# Semantic, Query, and Deductive Forgetting for ALC Ontologies in One Framework

Mostafa Sakr[0000−0003−4296−5831] and Renate A. Schmidt[0000−0002−6673−3333]

University of Manchester, United Kingdom.
Mostafa.Sakr,Renate.Schmidt@manchester.ac.uk

**Abstract.** In this paper we study the problem of forgetting concept names from ALC ontologies. We introduce a unified framework that computes faithful representations of the deductive, query, and semantic forgetting views, and extracts a set axioms indicating the information not preserved in each. This enables users to enhance forgetting views as needed with more information according to application requirements, and achieve a smooth transition between the three variants of forgetting views.

**Keywords:** Forgetting · Query · Ontology · Description Logic.

## 1 Introduction

Forgetting creates a restricted view of an ontology by eliminating subsets of concept and role names from the ontology, while preserving the information pertaining to the remainder of the signature [12,31,9]. Different variants of forgetting have been introduced in the literature to support different reasoning tasks. (i) *Deductive forgetting*, or *uniform interpolation* [38,15,31] where all concept descriptions over the non-forgetting symbols in the logic of the input ontology are preserved. This variant supports entailments checking and classification tasks. (ii) *Query forgetting* where the input ontology and the forgetting view allow for retrieving the same answers to arbitrary conjunctive queries against arbitrary ABoxes [22]. This variant supports ontology-based query answering applications. (iii) *Semantic forgetting* which preserves the interpretations of the non-forgetting symbols [28,27,36]. This variant supports all reasoning tasks over the non-forgetting symbols because it preserves the maximum amount of information [6].

In this paper we introduce a unified forgetting framework that eliminates concept names from input ontologies, and computes representations of views for the three forgetting variants. The framework is designed for $\mathcal{ALC}$ ontologies, and can thus operate on a wide range of real-life ontologies. In the first part of the framework, we compute an *intermediate ontology* which does not use any of the forgetting signature, and preserves the interpretations of the non-forgetting symbols. In the second part, the content of the intermediate ontology is customized according to the wanted variant of forgetting. We guarantee the existence of the intermediate ontology by allowing the use of auxiliary concept names, called definers. In the cases when the intermediate ontology does not use definers, it is a semantic forgetting view, and the customization processes are not needed. When definers are present in the intermediate ontology, it is a representation, or approximation, of the semantic forgetting views, and the elimination of the definers becomes the responsibility of the customization processes. We introduce two processes to extract deductive and query forgetting views from the intermediate ontology. To our knowledge the query customization provides the first query forgetting method for $\mathcal{ALC}$ ontologies. It computes two representations, an $\mathcal{ALC}$ representation, and an $\mathcal{ALCI}$ representation (with role inverse). The $\mathcal{ALC}$ representation uses definers which are eliminated in the $\mathcal{ALCI}$ representation. The $\mathcal{ALC}$ representation is also a representation of the deductive forgetting views. It is thus suitable for more reasoning tasks, and is usable with $\mathcal{ALC}$ reasoning tools. The forgetting framework also enables new use cases that are not supported by existing forgetting methods.

(i) Suppose agent A requests an extract of the ontology of agent B which decides to publish a forgetting view in order to hide sensitive portions of its ontology. The suitable variant of forgetting might not be known to agent A when the request is made, for instance if the complete list of reasoning tasks is only settled at a later stage. A solution is publishing the semantic forgetting view of the ontology omitting the sensitive information to agent A, but if it does not exist, an alternative is publishing the intermediate ontology computable by the first part of the framework, and customizing it later by agent A. This use case is not supported by any of the existing forgetting methods.

(ii) In addition to forgetting views, the unified framework creates sets of axioms indicating the content not preserved in each type of forgetting view. This enables users to enhance deductive and query forgetting views with more information based on their requirements, and achieve a smooth transition between the three variants of forgetting views. For instance, in the deductive customization we compute a set $\Delta^d$ of clauses indicating the content difference between the intermediate ontology and the deductive forgetting view. An $\mathcal{ALC}$ version of the query forgetting view is then obtained by enhancing a representation of the deductive view with axioms from $\Delta^d$.

Due to space restrictions, all proofs are provided in the long version.[1] A more detailed description of the first part of the framework, and the deductive customization is presented in [37].

## 2 Related Work

Forgetting was recognized as an important problem in artificial intelligence in [28], where it was found to coincide with the problem of second-order quantifier

[1] https://github.com/ALCForgetting/ForgettingFramework/blob/1bff4f5e39a37f811b0a3b95321cd90a0097bb21/paper/ForgettingFrameworkForALCOntologies-LongVersion.pdf

elimination, a highly undecidable problem concerned with eliminating second-order quantified symbols from second-order theories [1,13,14]. This notion of forgetting coincides with our notion of semantic forgetting.

Undecidability of semantic forgetting for first-order logic has led to the introduction of the weaker notion of deductive forgetting [42], which coincides with the problem of *uniform interpolation* [16,18,38]. Although uniform interpolation is also undecidable for several logics including $\mathcal{ALC}$ [31,43], empirical evaluations have shown that computing deductive forgetting views of real life ontologies is often feasible [23,45,29]. Subsequently, several deductive forgetting methods were developed for the DLs between $\mathcal{ALC}$ and $\mathcal{SHQ}$ [2,10,19,23,24,25,26,29,39,40].

In a modern view of forgetting, the different variants have been investigated based on *inseparability*, a notion originally introduced in [18] and rigorously developed in [7,8,20,21,30]. Forgetting computes a forgetting view that is inseparable from the input ontology. Inseparability here is dependent of the reasoning task of interest. For instance, an ontology and its deductive view are inseparable with respect to entailment checking, but may be separable with respect to instance checking or conjunctive query answering [30]. To address the last two tasks, the notions of *instance forgetting* and *query forgetting* were proposed and studied for extensions of the description logic $\mathcal{EL}$ [22].

For the description logic $\mathcal{ALC}$, semantic and query forgetting are insufficiently studied. Semantic forgetting methods were proposed in [46,47,48,49] for extensions of $\mathcal{ALC}$, but there are cases where they do not compute the forgetting view even if one exists [44]. Approximations to the semantic forgetting view were proposed in [28,36] where the forgetting view is enhanced with auxiliary concept names to avoid nonexistence issues. A similar idea is used in this paper where the intermediate ontology approximates the semantic forgetting view with definers. Query forgetting has not been studied in the context of $\mathcal{ALC}$. One challenge is that query inseparability, the inseparability notion underpinning query forgetting, is undecidable [8]. Although this adds complexity to query forgetting, developing $\mathcal{ALC}$ query forgetting methods remains open. The difference between the two problems is that the former decides whether any two ontologies are query inseparable or not, but the latter asks whether an arbitrary input $\mathcal{ALC}$ ontology and one specific query view are query inseparable or not.

## 3 Preliminaries and Basic Definitions

Let $N_c, N_r$ be two disjoint sets of concept symbols and role symbols. Concepts in $\mathcal{ALC}$ have the forms: $\bot \mid A \mid \neg C \mid C \sqcap D \mid \exists r.C$ where $A \in N_c, r \in N_r$ and $C$ and $D$ are general concept expressions. We use the following abbreviations: $\top \equiv \neg\bot, \forall r.C \equiv \neg\exists r.\neg C, C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$. An interpretation $\mathcal{I}$ in $\mathcal{ALC}$ is a pair $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ where the domain $\Delta^{\mathcal{I}}$ is a nonempty set and $\cdot^{\mathcal{I}}$ is an interpretation function that assigns to each concept symbol $A \in N_c$ a subset of $\Delta^{\mathcal{I}}$ and to each $r \in N_r$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Non-atomic concepts are interpreted according to: $\bot^{\mathcal{I}} := \emptyset, \quad (\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}, (C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}, (\exists r.C)^{\mathcal{I}} := \{x \in \Delta^{\mathcal{I}} | \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$.

The description logic $\mathcal{ALCI}$ extends $\mathcal{ALC}$ allowing concepts of the form $\exists r^-.C$, and the abbreviation $\forall r^-.C \equiv \neg\exists r.^-.\neg C$, where $r \in N_r$, $r^-$ is the *inverse* of $r$, and $C$ is an $\mathcal{ALCI}$ concept. An interpretation $\mathcal{I}$ in $\mathcal{ALCI}$ extends an $\mathcal{ALC}$ interpretation with the restriction that $(x, y) \in r^{-\mathcal{I}}$ if and only if $(y, x) \in r^{\mathcal{I}}$. The concept $\exists r^-.C$ is then interpreted as $\{x \in \Delta^{\mathcal{I}} | \exists y : (y, x) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$.

An $\mathcal{L}$-TBox, or an ontology, is a set of axioms $C \sqsubseteq D$ and $C \equiv D$, where $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$, and $C$ and $D$ are $\mathcal{L}$-concepts. $\mathcal{I}$ is a model of an ontology $\mathcal{O}$ if all axioms in $\mathcal{O}$ are true in $\mathcal{I}$, in symbols $\mathcal{I} \models C \sqsubseteq D$. And, $\mathcal{I} \models C \sqsubseteq D$ if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. We say that $C \sqsubseteq D$ is satisfiable with respect to $\mathcal{O}$ if and only if $\mathcal{I} \models C \sqsubseteq D$ for some model $\mathcal{I}$ of $\mathcal{O}$. We also say that $C \sqsubseteq D$ is a consequence (entailment) of $\mathcal{O}$, in symbols $\mathcal{O} \models C \sqsubseteq D$, if and only if $\mathcal{I} \models C \sqsubseteq D$ for every model $\mathcal{I}$ of $\mathcal{O}$. Let $C$ be a concept and $\mathcal{O}$ an ontology. We denote by $sig(C)$ the set of concept and role names appearing in $C$, and by $sig(\mathcal{O})$ the set $\bigcup_{C \sqsubseteq D \in \mathcal{O}} sig(C) \cup sig(D)$. We define the function $\mathcal{L}(\mathcal{O})$ to mean the logic used to express $\mathcal{O}$.

An ABox is a set of concept and role assertions over individuals. Let $N_I$ be a set of individuals disjoint with $N_c$, and $N_r$. Assertions in an ABox take the forms $A(a)$ and $r(a, b)$ where $A \in N_c$, $r \in N_r$, and $a, b \in N_I$. We denote by $sig(\mathcal{A})$ the concepts and role names in $\mathcal{A}$. An $\mathcal{ALC}$ ($\mathcal{ALCI}$) knowledge-base is a pair $(\mathcal{O}, \mathcal{A})$ of an $\mathcal{ALC}$ ($\mathcal{ALCI}$) ontology $\mathcal{O}$ and an ABox $\mathcal{A}$.

A conjunctive query $q(\vec{x})$ is a first order formula $\exists \vec{y}.\, q(\vec{x}, \vec{y})$ where $\vec{x} = x_1, \ldots, x_k$ are the *answer variables*, $q(\vec{x}, \vec{y})$ is a conjunction of atoms of the forms $A(u)$ and $r(u, v)$, $A \in N_c$, $r \in N_r$, $u, v \in N_I \cup \vec{x} \cup \vec{y}$. If $q$ does not have answer variables then $q$ is called a Boolean conjunctive query. An answer to a query $q(\vec{x})$ is $\vec{a} = a_1, \ldots, a_k$ with the same cardinality as $\vec{x}$ such that $\mathcal{K} \models q(\vec{a})$. That is, for every $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ we have $\mathcal{I} \models q(\vec{a})$. The answer to a Boolean conjunctive query is *yes* if $\mathcal{I} \models q$ for every model $\mathcal{I}$ of $\mathcal{K}$, and *no* otherwise.

**Definition 1.** *Let $\Sigma$ be a signature. Two models $\mathcal{I}$ and $\mathcal{J}$* $\Sigma$-coincide *iff $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$ and $p^{\mathcal{I}} = p^{\mathcal{J}}$ for every concept name or role name $p \in \Sigma$.*

**Definition 2.** *Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be two ontologies and $\Sigma \subseteq N_c \cup N_r$. We say that the two ontologies are:*

1. model inseparable wrt. *$\Sigma$, in symbols $\mathcal{O}_1 \equiv^{\mathcal{M}}_{\Sigma} \mathcal{O}_2$, iff for every model $\mathcal{I}_1$ of $\mathcal{O}_1$ there is a model $\mathcal{I}_2$ of $\mathcal{O}_2$, and vice versa, such that $\mathcal{I}_1$ and $\mathcal{I}_2$ $\Sigma$-coincide.*
2. deductively inseparable wrt. *$\Sigma$, in symbols $\mathcal{O}_1 \equiv^{C}_{\Sigma} \mathcal{O}_2$, iff for every axiom $\alpha$ over $\Sigma$ we have $\mathcal{O}_1 \models \alpha$ iff $\mathcal{O}_2 \ \models\ \alpha$, where $\mathcal{L}(\mathcal{O}_1) = \mathcal{L}(\mathcal{O}_2) = \mathcal{L}(\alpha)$.*
3. query inseparable wrt. *$\Sigma$, in symbols $\mathcal{O}_1 \equiv^{Q}_{\Sigma} \mathcal{O}_2$, iff for every ABox $\mathcal{A}$, and conjunctive query $q(\vec{x})$ over $\Sigma$, we have $(\mathcal{O}_1, \mathcal{A}) \models q(\vec{a})$ iff $(\mathcal{O}_2, \mathcal{A}) \models q(\vec{a})$.*

**Definition 3.** *Let $\mathcal{O}$ be an $\mathcal{ALC}$ ontology, and $\mathcal{F} \subseteq sig(\mathcal{O}) \cap N_c$ a* forgetting signature. *Let $\mathcal{V}$ be an ontology such that $sig(\mathcal{V}) \subseteq sig(\mathcal{O}) \backslash \mathcal{F}$. We say that $\mathcal{V}$ is:*

1. semantic forgetting view *of $\mathcal{O}$ wrt. $\mathcal{F}$ iff $\mathcal{O}$ and $\mathcal{V}$ are model inseparable.*
2. deductive forgetting view *of $\mathcal{O}$ wrt. $\mathcal{F}$ iff $\mathcal{O}$ and $\mathcal{V}$ are deductively inseparable.*
3. query forgetting view *of $\mathcal{O}$ wrt. $\mathcal{F}$ iff $\mathcal{O}$ and $\mathcal{V}$ are query inseparable.*

As observed in [6], semantic forgetting views are also deductive and query forgetting views because they preserve the interpretations of the non-forgetting symbols. In Horn logics, query forgetting views are deductive forgetting views [22], but this is not the case in non-Horn logics. Consider for instance the ontology consisting of the two axioms $A \sqsubseteq B, B \sqsubseteq C \sqcup D$. The empty ontology and the ontology $\{A \sqsubseteq C \sqcup D\}$ are respectively a query and a deductive forgetting views with respect to $\{B\}$, but the empty ontology is not a deductive forgetting view.

## 4 The Intermediate Ontology

This section describes the first part of the framework. The output of this part is the intermediate ontology $\mathcal{O}^{int}$ The first and second cards in Fig 1 depicts the computation process of $\mathcal{O}^{int}$. They are, the *normalize,* and the *forget,* processes. The *normalize* process takes the input ontology $\mathcal{O}$, and transforms it to an ontology $\mathcal{O}^{clausal}$ in normal form. Each axiom of $\mathcal{O}$ is transformed separately. First, the axiom is transformed to *negation normal form* where negations are directly applied to concept names and the symbols $\sqsubseteq$ and $\equiv$ are eliminated. Second, clauses below role restrictions that use forgetting symbols are extracted by the *structural transformation* technique in [4,34]. This step introduces auxiliary concept names, called *definers*, which represent subsets of role successors. For instance, if $B$ is a forgetting symbol the clause $A \sqcup \exists r.(B \sqcap C)$ is structurally transformed to the clauses $A \sqcup \exists r.D, \neg D \sqcup (B \sqcap C)$, where $D$ is a definer. Third, the clauses computed from the previous two steps are transformed to *conjunctive normal form.*

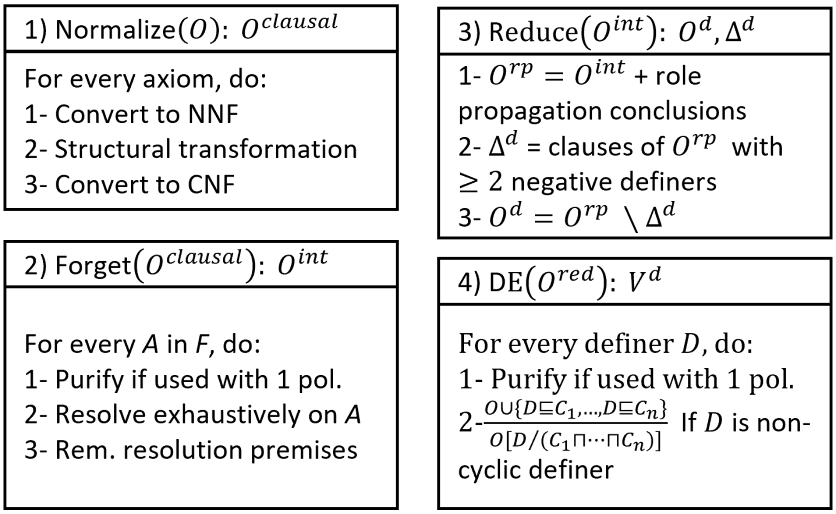
1) Normalize($O$): $O^{clausal}$
For every axiom, do:
1- Convert to NNF
2- Structural transformation
3- Convert to CNF
2) Forget($O^{clausal}$): $O^{int}$
For every $A$ in $F$, do:
1- Purify if used with 1 pol.
2- Resolve exhaustively on $A$
3- Rem. resolution premises
3) Reduce($O^{int}$): $O^{d}, \Delta^{d}$
1- $O^{rp} = O^{int}$ + role propagation conclusions
2- $\Delta^{d}$ = clauses of $O^{rp}$ with $\geq$ 2 negative definers
3- $O^{d} = O^{rp} \setminus \Delta^{d}$
4) DE($O^{red}$): $V^{d}$
For every definer $D$, do:
1- Purify if used with 1 pol.
2-$\frac{O \cup \{D \sqsubseteq C_1, \ldots, D \sqsubseteq C_n\}}{O[D/(C_1 \sqcap \cdots \sqcap C_n)]}$ If $D$ is non-cyclic definer

Fig. 1: Fine-grained forgetting framework.

The second process in the framework is the *forget* process. Its inputs are the ontology $\mathcal{O}^{clausal}$, and the forgetting signature $\mathcal{F}$. The output is an ontology $\mathcal{O}^{int}$, called *the intermediate ontology.* $\mathcal{O}^{int}$ is constructed from $\mathcal{O}^{clausal}$ by iteratively eliminating each forgetting concept name $A \in \mathcal{F}$. If $A$ occurs with a single polarity across all clauses, it is *purified*, i.e., replaced with $\top$ if it occurs positively, and $\bot$ if it occurs negatively. Otherwise, exhaustive resolution is performed on $A$-literals in all clauses before discarding these clauses. Standard optimizations such as *tautology deletion* which eliminates tautologous clauses, and *subsumption deletion* which deletes a clause if it is subsumed by another, are eagerly performed.

**Theorem 1.** *Let $\mathcal{O}$ be an input ontology, and $\mathcal{F}$ the given forgetting signature of concept names. Let $\mathcal{O}^{int}$ be the ontology obtained from the normalise and forget processes described above. The two ontologies $\mathcal{O}$ and $\mathcal{O}^{int}$ are model inseparable for the non-forgetting symbols $sig(\mathcal{O})\backslash\mathcal{F}$.*

## 5 The Deductive Customization

We present the deductive customization process of $\mathcal{O}^{int}$. The process is depicted by the third and fourth cards in Fig 1. The reduce process takes the intermediate ontology $\mathcal{O}^{int}$, and computes two sets $\mathcal{O}^{d}$ and $\Delta^{d}$ of axioms. $\mathcal{O}^{d}$ the *deductively reduced ontology* . It is deductively inseparable from the $\mathcal{O}$ and $\mathcal{O}^{int}$ with respect to the non-forgetting signature $sig(\mathcal{O})\backslash\mathcal{F}$. We will use $\mathcal{O}^{d}$ later to obtain the deductive forgetting view, and it will be the basis of the fine-grained views. $\Delta^{d}$ comprises the $\mathcal{O}^{int}$ clauses that use two or more distinct definers with negative polarity. It indicates the content difference between $\mathcal{O}^{int}$ and $\mathcal{O}^{d}$.

The ontology $\mathcal{O}^{d}$ is computed by first making some implicit consequences of $\mathcal{O}^{int}$ explicit. We denote this enhanced snapshot of $\mathcal{O}^{int}$ by $\mathcal{O}^{rp}$. Indeed, $\mathcal{O}^{rp}$ and $\mathcal{O}^{int}$ are logically equivalent, because they have the same models. The ontology $\mathcal{O}^{d}$ is then obtained as the set of clauses of $\mathcal{O}^{rp}$ that are not in $\Delta^{d}$. $\mathcal{O}^{rp}$ is obtained by saturating $\mathcal{O}^{int}$ with the conclusions computable by *role propagation* rule in Fig 2. The rule applies on four premises. The first is a $\Delta^{d}$ clause of the form $P_0 \sqcup C_0$, where $C_0$ is a concept that does not use negative definers, and $P_0$ is a disjunction of two or more negative definers. The clause $P_0 \sqcup C_0$ is called a *trigger clause*, and the definers occurring in $P_0$ *trigger definers*, because they influence the other premises of the role propagation rule. When $\mathcal{O}^{int}$ has several clauses $P_0 \sqcup C_0^1, P_0 \sqcup C_0^2, \ldots, P_0 \sqcup C_0^l$ with $l \geq 1$, we *combine* these clauses in a single clause $P_0 \sqcup (\bigsqcap_{i=1}^{l} C_0^i)$.

The second premise is a set of $m$ clauses $P_j \sqcup C_j$, with $1 \leq j \leq m$ and $m \geq 0$. Negative definers cannot occur in $C_j$, and the concepts $P_j$ are disjunctions of negative definers. The definers in $P_j$ must form a subset of the trigger definers.

The third and the fourth premises are clauses where trigger definers occur positively below role restrictions. The definer of the third premise can occur below existential or universal role restriction. The definers of the fourth premise can only occur below universal role restrictions. There can be $n + 1$ clauses in

**Role Propagation**

$$\frac{P_0 \sqcup C_0,\ \bigcup_{j=1}^{m}\{P_j \sqcup C_j\}.\ E_0 \sqcup \mathcal{Q}r.D_0,\ \bigcup_{i=1}^{n}\{E_i \sqcup \forall r.D_i\}}{(\bigsqcup_{i=0}^{n} E_i) \sqcup \mathcal{Q}r.(\bigsqcap_{j=0}^{m} C_j)}$$

where $P_0 = \bigsqcup_{i=0}^{n} \neg D_i$, $P_j$ is any concept in $P_0$, $\mathcal{Q} \in \{\exists, \forall\}$, and $C_0$ and $C_j$ do not contain a definer.

**Simple Definer Elimination**

$$\frac{\mathcal{O} \cup \{\neg D \sqcup C_1, ..., \neg D \sqcup C_n\}}{\mathcal{O}[D/\bigsqcap_{i=1}^{n} C_i]}$$

where $D \notin sig(C_i)$, $C_i$ does not contain any negative definers, and $\mathcal{O}$ does not contain $D$ negatively.

Fig. 2: Role Propagation and Simple Definer Elimination rules.

the third and the fourth premises. Each clause uses a different trigger definer. If $\mathcal{O}^{int}$ has many clauses where a trigger definer occurs positively, then we combine them, and use the combined clause in the inference.

The fourth process of the framework is the *DE* (or *Definer Elimination*) process. Its input is the deductively reduced ontology $\mathcal{O}^d$, and its output is an ontology $\mathcal{V}^d$ which is model inseparable to $\mathcal{O}^d$ with respect to $sig(\mathcal{O})\backslash\mathcal{F}$. The ontology $\mathcal{V}^d$ is computed from $\mathcal{O}^d$ by iteratively eliminating definers. Definers occurring with single polarity across all clauses are purified away. Otherwise, the simple definer elimination rule in Fig 2 is applied.

*Example 1.* Let $\mathcal{O} = \{A \sqsubseteq \forall r.B \sqcap \forall s.\neg B, G \sqsubseteq \exists r.(\neg B \sqcup C), B \sqsubseteq H\}$, and $\mathcal{F} = \{B\}$. The normalization process produces $\mathcal{O}^{clausal} = \{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg D_1 \sqcup B, \neg D_2 \sqcup \neg B, \neg D_3 \sqcup \neg B \sqcup C, \neg B \sqcup H\}$, where $D_1, D_2$, and $D_3$ are definers. The forget process performs resolution on the $B$-literals which computes the clauses $\{\neg D_1 \sqcup \neg D_2, \neg D_1 \sqcup \neg D_3 \sqcup C, \neg D_1 \sqcup H\}$, and discards the premises $\{\neg D_1 \sqcup B, \neg D_2 \sqcup \neg B, \neg D_3 \sqcup \neg B \sqcup C, \neg B \sqcup H\}$. Now, $\mathcal{O}^{int}$ is $\{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg D_1 \sqcup \neg D_2, \neg D_1 \sqcup \neg D_3 \sqcup C, \neg D_1 \sqcup H\}$, and $\Delta^d$ is the subset $\{\neg D_1 \sqcup \neg D_2, \neg D_1 \sqcup \neg D_3 \sqcup C\}$. We construct $\mathcal{O}^{rp}$ by saturating $\mathcal{O}^{int}$ using role propagation inferences. In this example, only one inference can be performed with the trigger clause $\neg D_1 \sqcup \neg D_3 \sqcup C$. The clause $\neg D_1 \sqcup \neg D_2$ cannot trigger a role propagation inference because $D_1$ and $D_2$ occur positively below different roles $r$ and $s$. The second premise of the inference is the empty set, the third is $\neg G \sqcup \exists r.D_3$, and the fourth is the set $\{\neg A \sqcup \forall r.D_1\}$. The conclusion of the inference is $\neg A \sqcup \neg G \sqcup \exists r.(C \sqcap H)$. The reduced ontology $\mathcal{O}^d$ comprises the clauses of $\mathcal{O}^{rp}$ that are not in $\Delta^d$. That is, $\mathcal{O}^d = \{\neg A \sqcup \forall r.D_1, \neg D_1 \sqcup H, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg A \sqcup \neg G \sqcup \exists r.(C \sqcap H)\}$. The definer $D_1$ is eliminated from $\mathcal{O}^d$ using a simple definer elimination inference, which replaces every occurrence of $D_1$ with $H$. The definers $D_2$ and $D_3$ are purified by replacing them with $\top$ in the third and fourth clauses of $\mathcal{O}^d$. The deductive forgetting view $\mathcal{V}^d$ of $\mathcal{O}$ with respect to $\mathcal{F}$ is therefore the set $\{\neg A \sqcup \forall r.H, \neg G \sqcup \exists r.\top, \neg A \sqcup \neg G \sqcup \exists r.(C \sqcap H)\}$.

