing $\mathcal{N}^{\mathcal{S}}$. All inferences in π_2 only involve clauses that only contain symbols from $\mathcal{S}$. Hence, they can also be performed starting from clauses in $\mathcal{N}^{\mathcal{S}}$. We obtain that if we can derive the empty clause from $\mathcal{N} \cup \mathcal{M}$, then we can also derive it from $\mathcal{N}^{\mathcal{S}} \cup \mathcal{M}$.

Hence, we have that $\mathcal{N} \cup \mathcal{M} \models \bot$ iff $\mathcal{N}^{\mathcal{S}} \cup \mathcal{M} \models \bot$. In other words, for any axiom or disjunctive concept assertion α with $\mathit{sig}(\alpha) \subseteq \mathcal{S}$, we have $\mathcal{N} \models \alpha$ iff $\mathcal{N}^{\mathcal{S}} \models \alpha$. This establishes $\mathcal{O} \models \alpha$ iff $\mathcal{O}^{\mathcal{S}} \models \alpha$, and that $\mathcal{O}^{\mathcal{S}}$ is the general $\mathcal{ALC}\nu$ uniform interpolant of $\mathcal{O}$ for $\mathcal{S}$.

Theorem 3. *Let $\mathcal{O}$ be any general $\mathcal{ALC}\nu$ ontology and $\mathcal{S}$ any signature. The described method always terminates and the returned ontology $\mathcal{O}^{\mathcal{S}}$ is the general $\mathcal{ALC}\nu$ uniform interpolant of $\mathcal{O}$ for $\mathcal{S}$.*

Having this theorem, and because the result of our procedure is a general $\mathcal{ALC}\nu$ ontology, we establish the first theorem of the paper.

Theorem 1. *Let $\mathcal{O}$ be any $\mathcal{ALC}\nu$ ontology, and $\mathcal{S}$ any signature. Then there exists a finite general $\mathcal{ALC}\nu$ ontology $\mathcal{O}^{\mathcal{S}}$ which is a uniform interpolant of $\mathcal{O}$ for $\mathcal{S}$.*

Correctness of the Approximation Method

Theorem 4. *Let $\mathcal{N}$ be any set of clauses. Then, the described method for approximating disjunctive concept assertions computes an $\mathcal{ALC}$ or $\mathcal{ALCO}$ ontology $\mathcal{O}$ with classical ABox, and we have for any $\mathcal{ALC}$ axiom α that $\mathcal{O} \models \alpha$ iff $\mathcal{N} \models \alpha$.*

Proof. Since our normal form transformation preserves all entailments modulo definers, we can again work on the normal form representation of each involved ontology. Denote by $\mathcal{N}$ the input clause set, and by $\mathcal{N}^{\mathit{conv}}$ the approximated clause set that only contains $\mathcal{ALCO}$ convertible clauses. Let α be any $\mathcal{ALC}$ axiom. We show that $\mathcal{N} \models \alpha$ iff $\mathcal{N}^{\mathit{conv}} \models \alpha$.

We assume without loss of generality that $\mathcal{N}$ is consistent. Since $\mathcal{N}^{\mathit{conv}}$ consists only of clauses inferred from $\mathcal{N}$, we have that $\mathcal{N}^{\mathit{conv}} \models \alpha$ implies $\mathcal{N} \models \alpha$. We therefore only have to show that $\mathcal{N} \models \alpha$ implies $\mathcal{N}^{\mathit{conv}} \models \alpha$. Assume $\mathcal{N} \models \alpha$. If α is a TBox axiom, since the approximating reduction only touches ABox clauses, $\mathcal{N}^{\mathit{conv}} \models \alpha$ holds. The same is true if α is a role assertion. Accordingly, we assume that α is a concept assertion $C(a)$.

$\mathcal{N} \models C(a)$ iff $\mathcal{N} \cup \{\neg C(a)\} \models \bot$. Let $\mathcal{M}$ be the normal form representation of $C(a)$. We show that $\mathcal{N} \cup \mathcal{M} \models \bot$ implies $\mathcal{N}^{\mathit{conv}} \cup \mathcal{M} \models \bot$. $\mathcal{M}$ has the following properties:

1. Every TBox clause in $\mathcal{M}$ is of the form $\neg D(x) \vee C$, where D does not occur in $\mathcal{N}$ or $\mathcal{N}^{\mathcal{S}}$.
2. Every ABox clause only contains the individual a occurring in $C(a)$.

Despite the second property, it is possible that clauses in $\mathcal{M}$ contribute to the derivation of ABox clauses containing more than one individual. This is due to the role instantiation rule. More precisely, if $\mathcal{M}$ contains a clause of the form $(\forall r.D)(a) \vee C$, and $\mathcal{N}$ contains a role assertion $r(a, b)$, $D(b) \vee C$ is derived. Inferences of this sort are however only possible through role assertions containing a. We say an individual b is *connected* to a if there is a chain of role assertions $r_0(a, a_1), r_1(a_1, a_2), \ldots, r_n(a_n, b)$. Let N_i^a denote all individuals in $\mathcal{N}$ that are connected to a. We refine the ordering $\prec_i$ used by $\mathit{Res}_{\prec}$ such that for each pair of individuals a_1, a_2 with $a_1 \in N_i^a$ and $a_2 \notin N_i^a$, we have $a_1 \prec_i a_2$. With this ordering, $\mathit{Res}_{\prec}$ gives precedence to ABox literals that contain an individual that is not connected to a.

$\mathcal{ALCO}$ convertible clauses are formally the clauses in which each individual is connected to some root individual. Therefore, every ABox clause that only contains individuals in N_i^a is by definition $\mathcal{ALCO}$ convertible, with a being the root individual. Every clause C that is not $\mathcal{ALCO}$ convertible must therefore contain at least one literal $L(a')$ with $a' \notin N_i^a$, which is maximal in C. Therefore, the only ABox clauses with which a rule application on a non $\mathcal{ALCO}$ convertible clause is possible, are ABox clauses in $\mathcal{N}$. Also, all TBox clauses in $\mathcal{M}$ contain at least one negative definer literal. Due to the side conditions of the rules, no inference between such a clause and an ABox clause is possible, unless the ABox clause contains a positive definer literal of the form $D(a)$, since the conclusion would otherwise contain an ABox literal of the form $\neg D(a)$. Since $\mathcal{M}$ does not share any definer symbols with $\mathcal{N}$, this means that no rule application is possible on a non $\mathcal{ALCO}$ convertible clauses and a clause from $\mathcal{M}$. Hence, if the empty clause can be derived from $\mathcal{N} \cup \mathcal{M}$, and if the derivation involves a clause C that is not $\mathcal{ALCO}$ convertible, then this derivation also involves a clause C' that can be derived from C using only clauses in $\mathcal{N}$. But these clauses are all included in $\mathcal{N}^{\mathit{conv}}$. Hence, if we can derive the empty clause from $\mathcal{N} \cup \mathcal{M}$, then we can also derive it from $\mathcal{N}^{\mathit{conv}} \cup \mathcal{M}$. We establish that for any classical $\mathcal{ALC}$ axiom α, we have $\mathcal{N} \models \alpha$ iff $\mathcal{N}^{\mathit{conv}} \models \alpha$. Hence, we have for any $\mathcal{ALC}$ axiom α that $\mathcal{O} \models \alpha$ iff $\mathcal{N} \models \alpha$. $\square$