The described method does not eliminate all definers. In particular, *cyclic definers*, i.e., definers occurring both positively and negatively in one or more clauses, in $\mathcal{O}^d$ are not attempted to be eliminated. Cyclic definers signal there are syntactic cycles in the original ontology over some forgetting symbols. In some cases, they are eliminable but may result in infinite forgetting views. In the remainder of the cases, finite representations are possible, but may be complex to find and unpractical. For example, for the ontology $\mathcal{O} = \{A \sqsubseteq B, B \sqsubseteq \forall r.B\}$, and forgetting signature $\mathcal{F} = \{B\}$, $\mathcal{O}^{int}$ and $\mathcal{O}^d$ coincide, and are represented by $\{\neg A \sqcup \forall r.D, \neg D \sqcup \forall r.D\}$. The definer $D$ is cyclic because it occurs positively and negatively in $\neg D \sqcup \forall r.D$. Eliminating it gives the clause $\beta = \neg A \sqcup \forall r.\forall r.\forall r.\ldots$ with infinite nesting of role restrictions. Suppose $\mathcal{O}$ contains the axiom $\alpha = A \sqsubseteq \forall r.\forall r.\bot$. In this case, the desired ontology will just comprise the axiom $\alpha$ because $\beta$ is redundant. Although $\alpha$ in this example is explicitly known, in real life it can be obscure, and inferring it can be expensive. In general eliminating cyclic definers can be double exponential in the size of the ontology [31]. The framework skips eliminating them. Empirical evaluations [36,37] shows that real-life ontologies contain relatively fewer cyclic definers.

**Theorem 2.** *Let $\mathcal{O}$ be an ontology, $\mathcal{F}$ a forgetting signature of concept names, $\mathcal{O}^d$ the deductively reduced ontology of $\mathcal{O}$ with respect to $\mathcal{F}$, and $\mathcal{V}^d$ the ontology obtained by eliminating definers from $\mathcal{O}^d$. The following hold. (i) The two ontologies $\mathcal{O}^d$ and $\mathcal{V}^d$ are model inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$. (ii) The two ontologies $\mathcal{O}$ and $\mathcal{V}^d$ are deductively inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$. (iii) If $\mathcal{V}^d$ does not use definers, then $\mathcal{V}^d$ is the deductive forgetting view of $\mathcal{O}$ with respect to $\mathcal{F}$.*

A distinguishing feature of the fine-grained framework is the computation of the set $\Delta^d$, which contains the clauses of $\mathcal{O}^{rp}$ that are not in $\mathcal{O}^d$. These clauses indicate the content difference between $\mathcal{O}$ and $\mathcal{O}^d$. For example, consider the clause $\neg D_1 \sqcup \neg D_2 \in \Delta^d$ in Example 1. It indicates that the set of $r$-successors and the set of $s$-successors of elements in the interpretation of $A$ are disjoint. An information that is not preserved in $\mathcal{O}^d$ and $\mathcal{V}^d$. The set $\Delta^d$ can thus give an understanding of the information relative to the non-forgetting symbols that is not preserved in the deductive forgetting views. We can use the $\Delta^d$ clauses to enhance $\mathcal{O}^d$ with more content. This reveals a spectrum of fine-grained views between the deductive and the semantic forgetting views.

## 6 The Query Customization

We present the query customization process, which is an application of the fine-grained process explained above. The first step in this process is the *query reduction* step which takes $\mathcal{O}^{int}$, and computes a snapshot $\mathcal{O}^q$ of $\mathcal{O}^d$ enhanced with clauses from $\Delta^d$ such that $\mathcal{O}$ and $\mathcal{O}^q$ are deductively and query inseparable. The second step eliminates the non-cyclic definers from $\mathcal{O}^q$.

### 6.1 Query Reduction

First, we compute $\mathcal{O}^q$ which is the union of $\mathcal{O}^d$ and a portion of $\Delta^d$ to be specified. For this we need to differentiate between existential and universal clauses.

**Definition 4.** *Let $\mathcal{O}^{int}$ be the intermediate ontology, and $D \in N_d$ a definer occurring in $\mathcal{O}^{int}$. We say $D$ is an* existential (universal) definer *if it occurs with positive polarity only below existential (universal) role restrictions in $\mathcal{O}^{int}$.*

*Let $\Delta^d$ be the set of clauses from $\mathcal{O}^{int}$ with two or more negative definers, and suppose $\mathcal{C} \in \Delta^d$. We say that $\mathcal{C}$ is* existential *if it contains at least one existential definer with negative polarity, and* universal *if all negative definers in $\mathcal{C}$ are universal. By $\Delta^e$ we denote the set of all existential clauses in $\Delta^d$, and by $\Delta^u$ the set of universal definers in $\Delta^d$.*

We now have the following.

$$\mathcal{O}^q = (\mathcal{O}^{rp} \backslash \Delta^d) \cup \Delta^u \tag{1}$$

This means that the query reduced ontology $\mathcal{O}^q$ is just the union of $\mathcal{O}^d$ and $\Delta^u$. The set $\Delta^e$ indicates the content difference between $\mathcal{O}^{rp}$ and $\mathcal{O}^q$. Since $\mathcal{O}^{rp}$ and $\mathcal{O}^{int}$ are logically equivalent, it indicates the content difference between $\mathcal{O}^{int}$ and $\mathcal{O}^q$.

**Lemma 1.** *Let $\mathcal{O}$ be an $\mathcal{ALC}$ ontology, $\mathcal{F}$ a forgetting signature consisting of concept names, and $\mathcal{O}^q$ the query reduced ontology obtained by (1). Then $\mathcal{O}$ and $\mathcal{O}^q$ are query inseparable and deductively inseparable with respect to the signature $sig(\mathcal{O}) \backslash \mathcal{F}$.*

### 6.2 Definer Elimination

There are three types of definers that may occur in $\mathcal{O}^q$: *cyclic definers, simple definers,* and *complex definers.* As described in Section 4, a definer is cyclic if it occurs in at least one clause of $\mathcal{O}^q$ with positive and negative polarities. We skip the elimination of cyclic definers for the existence and practicality reasons as explained in Section 4.

The remaining definers are either simple, or complex definers, which we define next.

**Definition 5.** *We say $D$ is a* complex definer*, if $D$ is not a cyclic definer, and one of the following is true about $D$. (i) $D$ occurs with negative polarity in a clause from the set $\Delta^u$. (ii) There is a clause of the following form in $\mathcal{O}^q$.*

$$C \sqcup \neg D \sqcup \forall r_1.D_1 \sqcup \cdots \sqcup \forall r_n.D_n, \tag{2}$$

*where $D_1, \ldots, D_n$ are complex definers, and $n \geq 1$. We use $N_d^+$ to denote the set of complex definers.*

*We say $D$ is a* simple definer *if it occurs negatively only in clauses where no other negative definer is used, and (ii) from above does not apply for $D$.*

Simple definers are eliminated using the method described in the definer elimination process of the deductive customization. That is, they are purified if they occur in $\mathcal{O}^q$ with single polarity, and eliminated using the simple definer elimination rule from Figure 2 when they occur with positive and negative polarities. The ontology obtained by eliminating simple definers from $\mathcal{O}^q$ is denoted by $\mathcal{V}^q_{ALC}$. It is the $\mathcal{ALC}$ representation of the query forgetting view.

**Theorem 3.** *Let $\mathcal{O}$ be an ontology, $\mathcal{F}$ a signature of concept names, $\mathcal{O}^q$ the query reduced ontology of $\mathcal{O}$ with respect to $\mathcal{F}$, and $\mathcal{V}^q_{ALC}$ the ontology obtained by eliminating simple definers from $\mathcal{O}^q$. The following hold.*

1. *$\mathcal{O}^q$ and $\mathcal{V}^q_{ALC}$ are model inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.*
2. *$\mathcal{O}$ and $\mathcal{V}^q_{ALC}$ are query inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.*
3. *$\mathcal{O}$ and $\mathcal{V}^q_{ALC}$ are deductively inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.*
4. *If no definers remain in $\mathcal{V}^q_{ALC}$, then $\mathcal{V}^q_{ALC}$ is a query and deductive forgetting view of $\mathcal{O}$ with respect to $\mathcal{F}$.*

From a computational point of view, the time to compute $\mathcal{V}^q_{ALC}$ only differs from the time to compute $\mathcal{V}^d$ by the time consumed to compute $\Delta^u$. The results of the empirical evaluation in [37] of the framework has shown that computing $\mathcal{V}^d$ is 2 times faster than using the state-of-the-art forgetting tool Lethe [23], the benchmark for many forgetting tools [2,3,45], on real life ontologies from the Bioportal repository [32,33,41]. Since the clauses of $\Delta^u$ are syntactically defined, extracting $\Delta^u$ is linear in the size of $\Delta^d$. As such, we may expect similar performance when computing $\mathcal{V}^q_{ALC}$ to that when computing $\mathcal{V}^d$ which is suggested to be practical in real life as noted. We attribute the simplicity of computing $\mathcal{O}^q$ and consequently $\mathcal{V}^q_{ALC}$ to the granularity of the axioms of $\mathcal{O}^{int}$, which in turn can be attributed to the normal form used in the framework.

Complex definers are harder to eliminate and require additional reasoning. We also allow role inverse to be used to find solutions more often. We categorize complex definers in two categories, *type 1* and *type 2* complex definers. The elimination of type 1 complex definers preserves model inseparability up to the eliminated definers, but the elimination of type 2 complex definers only preserves query inseparability.

**Definition 6.** *Let $D_1$ be a complex definer. We say $D$ is a* type 2 (complex) definer *if and only if there is a clause of the form $C \sqcup \forall r_1.D_1 \sqcup \forall r_2.D_2 \sqcup \cdots \sqcup \forall r_n.D_n$, where $C$ is a concept, $r_1, \ldots, r_n \in N_r$ are role names, $D_1, D_2, \ldots, D_n$ are complex definers, $n \geq 2$, and one of the following is true. (i) $D$ is a definer from the set $D_1, \ldots, D_n$. (ii) $D$ occurs with negative polarity in $C$. A complex definer $D$ is a* type 1 (complex) definer *if it occurs positively only in clauses of the form $C \sqcup \forall r.D$ such that $C$ does not contain negative definers.*

The rules for eliminating complex definers are listed in Figure 3. Only the LB rule is needed for eliminating type 1 definers, whereas the three rules are needed for eliminating type 2 definers. We will explain the rules first and their intuition, before explaining the elimination process.

The *upper bound extraction (UB)* rule computes clauses where definers are used as top level negative literals. The form of conclusions of UB inferences is

**Upper Bound Extraction (UB)**

$$\frac{C_2 \sqcup \forall s_1.D_1 \sqcup \cdots \sqcup \forall s_m.D_m, \quad \bigcup_{j=1}^{m} \neg D \sqcup \neg D_j \sqcup E_j}{\neg D \sqcup \forall s_1^-.C_2 \sqcup \cdots \sqcup \forall s_m^-.C_2 \sqcup E_1 \sqcup \cdots \sqcup E_m}$$

where $D, D_1, \ldots, D_m$ are type 2 definers, $D$ occurs in a clause $C_1 \sqcup \forall r_1.D \sqcup \forall r_2.D'$ with the type 2 definer $D'$, $m \geq 2$, negative definers do not occur in $C_1$, and definers do not occur in $C_2$.

**Lower Bound Extraction (LB)**

$$\frac{C \sqcup \forall r_1.D_1 \sqcup \cdots \sqcup \forall r_n.D_n, \quad \bigcup_{k=1}^{n}\{\neg D_k \sqcup C_k\}}{D_i \sqcup \forall r_i^-.(C \sqcup \bigsqcup_{j=1,j\neq i}^{n} \forall r_j.C_j)}$$

where $1 \leq i \leq n$, $n \geq 1$, $\{D_1, \ldots, D_n\} \subseteq N_d^+$, definers do not occur in $C$, and negative definers do not occur in $C_k$.

**Relaxing**

$$\frac{C \sqcup \forall r_1.D_1 \sqcup \cdots \sqcup \forall r_n.D_n, \quad \neg D_1 \sqcup E}{C \sqcup \forall r_1.E \sqcup \forall r_2.D_2 \sqcup \cdots \sqcup \forall r_n.D_n}$$

where $n \geq 2$, and $C$ and $E$ are concepts that do not use definers.

Fig. 3: Complex definer elimination rules

$D \sqsubseteq C$, where $C$ is the concept $\forall s_1^-.C_2 \sqcup \cdots \sqcup \forall s_m^-.C_2 \sqcup E_1 \sqcup \cdots \sqcup E_m$. The concept $C$ can be seen as an upper bound on $D$, because if a domain element is in the extension of $D$ then it is also in the extension of $C$. The first and second premises of the rule are clauses where at least two complex definers occur positively below universal role restrictions. The third premise is a set of $m$ clauses. Each uses a pair of negative definers $D$ and $D_j$, where $D$ is from the first premise, and $D_j$ is from the second premise. To understand the UB rule, think of the premises as inclusions.

(3) $$\neg C_2 \sqsubseteq \forall s_1.D_1 \sqcup \cdots \sqcup \forall s_m.D_m$$

(4) $$\bigcup_{j=1}^{m}\{D \sqcap D_j \sqsubseteq E_j\}$$

(5) $$D \sqcap \exists s_1^-.\neg C_2 \sqcap \cdots \sqcap \exists s_m^-.\neg C_2 \sqsubseteq E_1 \sqcup \cdots \sqcup E_m$$

The premises present as (3) and (4), and the conclusion as (5). Suppose $\mathcal{I}$ is a model, and $e \in \Delta^{\mathcal{I}}$ is a domain element. Assume $e$ is in the interpretation of the concept on the left hand side of the inclusion in (5). From the conjuncts $\exists s_i^-.\neg C_2$ with $1 \leq i \leq m$, there must exist a domain element $d \in \Delta^{\mathcal{I}}$ such that $d \notin C_2^{\mathcal{I}}$ and $(d, e) \in s_i^{\mathcal{I}}$. From (3) we know that $e$ is in the interpretation of one of $D_1, D_2, \ldots,$ or $D_m$. Since additionally $e \in D^{\mathcal{I}}$, we get from (4) that $e \in (E_1 \sqcup E_2 \sqcup \cdots \sqcup E_m)^{\mathcal{I}}$, which is expressed by the right hand side of the inclusion in (5).

The *lower bound extraction (LB)* rule computes clauses where a complex definer occurs positively as a top level literal. The conclusion of a LB inference can be written as $E \sqsubseteq D_i$, where $D_i$ is a definer, and $E$ is the negation of the concept $\forall r_i^-.(C \sqcup \bigsqcup_{j=1,j\neq i}^{n} \forall r_j.C_j)$. The concept $E$ is a lower bound of $D$ because a domain element in the extension of $E$ is necessarily in the extension of $D$. The first premise of a LB inference is a clause where the complex definers $D_1, \dots, D_n$ occur with positive polarity below universal role restrictions. We require the concept $C_1$ to be free of definers. This restriction on $C_1$ is important because $\mathcal{V}^q_{ALC}$ may contain a clause of the form (2). When $C_1$ contains a definer $D'$, the restriction on $C_1$ becomes a blocking constraint forcing either the elimination of $D'$ if it is universal, or performing further inferences such that $D_1, \dots, D_n$ are no longer complex definers.

The second premise of a LB inference is a set of $n$ clauses of the form $\neg D_k \sqcup C_k$ where $1 \leq k \leq n$. When several clauses $\neg D_k \sqcup C^l_k$ are present for a value of $k$, where $l \geq 2$, we combine them, and use the combined clause instead. Clauses from the set $\Delta^u$ cannot be used in the second premise. This restriction prevents negative definers from occurring below universal role restriction in the conclusion of the rule. If for some $D_k$, with $1 \leq k \leq n$, no clause in the required form exists, or if $D_k$ occurs negatively only in clauses from $\Delta^u$, we use the clause $\neg D_k \sqcup \top$.

The conclusion of a LB inference is a clause in which the complex definer $D_i$ occurs positively as a top level literal, and $1 \leq i \leq n$. Because of the use of role inverse, the conclusion is an $\mathcal{ALCI}$ concept. Let us explain the rule. Let us assume $i = 1$.

$$\neg C_1 \sqsubseteq \forall r_1.D_1 \sqcup \dots \sqcup \forall r_n.D_n \tag{6}$$

$$\bigcup_{i=2}^{n} \{D_i \sqsubseteq C_i\} \tag{7}$$

$$\exists r_1^-.(\neg C_1 \sqcap \sqcap_{i=2}^{n} \exists r_i.\neg C_i) \sqsubseteq D_1 \tag{8}$$

The axiom in (6) represents the first premise. (7) represents the second premise, and (8) represents the conclusion. Observe that we ignored the clause $D_1 \sqsubseteq C_1$ from (7) because it is not required to compute the conclusion. Suppose $\mathcal{I}$ is an interpretation, and a domain element $e$ is in the interpretation of the left hand side of (8). There is an $r_1$-predecessor $d$ of $e$, i.e., $(d, e) \in r_1^{\mathcal{I}}$. Furthermore, the following is true about $d$. (i) $d$ is not in the interpretation of $C_1$. (ii) For each $i$ where $2 \leq i \leq n$, $d$ has a successor $e_i$ through $r_i$, and $e_i$ is not in the interpretation of $C_i$. From (i) and (6) we know that there is $j$ where $1 \leq j \leq n$ where every $r_j$ successor of $d$ is in the interpretation of $D_j$. From (ii) and (7) we know that $j$ is not in the range from 2 to $n$. Therefore, $d$ is in the interpretation of $\forall r_1.D_1$. Since $e$ is a $r_1$ successor of $d$, we get that $e$ is in the interpretation of $D_1$, which is the right hand side of (8).

The *relaxing* rule is a form of resolution. The first premise is a clause of the form $C \sqcup \forall r_1.D_1 \sqcup \dots \sqcup \forall r_n.D_n$, where $n \geq 2$ and $C$ does not contain definers. The second premise is a clause of the form $D_1 \sqcup E$ where $E$ does not contain definers, if there are $m$ clauses $D_1 \sqcup E^i$ where $1 \leq i \leq m$, we combine them in the clause $\neg D_1 \sqcup (E^1 \sqcap \dots \sqcap E^m)$ and use the combined clause. The conclusion is a clause that resembles the first premise but replaces $D_1$ with $E$.

Type 1 definers are eliminated iteratively. In each iteration, LB inferences are performed exhaustively on all clauses that use the definer positively. The conclusions of the LB inferences will have the definer as a top level literal. We then resolve the obtained conclusions with the $\Delta^u$ clauses where the definer occurs. When all possible resolutions have been performed, we remove all clauses that use the definer. The following example explains some further subtleties.

*Example 2.* Consider the ontology $\mathcal{O} = \{A \sqsubseteq \forall r.B \sqcap \forall s.\neg B\}$, and let $\mathcal{F} = \{B\}$ be a forgetting signature. The query reduced ontology $\mathcal{O}^q$ of $\mathcal{O}$ with respect to $\mathcal{F}$ is the set $\{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg D_1 \sqcup \neg D_2\}$. Since $\mathcal{O}^q$ does not contain simple definers, we have $\mathcal{V}^q_{ALC} = \mathcal{O}^q$. The two definers $D_1$ and $D_2$ are type 1 definers. Suppose we eliminate $D_1$ first. We perform a LB inference with the clause $\neg A \sqcup \forall r.D_1$ being the first premise, and the clause $\neg D_1 \sqcup \top$ in the second premise. The conclusion is $D_1 \sqcup \forall r^-.\neg A$. We now resolve the obtained conclusion with the clause $\neg D_1 \sqcup \neg D_2$, obtain the clause $\neg D_2 \sqcup \forall r^-.\neg A$, and remove all clauses where $D_1$ occurs. In the second iteration, $D_2$ is eliminated. A LB inference is performed. The first premise of the inference is the clause $\neg A \sqcup \forall s.D_2$, and the second premise is $\neg D_2 \sqcup \forall r^-.\neg A$. The conclusion is $D_2 \sqcup \forall s^-.\neg A_2$, which we resolve with $\neg D_2 \sqcup \forall r^-.\neg A$ to obtain $\forall r^-.\neg A \sqcup \forall s^-.\neg A$. Next, the clauses, where $D_2$ is used, are removed. Since there are no more definers to eliminate, we obtain the query forgetting view $\mathcal{V}^q_{ALCI}$ of $\mathcal{O}$ with respect to $\mathcal{F}$ as the ontology consisting of the clause $\forall r^-.\neg A \sqcup \forall s^-.\neg A$.

Observe that the second premises of the performed LB inferences in the example are not used in computing the conclusion. The LB rule is typically used on the same premises to obtain $n$ conclusions. Each has one definer as top level literal. However, each of the $n$ inferences requires $n - 1$ clauses in the second premise to compute the conclusion.

Type 2 definers are eliminated in the same way as type 1 definers but with an additional step which uses the UB rule to obtain clauses where definers occur as top level negative literals. The computed UB conclusions are then used in the second premise of the LB inferences. One restriction on the UB rule is that $m$ clauses are required in the third premise. Each of them uses a different definer from the ones in the second premise. There may be cases where some but not all clauses of the third premise are available. In these cases, we use the Relaxing rule which may compute in versions of the second premise with less definers in them that can be used in UB inferences.

In the following example LB inferences are only possible when UB inferences have been performed.

*Example 3.* Consider the following ontology $\mathcal{O} = \{A_1 \sqsubseteq \forall r.B_1 \sqcup \forall r.B_2, A_2 \sqsubseteq \forall r.B_3 \sqcup \forall r.B_4, B_1 \sqcap B_3 \sqsubseteq C_1, B_1 \sqcap B_4 \sqsubseteq C_2, B_2 \sqcap B_3 \sqsubseteq C_3, B_2 \sqcap B_4 \sqsubseteq C_4\}$, and forgetting signature $\mathcal{F} = \{B_1, B_2, B_3, B_4\}$. The query reduced ontology $\mathcal{O}^q$ of $\mathcal{O}$ with respect to $\mathcal{F}$ is the set $\{\alpha_1, \alpha_2, \alpha_{1,3}, \alpha_{1,4}, \alpha_{2,3}, \alpha_{2,4}\}$, where $\alpha_1 = \neg A_1 \sqcup \forall r.D_1 \sqcup \forall r.D_2, \alpha_2 = \neg A_2 \sqcup \forall r.D_3 \sqcup \forall r.D_4, \alpha_{1,3} = \neg D_1 \sqcup \neg D_3 \sqcup C_1, \alpha_{1,4} = \neg D_1 \sqcup \neg D_4 \sqcup C_2, \alpha_{2,3} = \neg D_2 \sqcup \neg D_3 \sqcup C_3$, $\alpha_{2,4} = \neg D_2 \sqcup \neg D_4 \sqcup C_4$, and $D_1, D_2, D_3$, and $D_4$ are complex type 2 definers.

The conclusions of any LB inference would be tautologous, because the second premise of each such inference would only use tautologous clauses. For instance, suppose we perform one inference with $\alpha_1$ as first premise and $\neg D_2 \sqcup \top$ in the second premise. The conclusion is $\neg D_1 \sqcup \forall r^-.(\neg A_1 \sqcup \forall r.\top)$ which is a tautology. We can however perform UB inferences and use their conclusions in the second premises of LB inferences.

Four UB inferences can be performed. The first and second inferences use $\alpha_2$ as a first premise. The second premise of the two inferences is $\{\alpha_{1,3}, \alpha_{1,4}\}$ and $\{\alpha_{2,3}, \alpha_{2,4}\}$ respectively. The conclusions are $ub_1 = \neg D_1 \sqcup \forall r^-.\neg A_2 \sqcup C_1 \sqcup C_2$ and $ub_2 = \neg D_2 \sqcup \forall r^-.\neg A_2 \sqcup C_3 \sqcup C_4$, respectively. The third and fourth inferences use $\alpha_1$ as a first premise. The second premise of the two inference is $\{\alpha_{1,3}, \alpha_{2,3}\}$ and $\{\alpha_{1,4}, \alpha_{2,4}\}$ respectively. The conclusions are $ub_3 = \neg D_3 \sqcup \forall r^-.\neg A_1 \sqcup C_1 \sqcup C_3$ and $ub_4 = \neg D_4 \sqcup \forall r^-.\neg A_1 \sqcup C_2 \sqcup C_4$, respectively.

Next, four LB inferences are performed. The first inference has $\alpha_1$ and $ub_2$ as the first and second premises respectively, and concludes $D_1 \sqcup \forall r^-.(\neg A_1 \sqcup \forall r.(\forall r^-.\neg A_2 \sqcup C_3 \sqcup C_4))$. Similarly, the second inference has $\alpha_1$ and $ub_1$, and concludes $D_2 \sqcup \forall r^-.(\neg A_1 \sqcup \forall r.(\forall r^-.\neg A_2 \sqcup C_1 \sqcup C_2))$. The third inference has $\alpha_2$ and $ub_4$, and concludes $D_3 \sqcup \forall r^-.(\neg A_2 \sqcup \forall r.(\forall r^-.\neg A_1 \sqcup C_2 \sqcup C_4))$. The fourth inference has $\alpha_2$ and $ub_3$, and concludes $D_4 \sqcup \forall r^-.(\neg A_2 \sqcup \forall r.(\forall r^-.\neg A_1 \sqcup C_1 \sqcup C_3))$.

Next, we resolve exhaustively on the definers $D_1, D_2, D_3$, and $D_4$. When resolution is complete, the premises of the resolution inferences are removed. The four definers will remain only positively in the clauses $\alpha_1$ and $\alpha_2$, and will be purified by replacing them with $\top$. The final query forgetting view $\mathcal{V}^q_{ALCI}$ will consist of the clauses below.

$$\forall r^-.\neg A_2 \sqcup \forall r^-.(\neg A_1 \sqcup \forall r.(\forall r^-.\neg A_2 \sqcup C_3 \sqcup C_4)) \sqcup C_1 \sqcup C_2$$
$$\forall r^-.\neg A_2 \sqcup \forall r^-.(\neg A_1 \sqcup \forall r.(\forall r^-.\neg A_2 \sqcup C_1 \sqcup C_2)) \sqcup C_3 \sqcup C_4$$
$$\forall r^-.\neg A_1 \sqcup \forall r^-.(\neg A_2 \sqcup \forall r.(\forall r^-.\neg A_1 \sqcup C_2 \sqcup C_4)) \sqcup C_1 \sqcup C_3$$
$$\forall r^-.\neg A_1 \sqcup \forall r^-.(\neg A_2 \sqcup \forall r.(\forall r^-.\neg A_1 \sqcup C_1 \sqcup C_3)) \sqcup C_2 \sqcup C_4$$
$$\forall r^-.(\neg A_2 \sqcup \forall r.(\forall r^-.\neg A_1 \sqcup C_2 \sqcup C_4)) \sqcup \forall r^-.(\neg A_1 \sqcup \forall r.(\forall r^-.\neg A_2 \sqcup C_3 \sqcup C_4)) \sqcup C_1$$
$$\forall r^-.(\neg A_2 \sqcup \forall r.(\forall r^-.\neg A_1 \sqcup C_2 \sqcup C_4)) \sqcup \forall r^-.(\neg A_1 \sqcup \forall r.(\forall r^-.\neg A_2 \sqcup C_1 \sqcup C_2)) \sqcup C_3$$
$$\forall r^-.(\neg A_2 \sqcup \forall r.(\forall r^-.\neg A_1 \sqcup C_1 \sqcup C_3)) \sqcup \forall r^-.(\neg A_1 \sqcup \forall r.(\forall r^-.\neg A_2 \sqcup C_3 \sqcup C_4)) \sqcup C_2$$
$$\forall r^-.(\neg A_2 \sqcup \forall r.(\forall r^-.\neg A_1 \sqcup C_1 \sqcup C_3)) \sqcup \forall r^-.(\neg A_1 \sqcup \forall r.(\forall r^-.\neg A_2 \sqcup C_1 \sqcup C_2)) \sqcup C_4$$

Let us see how $\mathcal{V}^q_{ALCI}$ can be used to compute query answers. Consider the two ABoxes $\mathcal{A}_1$ and $\mathcal{A}_2$ modeled by the two graphs in Figure 4. Lower case letters represent individuals, and upper case labels represent concept assertions. For instance, we should understand from the left figure that $A_2(a_1)$ is an assertion in the ABox. For clarity, we abuse the notation by allowing negative assertions, for instance $\neg C_1(b_1)$. The assertion can be otherwise modeled by adding the axiom $E_2 \sqsubseteq \neg C_2$ to $\mathcal{O}$ (and $\mathcal{V}^q_{ALCI}$), and replacing it with the positive assertion $E_2(u)$. A directed arrow from $u$ to $v$ means that the role assertion $r(u, v)$ is in the ABox. Let us pose the query $q_1(x) = C_1(x)$ to the knowledge bases $(\mathcal{V}^q_{ALCI}, \mathcal{A}_1)$ and $(\mathcal{V}^q_{ALCI}, \mathcal{A}_2)$. The answers of the two knowledge bases to the query are $b_1$ and $b_2$ respectively, which coincide with those of $(\mathcal{O}, \mathcal{A}_1)$ and $(\mathcal{O}, \mathcal{A}_2)$. The answer $b_1$ is obtained from $(\mathcal{V}^q_{ALCI}, \mathcal{A}_1)$ directly from the first clause in the listing above, because $b_1$ is an instance of the negation of the concept $\forall r^-.\neg A_2 \sqcup \forall r^-.(\neg A_1 \sqcup \forall r.(\forall r^-.\neg A_2 \sqcup C_3 \sqcup C_4)) \sqcup C_2$. It follows from $(\mathcal{O}, \mathcal{A})$ because $a_2$ is an instance of $A_1$. As such, the $r$-successor $b_2$ of $a_2$ is either an instance of $B_1$ or $B_2$. Assume

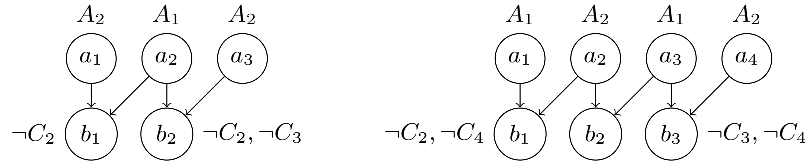

Fig. 4: Graph representation of ABox $\mathcal{A}_1$ (left) and ABox $\mathcal{A}_2$ (right).

it is $B_2$. Since $b_2$ is also a $r$-successor of $a_3$ which is an instance of $A_2$, we get that $b_2$ is either an instance of $B_2 \sqcap B_3$ or $B_2 \sqcap B_4$. From the last two axioms of $\mathcal{O}$, we get that $b_2$ is an instance of $C_3$ or $C_4$. Since neither of them are true, it can only be that all $r$ successors of $a_2$ are instances of $B_1$. One such successor is $b_1$ which is also a successor of $a_1$. Since $a_1$ is an instance of $A_2$, we get that $b_1$ is either an instance of $C_1$ or $C_2$. But we know from the ABox that it is not an instance of $C_2$, hence it is an instance of $C_1$.

The answer $b_2$ follows from $(\mathcal{V}^q_{ALCI}, \mathcal{A}_2)$ directly from the fifth clause of $\mathcal{V}^q_{ALCI}$ because it is an instance of the negation of the concept $\forall r^-.(\neg A_2 \sqcup \forall r.(\forall r^-.\neg A_1 \sqcup C_2 \sqcup C_4)) \sqcup \forall r^-.(\neg A_1 \sqcup \forall r.(\forall r^-.\neg A_2 \sqcup C_3 \sqcup C_4))$. It follows from $(\mathcal{O}, \mathcal{A}_2)$ because if all $r$-successors of $a_3$ are instances of $B_2$, then $b_3$ must be either an instance of $C_3$ or an instance of $C_4$. Since neither are true, we have that all $r$-successors of $a_3$ are instances of $B_1$, and $b_2$ is one such successor. In the same way, all $r$-successors of $a_2$, including $b_2$, must be instances of $B_3$. Hence $b_2$ is an instance of $B_1 \sqcap B_3$, and consequently an instance of $C_1$.

**Theorem 4.** *Let $\mathcal{O}$ be an ontology, $\mathcal{F}$ a forgetting signature of concept names, and $\mathcal{V}^q_{ALCI}$ the ontology computed as explained by the process above. The two ontologies $\mathcal{O}$ and $\mathcal{V}^q_{ALCI}$ are query inseparable with respect to the signature $sig(\mathcal{O}) \backslash \mathcal{F}$. Furthermore, if there are no definers present in $\mathcal{V}^q_{ALCI}$, then $\mathcal{V}^q_{ALCI}$ is a query forgetting view of $\mathcal{O}$ with respect to $\mathcal{F}$.*

## 7 Conclusions

We presented a unified framework for forgetting concept names from $\mathcal{ALC}$ ontologies. The framework computes representations of the semantic, and the deductive forgetting views, and extracts sets of axioms indicating the information not preserved by each, which allows for computing fine-grained forgetting views in-between. We identified the query forgetting view as one such fine-grained view, and we computed two representations of it. An $\mathcal{ALC}$ representation which is also a representation for the deductive forgetting views, and an $\mathcal{ALCI}$ representations which eliminates further definers from the $\mathcal{ALC}$ representation. We also computed a set $\Delta^e$ of axioms indicating the content difference between the $\mathcal{ALC}$ representation of the query forgetting view and the semantic forgetting view.

In future we plan to extend our framework with role forgetting, and evaluate query forgetting in real life ontology based data access applications.

## References

1. Ackermann, W.: Untersuchungen über das Eliminationsproblem der mathematischen Logik. Mathematische Annalen **110**, 390–413 (1935)
2. Alassaf, R., Schmidt, R.A.: Dls-forgetter: An implementation of the dls forgetting calculus for first-order logic. In: Calvanese, D., Iocchi, L. (eds.) GCAI 2019. Proceedings of the 5th Global Conference on Artificial Intelligence. EPiC Series in Computing, vol. 65, pp. 127–138. EasyChair (2019)
3. Alassaf, R., Schmidt, R.A., Sattler, U.: Saturation-based uniform interpolation for multi-agent modal logics. In: Fernández-Duque, D., Palmigiano, A., Pinchinat, S. (eds.) Advances in Modal Logic, Volume 14. pp. 37–58. College Publications, London (2022)
4. Baaz, M., Uwe, E., Leitsch, A.: Normal form transformations. In: Handbook of Automated Reasoning, pp. 275 – 332. North-Holland (2001)
5. Blackburn, P., van Benthem, J., Wolter, F.: Handbook of modal logic, Studies in logic and practical reasoning, vol. 3. Elsevier, 1st edn. (2007)
6. Botoeva, E., Konev, B., Lutz, C., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: Inseparability and conservative extensions of description logic ontologies: A survey. In: Reasoning Web 2016, pp. 27–89. Springer (2017)
7. Botoeva, E., Kontchakov, R., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: Games for query inseparability of description logic knowledge bases. Artificial Intelligence **234**, 78–119 (2016)
8. Botoeva, E., Lutz, C., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: Query inseparability for ALC ontologies. Artificial Intelligence **272**, 1–51 (2019)
9. Delgrande, J.: A Knowledge Level Account of Forgetting. Artificial Intelligence Research **60**, 1165–1213 (2017)
10. Delgrande, J.P., Wang, K.: A Syntax-Independent Approach to Forgetting in Disjunctive Logic Programs. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. pp. 1482–1488. AAAI'15, AAAI Press (2015)
11. Dummett, M.A.E., Lemmon, E.J.: Modal Logics Between S 4 and S 5. Mathematical Logic Quarterly **5**(14-24), 250–264 (1959), `https://onlinelibrary.wiley.com/doi/abs/10.1002/malq.19590051405`
12. Eiter, T., Kern-Isberner, G.: A Brief Survey on Forgetting from a Knowledge Representation and Reasoning Perspective. KI - Künstliche Intelligenz **33**(1), 9–33 (2019)
13. Gabbay, D., Ohlbach, H.J.: Quantifier elimination in second-order predicate logic. In: Proc. KR 1992. pp. 425–435. Morgan Kaufmann (1992)
14. Gabbay, D., Schmidt, R.A., Szałas, A.: Second-Order Quantifier Elimination. College Publications (2008)
15. Ghilardi, S.: An algebraic theory of normal forms. Ann. Pure Appl. Logic **71**(3), 189 – 245 (1995)
16. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? a case for conservative extensions in description logic. In: Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning. pp. 187–197. KR'06, AAAI Press (2006)
17. Glimm, B., Horrocks, I., Lutz, C., Sattler, U.: Conjunctive query answering for the description logic SHIQ. IJCAI International Joint Conference on Artificial Intelligence **31**(1), 399–404 (Jan 2007)
18. Henkin, L.: An Extension of the Craig-Lyndon Interpolation Theorem. The Journal of Symbolic Logic **28**(3), 201–216 (1963)

19. Herzig, A., Mengin, J.: Uniform interpolation by resolution in modal logic. In: Proceedings of the 11th European Conference on Logics in Artificial Intelligence. vol. 5293 LNAI, pp. 219–231. Springer, Berlin, Heidelberg (2008)
20. Konev, B., Kontchakov, R., Ludwig, M., Schneider, T., Wolter, F., Zakharyaschev, M.: Conjunctive query inseparability of OWL 2 QL TBoxes. In: Proceedings of the National Conference on Artificial Intelligence. vol. 1, pp. 221–226. AAAI Press, San Francisco (2011)
21. Konev, B., Lutz, C., Walther, D., Wolter, F.: Model-theoretic inseparability and modularity of description logic ontologies. Artificial Intelligence **203**, 66–103 (2013)
22. Konev, B., Walther, D., Wolter, F.: Forgetting and uniform interpolation in large-scale description logic terminologies. In: Proc. IJCAI 2009. p. 830–835. Morgan Kaufmann (2009)
23. Koopmann, P., Schmidt, R.A.: Implementation and evaluation of forgetting in ALC-ontologies. In: Proc. WoMo 2013. vol. 1081. CEUR-WS.org (2013)
24. Koopmann, P., Schmidt, R.A.: Uniform interpolation of $\mathcal{ALC}$-ontologies using fixpoints. In: Proc. FroCoS 2013. LNAI, vol. 8152, pp. 87–102. Springer (2013)
25. Koopmann, P., Schmidt, R.A.: Count and forget: Uniform interpolation of $\mathcal{SHQ}$-ontologies. In: Proc. IJCAR 2014. LNAI, vol. 8562, pp. 434–448. Springer (2014)
26. Koopmann, P., Schmidt, R.A.: Saturation-based forgetting in the description logic $\mathcal{SIF}$. In: Proc. DL 2015. CEUR (2015)
27. Lang, J., Liberatore, P., Marquis, P.: Propositional Independence - Formula-Variable Independence and Forgetting. Journal of Artificial Intelligence Research **18**, 391–443 (jun 2003)
28. Lin, F., Reiter, R.: Forget it! In: AAAI Fall Symp. on Relevance. pp. 154–159 (1994)
29. Ludwig, M., Konev, B.: Towards practical uniform interpolation and forgetting for $\mathcal{ALC}$ TBoxes. In: Proc. DL. CEUR (2013)
30. Lutz, C., Wolter, F.: Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. Symbolic Computation **45**, 194–228 (02 2010)
31. Lutz, C., Wolter, F.: Foundations for uniform interpolation and forgetting in expressive description logics. In: Proc. IJCAI 2011. p. 989–995 (2011)
32. Matentzoglu, N., Bail, S., Parsia, B.: A snapshot of the owl web. In: Proc. ISWC. pp. 331–346. Springer (2013)
33. Musen, M., Noy, N., Shah, N., Whetzel, P., Chute, C., Story, M.A., Smith, B.: The national center for biomedical ontology. Journal of the American Medical Informatics Association : JAMIA **19**, 190–5 (11 2011)
34. Nonnengart, A., Weidenbach, C.: Computing small clause normal forms. In: Handbook of Automated Reasoning, pp. 335 – 367. North-Holland (2001)
35. Piro, R.: Model Theoretic Characterisations of Description Logics. Ph.D. thesis, Department of Computer Science (2013)
36. Sakr, M., Schmidt, R.A.: Semantic forgetting in expressive description logics. In: Proc. FroCos. pp. 118–136. Springer (2021)
37. Sakr, M., Schmidt, R.A.: Fine-grained forgetting for the description logic alc. In: Proceedings of the 35th International Workshop on Description Logics. CEUR (2022)
38. Visser, A.: Uniform interpolation and layered bisimulation, LNL, vol. 6, p. 139–164. Springer (1996)
39. Wang, K., Wang, Z., Topor, R., Pan, J.Z., Antoniou, G.: Eliminating Concepts and Roles from Ontologies in Expressive Description Logics. Computational Intelligence **30**(2), 205–232 (May 2014)

40. Wang, Z., Wang, K., Topor, R., Pan, J.Z., Antoniou, G.: Uniform Interpolation for ALC Revisited. In: AI 2009: Advances in Artificial Intelligence. pp. 528–537. Springer, Melbourne (2009)
41. Whetzel, P.L., Noy, N., Shah, N.H., Alexander, P.R., Nyulas, C., Tudorache, T., Musen, M.A.: Bioportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. Nucleic Acids Research **39**, W541 – W545 (2011)
42. Zhang, Y., Zhou, Y.: Knowledge forgetting: Properties and applications. Artificial Intelligence **173**, 1525–1537 (2009)
43. Zhang, Y., Zhou, Y.: Forgetting revisited. In: Proc. KR. AAAI Press (2010)
44. Zhao, Y.: Automated Semantic Forgetting For Expressive Description Logics. Ph.D. thesis, University of Manchester (2018)
45. Zhao, Y., Feng, H., Alassaf, R., Del-Pinto, W., Schmidt, R.A.: The FAME Family-A Family of Reasoning Tools for Forgetting in Expressive Description Logics. In: Proceedings of the 30th International Workshop on Description Logics. Montpellier (2017)
46. Zhao, Y., Schmidt, R.A.: Concept forgetting in alcoi-ontologies using an ackermann approach. In: Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P.T., Dumontier, M., Heflin, J., Thirunarayan, K., Staab, S. (eds.) The Semantic Web, 14th International Semantic Web Conference, ISWC 2015. Lecture Notes in Computer Science, vol. 9366, pp. 587–602. Springer (2015)
47. Zhao, Y., Schmidt, R.A.: Forgetting concept and role symbols in $\mathcal{ALCOIH}\mu^+(\nabla,\sqcap)$-ontologies. In: Kambhampati, S. (ed.) Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016). pp. 1345–1352. AAAI Press/IJCAI (2016)
48. Zhao, Y., Schmidt, R.A.: Role forgetting for $\mathcal{ALCOQH}(\nabla)$-ontologies using an ackermann approach. In: Sierra, C. (ed.) Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI'17). pp. 1354–1361. AAAI Press/IJCAI (2017)
49. Zhao, Y., Schmidt, R.A.: On concept forgetting in description logics with qualified number restrictions. In: Lang, J. (ed.) Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI'18). pp. 1984–1990. AAAI Press/IJCAI (2018)

## Proof of Theorem 1

**Lemma 2.** *Let $\mathcal{O}$ be an $\mathcal{ALC}$ ontology, then $\mathcal{O} \equiv^{\mathcal{M}}_{sig(\mathcal{O})} \mathcal{O}^{clausal}$.*

*Proof.* Standard NNF and CNF transformations preserve logical equivalence. It remains to show $\equiv^{\mathcal{M}}_{sig(\mathcal{O})}$ equivalence is preserved by structural transformation. Let $\mathcal{O}_1$ be an ontology and $\mathcal{C} = \mathcal{Q}r.E$ be a concept in $\mathcal{O}_1$ where $\mathcal{Q} \in \{\exists, \forall\}$. After structural transformation we move to a new ontology $\mathcal{O}_2 = \mathcal{O}'_1 \cup \{\neg D \sqcup E\}$ where $\mathcal{O}'_1$ is equal to $\mathcal{O}_1$ but replaces $\mathcal{C}$ with $\mathcal{Q}r.D$ and $D \notin sig(\mathcal{O}_1)$. Assume that $\mathcal{I}$ is a model of $\mathcal{O}_2$. It is straight forward to see that $\mathcal{I}$ is also a model of $\mathcal{O}_1$. For the reverse direction, assume that $\mathcal{I}$ is a model of $\mathcal{O}_1$. We move to a new model $\mathcal{J}$ which extends $\mathcal{I}$ with $D$ and interprets it as $D^{\mathcal{J}} = \{y \in E^{\mathcal{I}} | (x,y) \in r^{\mathcal{I}} \wedge x \in \mathcal{C}^{\mathcal{I}}\}$. Then, we have that $\mathcal{J} \models \mathcal{O}_2$, and $\mathcal{I}$ and $\mathcal{J}$ $sig(\mathcal{O}_0)$-coincide. Altogether, we get that $\mathcal{O}_1 \equiv^{\mathcal{M}}_{sig(\mathcal{O}_1)} \mathcal{O}_2$. Given an ontology $\mathcal{O}$ in NNF and forgetting signature $\mathcal{F}$, we apply the above transformation exhaustively until no forgetting symbol is present under role restriction in $\mathcal{O}$. This gives a finite series of ontologies $\mathcal{O}_i$ where $0 \leq i \leq n$, $\mathcal{O}_0 = \mathcal{O}$, $\mathcal{O}_n = \mathcal{O}^{clausal}$ and $\mathcal{O}_i$ is generated by applying a single structural transformation step on $\mathcal{O}_{i-1}$. This series is finite and is bounded by the number of role restrictions in $\mathcal{O}$. By the above argument we have $\mathcal{O}_0 \equiv^{\mathcal{M}}_{sig(\mathcal{O}_0)} \cdots \equiv^{\mathcal{M}}_{sig(\mathcal{O}_{n-1})} \mathcal{O}_n$. But $sig(\mathcal{O}_i) \subseteq sig(\mathcal{O}_{i+1})$ because the transformation only introduces new symbols. So $\mathcal{O}_0 \equiv^{\mathcal{M}}_{sig(\mathcal{O}_0)} \cdots \equiv^{\mathcal{M}}_{sig(\mathcal{O}_0)} \mathcal{O}_n$. By transitivity it follows that $\mathcal{O}_0 \equiv^{\mathcal{M}}_{sig(\mathcal{O}_0)} \mathcal{O}_n$.

Since by Lemma 2 $\mathcal{O} \equiv^{\mathcal{M}}_{sig(\mathcal{O})} \mathcal{O}^{clausal}$, it suffices to prove that $\mathcal{O}^{clausal} \equiv^{\mathcal{M}}_{sig(\mathcal{O})\setminus\mathcal{F}}$ $\mathcal{O}^{int}$. Recall that $\mathcal{O}^{int}$ is generated from $\mathcal{O}^{clausal}$ by resolving on the forgetting symbols exhaustively followed by performing *purity deletion* with *purification* and *tautology deletion* operations applied eagerly throughout the process. Tautology deletion is a standard equivalence preserving method. Purification preserves the interpretation of the non-purified symbols. Therefore, it suffices to prove the correctness of the purity deletion step since resolution only introduces consequences of $\mathcal{O}^{clausal}$. The proof is adapted from [13]. Let $A$ be a forgetting symbol, then in the normal form of $\mathcal{O}^{clausal}$ $A$ does not occur under role restriction. Let $\mathcal{S}_1 = \{E, C \sqcup A, D \sqcup \neg A, C \sqcup D\}$ be the state before purity deletion where $C \sqcup A$ is a representative of all clauses that contain $A$ positively. This is viable since $A$ appears positively in $\mathcal{O}^{clausal}$ in the clauses $C_i \sqcup A$ for $1 \leq i \leq n$, and these clauses can be rewritten equivalently using the single formula $C \sqcup A$ where $C = \bigsqcap C_i$. In the same way, assume $E$ is a representative of all clauses that do not contain $A$, and $D \sqcup \neg A$ is a representative of all clauses that contain $A$ negatively. $C \sqcup D$ is the resolvent of $C \sqcup A$ and $D \sqcup \neg A$. Let $\mathcal{S}_2 = \{E, C \sqcup D\}$ be the state after purity deletion, omitting $\{C \sqcup A, D \sqcup \neg A\}$ from $\mathcal{S}_1$. Let $\mathcal{I}$ be a model of $\mathcal{S}_1$, then $\mathcal{I}$ is a model of $\mathcal{S}_2$ because $\mathcal{S}_2$ is a subset of $\mathcal{S}_1$. For the reverse direction, suppose that $\mathcal{I}$ is a model of $\mathcal{S}_2$. First, observe that for every $x \in \Delta^{\mathcal{I}}$ we have that $x \in (C \sqcup D)^{\mathcal{I}}$. We extend $\mathcal{I}$ with new concept symbol $A$ interpreted as follows. For every domain element $x \in \Delta^{\mathcal{I}}$:

1. if $x \in C^{\mathcal{I}}$ then $x \notin A^{\mathcal{I}}$, else

2. if $x \in D^{\mathcal{I}}$ then $x \in A^{\mathcal{I}}$

Call the extended model, $\mathcal{J}$. Evidently, $\mathcal{J} \models \mathcal{S}_1$, at the same time, $\mathcal{J}$ interprets every other symbol exactly the same as $\mathcal{I}$. Therefore, $\mathcal{O} \equiv^{\mathcal{M}}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^{int}$.

## Proof of Theorem 2

(iii) is a directly follows from (ii) and Definition 3. If (i) is true, then $\mathcal{O}^d$ and $\mathcal{V}^d$ are deductively inseparable because model inseparability implies deductive inseparability. If we then prove that $\mathcal{O}$ and $\mathcal{O}^d$ are deductively inseparable with respect to $sig(\mathcal{O})\setminus\mathcal{F}$, then we will have (ii) proved, because inseparability relations are transitive.

First we prove (i). The Simple Definer Elimination rule can be seen as a two step operation. The first replaces all clauses of the form $\neg D \sqcup C_i$ with a single clause $\neg D \sqcup C$ where $C = \sqcap C_i$. This step clearly preserves equivalence. The second replaces every $D$ in $\mathcal{O}$ with the concept $C$. This step is the inverse of structural transformation. Therefore, by Lemma 2 we get that $\mathcal{O}^d \equiv^{\mathcal{M}}_{sig(\mathcal{V})} \mathcal{V}$.

Now we prove that $\mathcal{O}$ and $\mathcal{O}^d$ are deductively inseparable with respect to $sig(\mathcal{O})\setminus\mathcal{F}$. Since $\mathcal{O}$ and $\mathcal{O}^{int}$ are model inseparable with respect to $sig(\mathcal{O})\setminus\mathcal{F}$, it suffices to prove that $\mathcal{O}^{int}$ and $\mathcal{O}^d$ are deductively inseparable with respect to $sig(\mathcal{O})\setminus\mathcal{F}$. Also, since $\mathcal{O}^{rp}$ and $\mathcal{O}^{int}$ are logically equivalent, we only need to prove that $\mathcal{O}^{rp}$ and $\mathcal{O}^d$ are deductively inseparable with respect to $sig(\mathcal{O})\setminus\mathcal{F}$.

**Lemma 3.** *The conclusion of the* Role Propagation *rule in Figure 2 is entailed by the premises.*

*Proof.* Let $\mathcal{I}^{int}$ be an arbitrary model of $\mathcal{O}^{int}$ and $d$ be a domain element in $\Delta^{\mathcal{I}^{int}}$. If $d \notin (E_0 \sqcup ... \sqcup E_n)^{\mathcal{I}^{int}}$, then it must be the case that $d \in (\mathcal{Q}r.D_0 \sqcap \forall r.D_1 \sqcap ... \sqcap \forall r.D_n)^{\mathcal{I}^{int}}$. This is equivalent to saying $d \in (\mathcal{Q}r.D_0 \sqcap \forall r.(D_1 \sqcap ... \sqcap D_n))^{\mathcal{I}^{int}}$. Let $\mathcal{Q} = \exists$, then there is $e \in D_0^{\mathcal{I}^{int}}$ such that $(d,e) \in r^{\mathcal{I}^{int}}$. It must also be that $e \in (D_0 \sqcap ... \sqcap D_n)^{\mathcal{I}^{int}}$. Observe that $P_0 \equiv \neg(D_0 \sqcap ... \sqcap D_n)$, so $e \notin P_0^{\mathcal{I}^{int}}$. But since $\mathcal{I}^{int} \models P_0 \sqcup C_0$ we get that $e \in C_0^{\mathcal{I}^{int}}$. Similarly, since $P_j \sqsubseteq P_0$, we have $e \in C_j^{\mathcal{I}^{int}}$. Altogether, $d \in (E_0 \sqcup ... \sqcup E_n \sqcup \exists r.(C_0 \sqcap ... \sqcap C_m))^{\mathcal{I}^{int}}$ for any domain element $d$. If $\mathcal{Q} = \forall$, then $d \in \forall r.(D_0 \sqcap ... \sqcap D_n))^{\mathcal{I}^{int}}$ which is equivalent to saying that $d \in (\forall r.\neg P_0)^{\mathcal{I}^{int}}$. It follows that $d \in (\forall r.C_0)^{\mathcal{I}^{int}}$. Additionally, it must be the case that $d \in (\forall r.\neg P_j)^{\mathcal{I}^{int}}$ because $\forall r.\neg P_j$ subsumes $\forall r.\neg P_0$. So $d \in (\forall r.C_j)^{\mathcal{I}^{int}}$ Altogether, $d \in (E_0 \sqcup ... \sqcup E_n \sqcup \forall r.(C_0 \sqcap ... \sqcap C_m))^{\mathcal{I}^{int}}$ for any domain element $d$.

**Lemma 4.** $\mathcal{O}^{rp} \equiv^{C}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^d$.

*Proof.* We incrementally build the proof through the remaining definitions and lemmas in this section.

**Definition 7.** *Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be any two ontologies. By* mDiff$(\mathcal{O}_1, \mathcal{O}_2)$ *we mean the set of models of $\mathcal{O}_1$ that are not models of $\mathcal{O}_2$.*

**Lemma 5.** *mDiff($\mathcal{O}^{rp}, \mathcal{O}^d$) $= \emptyset$, but in general mDiff($\mathcal{O}^d, \mathcal{O}^{rp}$) $\neq \emptyset$.*

*Proof.* (1) mDiff($\mathcal{O}^{rp}$, $\mathcal{O}^d$) $= \emptyset$: Let $\mathcal{I}$ be any model of $\mathcal{O}^{rp}$. Since $\mathcal{O}^d \subseteq \mathcal{O}^{rp}$, it must be that $\mathcal{I} \models \mathcal{O}^d$.

(2) mDiff($\mathcal{O}^d$, $\mathcal{O}^{rp}$) $\neq \emptyset$: We prove this by giving an example. Let $\mathcal{O}^{rp} = \{\exists r.D_1, \exists r.D_2, \neg D_1 \sqcup \neg D_2\}$ where $D_1$ and $D_2$ are definer symbols. Then, $\mathcal{O}^d = \{\exists r.D_1, \exists r.D_2\}$. Let $\mathcal{I}$ be the interpretation with $\Delta^{\mathcal{I}} = \{a, b\}$, and $D_1^{\mathcal{I}} = D_2^{\mathcal{I}} = \{b\}, r^{\mathcal{I}} = \{(a, b)\}$. Clearly $\mathcal{I}$ is a model of $\mathcal{O}^d$ but is not a model of $\mathcal{O}^{rp}$.

**Lemma 6.** *$\mathcal{O}^{rp} \not\equiv^C_{sig(\mathcal{O})\backslash\mathcal{F}} \mathcal{O}^d$ iff there exists a concept inclusion $\alpha$ over $sig(\mathcal{O})\backslash\mathcal{F}$ such that $\mathcal{O}^{rp} \models \alpha$ and $\mathcal{O}^d \not\models \alpha$.*

*Proof.* The right to left direction is obvious. Left to right: Suppose $\mathcal{O}^{rp} \not\equiv^C_{sig(\mathcal{O})\backslash\mathcal{F}}$ $\mathcal{O}^d$. By definition there must be a concept inclusion $\alpha = C \sqsubseteq D$ over $sig(\mathcal{O})\backslash\mathcal{F}$ such that:

1. $\mathcal{O}^{rp} \models \alpha$ and $\mathcal{O}^d \not\models \alpha$, or
2. $\mathcal{O}^{rp} \not\models \alpha$ and $\mathcal{O}^d \models \alpha$.

Consider the second case, there must be a model $\mathcal{I}$ of $\mathcal{O}^{rp}$ that satisfies $C \sqcap \neg D$. By Lemma 5 $\mathcal{I}$ is a model of $\mathcal{O}^d$ which contradicts the assumption that $\mathcal{O}^d \models \alpha$.

To proceed, we need to define the notions of *bisimulation* and *tree unravelling.*

**Definition 8.** *A* pointed interpretation *$(\mathcal{I}, d)$ is an interpretation $\mathcal{I}$ generated by $d \in \Delta^{\mathcal{I}}$. $(\mathcal{I}, d)$ is a directed graph with the root $d$, and for any $e_1, e_2 \in \Delta^{\mathcal{I}}$, there is a transition, or an edge, from $e_1$ to $e_2$ iff $(e_1, e_2) \in r^{\mathcal{I}}$ where $r \in N_r$.*

**Definition 9.** *Let $(\mathcal{I}, d_1)$ and $(\mathcal{J}, d_2)$ be two pointed interpretations, and $\Sigma$ a signature. $(\mathcal{I}, d_1), (\mathcal{J}, d_2)$ are $\Sigma$-*bisimilar*, in symbols $(\mathcal{I}, d_1) \sim_\Sigma (\mathcal{J}, d_2)$, iff there is a relation $R \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$ where $(d_1, d_2) \in R$ and for every $(d, d') \in R$ the following hold:*

1. *$d \in A^{\mathcal{I}}$ iff $d' \in A^{\mathcal{J}}$ for all concept names $A \in \Sigma$.*
2. *if $(d, e) \in r^{\mathcal{I}}$ then there is $e' \in \Delta^{\mathcal{J}}$ such that $(d', e') \in r^{\mathcal{J}}$ for every role name $r \in \Sigma$ and $(e, e') \in R$.*
3. *if $(d', e') \in r^{\mathcal{J}}$ then there is $e \in \Delta^{\mathcal{I}}$ such that $(d, e) \in r^{\mathcal{I}}$ for every role name $r \in \Sigma$ and $(e, e') \in R$.*

It is well known that bisimilar interpretations coincide on all $\mathcal{ALC}$-consequences. This is stated in the following lemma borrowed from [31].

**Lemma 7.** *[31] Let $(\mathcal{I}, d_1), (\mathcal{J}, d_2)$ be two pointed interpretations, and let $\Sigma$ be some signature. If $(\mathcal{I}, d_1), (\mathcal{J}, d_2)$ are $\Sigma$-bisimilar then for every $\mathcal{ALC}$ concept $C$ where $sig(C) \subseteq \Sigma$ we have that $d_1 \in C^{\mathcal{I}}$ iff $d_2 \in C^{\mathcal{J}}$, in symbols $(\mathcal{I}, d_1) \equiv_\Sigma (\mathcal{J}, d_2)$.*

Every $\mathcal{ALC}$ pointed interpretation $(\mathcal{I}, d)$ can be *unravelled* into a *tree interpretation* $(\mathcal{I}', d)$ [11,5,35] with $\Delta^{\mathcal{I}'}$ defined as follows:

1. $d \in \Delta^{\mathcal{I}'}$; and

2. The word $w = d.r_1.d_1.r_2.d_2.\dots.r_n.d_n$ is in $\Delta^{\mathcal{I}'}$ if and only if there is a path in $(\mathcal{I}, d)$ from $d$ to $d_n$ along the edges $r_i \in N_r$ and the nodes $d_i \in \Delta^{\mathcal{I}}$ where $1 \leq i \leq n$.

For every concept name $A \in N_c \cup N_d$ and role name $r \in N_r$:

1. $A^{\mathcal{I}'} = \{d.r_1 \dots r_n.d_n \in \Delta^{\mathcal{I}'} \mid d_n \in A^{\mathcal{I}}\}$, and
2. $r^{\mathcal{I}'} = \{(w_1, w_1.r.d') \mid w_1, w_1.r.d' \in \Delta^{\mathcal{I}'} \wedge d' \in \Delta^{\mathcal{I}}\}$.

$(\mathcal{I}', d)$ can be seen as a tree whose nodes are the elements of $\Delta^{\mathcal{I}'}$ and edges are the role names in $N_r$. By construction, $(\mathcal{I}', d)$ has the following properties:

1. Every node in $(\mathcal{I}', d)$ has exactly one predecessor, except the root node $d$ which does not have a predecessor.
2. $r^{\mathcal{I}'} \cap s^{\mathcal{I}'} \neq \emptyset$ if and only if $r = s$ for all $r, s \in N_r$.
3. For every $e \in \Delta^{\mathcal{I}}$ there exists $e' \in \Delta^{\mathcal{I}'}$ and vice versa such that $e \in C^{\mathcal{I}}$ if and only if $e' \in C^{\mathcal{I}'}$ where $C$ is an $\mathcal{ALC}$ concept.
4. If $(I, d)$ is a cyclic graph, then $(\mathcal{I}', d)$ is an acyclic tree with infinite depth.

In proving Theorem 4, suppose for the sake of contradiction that $\mathcal{O}^{rp} \not\equiv^{C}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^d$, or by Lemma 6 there is a concept inclusion $\alpha = C \sqsubseteq E$ with $\mathcal{O}^{rp} \models \alpha$ and $\mathcal{O}^d \not\models \alpha$. Let $\mathcal{I}$ be a model of $\mathcal{O}^d$ generated by an arbitrary $d \in C^{\mathcal{I}} \cap \neg E^{\mathcal{I}}$, then $\mathcal{I} \in \text{mDiff}(\mathcal{O}^d, \mathcal{O}^{rp})$. The following Lemma sets a condition on $\mathcal{I}$.

**Lemma 8.** *Suppose $\mathcal{O}^{rp} = \mathcal{O}^d \cup \{\neg D_1 \sqcup \dots \sqcup \neg D_n \sqcup F\}$ where $n > 1$, and let $(\mathcal{I}, d)$ be as above. Then, $\mathcal{I} \in mDiff(\mathcal{O}^d, \mathcal{O}^{rp})$ iff there is $e \in \Delta^{\mathcal{I}}$ that is reachable from $d$ where $e \in D_1^{\mathcal{I}} \cap \dots \cap D_n^{\mathcal{I}}$ and $e \notin F^{\mathcal{I}}$.*

*Proof.* Right to left is obvious. Left to right: Suppose there is no such $e$, then $\mathcal{I} \models \neg D_1 \sqcup \dots \sqcup \neg D_n \sqcup F$. But also $\mathcal{I} \models \mathcal{O}^d$, so we get that $\mathcal{I} \models \mathcal{O}^{rp}$ which contradicts that $\mathcal{I} \in \text{mDiff}(\mathcal{O}^d, \mathcal{O}^{rp})$.

**Definition 10.** *Let $\mathcal{I}$ be a model and $e \in \Delta^{\mathcal{I}}$. Recall that $N_d$ is the set of definer symbols, and define $\mathcal{C}_{\mathcal{I}}(e)$ to be the closure under single negation of the symbols in $N_c \cup N_d$ that contain $e$ in their interpretation.*

**Lemma 9.** *Let $\mathcal{O}^{rp}$, $\mathcal{O}^d$, and $\mathcal{I}$ be defined as in Lemma 8. There is a model $\mathcal{J}$ generated by $d$ such that:*

1. *$\mathcal{J} \models \mathcal{O}^d$;*
2. *$(\mathcal{I}, d) \sim_{sig(\mathcal{O}^d)\setminus N_d} (\mathcal{J}, d)$; and*
3. *There is no $e \in \Delta^{\mathcal{J}}$ such that $e \in D_1^{\mathcal{J}} \cap \dots \cap D_n^{\mathcal{J}} \cap (\neg F)^{\mathcal{J}}$.*

*Proof.* Let $\mathcal{I}_0$ be a model that coincides with $\mathcal{I}$ on everything but reinterprets the definer symbols as follows:

$$(9) \quad D^{\mathcal{I}_0} := D^{\mathcal{I}} \cap \{y \in \Delta^{\mathcal{I}_0} \mid (x, y) \in r^{\mathcal{I}_0} \text{ and } x \notin G^{\mathcal{I}_0} \text{ and } (\mathcal{O}^d \models G \sqcup \exists r.D, \text{ or } \mathcal{O}^d \models G \sqcup \forall r.D)\}$$

where $D \in N_d, r \in N_r$, and $G$ is an $\mathcal{ALC}$ concept.

The idea of (9) is to restrict the elements in the extension of $D$ to the minimum set that is required for $\mathcal{O}^d$. For example, suppose $\mathcal{O}^d = \{\neg A \sqcup \forall r.D_1, \neg B \sqcup \forall r.D_2, \neg D_1 \sqcup \neg D_2, \neg A \sqcup \neg B \sqcup \forall r.\bot\}$. The model $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{a, b\}, A^{\mathcal{I}} = \{a\}, B^{\mathcal{I}} = \{\}, D_1^{\mathcal{I}} = D_2^{\mathcal{I}} = \{b\}, r^{\mathcal{I}} = \{(a, b)\}$ is a tree model of $\mathcal{O}^d$. But $D_2^{\mathcal{I}} = \{b\}$, which violates the intuition that $D_2$ represents the $r$-successors of elements in $B^{\mathcal{I}}$. Re-interpreting $D_2$ using (9) removes $b$ from $D_2^{\mathcal{I}}$. We shall find this necessary when constructing $\mathcal{J}$.

Recall that in $\mathcal{O}^{rp}$ a definer symbol $D$ exists positively only in clauses of the form $C \sqcup \exists r.D$ or $C \sqcup \forall r.D$, and negatively only in clauses of the form $\neg D \sqcup C$. Then, it can be seen that $\mathcal{I}_0$ is a model of $\mathcal{O}^d$ since all clauses of $\mathcal{O}^d$ on the forms of $G \sqcup \exists r.D$ and $G \sqcup \forall r.D$ are true in $\mathcal{I}_0$ by the side conditions of (9). Also, since (9) only removes elements from the extension of $D$, clauses on the form of $\neg D \sqcup H$ remain satisfied in $\mathcal{I}_0$. As (9) only modifies the interpretations of definer symbols, it also follows that $(\mathcal{I}, d) \sim_{sig(\mathcal{O}^d)\setminus N_d} (\mathcal{I}_0, d)$.

We shall now construct a sequence of interpretations $\mathcal{I}_1, \mathcal{I}_2, \ldots$. The interpretations are constructed such that $\mathcal{I}_{k+1}$ is a transformation of $\mathcal{I}_k$ that eliminates one domain element $e \in D_1^{\mathcal{I}_k} \cap ... \cap D_n^{\mathcal{I}_k} \cap (\neg F)^{\mathcal{I}_k}$ where $k \geq 0$, and $(\mathcal{I}_k, d) \sim_{sig(\mathcal{O}^d)\setminus N_d} (\mathcal{I}_{k+1}, d)$. The limit of this sequence is $\mathcal{J}$. It then follows by transitivity of $\sim$ that $(\mathcal{I}, d) \sim_{sig(\mathcal{O}^d)\setminus N_d} (\mathcal{J}, d)$, and there is no $e \in \Delta^{\mathcal{J}}$ such that $e \in D_1^{\mathcal{J}} \cap ... \cap D_n^{\mathcal{J}} \cap (\neg F)^{\mathcal{J}}$.

Suppose there is a domain element $e \in \Delta^{I_k}$ such that $e \in D_1^{\mathcal{I}_k} \cap ... \cap D_n^{\mathcal{I}_k} \cap (\neg F)^{\mathcal{I}_k}$. It is guaranteed by (9) that $e$ has a predecessor, call it $e_{pre}$, such that for every $i$ with $1 \leq i \leq n$ we have $\mathcal{O}^d \models G_i \sqcup \exists r.D_i$ or $\mathcal{O}^d \models G_i \sqcup \forall r.D_i$, and $e_{pre} \notin G_i^{\mathcal{I}_k}$. Note that the case when $\mathcal{O}^d \models G_i \sqcup \forall r.D_i$ for all $i$ with $1 \leq i \leq n$ is not possible since by the construction of $\mathcal{O}^d$ the clause $G_1 \sqcup \cdots \sqcup G_n \sqcup \forall r.F$ must have been introduced in $\mathcal{O}^d$ by the *Role Propagation* rule in Figure 2. Since $e_{pre} \notin G_i^{\mathcal{I}_k}$ with $1 \leq i \leq n$, it must be that $e_{pre} \in (\forall r.F)^{\mathcal{I}_k}$, hence $e \in F^{\mathcal{I}_k}$ which contradicts the hypothesis that $e \notin F^{\mathcal{I}_k}$. For the other cases we transform $\mathcal{I}_k$ to $\mathcal{I}_{k+1}$ as follows:

(I) Suppose that there is an $l$ such that $1 \leq l \leq n$ with $\mathcal{O}^d \models G_l \sqcup \exists r.D_l$ and $\mathcal{O}^d \models G_i \sqcup \forall r.D_i$ where $1 \leq i \leq n$ and $i \neq l$. Again from the *Role Propagation* rule we must have $\mathcal{O}^d \models G_1 \sqcup \ldots G_n \sqcup \exists r.(F \sqcap \bigsqcap_{j=1}^m F_j)$, where $\mathcal{O}^{rp} \models P_j \sqcup F_j$ and $P_j$ is any sub-concept of $\neg D_1 \sqcup \cdots \sqcup \neg D_n$ (these are the second premise of the *Role Propagation* rule). Since $e_{pre} \notin G_i^{\mathcal{I}_k}$ with $1 \leq i \leq n$, it must be that $e_{pre} \in (\exists r.(F \sqcap \bigsqcap_{j=1}^m F_j))^{\mathcal{I}_k}$. That is, there is a child $e'$ of $e_{pre}$ via $r$, such that $e' \in D_i^{\mathcal{I}_k}$ where $1 \leq i \leq n$ and $i \neq l$, and $e' \in F^{\mathcal{I}_k}$. We construct a model $\mathcal{I}_{k+1}$ which is equivalent to $\mathcal{I}_k$ in everything except $D_l$ which it interprets differently: If $e' \in D_l^{\mathcal{I}_k}$, then $D_l^{\mathcal{I}_{k+1}} := D_l^{\mathcal{I}_k} \setminus \{e\}$. If $e' \notin D_l^{\mathcal{I}_k}$, then $D_l^{\mathcal{I}_{k+1}} := (D_l^{\mathcal{I}_k} \cup \{e'\}) \setminus \{e\}$. We note that $\mathcal{I}_{k+1}$ is a model of $\mathcal{O}^d$: (1) By removing $e$ from the interpretation of $D_l$, we need only to worry about clauses of $\mathcal{O}^d$ where $D_l$ appears positively, that is, we show that clauses of the form $G \sqcup \exists r.D_l$ are satisfied at $e_{pre}$. Since by the transformation above $e' \in D_l^{\mathcal{I}_{k+1}}$ and $(e_{pre}, e') \in r^{\mathcal{I}_{k+1}}$, we have $e_{pre} \in (\exists r.D_l)^{\mathcal{I}_{k+1}}$. Therefore $e_{pre} \in (G \sqcup \exists r.D_l)^{\mathcal{J}_d}$ for any $\mathcal{ALC}$ concept $G$. (2) By adding $e'$ to the interpretation of $D_l$, we need only to worry about clauses of $\mathcal{O}^d$ where $D_l$ appears negatively, that is, we show that $e' \in (\neg D_l \sqcup G)^{\mathcal{I}_{k+1}}$ for any

$\mathcal{ALC}$ concept $G$. Since $\neg D_l \sqcup G \in \mathcal{O}^d$, it must have been a premise for the *Role Propagation* rule, i.e., there must be a $j$ such that $1 \leq j \leq m$ such that $G = F_j$. Since $e' \in (F \sqcap \bigsqcap_{j=1}^{m} F_j)^{\mathcal{I}_{k+1}}$, then $e' \in F_j$, consequently $e' \in (\neg D \sqcup G)^{\mathcal{I}_{k+1}}$.

As the transformation only changes the interpretation of $D_l$, it can be seen that $\mathcal{I}_k \sim_{sig(\mathcal{O}^d)\backslash\{D_l\}} \mathcal{I}_{k+1}$.

(II) Suppose that $\mathcal{O}^d \models G_i \sqcup \exists r.D_i$ with $1 \leq i \leq p$ and $2 \leq p \leq n$, and $\mathcal{O}^d \models G_j \sqcup \forall r.D_j$ with $p < j \leq n$. That is, two or more definers in $D_1, \ldots, D_n$ occur under existential role restriction. In this case the *Role Propagation* rule in Figure 2 does not apply. We construct a model $\mathcal{I}_{k+1}$ which is equivalent to $\mathcal{I}_k$ in everything except that it replaces $e$ with fresh domain elements $e_i$ where $1 \leq i \leq p$ such that $e_i \notin D_i^{\mathcal{I}_{k+1}}$, $e_i \in D_j^{\mathcal{I}_{k+1}}$ with $1 \leq j \leq p$ and $j \neq i$, and:

1. $e_i \in A^{\mathcal{I}_{k+1}}$ iff $A \in \mathcal{C}_{\mathcal{I}_k}(e)\backslash\{D_i\}$ for every $A \in N_c \cup N_d$;
2. $(e_{pre}, e_i) \in r^{\mathcal{I}_{k+1}}$;
3. $(e_i, e') \in r^{\mathcal{I}_{k+1}}$ iff $(e, e') \in r^{\mathcal{I}_k}$.

where $r \in N_r$.

First we show that $\mathcal{I}_{k+1} \models \mathcal{O}^d$: (1) As before, by removing $e$ from $\Delta^{\mathcal{I}_{k+1}}$ we need only to show that clauses of $\mathcal{O}^d$ of the form $G_i \sqcup \exists r.D_i$ where $1 \leq i \leq p$ are satisfied at $e_{pre}$. By construction, $e$ is replaced with $e_i$ where $1 \leq i \leq p$ such that $e_i \in D_j^{\mathcal{I}_{k+1}}$ with $1 \leq j \leq p$ and $j \neq i$. So for every $D_i$ with $1 \leq i \leq p$ there are $p-1$ new domain elements $e_j$ such that $e_j \in D_i^{\mathcal{I}_{k+1}}$ and $(e_{pre}, e_j) \in r^{\mathcal{I}_{k+1}}$. Therefore, $e_{pre} \in (\exists r.D_i)^{\mathcal{I}_{k+1}}$ consequently $(G_i \sqcup \exists r.D_i)^{\mathcal{I}_{k+1}}$ for $1 \leq i \leq p$. (2) By introducing new elements $e_i$ such that $e_i \notin D_i^{\mathcal{I}_{k+1}}$ with $1 \leq i \leq p$, we need to show that if $\neg D_i \sqcup G_i \in \mathcal{O}^d$ then $e_i \in G_i^{\mathcal{I}_{k+1}}$. Since $e \in G_i^{\mathcal{I}_k}$, it follows from the first condition of the transformation that $e_i \in G_i^{\mathcal{I}_{k+1}}$. Altogether, $\mathcal{I}_{k+1} \models \mathcal{O}^d$.

Second we prove that $(\mathcal{I}_k, d) \sim_{sig(\mathcal{O}^d)\backslash N_d} (\mathcal{I}_{k+1}, d)$ by induction from bottom to top. Assume that $e$, (hence $e_i$) is reachable from $d$ in $m$ transitions. Let $\Sigma = sig(\mathcal{O}^d)\backslash\{D_1, \ldots, D_n\}$. By construction, for any node $u \in (\mathcal{I}_k, d)$ or $(\mathcal{I}_{k+1}, d)$ such that $u$ is reachable from $d$ in at least $m+1$ transitions, we have $(\mathcal{I}_k, u) \sim_\Sigma (\mathcal{I}_{k+1}, u)$. That is, nothing below $e$ and $e_i$ has changed. Second, for every $u \in \mathcal{I}_k$ at depth $m$ we have: (1) If $u = e$, then by construction $\mathcal{C}_{\mathcal{I}_k}(u)\backslash\{D_i\} = \mathcal{C}_{\mathcal{I}_{k+1}}(e_i)$. Also, since the third condition of the transformation guarantees that everything below $u$ and $e_i$ is the same, it follows that $(\mathcal{I}_k, u) \sim_\Sigma (\mathcal{I}_{k+1}, e_i)$. (2) If $u \neq e$, then $u$ is also a node in $(\mathcal{I}_{k+1}, d)$, hence $(\mathcal{I}_k, u) \sim_\Sigma (\mathcal{I}_{k+1}, u)$. Third, for every $u \in (\mathcal{I}_k, d)$ such that $u$ is reachable in at most $m-1$ transitions from $d$ we have that $u \in (\mathcal{I}_{k+1}, d)$, and we have the following cases: (1) $e \notin (\mathcal{I}_k, u)$. Then, as before $(\mathcal{I}_k, u) \sim_\Sigma (\mathcal{I}_{k+1}, u)$. (2) $e \in (\mathcal{I}_k, u)$. Condition 1 in the transformation guarantees that $\mathcal{C}_{\mathcal{I}_k}(u) = \mathcal{C}_{\mathcal{I}_{k+1}}(u)$ on all $C \in (N_c \cup N_d)$. Suppose that $u \in (\exists \vec{r}.C)^{\mathcal{I}_k}$ where $\exists \vec{r} = \exists r_1.\exists.r_2....\exists.r_l$, $r_i \in N_r$, and $C$ is any $\mathcal{ALC}$ concept over $\Sigma$. Then there must be a path $ur_1u_1r_2u_2r_3...r_lu_l$ in $\mathcal{I}_k$ such that $u_l \in C^{\mathcal{I}_k}$. If $u_i \neq e$ for every $i \in [1..l]$ then by construction this path also exists in $(\mathcal{I}_{k+1}, d)$ and $u \in (\exists \vec{r}.C)^{\mathcal{I}_{k+1}}$. If $u_i = e$ for any $i \in [1..l]$, then again by construction there is a path $ur_1u_1r_2...r_ivr_{i+1}...r_lu_l$ and we have the choice to set $v$ to any of $\{e_1, ..., e_n\}$. In particular, we have $u_l \in C^{\mathcal{I}_{k+1}}$ and $u \in (\exists \vec{r}.C)^{\mathcal{I}_{k+1}}$. We get that

$(\mathcal{I}_k, u) \sim_\Sigma (\mathcal{I}_{k+1}, u)$ for every $u$ reachable in at most $m-1$ transitions from $d$. The same argument can be used to show that every node in $(\mathcal{I}_{k+1}, d)$ has a bisimilar node in $(\mathcal{I}_k, d)$. Altogether, we get that $\mathcal{I}_k \sim_\Sigma (\mathcal{I}_{k+1}, d)$.

In the sequence of constructed models, it follows that:

1. $\mathcal{J} \models \mathcal{O}^d$;
2. $\mathcal{I} \sim_{sig(\mathcal{O}^d)\backslash N_d} \mathcal{J}$; and
3. $\mathcal{J} \not\models D_1 \sqcap D_2 \sqcap ... \sqcap D_n \sqcap \neg F$.

Since $\mathcal{J} \models \mathcal{O}^d$ and there is no $e \in \Delta^\mathcal{J}$ such that $e \in D_1^\mathcal{J} \cap ... \cap D_n^\mathcal{J} \cap (\neg F)^\mathcal{J}$, it follows by Lemma 8 that $\mathcal{J} \models \mathcal{O}^{rp}$. Also, since $\mathcal{I} \sim_{sig(\mathcal{O}^d)\backslash N_d} \mathcal{J}$, and $sig(\mathcal{O}^d)\backslash N_d = sig(\mathcal{O})\backslash \mathcal{F}$, it follows by Lemma 7 that $\mathcal{J} \models C \sqcap \neg E$. But this contradicts the assumption that $\mathcal{O}^{rp} \models C \sqsubseteq E$ implying that the assumption that there is $\alpha = C \sqsubseteq E$ with $\mathcal{O}^{rp} \models \alpha$ and $\mathcal{O}^d \not\models \alpha$ is incorrect, and $\mathcal{O}^{rp} \equiv^C_{sig(\mathcal{O}^d)\backslash N_d} \mathcal{O}^d$.

Finally assume $\mathcal{O}^{rp} = \mathcal{O}^d \cup \Delta$ with $\Delta$ being a set of $n$ clauses of the form $\neg D_1 \sqcup \cdots \sqcup D_n \sqcup F$. We construct $n$ ontologies $\mathcal{O}_i^{rp}$ where $\mathcal{O}_n^{rp} = \mathcal{O}^{rp}$, $\mathcal{O}_k^{rp} = \mathcal{O}_{k-1}^{rp} \cup \{\mathcal{S}\}$ where $\mathcal{S}$ is a clause in $\Delta$ and $\mathcal{O}_0^{rp} = \mathcal{O}^d$. By the above proof we have $\mathcal{O}_k^{rp} \equiv^C_{sig(\mathcal{O}^d)\backslash N_d} \mathcal{O}_{k-1}^{rp}$ for $1 \leq k \leq n$. By transitivity of $\equiv^C$ and because $sig(\mathcal{O}^d)\backslash N_d = sig(\mathcal{O})\backslash\mathcal{F}$ we conclude $\mathcal{O}^{rp} \equiv^C_{sig(\mathcal{O})\backslash\mathcal{F}} \mathcal{O}^d$.

## Proof of Lemma from Section 6.1

Deductive inseparability of $\mathcal{O}$ and $\mathcal{O}^q$ directly follows because $\mathcal{O}^q$ only increments $\mathcal{O}^d$ with axioms from $\Delta^d$. For query inseparability, suppose $\Sigma = sig(\mathcal{O})\backslash\mathcal{F}$ and $\Sigma' = sig(\mathcal{O}^{int})\backslash N_d$. Recall that the method of computing $\mathcal{O}^{int}$ may eliminate clauses following exhaustive resolution, or through the purification rule. Symbols from $sig(\mathcal{O})\backslash\mathcal{F}$ may get eliminated by deletion of redundant clauses or purification. Therefore, there might be non-forgetting symbols in $\Sigma$ not present in $\Sigma'$. As taking account of this technicality would unnecessarily complicate the proof, we assume in this section only that $\mathcal{O}^{int}$ is incremented with the axioms $A \sqsubseteq \top$ for every concept name $A \in \Sigma\backslash\Sigma'$, and $\top \sqsubseteq \forall r.\top$ for every role name $r \in \Sigma\backslash\Sigma'$. The axioms are only added so that the signature of $\mathcal{O}^{int}$ modulo definers coincides with the non-forgetting symbols $sig(\mathcal{O})\backslash\mathcal{F}$ in the original ontology. Because these axioms do not change the content of $\mathcal{O}^{int}$, $\mathcal{O}$ and $\mathcal{O}^{int}$ are model inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.

Let $\mathcal{O}^{rp}$ be $\mathcal{O}^{int}$ extended with the conclusions of the *Role Propagation* inferences performed at line **??** of Algorithm **??**. This implies $\mathcal{O}^q$ is a subset of $\mathcal{O}^{rp}$ and

$$\mathcal{O}^{rp} = \mathcal{O}^q \cup \Delta^q \tag{10}$$

We have the following lemma.

**Lemma 10.** *Consider $\mathcal{O}^{int}$ and $\mathcal{O}^{rp}$ from above. The following are equivalent.*

$$\mathcal{O}^{int} \equiv^Q_{sig(\mathcal{O})\backslash\mathcal{F}} \mathcal{O}^q \tag{11}$$

$$\mathcal{O}^{rp} \equiv^Q_{sig(\mathcal{O})\backslash\mathcal{F}} \mathcal{O}^q \tag{12}$$

*Proof.* We can see that $\mathcal{O}^{rp}$ is logically equivalent to $\mathcal{O}^{int}$ because $\mathcal{O}^{int}$ is a subset of $\mathcal{O}^{rp}$, and the remaining clauses of $\mathcal{O}^{rp}$ are conclusions of the clauses of $\mathcal{O}^{int}$. Recall from Lemma **??** that query inseparability is transitive. We have the following conclusions. The last two proves the lemma.

1. Since $\mathcal{O}^{rp}$ and $\mathcal{O}^{int}$ are logically equivalent as shown above, we see from Lemmas **??** and **??** that $\mathcal{O}^{rp}$ and $\mathcal{O}^{int}$ are model inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.
2. From the previous conclusion and Lemma **??** which says that model inseparability implies query inseparability, we have that $\mathcal{O}^{rp}$ and $\mathcal{O}^{int}$ are query inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.
3. Assume (11) is true, from the previous conclusion and Lemma **??** regarding the transitivity of query inseparability, we conclude that (12) is also true. In other words, (11) implies (12).
4. In the same way of 3, we see that (12) implies (11).

To prove Theorem **??** we prove (12) instead of (11). Suppose we have $q(\vec{x})$ as an arbitrary conjunctive query. We need to prove the following.

1. An answer $\vec{a}$ for $q(\vec{x})$ in $(\mathcal{O}^q, \mathcal{A})$ is also an answer to the same query in $(\mathcal{O}^{rp}, \mathcal{A})$. In symbols, $(\mathcal{O}^q, \mathcal{A}) \models q(\vec{a})$ implies $(\mathcal{O}^{rp}, \mathcal{A}) \models q(\vec{a})$.
2. An answer $\vec{a}$ for $q(\vec{x})$ in $(\mathcal{O}^{rp}, \mathcal{A})$ is also an answer to the same query in $(\mathcal{O}^q, \mathcal{A}))$. In symbols, $(\mathcal{O}^{rp}, \mathcal{A}) \models q(\vec{a})$ implies $(\mathcal{O}^q, \mathcal{A}) \models q(\vec{a})$.

The first statement is always true. It can be proved by contradiction. Suppose that $\vec{a}$ is an answer for $q(\vec{x})$ in $(\mathcal{O}^q, \mathcal{A})$, and there is a model $\mathcal{I}$ of $(\mathcal{O}^{rp}, \mathcal{A})$ where $q(\vec{a})$ is not true. Recall that $\mathcal{I}$ is a model of the knowledge base $(\mathcal{O}^{rp}, \mathcal{A})$ if and only if $\mathcal{I}$ is a model of $\mathcal{O}^{rp}$, and $\mathcal{I}$ is a model of $\mathcal{A}$. We observe from this that $\mathcal{I}$ is a model of $\mathcal{O}^q$, because $\mathcal{O}^q \subseteq \mathcal{O}^{rp}$. But then we have that $\mathcal{I}$ is a model of the knowledge base $(\mathcal{O}^q, \mathcal{A})$. We have now established the following, which yield a contradiction.

1. $\mathcal{I} \not\models q(\vec{a})$ by assumption.
2. $\mathcal{I} \models q(\vec{a})$ because $\mathcal{I} \models (\mathcal{O}^q, \mathcal{A})$ and $(\mathcal{O}^q, \mathcal{A}) \models q(\vec{a})$.

Therefore, there is no such model of $(\mathcal{O}^{rp}, \mathcal{A})$ where $q(\vec{a})$ is not true. In other words, $(\mathcal{O}^q, \mathcal{A}) \models q(\vec{a})$ implies $(\mathcal{O}^{rp}, \mathcal{A}) \models q(\vec{a})$. This proves the following lemma.

**Lemma 11.** *$\mathcal{O}^q$ and $\mathcal{O}^{rp}$ are not query inseparable if and only if $(\mathcal{O}^{rp}, \mathcal{A}) \models q(\vec{a})$ and $(\mathcal{O}^q, \mathcal{A}) \not\models q(\vec{a})$ where $\vec{a}$ is an answer to a query $q(\vec{x})$.*

The following lemma finds the root cause of query inseparability from a model theoretic perspective when $\Delta^q$ consists of one clause.

**Lemma 12.** *Suppose $\Delta^q = \{\neg D_1 \sqcup \cdots \sqcup \neg D_n \sqcup E\}$ where $D_1$ is existential. If we have $\mathcal{O}^{rp} \not\equiv^Q_{sig(\mathcal{O})\backslash\mathcal{F}} \mathcal{O}^q$ then there is a model $\mathcal{I}$ of $(\mathcal{O}^q, \mathcal{A})$ where $D_1 \sqcap \cdots \sqcap D_n \sqcap \neg E$ is satisfied in $\mathcal{I}$.*

*Proof.* We prove the lemma by contradiction. Assume $\mathcal{O}^{rp} \not\equiv^Q_{sig(\mathcal{O})\backslash\mathcal{F}} \mathcal{O}^q$, and $\neg D_1 \sqcup \cdots \sqcup \neg D_n \sqcup E$ is true in every model $\mathcal{I}$ of $(\mathcal{O}^q, \mathcal{A})$. From the first assumption and Lemma 11 we have that $(\mathcal{O}^{rp}, \mathcal{A}) \models q(\vec{a})$ and $(\mathcal{O}^q, \mathcal{A}) \not\models q(\vec{a})$ where $\vec{a}$ is an answer to $q(\vec{x})$. Let $\mathcal{I}$ be an arbitrary model of $(\mathcal{O}^q, \mathcal{A})$ but $q(\vec{a})$ is not true in $\mathcal{I}$. We have that $\mathcal{I}$ is a model of $\mathcal{O}^q$ because $\mathcal{I} \models (\mathcal{O}^q, \mathcal{A})$. Furthermore, $\mathcal{I} \models \neg D_1 \sqcup \cdots \sqcup D_n \sqcup E$ by assumption. Putting both together, we have $\mathcal{I} \models \mathcal{O}^{rp}$. So, $\mathcal{I}$ is a model of the knowledge base $(\mathcal{O}^{rp}, \mathcal{A})$ which yields a contradiction because both the following are true in $\mathcal{I}$.

1. $\mathcal{I} \not\models q(\vec{a})$ by construction of $\mathcal{I}$.
2. $\mathcal{I} \models q(\vec{a})$ because $\mathcal{I} \models (\mathcal{O}^{rp}, \mathcal{A})$ and $(\mathcal{O}^{rp}, \mathcal{A}) \models q(\vec{a})$.

Therefore, either $\mathcal{O}^{rp} \equiv^{Q}_{sig(\mathcal{O})\backslash\mathcal{F}} \mathcal{O}^q$ or $D_1 \sqcap \cdots \sqcap D_n \sqcap \neg E$ is true in $\mathcal{I}$. If $\mathcal{O}^{rp} \not\equiv^{Q}_{sig(\mathcal{O})\backslash\mathcal{F}} \mathcal{O}^q$, it must be that $D_1 \sqcap \cdots \sqcap D_n \sqcap \neg E$ is true in $\mathcal{I}$ for every $\mathcal{I}$ where $q(\vec{a})$ is not true.

We now try to prove that for every model $\mathcal{I}$ of $(\mathcal{O}^q, \mathcal{A})$ where $D_1 \sqcap \cdots \sqcap D_n \sqcap \neg E$ is satisfied, there is a model $\mathcal{J}$ of $(\mathcal{O}^{rp}, \mathcal{A})$ that coincides with $\mathcal{I}$ on all answers to queries over the non-forgotten symbols $sig(\mathcal{O})\backslash\mathcal{F}$. The proving of this statement together with Lemma 11, imply query inseparability between $\mathcal{O}^{rp}$ and $\mathcal{O}^q$ with respect to $sig(\mathcal{O})\backslash\mathcal{F}$. This part is of the proof parallels the proof of Lemma 9, where the model $\mathcal{J}$ was constructed using model transformations applied on $\mathcal{I}$.

There are two concerns that require settling before building this proof. One concern is the data structure used to represent the models of knowledge bases. The other concern is the comparison method that ensures coincidence on the answers to queries. While tree models and bisimulation were used in of Lemma 9, they are not suitable when an ABox is present. A new data structure, namely *pseudo forests*, is needed. Furthermore, the definition of bisimulation is adapted to use pseudo forests instead of pointed interpretations. The following example shows why tree models are not convenient when an ABox is present.

*Example 4.* Consider a knowledge base $\mathcal{K} = (\emptyset, \mathcal{A})$ consisting of the empty ontology and the following ABox $\mathcal{A}$.

(13) $r(a_1, b)$

(14) $r(a_2, b)$

Let $\mathcal{I}$ be a model of $\mathcal{K}$ over the domain $\Delta^{\mathcal{I}} = \{a_1, a_2, b\}$ with the following interpretations.

(15) $a_1^{\mathcal{I}} = a_1$

(16) $a_2^{\mathcal{I}} = a_2$

(17) $b^{\mathcal{I}} = b$

(18) $r^{\mathcal{I}} = \{(a_1, b), (a_2, b)\}$

We can visualize $\mathcal{I}$ with the graph in Figure 5, where domain elements are nodes in the graph, and a directed edge is drawn between two nodes $x$ and $y$ if $(x, y) \in r^{\mathcal{I}}$ for a role name $r$. If we use the tree unravelling algorithm explained in Section **??**, we would obtain the tree model in Figure 6. We can observe that individual $b$ is represented by the two domain elements $a_1.r.b$ and $a_2.r.b$ in the unravelled model. This representation is wrong as the individuals $a_1$ and $a_2$ should be mapped to a single domain element $b$ in the interpretation.

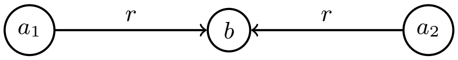

Fig. 5: Graph representation of $\mathcal{I}$

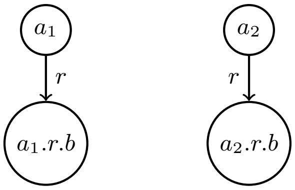

Fig. 6: Tree model obtained from $\mathcal{I}$ by tree unravelling

Pseudo forest can intuitively be described as several separate trees with the following being satisfied.

1. Individuals from the ABox $\mathcal{A}$ occur only as root nodes.
2. Edges connecting different trees can only be between the roots.

For instance, the model $\mathcal{I}$ in Figure 5 is a pseudo forest, because it consists of three tree structures, and each tree comprises only one node which is the root of the tree.

We explain the construction of pseudo forests in the following. We first define the notion of sub-interpretation rooted at an individual $a \in ind(\mathcal{A})$.

**Definition 11.** *Let $\mathcal{K} = (\mathcal{O}, \mathcal{A})$ be a knowledge base, and $\mathcal{I}$ a model of $\mathcal{K}$ where $x^{\mathcal{I}} = x$ for every individual $x \in ind(\mathcal{A})$. Further, let $a \in ind(\mathcal{A})$ be an individual. For a model $\mathcal{J}$, we say $\mathcal{J}$ is a* sub-interpretation of $\mathcal{I}$ rooted at *$a$ if the following are true about $\mathcal{J}$.*

*1. $\Delta^{\mathcal{J}} = \{a\} \cup \{d \in \Delta^{\mathcal{I}} \mid d \notin ind(\mathcal{A})$, there is a path from $a$ to $d$ along $\bigcup_{r \in N_r} r^{\mathcal{I}}\}$,*
*2. $a^{\mathcal{J}} = a$,*
*3. $A^{\mathcal{J}} = A^{\mathcal{I}} \cap \Delta^{\mathcal{J}}$,*
*4. $r^{\mathcal{J}} = r^{\mathcal{I}} \cap (\Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}})$,*

*where in 1 there is a path from $a$ to $d$ along $\bigcup_{r \in N_r} r^{\mathcal{I}}$ if $(a, d) \in r^{\mathcal{I}}$ where $r \in N_r$, or we have $r, r_1, r_2, \ldots r_n \in N_r$ and $e_1, e_2, \ldots e_n \in \Delta^{\mathcal{I}}$ such that $(a, e_1) \in r_1^{\mathcal{I}}$, $(e_i, e_{i+1}) \in r_{i+1}^{\mathcal{I}}$, and $(e_n, d) \in r^{\mathcal{I}}$, where $n \geq 1$ and $1 \leq i \leq n-1$.*

*By $\mathcal{M}$ we denote the set of all sub-interpretations of $\mathcal{I}$ rooted at the individuals from $ind(\mathcal{A})$.*

Recall from Section **??** the tree unravelling algorithm for constructing tree interpretations. We use this algorithm to construct a tree interpretation for each sub-interpretation in $\mathcal{M}$.

**Definition 12.** *Let $\mathcal{I}$ be a model of a knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{A})$ as in Definition 11. Let $\mathcal{M}$ be the set of all sub-interpretations of $\mathcal{I}$ rooted at the individuals from $ind(\mathcal{A})$. Let $tree(\mathcal{I})$ be a function that unravels an interpretation $\mathcal{I}$ into a tree interpretation $\mathcal{I}^t$. By $\mathcal{M}^t = \{\mathcal{I}^t = tree(\mathcal{I}) \mid \mathcal{I} \in \mathcal{M}\}$ we denote the set of tree interpretations corresponding to the sub-interpretations in $\mathcal{M}$.*

A pseudo forest is defined using Definition 12 as follows.

**Definition 13.** *Let $\mathcal{I}$ be a model of a knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{A})$ as in Definitions 11 and 12. Let $\mathcal{M}^t$ be the set of tree interpretations constructed from the sub-interpretations of $\mathcal{I}$ rooted at the individuals from $ind(\mathcal{A})$. Suppose there are $n$ tree models $\mathcal{I}_1, \ldots, \mathcal{I}_n$ with $n \geq 1$ in $\mathcal{M}^t$.* A pseudo forest model *$\mathcal{J}$ is the model constructed from the tree models in $\mathcal{M}^t$ as follows. The operator $\sqcup$ used below means the disjoint union.*

1. *$\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}_1} \sqcup \cdots \sqcup \Delta^{\mathcal{I}_n}$.*
2. *$A^{\mathcal{J}} = A^{\mathcal{I}_1} \sqcup \cdots \sqcup A^{\mathcal{I}_n}$ for every concept name $A \in N_c \cup N_d$.*
3. *$r^{\mathcal{J}} = r^{\mathcal{I}_1} \sqcup \cdots \sqcup r^{\mathcal{I}_n} \cup \{(x, y) \in r^{\mathcal{I}} \mid x, y \in ind(\mathcal{A})\}$ for every role name $r \in N_r$.*

We explain the above definition of a pseudo forest with the following example.

*Example 5.* Consider the following ontology $\mathcal{O}$.

$$A_1 \sqsubseteq \exists r.B \tag{19}$$

$$A_2 \sqsubseteq \exists r.B \tag{20}$$

Let $\mathcal{A}$ be the following ABox.

$$A_1(a_1) \tag{21}$$

$$A_2(a_2) \tag{22}$$

Suppose we have a model $\mathcal{I}$ of the knowledge base $(\mathcal{O}, \mathcal{A})$. Assume the domain of interpretation of $\mathcal{I}$ is the set $\{a_1, a_2, b\}$, and the interpretation function is defined by the following.

$$a_1^{\mathcal{I}} = a_1 \tag{23}$$

$$a_2^{\mathcal{I}} = a_2 \tag{24}$$

$$A_1^{\mathcal{I}} = \{a_1\} \tag{25}$$

$$A_2^{\mathcal{I}} = \{a_2\} \tag{26}$$

$$B^{\mathcal{I}} = \{b\} \tag{27}$$

$$r^{\mathcal{I}} = \{(a_1, b), (a_2, b), (a_1, a_2)\} \tag{28}$$

The model $\mathcal{I}$ is visualized in the lift graph of Figure 7. Let $\mathcal{I}_1$ be the sub-interpretation of $\mathcal{I}$ rooted at $a_1$, and $\mathcal{I}_2$ the sub-interpretation of $\mathcal{I}$ rooted at $a_2$. They are defined as follows.

$$\begin{aligned}
\Delta^{\mathcal{I}_1} &= \{a_1, b\} & \Delta^{\mathcal{I}_2} &= \{a_2, b\} \\
a_1^{\mathcal{I}_1} &= a_1 & a_2^{\mathcal{I}_2} &= a_2 \\
A_1^{\mathcal{I}_1} &= \{a_1\} & A_1^{\mathcal{I}_2} &= \emptyset \\
A_2^{\mathcal{I}_1} &= \emptyset & A_2^{\mathcal{I}_2} &= \{a_2\} \\
B^{\mathcal{I}_1} &= \{b\} & B^{\mathcal{I}_2} &= \{b\} \\
r^{\mathcal{I}_1} &= \{(a_1, b)\} & r^{\mathcal{I}_2} &= \{(a_2, b)\}
\end{aligned}$$

The two models $\mathcal{I}_1$ and $\mathcal{I}_2$ are tree models. So tree unravelling is not required in this case, and we may define the set $\mathcal{M}^t$ as the set consisting of $\mathcal{I}_1$ and $\mathcal{I}_2$. The pseudo forest model $\mathcal{J}$ is constructed as the disjoint union of $\mathcal{I}_1$ and $\mathcal{I}_2$. This means that common domain elements of $\mathcal{I}_1$ and $\mathcal{I}_2$ are copied twice to $\Delta^{\mathcal{J}}$ as different elements. Let $a_1.r.b$ be the image of $b \in \Delta^{\mathcal{I}_1}$ in $\Delta^{\mathcal{J}}$, and $a_2.r.b$ be the image of $b \in \Delta^{\mathcal{I}_2}$ in $\Delta^{\mathcal{J}}$. The pseudo forest model $\mathcal{J}$ is defined as follows.

$$\Delta^{\mathcal{J}} = \{a_1, a_2, a_1.r.b, a_2.r.b\} \tag{29}$$

$$a_1^{\mathcal{J}} = a_1 \tag{30}$$

$$a_2^{\mathcal{J}} = a_2 \tag{31}$$

$$A_1^{\mathcal{J}} = \{a_1\} \tag{32}$$

$$A_2^{\mathcal{J}} = \{a_2\} \tag{33}$$

$$B^{\mathcal{J}} = \{a_1.r.b, a_2.r.b\} \tag{34}$$

$$r^{\mathcal{J}} = \{(a_1, a_1.r.b), (a_2, a_2.r.b), (a_1, a_2)\} \tag{35}$$

The pseudo forest model $\mathcal{J}$ is depicted in the right graph of Figure 7.

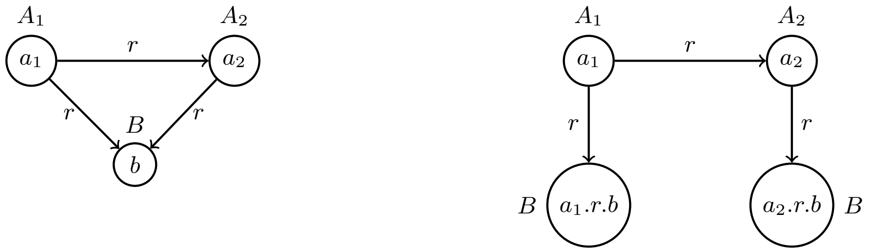

Fig. 7: Tree model obtained from $\mathcal{I}$ by tree unravelling

Definitions 11, 12, and 13 require that individuals are interpreted as themselves. Thus, we make the unique name assumption [8]. It is known from the literature that query answering in $\mathcal{ALC}$ is not impacted by the unique name assumption. The following lemma allows us to consider only pseudo forest models. The lemma is known from the literature, and the proof can be found in [17,7,8] for example.

**Lemma 13.** *Let $\mathcal{K} = (\mathcal{O}, \mathcal{A})$ be a knowledge base, and $q(\vec{x})$ a conjunctive query. $\mathcal{K} \not\models q(\vec{a})$ if and only if there is a pseudo forest model $\mathcal{I}$ of $\mathcal{K}$ but $\mathcal{I} \not\models q(\vec{a})$, for any tuple $\vec{a}$ over ind$(\mathcal{A})$ with the same length as $\vec{x}$.*

We discuss bisimulation as a comparison method that ensures coincidence of models on answers to queries. Definition 9 of bisimulation considers pointed interpretations which have one root element. We adapt the definition to pseudo forests.

**Definition 14.** *Let $\gamma$ be a set of individual names, and $\Sigma$ a signature. Suppose $\mathcal{I}$ and $\mathcal{J}$ are two pseudo forest models. $\mathcal{I}$ and $\mathcal{J}$ are* bisimilar with respect to $\Sigma$ and $\gamma$, *in symbols $\mathcal{I} \sim^{\gamma}_{\Sigma} \mathcal{J}$, if and only if there is a relation $R \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$ where $(a, a) \in R$ for every individual in $\gamma$, and for every $(d, d') \in R$ the following hold:*

1. *$d \in A^{\mathcal{I}}$ iff $d' \in A^{\mathcal{J}}$ for all concept names $A \in \Sigma$.*
2. *if $(d, e) \in r^{\mathcal{I}}$ then there is $e' \in \Delta^{\mathcal{J}}$ such that $(d', e') \in r^{\mathcal{J}}$ for every role name $r \in \Sigma$ and $(e, e') \in R$.*
3. *if $(d', e') \in r^{\mathcal{J}}$ then there is $e \in \Delta^{\mathcal{I}}$ such that $(d, e) \in r^{\mathcal{I}}$ for every role name $r \in \Sigma$ and $(e, e') \in R$.*

Bisimulation ensures coincidence on answers to arbitrary conjunctive queries. The following lemma proves this statement.

**Lemma 14.** *Let $\mathcal{I}$ and $\mathcal{J}$ be two bisimilar models with respect to a set $\Sigma$ of symbols, and a set $\gamma$ of individuals. Suppose $q(\vec{x})$ is a conjunctive query where $sig(q) \subseteq \Sigma$, and $\vec{a}$ is an answer to $q(\vec{x})$ where $\vec{a} \subseteq \gamma$. Then $q(\vec{a})$ is true in $\mathcal{I}$ if and only if $q(\vec{a})$ is true in $\mathcal{J}$ for any answer $\vec{a}$ of $q(\vec{x})$.*

*Proof.* Suppose $\mathcal{I} \models q(\vec{a})$, then there is a mapping $\pi$ that maps each anonymous variable in $q$ to an element in $\Delta^{\mathcal{I}}$, and each answer variable $x$ at position $i$ in $\vec{x}$ to an individual in $a$ at the same position in $\vec{a}$. Furthermore, for every two variables $u$ and $v$ occurring in $q$, the following must be true.

1. If $A(u)$ is an atom in $q$, then $\pi(u) \in A^{\mathcal{I}}$, for any $A \in \Sigma$.
2. If $r(u, v)$ is an atom in $q$, then $(\pi(u), \pi(v)) \in r^{\mathcal{I}}$, for any $r \in \Sigma$.

Let $\sigma$ be a function that maps a domain element from $\Delta^{\mathcal{I}}$ to a bisimilar element in $\Delta^{\mathcal{J}}$. The function $\sigma$ is a total function, because each domain element in $\Delta^{\mathcal{I}}$ is either an individual $a$ which has a map in $R$, or an element reachable from $a$ through $\bigcup r^{\mathcal{I}}$ for every $r \in N_r$. The latter elements are guaranteed to have a map in $R$ by Conditions 2 and 3 of Definition 14.

Consider the mapping $\beta = \sigma \circ \pi$ which is the composition of $\pi$ and $\sigma$. We can see that following about $\beta$.

1. $\beta$ maps each answer variable $x$ at position $i$ in $\vec{x}$ to an individual in $a$ at the same position in $\vec{a}$, because $a = \pi(x)$ and $a = \sigma(a)$.

2. $\beta$ maps each anonymous variable occurring in $q$ to a domain element in $\Delta^{\mathcal{J}}$, because $\sigma$ is a total function.
3. If $A(u)$ is an atom in $q$, then $\beta(u) \in A^{\mathcal{J}}$, because $\pi(u) \in A^{\mathcal{I}}$, and $\pi(u)$ must be mapped through $\sigma$ to an element in the extension of $A$ in $\mathcal{J}$.
4. If $r(u, v)$ is an atom in $q$. then $(\beta(u), \beta(v)) \in r^{\mathcal{J}}$, because $\pi(u)$ must be mapped through $\sigma$ to an element $d \in \Delta^{\mathcal{J}}$, and $v$ to an element $e$ where $(d, e) \in r^{\mathcal{J}}$.

So, if $q(\vec{a})$ is true in $\mathcal{I}$, then $q(\vec{a})$ is true in $\mathcal{J}$. The same argument shows that if $q(\vec{a})$ is true in $\mathcal{J}$, then $q(\vec{a})$ is true in $\mathcal{I}$.

*Example 6.* Consider the following ontology $\mathcal{O}$.

(36) $$A \sqsubseteq \forall r.B_1 \sqcup \forall r.B_2$$

(37) $$B_1 \sqcup B_2 \sqsubseteq \exists s.C$$

(38) $$B_1 \sqcap B_2 \sqsubseteq \bot$$

and the following ABox $\mathcal{A}$

(39) $$A(a)$$

(40) $$r(a, b)$$

Two pseudo forest models $\mathcal{I}, \mathcal{J}$ of the knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{A})$ are depicted in Figure 8. Both models are interpretations over the domain $\{a, b, c\}$, where $a, b \in ind(\mathcal{A})$. Let $q_1(x)$ and $q_2(x)$ be the following queries.

(41) $$q_1(x) = \exists y, z.A(x) \wedge r(x, y) \wedge s(y, z) \wedge C(z)$$

(42) $$q_2(x) = B_1(x)$$

The two models coincide on the answer 'a' for $q_1$ because they are bisimilar with respect to the symbols $\{A, C, r, s\}$ and the individuals $\{a, b\}$. The bisimulation relation is $R = \{(a, a), (b, b), (c, c)\}$. However the two models do not coincide on their answers to $q_2$ as $\mathcal{I} \models B_1(b)$ but no answers for $q_2$ exist in $\mathcal{J}$. The models $\mathcal{I}$ and $\mathcal{J}$ are different on the answers of $q_2(x)$ because they are not bisimilar with respect to any signature containing $B_1$ and the individuals $\{a, b\}$.

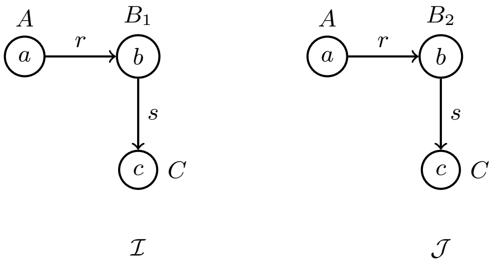

Fig. 8: $I_1$ and $I_2$ are bisimilar with respect to $\{A, r\}$ and $\{a, b\}$.

The following lemma asserts important properties of bisimulation.

**Lemma 15.** *Bisimulation is a transitive operation. Furthermore, if $\Sigma_1$ and $\Sigma_2$ are two sets of symbols such that $\Sigma_1 \subseteq \Sigma_2$ then $\mathcal{I} \sim^{\gamma}_{\Sigma_2} \mathcal{J}$ implies $\mathcal{I} \sim^{\gamma}_{\Sigma_1} \mathcal{J}$, where $\mathcal{I}$ and $\mathcal{J}$ are pseudo forest models, and $\gamma$ is a set of individuals.*

We have shown by Lemma 11 that $\mathcal{O}^{rp}$ and $\mathcal{O}^{q}$ are query separable if $(\mathcal{O}^{rp}, \mathcal{A}) \models q(\vec{a})$ and $(\mathcal{O}^{q}, \mathcal{A}) \not\models q(\vec{a})$, where $q(\vec{x})$ is a conjunctive query over the non-forgotten symbols $sig(\mathcal{O})\backslash\mathcal{F}$, and $\vec{a}$ is an answer to $q(\vec{x})$. When $\Delta^q$ contains only the clause $\neg D_1 \sqcup \cdots \sqcup \neg D_n \sqcup E$ where $D_1$ is an existential definer, Lemma 12 showed that $\mathcal{O}^{rp}$ and $\mathcal{O}^{q}$ are query separable if and only if there is a model $\mathcal{I}$ of $(\mathcal{O}^{q}, \mathcal{A})$ where $D_1 \sqcap \cdots \sqcap D_n \sqcap \neg E$ is true. We may assume that $\mathcal{I}$ is a pseudo forest model by Lemma 13. We show that there is a model $\mathcal{J}$ of $(\mathcal{O}^{rp}, \mathcal{A})$ that coincides with $\mathcal{I}$ on all answers to queries over the non-forgotten symbols $sig(\mathcal{O})\backslash\mathcal{F}$. This will cause a contradiction because the following will be true about $\mathcal{J}$.

1. $\mathcal{J} \not\models q(\vec{a})$, because $\mathcal{J}$ coincides with $\mathcal{I}$ on all answers to queries over $sig(\mathcal{O})\backslash\mathcal{F}$.
2. $\mathcal{J} \models q(\vec{a})$ because $\mathcal{J}$ is a model of $(\mathcal{O}^{rp}, \mathcal{A})$ and $(\mathcal{O}^{rp}, \mathcal{A}) \models q(\vec{a})$.

The contradiction implies that $\mathcal{O}^{rp}$ and $\mathcal{O}^{q}$ are query inseparable. The case when $\Delta^q$ contains $n$ clauses where $n$ is arbitrary, can be generalized from the above by induction. We construct a sequence of $n$ ontologies where each ontology excludes only one clause from $\Delta^q$ such that the first ontology coincides with $\mathcal{O}^{rp}$ and the last coincides with $\mathcal{O}^{q}$. The above argument shows that every two consecutive ontologies in this sequence are query inseparable, and by transitivity of query inseparability (see Lemma **??** we prove query inseparability of $\mathcal{O}^{rp}$ and $\mathcal{O}^{q}$ with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.

The central part as discussed above is finding the model $\mathcal{J}$. We construct $\mathcal{J}$ from $\mathcal{I}$ by eliminating all occurrences of domain elements satisfying $D_1 \sqcap \cdots \sqcap D_n \sqcap \neg E$. Suppose there are $m$ domain elements in $\mathcal{I}$ satisfying $D_1 \sqcap \cdots \sqcap D_n \sqcap \neg E$. We construct $m$ models $\mathcal{I}_1, \mathcal{I}_2, \ldots$ where each model eliminates exactly one domain element satisfying $D_1 \sqcap \cdots \sqcup D_n \sqcap \neg E$ from the previous model, where $\mathcal{I}_0 = \mathcal{I}$ and $I_m = \mathcal{J}$. Each two consecutive models in this sequence are bisimilar with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$. By transitivity of bisimulation, we obtain that $\mathcal{I}$ and $\mathcal{J}$ are bisimilar with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$.

**Lemma 16.** *Let $\mathcal{K} = (\mathcal{O}, \mathcal{A})$ be a knowledge base. Let $\mathcal{O}^{rp}$ and $\mathcal{O}^{q}$ be as in Lemma 12 where $\mathcal{O}^{rp}\backslash\mathcal{O}^{q} = \{\neg D_1 \sqcup \cdots \sqcup \neg D_n \sqcup E\}$, $D_1, \ldots, D_n$ are definers, and $D_1$ is an existential definer. Suppose $\mathcal{I}$ is a pseudo forest model of $(\mathcal{O}^{q}, \mathcal{A})$ and $D_1 \sqcap \cdots \sqcap D_n \sqcap \neg E$ is true in $\mathcal{I}$. There is a model $\mathcal{J}$ with the following properties:*

*1. $\mathcal{J} \models \mathcal{O}^{q}$;*
*2. There is no $e \in \Delta^{\mathcal{J}}$ such that $e \in D_1^{\mathcal{J}} \cap \ldots \cap D_n^{\mathcal{J}} \cap (\neg E)^{\mathcal{J}}$; and*
*3. $\mathcal{I}$ and $\mathcal{J}$ are bisimilar with respect to $(sig(\mathcal{O})\backslash\mathcal{F})$ and $ind(\mathcal{A})$.*

*Proof.* Recall from the normal form of $\mathcal{O}^{q}$ that a definer occurs with positive polarity only below existential role restrictions, or only below universal role restrictions. It also occurs with negative polarity only as a top level negated disjunct. Let $\mathcal{I}_0$ be a model that coincides with $\mathcal{I}$ on everything but reinterprets every definer $D$ as below. We assume $\mathcal{O}^q \models G \sqcup \exists r.D$ or $\mathcal{O}^q \models G \sqcup \forall r.D$ where $G$ is an $\mathcal{ALC}$ concept. If $\mathcal{O}^q$ has several consequences of these forms, e.g., $\mathcal{O}^q \models G_i \sqcup \exists r.D$ with $1 \leq i \leq n$ and $n > 1$, then we assume $G$ is the conjunction of all such $G_i$. The definer $D$ is interpreted in $\mathcal{I}_0$ by the following.

$$D^{\mathcal{I}_0} := D^{\mathcal{I}} \cap \{y \in \Delta^{\mathcal{I}} \mid \exists x.(x, y) \in r^{\mathcal{I}} \text{ and } x \notin G^{\mathcal{I}} \text{ where } (\mathcal{O}^q \models G \sqcup \exists r.D, \text{ or } \mathcal{O}^q \models G \sqcup \forall r.D)\} \tag{43}$$

where $D \in N_d, r \in N_r$, and $G$ is an $\mathcal{ALC}$ concept.

The idea of (43) is to restrict the elements in the extension of $D$ to the minimum set required to satisfy $(\mathcal{O}^q, \mathcal{A})$. For example, suppose $\mathcal{O}^q$ is the following ontology.

$$\neg A \sqcup \forall r.D_1 \tag{44}$$

$$\neg B \sqcup \forall r.D_2 \tag{45}$$

$$\neg A \sqcup \neg B \sqcup \forall r.\bot \tag{46}$$

and $\mathcal{A}$ is the following ABox.

$$A(a) \tag{47}$$

$$r(a, b) \tag{48}$$

The model $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{a, b\}, A^{\mathcal{I}} = \{a\}, B^{\mathcal{I}} = \{\}, D_1^{\mathcal{I}} = D_2^{\mathcal{I}} = \{b\}, r^{\mathcal{I}} = \{(a, b)\}$ is a model of $(\mathcal{O}^q, \mathcal{A})$. That $D_2^{\mathcal{I}} = \{b\}$, violates the intuition that $D_2$ represents the $r$-successors of elements in $B^{\mathcal{I}}$. Re-interpreting $D_2$ using (43) removes $b$ from $D_2^{\mathcal{I}}$, and constructs $\mathcal{I}_0$ as the model over $\{a, b\}$ with $A^{\mathcal{I}_0} = \{a\}, B^{\mathcal{I}_0} = D_2^{\mathcal{I}_0} = \{\}, D_1^{\mathcal{I}_0}\{b\}, r^{\mathcal{I}_0} = \{(a, b)\}$.

Reinterpreting definers using (43) allows assuming information about predecessors of $e$ in the graph structure of $\mathcal{I}_0$, and simplifies the model transformations used below. (43) has two important properties. The first is that the obtained interpretation $\mathcal{I}_0$ is a model of $(\mathcal{O}^q, \mathcal{A})$. The second is that $\mathcal{I}$ and $\mathcal{I}_0$ are bisimilar with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and the individuals $ind(\mathcal{A})$. The first property holds because all clauses of $\mathcal{O}^q$ of the forms $G \sqcup \exists r.D$ or $G \sqcup \forall r.D$ are true in $\mathcal{I}_0$ by the side conditions of (43). Also, since (43) only removes elements from the extension of $D$, clauses of the form $\neg D \sqcup H$, where $H$ is an $\mathcal{ALC}$ concept, remain satisfied in $\mathcal{I}_0$. The second property holds because only the interpretations of definers are modified by (43).

We now construct a sequence of interpretations $\mathcal{I}_1, \mathcal{I}_2, \ldots$. The interpretations are constructed such that $\mathcal{I}_{k+1}$ eliminates one domain element $e \in D_1^{\mathcal{I}_k} \cap \ldots \cap D_n^{\mathcal{I}_k} \cap (\neg E)^{\mathcal{I}_k}$ where $k \geq 0$, and $\mathcal{I}_k$ and $\mathcal{I}_{k+1}$ are bisimilar with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$. The limit of this sequence is $\mathcal{J}$. The construction considers the following six conditions.

1. $e$ is an individual from $ind(\mathcal{A})$.

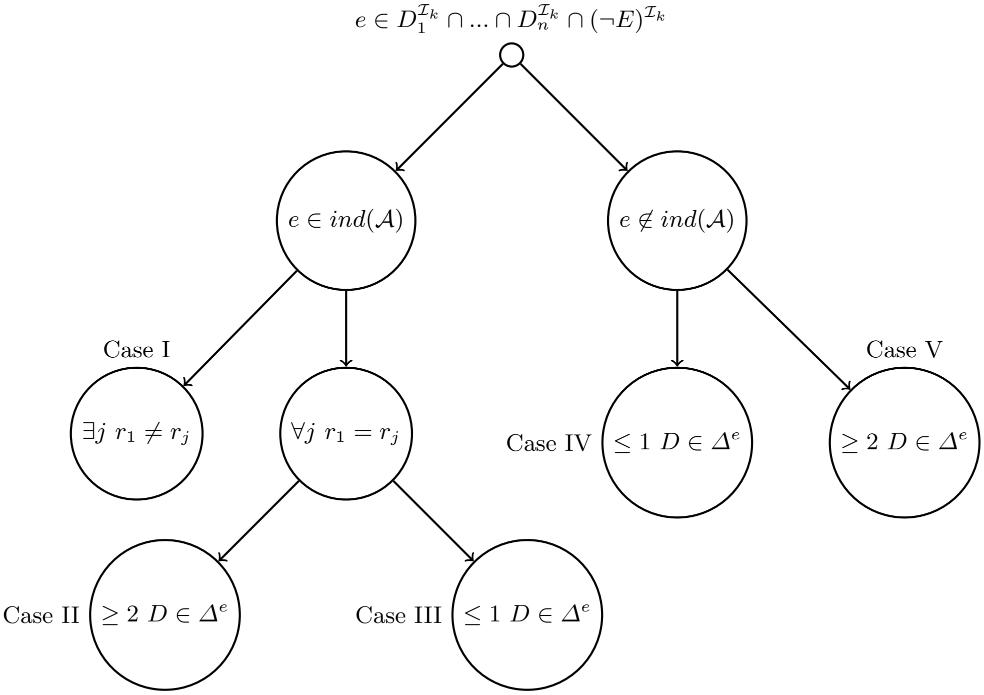
$e \in D_1^{\mathcal{I}_k} \cap ... \cap D_n^{\mathcal{I}_k} \cap (\neg E)^{\mathcal{I}_k}$
$e \in ind(\mathcal{A})$
$e \notin ind(\mathcal{A})$
Case I
$\exists j \; r_1 \neq r_j$
$\forall j \; r_1 = r_j$
Case IV
$\leq 1 \; D \in \Delta^e$
Case V
$\geq 2 \; D \in \Delta^e$
Case II
$\geq 2 \; D \in \Delta^e$
Case III
$\leq 1 \; D \in \Delta^e$

Fig. 9: Cases satisfied by $e$ in the construction of $\mathcal{I}_{k+1}$

2. $e$ is not an individual from $ind(\mathcal{A})$.
3. There is a $j$ where $1 \leq j \leq n$ and $r_1 \neq r_j$. So the definers $D_1, D_2, \ldots, D_n$ do not occur positively below the same role in $\mathcal{O}^q$.
4. For every $j$ where $1 \leq j \leq n$ and $r_1 = r_j$. So the definers $D_1, D_2, \ldots, D_n$ occur positively below the same role in $\mathcal{O}^q$.
5. Two or more definers from $D_1, D_2, \ldots, D_n$ are existential.
6. Only one definer, which is $D_1$, from $D_1, D_2, \ldots, D_n$ is existential.

Five cases are obtained as different combinations of the above conditions. Figure 9 explains them in more detail. Case I is a combination of conditions 1, and 3. Case II is a combination of conditions 1, 4, and 5. Case III is a combination of conditions 1, 4, and 6. Case IV is a combination of conditions 2, and 6. Case V is a combination of conditions 2, and 5. We consider all five cases. A transformation which ensures $e \notin D_1^{\mathcal{I}_{k+1}} \cap ... \cap D_n^{\mathcal{I}_{k+1}} \cap (\neg E)^{\mathcal{I}_{k+1}}$ is shown for each case.

Suppose first that $e \in ind(\mathcal{A})$, and consider the cases I, II, and III. By 43, it is guaranteed that $e$ has $n$ predecessors $e_{pre}^i$ such that $\mathcal{O}^q \models G \sqcup \mathcal{Q}r_i.D_i$, $e_{pre}^i \notin G_i^{\mathcal{I}_k}$, and $(e_{pre}^i, e) \in r_i^{\mathcal{I}_k}$, where $1 \leq i \leq n$. Denote by $\mathcal{P}$ the set of predecessors $e_{pre}^i \notin G_1^{\mathcal{I}_k}$. We construct $\mathcal{I}_{k+1}$ as follows.
Case I: $e \in ind(\mathcal{A})$ and $r_1 \neq r_j$ for some $j$ where $1 \leq j \leq n$:

Extend $\mathcal{I}_k$ to $\mathcal{I}_{k+1}$ as follows. For every $d \in \mathcal{P}$, a new domain element $e'$ is introduced in $\Delta^{\mathcal{I}_{k+1}}$. Additionally, the following updates are performed:

1. $e' \in A^{\mathcal{I}_{k+1}}$ if $e \in A^{\mathcal{I}_k}$ for every concept name $A \in sig(\mathcal{O})\backslash\mathcal{F}$;
2. $(d, e') \in r_1^{\mathcal{I}_{k+1}}$;
3. $(e', f) \in r^{\mathcal{I}_{k+1}}$ if $(e, f) \in r^{\mathcal{I}_k}$ for every role name $r \in sig(\mathcal{O})\backslash\mathcal{F}$;
4. $e' \in D^{\mathcal{I}_{k+1}}$ if $d \in (\forall r_1.D)^{\mathcal{I}_k}$;
5. $D_1^{\mathcal{I}_{k+1}} := (D_1^{\mathcal{I}_k}\backslash\{e\}) \cup \{e'\}$.

The fourth update above makes $e'$ in the interpretations of a universal definer $D$ if $d \in (\forall r_1.D)^{\mathcal{I}_k}$. The fifth update ensures that $d$ remains an element in the interpretation of the concept $\exists r_1.D_1$. The two updates qualify $\mathcal{I}_{k+1}$ as a model of $(\mathcal{O}^q, \mathcal{A})$ because they make the clauses of the forms $G \sqcup \forall r_1.D$ and $G \sqcup \exists r_1.D$ valid in $\mathcal{I}_{k+1}$. So, they contribute towards the first property of $\mathcal{J}$ in the body of Lemma 16.

The fourth update additionally removes $e$ from the interpretation of the concept $D_1 \sqcap \cdots \sqcap D_n \sqcap \neg E$, because it removes it from the interpretation of $D_1$. So the update contributes towards achieving the second property of $\mathcal{J}$ in the body of Lemma 16.

1, 2, and 3 in the transformations above ensure that the pair $(e', e)$ satisfy the bisimulation conditions 1, 2, and 3 of Definition 14. The three updates are necessary to show bisimulation between $\mathcal{I}_k$ and $\mathcal{I}_{k+1}$ with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$. The relation $R$ defined by (49) is a bisimulation relation with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$, because the new domain elements $e'$ mirrors $e$ in everything except the interpretations of definers, and other domain elements were left unchanged.

$$(49) \quad R = \{(d, d) \in \Delta^{\mathcal{I}_k} \times \Delta^{\mathcal{I}_{k+1}} \mid d \in \Delta^{\mathcal{I}_k} \cap \Delta^{\mathcal{I}_{k+1}}\} \\ \cup \{(e', e) \mid \text{for every } e' \text{ introduced by the transformation}\}$$

We explain the above transformation through the example depicted in Figure 10. The left graph of the figure shows a model $\mathcal{I}_k$ where an individual $b$ has a predecessor $a$ via $r$, and a predecessor $c$ via $s$. The individual $b$ is labelled with two definers $D_1$ and $D_2$. The definer $D_1$ is existential, and $D_2$ is universal. The domain element $a$ is in the interpretation of $\exists r.D$, and $c$ is in the interpretation of $\forall s.D_2$.

In the model $\mathcal{I}_{k+1}$, a new successor $e$ of $a$ via $r$ is introduced by the above transformation. New edges are created to connect $e$ to the successors of $b$. The edges represent the updates to the interpretations of role symbols defined by the third transformation. Finally, $e$ substitutes $b$ in the interpretation of $D_1$.

The dotted lines indicate bisimilar elements in the relation $R$. For instance, the dotted line between $e$ and $b$ indicates that $(e, b) \in R$.

The transformation applied in Case I is performed only when $D_1$ and $D_j$ occur positively below different roles where $2 \leq j \leq n$. We can explain the importance of this restriction on the previous example. Suppose this restriction was not present, and $D_2$ occurred positively below $r$. Further, assume $a$ is in the

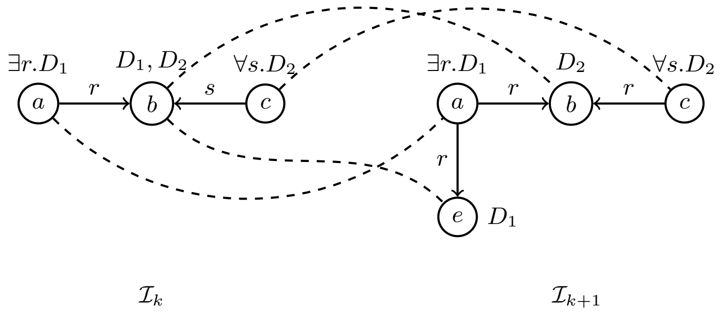

Fig. 10: Example of the transformation of Case I

interpretations of $\exists r.D_1$ and $\forall r.D_2$. When the transformation is performed, we get the following.

1. From 5, we get $e \in D_1$.
2. From 4 we get $e \in D_2$

So, the transformation would introduce a new element in both the interpretations of $D_1$ and $D_2$, which fails its main purpose.

Case II: $e \in ind(\mathcal{A})$, all definers occur below $r_1$, and two or more are existential: When $D_1, D_2, \ldots, D_n$ occur positively below the same role, and at least two of them are existential, we repeat the same transformation in Case I.

We explain this transformation through the example in Figure 11. The example shows a slight modification of that in Figure 10. The individual $c$ is in the interpretation of $\exists r.D_2$ instead of $\forall s.D_2$. The output of the transformation remains unchanged. A new $r$-successor $e$ of $a$ is introduced. The interpretation of $D_1$ is updated by substituting $e$ for $b$.

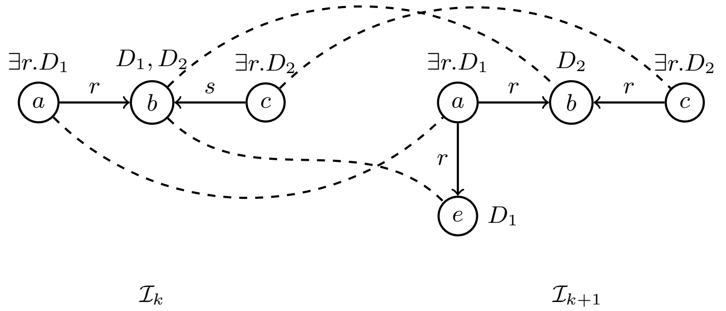

Fig. 11: Example of the transformation of Case II

Although $D_1, D_2, \ldots, D_n$ occur positively below the same role in $\mathcal{O}^q$, the concern discussed after the example in Case I is not considered a problem in our case. The reason is that step 4 of the transformation, only updates the interpretations of universal definers. Because two or more of the definers $D_1, \dots, D_n$ are existential, the newly introduced element will not be in the interpretations of $D_1, D_2, \dots, D_n$ all together.

Case III: $e \in ind(\mathcal{A})$, all definers occur below $r_1$, and only $D_1$ is existential:

We explain the transformation performed when the definers $D_1, D_2, \dots, D_n$ occur positively below the same role, and they all are universal except $D_1$. Note that in this case the concern expressed at the end of Case I is valid.

Since all definers, except $D_1$, are universal, it must be true that $\mathcal{O}^q \models G_j \sqcup \forall r_1.D_j$ for every $2 \leq j \leq n$. Further, it must be true that $\mathcal{O}^q \models G_1 \sqcup \exists r_1.D_1$, because $D_1$ is existential. But then the clause $G_1 \sqcup \dots G_n \sqcup \exists r_1.(E \sqcap \bigsqcap_{s=1}^{l} E_s)$ must have been preserved in $\mathcal{O}^q$ by a *Role Propagation* inference, where $\mathcal{O}^{rp} \models P_s \sqcup E_s$, and $P_s$ is any concept in $\neg D_1 \sqcup \dots \sqcup \neg D_n$ (The clauses $P_s \sqcup E_s$ are the clauses of the type-2 premise of the *Role Propagation* rule).

Denote by $\mathcal{P}'$ the predecessors $e_{pre}$ from $\mathcal{P}$ such that $e_{pre} \notin G_j^{\mathcal{I}_k}$ with $2 \leq j \leq n$. The set $\mathcal{P}'$ includes all domain elements $e_{pre}$ whose $r_1$ successors would be in the interpretations of $D_1, \dots, D_n$ if the transformation in Case I is performed. Since $\mathcal{O}^q \models G_1 \sqcup \dots G_n \sqcup \exists r_1.(E \sqcap \bigsqcap_{s=1}^{l} E_s)$, every $p_{pre} \in \mathcal{P}'$ must be in the interpretation of $\exists r_1.(E \sqcap \bigsqcap_{s=1}^{l} E_s)$ in $\mathcal{I}_k$. So, there is a child $e'$ of $e_{pre}$ via $r_1$, such that $e' \in D_i^{\mathcal{I}_k}$ where $2 \leq i \leq n$, and $e' \in E^{\mathcal{I}_k}$.

We extend $\mathcal{I}_k$ to $\mathcal{I}_{k+1}$ by applying the following transformations.

1. For every $e'$ identified above, we set $e' \in D_1^{\mathcal{I}_{k+1}}$.
2. For every $e_{pre} \in \mathcal{P}$ and $e_{pre} \notin \mathcal{P}'$ we apply the transformation described in Case I.
3. $D_1^{\mathcal{I}_{k+1}} := D_1^{\mathcal{I}_k} \backslash \{e\}$.

The argument of Case I shows that $\mathcal{I}_{k+1} \models (\mathcal{O}^q, \mathcal{A})$. Moreover, we see that $\mathcal{I}_{k+1}$ and $\mathcal{I}_k$ are bisimilar with respect to $sig(\mathcal{O}) \backslash \mathcal{F}$ and $ind(\mathcal{A})$. The relation $R$ defined by (49) is the bisimulation relation between $\mathcal{I}_{k+1}$ and $\mathcal{I}_k$.

Suppose $e \notin ind(\mathcal{A})$. We consider cases IV and V. When $e$ is not an individual from $ind(\mathcal{A})$, we have $r_j = r_1$ for every $j$ where $1 \leq j \leq n$ because $\mathcal{I}_k$ is a pseudo forest. Also, since $e$ has only one predecessor $e_{pre}$, the sets $\mathcal{P}$ and $\mathcal{P}'$ would contain only $e_{pre}$. We use the transformation discussed for Case III in Case IV, and the transformation discussed for Case II in Case V. This completes the proof of the lemma.

## Proof of Theorem 3

The ontology $\mathcal{V}^q_{ALC}$ is obtained by eliminating simple definers from $\mathcal{O}^q$. The elimination process has been shown to preserve model inseparability (see proof of Theorem 2), which proves 1. Since $\mathcal{O}^q$ is deductive inseparable and query inseparable from the input ontology $\mathcal{O}$, we get that 2 and 3 hold. Finally, 4 directly follows from 2, 3, and Definition 3.

## Proof of Theorem 4

**Lemma 17.** *Let $\mathcal{C}_1$ be the clause $C \sqcup \forall r.D$ be a clause, where $C$ is a concept, $r$ a role name, and $D$ a definer. Let $\mathcal{C}_2$ be the conclusion $\forall r^-.C \sqcup D$ obtained from a LB inference, where the first premise of the inference is $\mathcal{C}_1$ and the second premise is $\emptyset$. Then the two ontologies $\{\mathcal{C}_1\}$ and $\{\mathcal{C}_2\}$ are logically equivalent.*

*Proof.* Suppose we have the following.

$$\mathcal{O}_1 = \{C \sqcup \forall r.D\} \tag{50}$$

$$\mathcal{O}_2 = \{D \sqcup \forall r^-.C\} \tag{51}$$

To prove logical equivalence we need to show the following.

$$\mathcal{O}_1 \models \mathcal{O}_2 \tag{52}$$

$$\mathcal{O}_2 \models \mathcal{O}_1 \tag{53}$$

From Lemma **??** we know that (52) is true.

We show that (53) by obtaining a contradiction from $\mathcal{O}_2$ and the negation of $\mathcal{O}_1$. The negation of $\mathcal{O}_1$ is the following set of clauses.

$$\neg A(a)$$

$$(\exists r.\neg D)(a)$$

where $a$ is a witness instance. We now have:

$$D \sqcup \forall r^-.A \tag{54}$$

$$\neg A(a) \tag{55}$$

$$(\exists r.\neg D)(a) \tag{56}$$

From (56) there must be an instance $b$ satisfying the following.

$$r(a, b) \tag{57}$$

$$\neg D(b) \tag{58}$$

From (54) and (58), we get the following.

$$(\forall r^-.A)(b) \tag{59}$$

From (57) and (59) we get:

$$A(a) \tag{60}$$

A contradiction is now obtained from (60) and (55).

Lemma 17 can be used to prove that the elimination process of Type 1 definers, described in Algorithm **??**, preserves model equivalence with respect to the non-forgotten signature $sig(\mathcal{O})\backslash\mathcal{F}$.

**Lemma 18.** *Let $\mathcal{O}$ be an ontology, and $\mathcal{N}$ a set of Type 1 definers occurring in $\mathcal{O}$. Let $\mathcal{V}$ be the ontology obtained by eliminating the Type 1 definers from $\mathcal{O}$ using the process explained in Algorithm* **??**. *The two ontologies $\mathcal{O}$ and $\mathcal{V}$ are model inseparable with respect to the signature $sig(\mathcal{O})\backslash\mathcal{N}$.*

*Proof.* We can describe Algorithm **??** as follows. Algorithm **??** iterates over the Type 1 definers. Every iteration eliminates one Type 1 definer. An iteration eliminating a definer $D$ replaces every clause of the following form with the conclusion of the LB inference performed on the clause.

$$C \sqcup \forall r.D \tag{61}$$

When all clauses of the form (61) are replaced, Algorithm **??** is called to eliminate $D$. When all Type 1 definers have been eliminated, simple definers occurring in $\mathcal{O}$ are eliminated using the process explained in Algorithm **??**.

Lemma **??** shows that the call to Algorithm **??** preserves model inseparability with respect to $sig(\mathcal{O})\backslash\mathcal{N}$. Lemma **??** shows that the call to Algorithm **??** preserves model inseparability with respect to $sig(\mathcal{O})\backslash\mathcal{N}$. We show that the substitution of the conclusions of the LB inferences for the premises of the inferences preserves logical equivalence.

Consider first the substitution of the conclusion of a LB inference for a clause of the form (61). Let $\mathcal{C}_1$ be the clause of the form (61) being substituted for, and $\mathcal{C}_2$ the conclusion of the LB inference. Let $\mathcal{O}_1$ be a snapshot of the ontology before the substitution, and $\mathcal{O}_2$ be the snapshot after the substitution, and assume that the two ontologies are consistent. We show that $\mathcal{O}_1$ and $\mathcal{O}_2$ are logically equivalent by showing that every model of $\mathcal{O}_1$ is a model of $\mathcal{O}_2$, and every model of $\mathcal{O}_2$ is a model of $\mathcal{O}_1$. Let $\mathcal{O}_0$ be the subset of $\mathcal{O}_1$ which does not contain $\mathcal{C}_1$. The following is true about $\mathcal{O}_1$ and $\mathcal{O}_2$.

$$\mathcal{O}_1 = \mathcal{O}_0 \cup \{\mathcal{C}_1\} \tag{62}$$

$$\mathcal{O}_2 = \mathcal{O}_0 \cup \{\mathcal{C}_2\} \tag{63}$$

Let $\mathcal{I}$ be a model of $\mathcal{O}_1$, then the following is true about $\mathcal{I}$.

$$\mathcal{I} \models \mathcal{O}_0 \tag{64}$$

$$\mathcal{I} \models \{\mathcal{C}_1\} \tag{65}$$

From (65) and Lemma 17, we get that $\mathcal{I}$ is a model of $\{\mathcal{C}_2\}$. Because $\mathcal{I}$ is a model of $\mathcal{O}_0$ as well, we get that $\mathcal{I}$ is a model of $\mathcal{O}_2$. The same argument proves that a model $\mathcal{I}$ of $\mathcal{O}_2$ is a model of $\mathcal{O}_1$.

**Theorem 5.** *Let $\mathcal{O}$ be an ontology, and $\mathcal{F}$ a forgetting signature. Suppose $\mathcal{O}^q$ is the reduced ontology obtained from $\mathcal{O}$ with respect to $\mathcal{F}$, and $\mathcal{V}^q_{ALCI}$ is the ontology obtained by eliminating simple and complex definers from $\mathcal{O}^q$. We have:*

1. *$\mathcal{O} \equiv^Q_{sig(\mathcal{O})\backslash\mathcal{F}} \mathcal{V}^q_{ALCI}$, and*
2. *If $sig(\mathcal{V}^q_{ALCI}) \cap N_d = \emptyset$, then $\mathcal{V}^q_{ALCI}$ is a query forgetting view of $\mathcal{O}$ with respect to $\mathcal{F}$.*

The proof of the theorem is delivered through the discussions and lemmas in the remainder of this section. Note that 2 in Theorem 5 directly follows from 1 and Definition **??**. So we only need to prove 1.

We know from Theorem **??** that $\mathcal{O}$ and $\mathcal{O}^q$ are query inseparable with respect to $\mathcal{F}$. The ontology $\mathcal{V}^q_{ALC}$ obtained from $\mathcal{O}^q$ by eliminating simple definers is also query inseparable from $\mathcal{O}$ with respect to $\mathcal{F}$. A proof for this was given in Lemma **??**. Furthermore, the elimination of Type 1 definers preserves model inseparability with respect to the non-forgotten signature $sig(\mathcal{O})\backslash\mathcal{F}$. From Lemma **??** we can see that the elimination of Type 1 definers also preserves query inseparability with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ It remains to show that the complex definer elimination process preserves query inseparability. It remains to consider the elimination of Type 2 definers.

We analyse each component in elimination process of Type 2 definers separately. There are five components to consider.

1. The LB and the UB inferences: This component preserves logical equivalence, because it only makes some implicit entailments of $\mathcal{V}^q_{ALCI}$ explicit. From Lemmas **??** and **??**, we see that logical equivalence implies query inseparability. So this component preserves query inseparability with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.
2. *Simple definer elimination*: This component relies only on syntactic transformation as well as purification rules. Lemma **??** showed that elimination of simple definers preserves query inseparability with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.
3. Resolution: Resolution makes implicit information explicit, so it preserves logical equivalence.
4. Elimination of resolution premises: Theorem **??** showed that removing the premises of resolution inferences preserves model inseparability. However, the result of the theorem cannot be admitted in our case because it was obtained assuming a specific normal form of the input $\mathcal{O}^{clausal}$. In particular, the eliminated symbols are assumed to occur only as top level, possibly negated, disjuncts, whereas Type 2 definers can additionally be used below role restrictions. Therefore, query inseparability with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ would need to be verified when the premises of resolution inferences are removed.
5. Elimination of clauses of $\Delta^u$: Query inseparability with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ is needed to be verified for this component.

Based on the above analysis we focus on proving query inseparability with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ when clauses are removed in the components 4 and 5.

We consider 4 first. Resolution is performed iteratively. Each iteration resolves the clauses that use a certain definer, and eliminates the premises of the resolution inferences when resolution has completed. We consider a single iteration where the clauses that use a definer $D_1$ are resolved together on $D_1$, then eliminated. Let $\mathcal{V}^{pre}$ be the snapshot of $\mathcal{V}^q_{ALCI}$ before removing the clauses. Suppose $\mathcal{V}^{pre}$ has a set of clauses $C_i \sqcup D_1$ with $1 \leq i \leq n$ and $n \geq 1$. We replace these clauses with their combined form $C \sqcup D_1$ where $C = \bigsqcap_{i=1}^{n} C_i$. In the same way, if there are $m$ clauses of the form $E_j \sqcup \neg D_1$ with $1 \leq j \leq m$ in

$\mathcal{V}^{pre}$, we replace them with the combined form $E \sqcup \neg D_1$ where $E = \bigsqcap_{j=1}^{m} E_j$. Combining clauses is a technique that we have used in numerous places, and was shown to preserve logical equivalence, see for instance the proof of Lemma **??**. Note that the clauses $C_i \sqcup D_1$ are the conclusions of LB inferences. The concept $C_i = \forall r_1^{-}.(G \sqcup \bigsqcup_{j=2}^{n} \forall r_j.F_j)$ is a conjunct in $C$ if and only if a LB inference was performed with $G \sqcup \forall r_1.D_1 \sqcup \cdots \sqcup \forall r_n.D_n$ as a first premise, and a set of $n-1$ clauses $\neg D_j \sqcup F_j$ as a second premise, where $2 \leq j \leq n$. We can view $\mathcal{V}^{pre}$ as follows.

$$\mathcal{V}^{pre} = \mathcal{V}_0 \cup \mathcal{V}_\forall \cup \{C \sqcup D_1, E \sqcup \neg D_1, C \sqcup E\} \tag{66}$$

where $\mathcal{V}_0$ is a set of clauses that does not use $D_1$, and $\mathcal{V}_\forall$ is a set of clauses of the form $G \sqcup \forall r_1.D_1 \sqcup \cdots \sqcup \forall r_n.D_n$. The clauses $C \sqcup D_1$ and $E \sqcup \neg D_1$ are as above, and $C \sqcup E$ is their resolvent.

Let $\mathcal{V}^{post}$ be the snapshot of $\mathcal{V}^{q}_{ALCI}$ after removing the clauses used in resolution. We have the following.

$$\mathcal{V}^{post} = \mathcal{V}_0 \cup \mathcal{V}_\forall \cup \{C \sqcup E\} \tag{67}$$

Let $\mathcal{A}$ be an arbitrary ABox, $q(\vec{x})$ a conjunctive query with $\vec{x} = x_1, x_2, \ldots, x_k$, and $\vec{a} = a_1, a_2, \ldots, a_k$ an answer to $q(\vec{x})$ where $k \geq 0$. Our aim is to show that $\mathcal{V}^{pre} \equiv^{Q}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{V}^{post}$, which requires proving the following.

$$(\mathcal{V}^{pre}, \mathcal{A}) \models q(\vec{a}) \quad \text{implies} \quad (\mathcal{V}^{post}, \mathcal{A}) \models q(\vec{a}) \tag{68}$$

$$(\mathcal{V}^{post}, \mathcal{A}) \models q(\vec{a}) \quad \text{implies} \quad (\mathcal{V}^{pre}, \mathcal{A}) \models q(\vec{a}) \tag{69}$$

The following lemma shows that (69) is always true.

**Lemma 19.** *Let $\mathcal{V}^{pre}$, $\mathcal{V}^{post}$, $\mathcal{A}$, and $q(\vec{x})$ as described above. Then, $(\mathcal{V}^{post}, \mathcal{A}) \models q(\vec{a})$ implies $(\mathcal{V}^{pre}, \mathcal{A}) \models q(\vec{a})$, where $\vec{a}$ is an answer to $q(\vec{x})$.*

*Proof.* Suppose $(\mathcal{V}^{post}, \mathcal{A}) \models q(\vec{a})$. Also suppose $\mathcal{I}$ is a model of $(\mathcal{V}^{pre}, \mathcal{A})$ such that $q(\vec{a})$ is false in $\mathcal{I}$. Since $\mathcal{I} \models (\mathcal{V}^{pre}, \mathcal{A})$ we have that $\mathcal{I}$ is a model of $\mathcal{V}^{pre}$, and a model of $\mathcal{A}$. From the construction of $\mathcal{V}^{pre}$ and $\mathcal{V}^{post}$, we can see the following.

$$\mathcal{V}^{post} \subseteq \mathcal{V}^{pre} \tag{70}$$

The subsumption (70) implies that $\mathcal{I}$ is also a model of $\mathcal{V}^{post}$, because removing clauses does not invalidate the models. We now have $\mathcal{I} \models \mathcal{V}^{post}$ and $\mathcal{I} \models \mathcal{A}$. So, $\mathcal{I}$ is a model of $(\mathcal{V}^{post}, \mathcal{A})$. The two following contradicting remarks are true about $\mathcal{I}$.

1. $\mathcal{I} \not\models q(\vec{a})$ by assumption.
2. $\mathcal{I} \models q(\vec{a})$ because $\mathcal{I} \models (\mathcal{V}^{post}, \mathcal{A})$ and $(\mathcal{V}^{post}, \mathcal{A}) \models q(\vec{a})$.

From the contradiction we get that if $(\mathcal{V}^{post}, \mathcal{A}) \models q(\vec{a})$ and $\mathcal{I} \models (\mathcal{V}^{pre}, \mathcal{A})$ then $\mathcal{I} \models q(\vec{a})$.

Next, we show that (68) is true. Suppose $(\mathcal{V}^{pre}, \mathcal{A}) \models q(\vec{a})$, and let $\mathcal{I}$ be a model of $(\mathcal{V}^{post}, \mathcal{A})$ where $q(\vec{a})$ is false. First, we highlight cases where $\mathcal{I}$ can be transformed to a model $\mathcal{J}$, such that $\mathcal{J} \models (\mathcal{V}^{pre}, \mathcal{A})$ and $\mathcal{I}$ and $\mathcal{J}$ are bisimilar with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$. In these cases the following contradicting observations would occur.

1. $\mathcal{J} \models q(\vec{a})$ because $\mathcal{J} \models (\mathcal{V}^{pre}, \mathcal{A})$ and $(\mathcal{V}^{pre}, \mathcal{A}) \models q(\vec{a})$.
2. $\mathcal{J} \not\models q(\vec{a})$, because $I$ and $\mathcal{J}$ are bisimilar with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$, and $\mathcal{I} \not\models q(\vec{a})$.

For the other cases where $\mathcal{I}$ cannot be transformed to a model $\mathcal{J}$ of $(\mathcal{V}^{pre}, \mathcal{A})$, we transform $\mathcal{I}$ to $\mathcal{J}$ such that $\mathcal{I}$ and $\mathcal{J}$ are bisimilar with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$, and we show that there is a map $\pi$ with $\pi(x_i) = a_i$ for $1 \leq i \leq k$, and for every anonymous variable $y$ in $q$, $\pi$ maps $y$ to an element in $\Delta^{\mathcal{J}}$ such that the following is true.

1. If $A(u)$ is an atom in $q$, then $\pi(u) \in A^{\mathcal{J}}$.
2. If $r(u, v)$ is an atom in $q$, then $(\pi(u), \pi(v)) \in r^{\mathcal{J}}$.

A contradiction would be obtained as the following will occur.

1. $\mathcal{J} \not\models q(\vec{a})$ because $\mathcal{I}$ and $\mathcal{J}$ are bisimilar with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$, and $\mathcal{I} \not\models q(\vec{a})$.
2. $\mathcal{J} \models q(\vec{a})$ by construction of the map $\pi$ described above.

The two contradictions imply that if $(\mathcal{V}^{pre}, \mathcal{A}) \models q(\vec{a})$ and $\mathcal{I}$ is a model of $(\mathcal{V}^{post}, \mathcal{A})$, then $\mathcal{I} \models q(\vec{a})$, which proves (68).

The following definitions support the discussion below.

**Definition 15.** *Let $\mathcal{I}$ be a model, and $b \in \Delta^{\mathcal{I}}$ a domain element.*

1. *By $\Gamma_D$ we denote the set of clauses $G \sqcup \forall r_1.D \sqcup \forall r_2.D_2 \sqcup \cdots \sqcup \forall r_n.D_n$.*
2. *By $\beta_D$ we denote the set of concepts $G$ occurring in the clauses of $\Gamma_D$.*
3. *By $\gamma_{b,D}$ we denote the set of domain elements $a \in \Delta^{\mathcal{I}}$ where $(a, b) \in r_1^{\mathcal{I}}$ and $a \notin G$ for any concept $G$ from $\beta_D$.*

Let $\mathcal{I}$ be as above. We transform $\mathcal{I}$ to a model $\mathcal{I}_0$ where for every $b \in \Delta^{I_0}$ we have $b \in D_1^{\mathcal{I}_0}$ if and only if $b \in D_1^{\mathcal{I}}$ and $\gamma_{b,D_1} \neq \emptyset$. Intuitively, the transformation removes $b$ from the interpretation of $D_1$ if $\gamma_{b,D_1}$ is empty.

**Lemma 20.** *Let $\mathcal{I}$ and $\mathcal{I}_0$ be as above. Then $\mathcal{I}_0 \models (\mathcal{V}^{post}, \mathcal{A})$ and $\mathcal{I}_0 \not\models q(\vec{a})$.*

*Proof.* From (67), $\mathcal{I}_0 \models \mathcal{V}^{post}$ if and only if $\mathcal{I}_0 \models \mathcal{V}_0$, $\mathcal{I}_0 \models \mathcal{V}_\forall$, and $\mathcal{I}_0 \models C \sqcup E$.

$\mathcal{I}_0 \models \mathcal{V}_0$ because $\mathcal{I}_0$ was obtained from $\mathcal{I}$ by changing the interpretation of $D_1$, and $D_1$ does not occur in $\mathcal{V}_0$.

$\mathcal{I}_0 \models \mathcal{V}_\forall$ if and only if for every domain element $b \in \Delta^{\mathcal{I}_0}$ such that $b \in D_1^{\mathcal{I}}$, $b \notin D_1^{\mathcal{I}_0}$, we have $a \in \mathcal{C}^{\mathcal{I}_0}$ for every $a \in \gamma_{b,D_1}$ and every $\mathcal{C} \in \Gamma_{D_1}$. However, from the construction of $\mathcal{I}_0$ we have $b \in D_1^{\mathcal{I}}$ and $b \notin D_1^{\mathcal{I}_0}$ implies $\gamma_{b,D_1} = \emptyset$. So, we have $\mathcal{I}_0 \models \mathcal{V}_\forall$.

To show that $\mathcal{I}_0 \models C \sqcup E$, it suffices to show that for every element $b \in D_1^{\mathcal{I}}$ if $b \notin D_1^{\mathcal{I}_0}$ then $b \in (C \sqcup E)^{\mathcal{I}}_0$. Recall from the construction of $\mathcal{I}_0$ that $b$ is removed from $D_1^{\mathcal{I}_0}$ only if $\gamma_{b,D_1}$ is empty. Therefore, $b \in (\forall r_1^-.G)^{\mathcal{I}_0}$ for every $G \in \beta_{D_1}$. This observation, and the discussion above regarding the form of the concept $C$ imply that $b \in C^{\mathcal{I}_0}$. So, $b \in (C \sqcup E)^{\mathcal{I}_0}$.

From the above we see that $\mathcal{I}_0 \models \mathcal{V}^{post}$. Since $\mathcal{I}_0$ is constructed by removing elements from the interpretation of $D_1$, we have $\mathcal{I}_0 \models \mathcal{A}$ because $D_1 \notin \mathit{sig}(\mathcal{A})$. Therefore $\mathcal{I}_0 \models (\mathcal{V}^{post}, \mathcal{A})$.

The two models $\mathcal{I}$ and $\mathcal{I}_0$ are bisimilar with respect to $\mathit{sig}(\mathcal{V}^{post})\backslash\{D_1\}$ and $\mathit{ind}(\mathcal{A})$, because $\mathcal{I}_0$ differs from $\mathcal{I}$ only on the interpretation of $D_1$. Since $\mathcal{I} \not\models q(\vec{a})$, and $\mathit{sig}(q) \subseteq \mathit{sig}(\mathcal{V}^{post})\backslash\{D_1\}$, we have from Lemma 14 that $\mathcal{I}_0 \not\models q(\vec{a})$.

Next, we discuss cases where $\mathcal{I}_0$ can be transformed to the model $\mathcal{J}$ of $(\mathcal{V}^{pre}, \mathcal{A})$ described before. The construction of $\mathcal{V}^{post}$ and $\mathcal{V}^{pre}$ implies that $\mathcal{J}$ is a model of $(\mathcal{V}^{pre}, \mathcal{A})$ if the following three conditions are satisfied: $\mathcal{J} \models (\mathcal{V}^{post}, \mathcal{A})$, $\mathcal{J} \models C \sqcup D_1$, and $\mathcal{J} \models E \sqcup \neg D_1$. Let $b \in \Delta^{\mathcal{I}_0}$ be a domain element. Since $C \sqcup E$ is a clause in $\mathcal{V}^{post}$, and $\mathcal{I}_0 \models \mathcal{V}^{post}$, one of the following must be true.

(71) $b \in E^{\mathcal{I}}$ and $b \notin C^{\mathcal{I}}$

(72) $b \notin E^{\mathcal{I}}$ and $b \in C^{\mathcal{I}}$

(73) $b \in E^{\mathcal{I}}$ and $b \in C^{\mathcal{I}}$

We have the following cases.
Case 1: We perform the following transformation if (71) is true, or (73) is true. Let $\mathcal{J}$ be the model that extends $\mathcal{I}_0$ by adding $b$ to the interpretation of $D_1$. We have:

1. $\mathcal{J}$ is a model of $(\mathcal{V}^{post}, \mathcal{A})$, because adding elements to the interpretation of $D_1$ only impacts clauses where $D_1$ occurs with negative polarity. Since $D_1$ does not occur with negative polarity in $\mathcal{V}^{post}$, we have that $\mathcal{J}$ is a model of $\mathcal{V}^{post}$. Moreover, $\mathcal{J} \models \mathcal{A}$, because $D_1 \notin \mathit{sig}(\mathcal{A})$. So $\mathcal{J}$ is a model of the knowledge base $(\mathcal{V}^{post}, \mathcal{A})$.
2. $\mathcal{I}_0$ and $\mathcal{J}$ are bisimilar with respect to $\mathit{sig}(\mathcal{V}^{post})\backslash\{D_1\}$ and $\mathit{ind}(\mathcal{A})$, because only the interpretation of $D_1$ is modified by the transformation.

Case 2: We assume (72) is true. Assume that for every clause $\mathcal{C} = G \sqcup \forall r_1.D_1 \sqcup \cdots \sqcup \forall r_n.D_n$ in $\Gamma_{D_1}$, and every $a \in \gamma_{b,D_1}$ such that $a \notin G^{\mathcal{I}_0}$, the following is true.

(74) $a \in (\forall r_j.D_j)^{\mathcal{I}_0}$ for some $j$ with $2 \leq j \leq n$

We transform $\mathcal{I}$ to $\mathcal{J}$ by removing $b$ from the interpretation of $D_1$.

First, we show that $\mathcal{J}$ is a model of $(\mathcal{V}^{post}, \mathcal{A})$. Consider all clauses of $\mathcal{V}^{post}$ where $D_1$ occurs positively. Since $D_1$ occurs positively only in the clauses $\mathcal{C}$ below role restrictions, we need to verify that for every $a \in \gamma_{b,D_1}$ we have $a \in \mathcal{C}^{\mathcal{J}}$. But this is always true because of the condition in (74).

Second, we can see that $\mathcal{I}_0$ and $\mathcal{J}$ are bisimilar with respect to $\mathit{sig}(\mathcal{O})\backslash\mathcal{F}$ and $\mathit{ind}(\mathcal{A})$, because only the interpretation of $D_1$ is modified and $D_1 \notin \mathit{sig}(\mathcal{O})\backslash\mathcal{F}$.
Case 3: We assume (72) is true, and suppose there is a clause $\mathcal{C} \in \Gamma_{D_1}$ where

$\mathcal{C} = G \sqcup \forall r_1.D_1 \sqcup \cdots \sqcup \forall r_n.D_n$, and a domain element $a \in \gamma_{b,D_1}$ such that (75), (76), and (77) below are true.

$$a \notin G^{\mathcal{I}_0} \tag{75}$$

$$a \in (\forall r_1.D_1)^{\mathcal{I}_0} \tag{76}$$

$$a \notin (\forall r_j.D_j)^{\mathcal{I}_0} \text{ for every } j \text{ with } 2 \leq j \leq n. \tag{77}$$

then removing $b$ from the interpretation of $D_1$ makes $\mathcal{C}$ not valid in $\mathcal{J}$. Suppose (75), (76), and (77) are true. We observe the following.

1. From (77), there exist domain elements $b_j$, such that $(a, b_j) \in r_j^{\mathcal{I}_0}$, and $b_j \notin D_j^{\mathcal{I}}$ for every $j$ where $2 \leq j \leq n$.
2. By assumption we have $b \in C^{\mathcal{I}_0}$, where $C$ is the conjunction of concepts of the form $\forall r_1^-.(H \sqcup \bigsqcup_{i=2}^n \forall r_i.F_i)$, $H \in \beta_{D_1}$, and $\neg D_i \sqcup F_i$ are clauses in $\mathcal{V}^{post}$ with $2 \leq i \leq n$.
3. One of the conjuncts of $C$ is the concept $\forall r_1^-.(G \sqcup \bigsqcup_{i=2}^n \forall r_i.F_i)$ obtained from a LB inference whose first premise is $\mathcal{C}$.
4. From the previous point, we have $b \in \forall r_1^-.(G \sqcup \bigsqcup_{i=2}^n \forall r_i.F_i)^{\mathcal{I}_0}$, and $a \in (G \sqcup \bigsqcup_{i=2}^n \forall r_i.F_i)^{\mathcal{I}_0}$ because $(a, b) \in r_1^{\mathcal{I}_0}$.
5. From (75) and the previous point, we have $a \in (\bigsqcup_{i=2}^n \forall r_i.F_i)^{\mathcal{I}_0}$.
6. From the previous point and 1, there exists a subset $\sigma$ of the definers $D_2$, $\ldots$, $D_n$ where for each definer $D_j \in \sigma$ there is a domain element $b_j \notin D_j$, such that $(a, b_j) \in r_j$, and $b_j \in F_j$.

Suppose $\sigma$ has only one definer $D_k$ with $2 \leq k \leq n$ where $D_k$ occurs in the disjunct $\forall r_k.D_k$ in $\mathcal{C}$. Then, there are no $r_k$-successors of $a$ in the interpretation of $\forall r_k^-.(G \sqcup \bigsqcup_{l=1, l \neq k}^n \forall r_l.F_l)$, and we have the following.

1. If $D_k$ has been eliminated in an earlier resolution iteration, then $\mathcal{V}^{post}$ must have a clause $C_k \sqcup E_k$ where $\forall r_k^-.(G \sqcup \bigsqcup_{l=1, l \neq k}^n \forall r_l.F_l)$ is a conjunct in $C_k$. A $r_k$-successor $b_k$ of $a$ is not in the interpretation of $C_k$ because it is not in the interpretation of the just mentioned conjunct. Furthermore, $b_k \in E_k^{\mathcal{I}_0}$ because $b_k \in (C_k \sqcup E_k)^{\mathcal{I}_0}$ and $b_k \notin C_k^{\mathcal{I}_0}$. We transform $I_0$ to $\mathcal{J}$ such that $a \in \forall r_k.D_k$, and remove $b$ from the interpretation of $D_1$.
2. If $D_k$ has not been eliminated in an earlier resolution iteration, then $\mathcal{V}^{post}$ must have a clause $D_k \sqcup C_k$. Since every $r_k$-successor of $a$ is not in the interpretation of $C_k$, we have $a \in \forall r_k.D_k$. This case cannot happen because it contradicts the assumption (77).

Case 4: Assume the conditions in Case 3 are true, except $\sigma$ which we assume has two or more definers. This case indicates the existence of a set of models of $(\mathcal{V}^{pre}, \mathcal{A})$, each uses a definer $D$ from $\sigma$ such that the domain element $a$ is in the interpretation of $\forall r.D$.

**Definition 16.** *Let $\mathcal{I}_0$ be a model of $(\mathcal{V}^{post}, \mathcal{A})$ as before. Let $\mathcal{I}$ be a model of $(\mathcal{V}^{pre}, \mathcal{A})$. We say $\mathcal{I}$ is* extendable *from $\mathcal{I}_0$ if $\Delta^{\mathcal{I}_0} \subseteq \Delta^{\mathcal{I}}$, and for every concept name $C$ and role $r$, we have:*

*1.* $C^{\mathcal{I}_0} \subseteq C^{\mathcal{I}}$*, and*
*2.* $r^{\mathcal{I}_0} \subseteq r^{\mathcal{I}}$*.*

*The set* $\mathcal{M}$ *denote the set of models of* $(\mathcal{V}^{pre}, \mathcal{A})$ *that are extendable from* $\mathcal{I}_0$*.*

Let $D$ be as in the previous paragraph, and $\alpha$ denote the set of $r$-successors of the domain element $a$ that are not in the interpretation of $D$. Suppose one of the following is true.

1. $\mathcal{V}^{post}$ does not have a clause of the form $\neg D \sqcup \neg D' \sqcup C$, or
2. $d \notin D'^{\mathcal{I}_0}$ implies $d \in C^{\mathcal{I}_0}$ for every $d \in \alpha$.

where $D' \neq D$ is a definer.

We extend $\mathcal{I}_0$ to $\mathcal{J} \in \mathcal{M}$ by adding all elements occurring in the set $\alpha$ to the interpretation of $D$, and remove every $b \in \Delta^{\mathcal{J}}$ satisfying (72) from the interpretation of $D_1$.

First, We show that $\mathcal{J}$ is a model of $(\mathcal{V}^{post}, \mathcal{A})$.

1. Since the transformation adds $r$-successors of the domain element $a$ to the interpretation of $D$, we need only to consider clauses where $D$ occurs negatively. Let $\neg D \sqcup C$ be a clause in $\mathcal{V}^{post}$, and $d$ a domain element added to the interpretation of $D$ by the transformation. Since $D \in \sigma$, we have $d \in C^{\mathcal{I}_0}$. So, $d \in \neg D \sqcup C$. Suppose $D \sqcup D' \sqcup C$ is a clause in $\mathcal{V}^{post}$, the conditions above require that $d \notin D'^{\mathcal{I}_0}$ or $d \in C^{\mathcal{I}_0}$. So, $d \in (\neg D \sqcup \neg D' \sqcup C)^{\mathcal{J}}$.
2. Since the transformation removes $b$ from the interpretation of $D_1$, we need to show that $a \in (G \sqcup \forall r.D \sqcup \forall r_1.D_1 \sqcup \cdots \sqcup \forall r_n.D_n)^{\mathcal{J}}$. Observe that adding domain elements to the interpretation of $D$ makes $a \in (\forall r.D)^{\mathcal{J}}$. So, $a \in (G \sqcup \forall r.D \sqcup \forall r_1.D_1 \sqcup \cdots \sqcup \forall r_n.D_n)^{\mathcal{J}}$ even though $b$ is removed from the interpretation of $D_1$.

Second, the two models $\mathcal{I}_0$ and $\mathcal{J}$ are bisimilar with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$ because we only change the interpretations of definers.
Case 5: Assume the conditions in Case 3 are true, except $\sigma$ which we assume has two or more definers. Further, assume no definer from $\sigma$ satisfy the two conditions in Case 4.

Let $D_p$ and $D_q$ be any two definers from $\sigma$, $b_p$ be a successor of $a$ via $r_p$, and $b_q$ a successor of $a$ via $r_q$. Assume $b_p \in D_{p,1}^{\mathcal{I}_0}$ where $D_{p,1}$ is a definer, $\neg D_{p,1} \sqcup \neg D_p \sqcup C_p$ is a clause in $\mathcal{V}^{post}$, and $b_p \notin C_p^{\mathcal{I}_0}$. In parallel, assume $b_q \in D_{q,1}^{\mathcal{I}_0}$ where $D_{q,1}$ is a definer, $\neg D_{q,1} \sqcup \neg D_q \sqcup C_q$ is a clause in $\mathcal{V}^{post}$, and $b_p \notin C_p^{\mathcal{I}_0}$.

Since $\mathcal{V}^{post}$ only has Type 2 definers, $D_{p,1}$ must be a type 2 definer. Furthermore, $D_{p,1}$ must be a universal definer because existential definers cannot occur in $\mathcal{V}^{post}$ with negative polarity in clauses where two negative definers occur. So, $\mathcal{V}^{post}$ must have a clause $G_p \sqcup \forall s_1.D_{p,1} \sqcup \cdots \sqcup \forall s_u.D_{p,u}$ in $\mathcal{V}^{post}$, and a domain element $a_p$ such that $(a_p, b_p) \in s_1^{\mathcal{I}_0}$, $a_p \notin G_p^{\mathcal{I}_0}$ and $a_p \in (\forall s_1.D_{p,1})^{\mathcal{I}_0}$. In the same way, $\mathcal{V}^{post}$ must have a clause $G_q \sqcup \forall t_1.D_{q,1} \sqcup \cdots \sqcup \forall t_v.D_{q,v}$ in $\mathcal{V}^{post}$, and a domain element $a_q$ such that $(a_q, b_q) \in t_1^{\mathcal{I}_0}$, $a_q \notin G_q^{\mathcal{I}_0}$ and $a_q \in (\forall t_1.D_{q,1})^{\mathcal{I}_0}$.

**Lemma 21.** *There exist two models* $\mathcal{I}_p$ *and* $\mathcal{I}_q$ *of* $(\mathcal{V}^{pre}, \mathcal{A})$ *in* $\mathcal{M}$ *where:*

1. *$b_p \notin D_{p,1}^{\mathcal{I}_p}$ and $b_p \notin C_p^{\mathcal{I}_p}$.*
2. *$b_q \notin D_{q,1}^{\mathcal{I}_q}$ and $b_q \notin C_q^{\mathcal{I}_q}$.*

*Proof.* Recall $a_p \notin G_p^{\mathcal{I}_0}$, where $G_p \sqcup \forall s_1.D_{p,1} \sqcup \cdots \sqcup \forall s_u.D_{p,u}$ is a clause in $\mathcal{V}^{post}$.

We note that $u \geq 2$, because if $u = 1$, then we have $(\mathcal{V}^{post}) \models D_{p,1} \sqcup \forall s_1^-.G_p$, and we would have that $\mathcal{V}^{post} \models \neg D_p \sqcup \forall s_1^-.G_p \sqcup C_p$. By the assumption made in 6 in Case 3, we get that $b_p \in C_p^{\mathcal{I}_0}$ which contradicts our assumption.

Consider all models of $(\mathcal{V}^{pre}, \mathcal{A})$ from $\mathcal{M}$ which are extended from $\mathcal{I}_0$. Suppose in every model $\mathcal{I} \in \mathcal{M}$ we have $a_p \in (\forall s_1.D_{p,1})^{\mathcal{I}}$ and $a_p \notin (\forall s_i.D_{p,i})^{\mathcal{I}}$ with $2 \leq i \leq u$. In this case, we must have $(\mathcal{V}^{pre}, \mathcal{A}) \models D_{p,i} \sqsubseteq C_{p,i}$, and domain elements $b_{p,i}$ such that $(a_p, b_{p,i}) \in s_i^{\mathcal{I}}$ and $b_{p,i} \notin C_{p,i}^{\mathcal{I}}$. Then $\mathcal{V}^{pre} \models D_{p,1} \sqcup \forall s_1^-(G_p \sqcup \bigsqcup_{i=2}^u \forall s_i.C_{p,i})$. In this derivation we did not rely on clauses where $D_1$ occur. So, by construction we have $\mathcal{V}^{post} \models D_{p,1} \sqcup \forall s_1^-(G_p \sqcup \bigsqcup_{i=2}^u \forall s_i.C_{p,i})$. We also have $\mathcal{V}^{post} \models \neg D_p \sqcup \neg D_{p,1} \sqcup C_p$ by assumption. So, $\mathcal{V}^{post} \models \neg D_p \sqcup \forall s_1^-(G_p \sqcup \bigsqcup_{i=2}^u \forall s_i.C_{p,i}) \sqcup C_p$. From 6 we get that $b_p \in (\forall s_1^-(G_p \sqcup \bigsqcup_{i=2}^u \forall s_i.C_{p,i}) \sqcup C_p)^{\mathcal{I}_0}$. Since $a_p \notin (G_p \sqcup \bigsqcup_{i=2}^u \forall s_i.C_{p,i})^{\mathcal{I}_0}$ we get that $b_p \in C_p^{\mathcal{I}_0}$ which contradicts the assumption that $b_p \notin C_p^{\mathcal{I}_0}$.

We conclude from the above that there are models in $\mathcal{M}$ where $a_p$ is in the interpretation of $\forall s_i.D_{p,i}$ for some values of $i$ with $2 \leq i \leq u$. We transform all such models by removing every $s_1$ successor of $a_p$ from the interpretation of $D_{p,1}$, and denote the transformed models by $\mathcal{N}$. We can see that the models of $\mathcal{N}$ are models of $(\mathcal{V}^{pre}, \mathcal{A})$ because in each model $\mathcal{I} \in \mathcal{N}$ we have $a_p \in (\forall s_i.D_{p,i})^{\mathcal{I}}$ for at least one value of $i$ with $2 \leq i \leq u$.

If $b_p$ is in the interpretation of $C_p$ in every model from $\mathcal{N}$, then one of the following must be true.

1. $b_p \in ind(\mathcal{A})$, and $\mathcal{A} \models C_p(b_p)$.
2. $\mathcal{V}_0 \models \neg C'_p \sqcup C_p$, and for every model $\mathcal{I}$ of $(\mathcal{V}_0, \mathcal{A})$ we have $C'_p(b_p)$.
3. $(a_p, b_p) \in s_i^{\mathcal{I}}$ for some value of $i$ with $2 \leq i \leq u$ and every $\mathcal{I} \in \mathcal{N}$, where $\neg D_p \sqcup \neg D_{p,i} \sqcup C_{p,i}$ are clauses in $\mathcal{V}^{pre}$, and $\mathcal{V}_0 \models C_{p,i} \sqsubseteq C_p$.

The conditions 1 and 2 imply that $b_p \in C_p^{\mathcal{I}_0}$ which contradicts the earlier assumption that $b_p \notin C_p^{\mathcal{I}_0}$. When the side conditions of 3 are satisfied, we must have had the following conclusion obtained from a UB inference.

$$(78) \qquad \neg D_p \sqcup \forall s_1^-.G_p \sqcup \cdots \sqcup \forall s_u^-.G_p \sqcup C_p \sqcup C_{p,2} \sqcup \cdots \sqcup C_{p,u}$$

The clause (78) must be in $\mathcal{V}^{post}$, because we require ordering of the inferences that computes it before performing the LB inference whose conclusion uses $D$ as a top level positive literal. (78) implies that $b_p \in C_p^{\mathcal{I}_0}$, which contradicts our initial assumption.

Therefore, there must exist a model $\mathcal{I}_p$ of $\mathcal{V}^{pre}, \mathcal{A}$ where $b_p \notin C_p^{\mathcal{I}_p}$. The same argument shows that there is a model $\mathcal{I}_q$ of $(\mathcal{V}^{pre}, \mathcal{A})$ over $\Delta^{\mathcal{I}_0}$ where $b_q \notin C_q^{\mathcal{I}_q}$.

The result established by Lemma 21 is that some but not all models of $(\mathcal{V}^{pre}, \mathcal{A})$ have $b_p$ in the interpretation of $C_p$. In the same way some but not all models of $(\mathcal{V}^{pre}, \mathcal{A})$ have $b_q$ in the interpretation of $C_q$. We use this result in the following

Lemma to show that if $(\mathcal{V}^{pre}, \mathcal{A}) \models q(\vec{a})$ then $\mathcal{I}_0 \models q(\vec{a})$. With this we will have the contradiction described before Definition 15 established.

**Lemma 22.** *Let $q(\vec{x})$ be a conjunctive query, and $\vec{a}$ an answer to the query in $(\mathcal{V}^{pre}, \mathcal{A})$. That is, $(\mathcal{V}^{pre}, \mathcal{A}) \models q(\vec{a})$. We have $\mathcal{I}_0 \models q(\vec{a})$.*

*Proof.* Assume $\mathcal{I}_0 \not\models q(\vec{a})$. Let $\mathcal{J}$ be the model obtained from $\mathcal{I}_0$ by performing the transformations in Cases 1, 2, 3, and 4 when the relevant conditions are true. When the conditions of Case 5 are true for a domain element $b$, we remove $b$ from the interpretation of $D_1$, and add every $r_p$ successor of $a$ to the interpretation of $D_p$. Denote by $\alpha$ the $r_p$ successors of $a$ which are added to the interpretation of $D_p$, and are not in the interpretation of $C_p$. The models $\mathcal{I}_0$ and $\mathcal{J}$ are bisimilar with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$, because all applied transformations change the interpretations of definers which are not in the set $sig(\mathcal{O})\backslash\mathcal{F}$. So, we have that $\mathcal{J} \not\models q(\vec{a})$. If $\alpha = \emptyset$, then only the transformations in Cases 1, 2, 3, and 4 were performed, and we have $\mathcal{J} \models (\mathcal{V}^{pre}, \mathcal{A})$. Moreover, the following contradicting remarks would be true about $\mathcal{J}$.

1. $\mathcal{J} \models q(\vec{a})$, because $\mathcal{J} \models (\mathcal{V}^{pre}, \mathcal{A})$ and $(\mathcal{V}^{pre}, \mathcal{A}) \models q(\vec{a})$.
2. $\mathcal{J} \not\models q(\vec{a})$, because $\mathcal{J} \not\models q(\vec{a})$ as explained above.

The contradiction implies that if $\alpha = \emptyset$ then $\mathcal{I}_0 \models q(\vec{a})$.

Suppose $\alpha \neq \emptyset$. The model $\mathcal{J}$ is not a model of $(\mathcal{V}^{pre}, \mathcal{A})$ because the domain elements added to the interpretation of $D_p$ are not in the interpretation of $C_p$. We can nevertheless show that $\mathcal{J} \models q(\vec{a})$.

Recall that $(\mathcal{V}^{pre}, \mathcal{A}) \models q(\vec{a})$, where $q(\vec{x})$ is a conjunctive query, $\vec{x} = x_1, \ldots, x_k$ are the answer variables, and $\vec{a} = a_1, a_2, \ldots, a_k$ is an answer to the query. There must be a map $\pi$ with $\pi(x_i) = a_i$ for $1 \leq i \leq k$, and for every anonymous variable $y$ in $q$, and every model $\mathcal{I}$ of $(\mathcal{V}^{pre}, \mathcal{A})$, $\pi$ maps $y$ to an element in $\Delta^{\mathcal{I}}$ such that the following is true.

1. If $A(u)$ is an atom in $q$, then $\pi(u) \in A^{\mathcal{I}}$.
2. If $r(u, v)$ is an atom in $q$, then $(\pi(u), \pi(v)) \in r^{\mathcal{I}}$.

From the construction of $\mathcal{J}$ we have $\mathcal{J} \not\models q(\vec{a})$ implies the following conditions are true.

1. $u$ is a variable in $q$, and
2. For every model $\mathcal{I} \in \mathcal{M}$, there is a map $\pi_{\mathcal{I}}$ as above that maps $u$ to $b \in \alpha$.

We prove the above implication by contradiction. Let $\mathcal{I}$ be a model from $\mathcal{M}$. We have $\mathcal{I} \models q(\vec{a})$, because $\mathcal{I} \models (\mathcal{V}^{pre}, \mathcal{A})$. Therefore, there must be a map $\pi$ that maps $x_i$ to $a_i$ for every $1 \leq i \leq k$, such that the conditions noted above about $\pi$ are true. Suppose that for every variable $u$ occurring in $q$, we have $\pi(u) \notin \alpha$. Then, $\pi$ maps every variable in $q$ to elements in $\Delta^{\mathcal{J}}$. That is, $\mathcal{J} \models q(\vec{a})$ which contradicts the assumption above.

Now, Lemma 21 shows that there exist a model in $\mathcal{M}$ where $\alpha = \emptyset$. Thus, from the contrapositive of the above implication, we get that $\mathcal{J} \models q(\vec{a})$.

Since $\mathcal{I}_0$ and $\mathcal{J}$ are bisimilar with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$, we have $\mathcal{I}_0 \models q(\vec{a})$.

It remains to consider the elimination of the clauses from $\varDelta^u$ where two or more negative definers occur. This step is performed after all resolution inferences has been performed. Recall that clauses used in resolution inferences are removed when resolution has been performed. Therefore, the clauses removed by this step have not been used in resolution inferences. Suppose a clause $\mathcal{C} = \neg D_1 \sqcup \cdots \sqcup \neg D_n \sqcup C$ is removed. No LB inferences could have been performed to obtain clauses where the definers $D_1, \ldots, D_n$ occur positively as top-level literals. If such clauses were obtained, then resolution would have been performed, and $\mathcal{C}$ would have been removed.

The clause $\mathcal{C}$ cannot be used in LB inferences if one of the following is true.

1. A negative existential definer $D$ occurs in $G$.
2. There are two definers, say $D_1$ and $D_2$, where $\mathcal{V}^q_{ALC} \not\models D_1 \sqsubseteq C_1$ and $\mathcal{V}^q_{ALC} \not\models D_2 \sqsubseteq C_2$ for any concepts $C_1 \not\equiv \top$ and $C_2 \not\equiv \top$.

When the clauses which use two or more negative definers has been eliminated all remaining definers would be simple definers. Consequently, they will be eliminated by the simple definer elimination process. If 2 is satisfied, then the two definers $D_1$ and $D_2$ would be eliminated using purification, and $\mathcal{C}$ will be eliminated as it becomes a tautology. If 1 is satisfied, and for every definer $D_i$ $\mathcal{V}^q_{ALCI}$ has a clause $\neg D_i \sqcup C$ where $1 \leq i \leq n$, then the definers will be eliminated by *Non-cyclic Definer Elimination* inferences.

Denote by $D_e$ the set of definers occurring positively in a clause $\mathcal{C}$ satisfying 1. We assume that the elimination of clauses from $\varDelta^u$ and the elimination of simple definers have been performed in two consecutive steps.

The first step removes the clauses of $\varDelta^u$ that use definers from $D_e$ with negative polarity, and eliminates simple definers immediately after the clauses have been removed. Let $\mathcal{V}_e$ be the snapshot of the ontology obtained after eliminating the simple definers.

The second step removes the remainder of the clauses of $\varDelta^u$, and purifies the definers. Let $\mathcal{V}_\top$ be the snapshot of $\mathcal{V}^q_{ALCI}$ obtained after the step has completed.

We consider the first step. Let $\mathcal{V}_1$ be the snapshot of $\mathcal{V}^q_{ALCI}$ before removing a clause from $\varDelta^u$ that uses a definer from $D_e$ with negative polarity, and $\mathcal{V}_2$ be the snapshot after removing the clause. We show that $\mathcal{V}_1$ and $\mathcal{V}_2$ are query inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ by showing the following.

1. Every model $\mathcal{I}$ of $(\mathcal{V}_1, \mathcal{A})$ is a model of $(\mathcal{V}_2, \mathcal{A})$.
2. Every model $\mathcal{I}$ of $(\mathcal{V}_2, \mathcal{A})$ can be transformed to a model $\mathcal{J}$ of $(\mathcal{V}_1, \mathcal{A})$ such that $\mathcal{I}$ and $\mathcal{J}$ are bisimilar with respect to $sig(\mathcal{O})\backslash\mathcal{F}$ and $ind(\mathcal{A})$.

To prove 1, let $\mathcal{I}$ be a model of $(\mathcal{V}_1, \mathcal{A})$. So, $\mathcal{I} \models \mathcal{V}_1$ and $\mathcal{I} \models \mathcal{A}$. Since $\mathcal{V}_2 \subseteq \mathcal{V}_1$ we have $\mathcal{I} \models \mathcal{V}_2$. Hence $\mathcal{I} \models (\mathcal{V}_2, \mathcal{A})$.

To prove 2, let $\mathcal{I}$ be a model of $(\mathcal{V}_2, \mathcal{A})$. Suppose $\neg D_i \sqcup D'_i \sqcup C_i$ is the clause removed from $\mathcal{V}_2$ such that the following clauses are present in $\mathcal{V}_2$.

$$G \sqcup \exists r.D \tag{79}$$

$$\neg D \sqcup H \sqcup \forall r_1.D_1 \sqcup \cdots \sqcup \forall r_n.D_n \tag{80}$$

$$\neg D_1 \sqcup \neg D'_1 \sqcup C_1 \tag{81}$$

$$\neg D_1 \sqcup \neg D'_1 \sqcup C_1 \tag{82}$$

$$\dots \tag{83}$$

$$\neg D_n \sqcup \neg D'_n \sqcup C_n \tag{84}$$

$$\tag{85}$$

where $1 \leq i \leq n, G, H, C_1, \dots, C_n$ are $\mathcal{ALC}$ concepts, $D_1, \dots, D_n$ and $D'_1, \dots, D'_n$ are complex universal definers, and $D$ is an existential definer.

Suppose for every $e \in D_i^{\mathcal{I}}$ we have $e \notin D_i'^{\mathcal{I}}$ or $e \in C_i^{\mathcal{I}}$, then $\mathcal{I} \models \neg D_i \sqcup \neg D'_i \sqcup C_i$. Since $\mathcal{I} \models \mathcal{V}_1$ and $\mathcal{I} \models \mathcal{A}$, we would have that that $\mathcal{I} \models (\mathcal{V}_2, \mathcal{A})$.

Suppose $\mathcal{I} \not\models (\mathcal{V}_2, \mathcal{A})$. We must have $d_1, d_2, e_i \in \Delta^{\mathcal{I}}$ such that:

$$d_1 \notin G^{\mathcal{I}} \tag{86}$$

$$d_2 \in D^{\mathcal{I}} \tag{87}$$

$$(d_1, d_2) \in r^{\mathcal{I}} \tag{88}$$

$$d_2 \in (\forall r_i.D_i)^{\mathcal{I}} \tag{89}$$

$$(d_2, e_i) \in r_i^{\mathcal{I}} \tag{90}$$

$$e_i \notin C_i^{\mathcal{I}} \tag{91}$$

$$e_i \in D_i'^{\mathcal{I}} \tag{92}$$

We extend $\mathcal{I}$ to $\mathcal{J}$ such that $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}} \cup \{d'_2, e'_i\}$, and the following is true.

1. $(d_1, d'_2) \in r^{\mathcal{J}}$,
2. $d'_2 \in D^{\mathcal{J}}$
3. $d'_2 \in A^{\mathcal{J}}$ if and only if $d_2 \in A^{\mathcal{I}}$ for every concept name $A$,
4. $e'_i \in A^{\mathcal{J}}$ if and only if $e_i \in A^{\mathcal{I}}$ for every concept name $A$,
5. $d'_2 \in D^{\mathcal{J}}$ if and only if $d_2 \in D^{\mathcal{I}}$ for every definer $D$,
6. $e'_i \in D^{\mathcal{J}}$ if and only if $e_i \in D^{\mathcal{I}}$ for every definer $D \neq D'_i$,
7. $(d'_2, e) \in s^{\mathcal{J}}$ if and only if $(d_2, e) \in s^{\mathcal{I}}$, for every domain element $e \neq e_i$ and role name $s$,
8. $(d'_2, e'_i) \in r_i^{\mathcal{J}}$,
9. $(e'_i, f) \in s^{\mathcal{J}}$ if and only if $(e_i, f) \in s^{\mathcal{I}}$, for every domain element $f$ and role name $s$, and
10. $d_2 \notin D^{\mathcal{J}}$.

The transformation above copies $d_2$ to $d'_2$, and $e_i$ to $e'_i$. Then, it removes $d_2$ from the interpretation of $D$, and $e'_i$ from the interpretation of $D'_i$. We can see that $\mathcal{J} \models (\mathcal{V}, \mathcal{A})$. Moreover, $\mathcal{J}$ and $\mathcal{I}$ are bisimilar with respect to $sig(\mathcal{O}) \backslash \mathcal{F}$ and $ind(\mathcal{A})$, because the transformation creates copies of elements which preserves the bisimulation, and changes the interpretations of $D$ and $D'_i$ which are not in $sig(\mathcal{O}) \backslash \mathcal{F}$.

From the above discussion we get that $\mathcal{V}_1$ and $\mathcal{V}_2$ are query inseparable with respect to $sig(\mathcal{O}) \backslash \mathcal{F}$. Since we eliminate simple definers from $\mathcal{V}_2$, and the elimination of simple definers preserves query inseparability with respect to $sig(\mathcal{O}) \backslash \mathcal{F}$, we get from Lemma **??** the required result.

We consider the second step that removes the clauses of $\Delta^u$ that do not use definers from $D_e$, and eliminates the simple definers. Let $\mathcal{V}_1$ be the snapshot of $\mathcal{V}_e$ before removing the clauses of $\Delta^u$. We can view $\mathcal{V}_1$ as follows.

$$\mathcal{V}_1 = \mathcal{V}_0 \cup \mathcal{V}_\forall \cup \mathcal{V}_D \cup \mathcal{V}_D^2 \tag{93}$$

where:

1. $\mathcal{V}_0$ is the set of clauses that do not use definers.
2. $\mathcal{V}_\forall$ is the set of clauses of the form $G \sqcup \forall r_1.D_1 \sqcup \cdots \sqcup \forall r_n.D_n$ where $D_1, \ldots, D_n$ are definers.
3. $\mathcal{V}_D$ is the set of clauses of the form $\neg D \sqcup C$ where $D$ is a definer, and $C$ is a concept that does not use definers.
4. $\mathcal{V}_D^2$ is the set of clauses of the form $\neg D_1 \sqcup \neg D_2 \sqcup C$ where $D_1$ and $D_2$ are definers, and $C$ is a concept.

The output of this step is $\mathcal{V}_0$. We obtain the same output in a different way that we show to preserve query inseparability between $\mathcal{V}_1$ and $\mathcal{V}_0$ with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.

Recall that $\mathcal{V}_1$ and the input ontology $\mathcal{O}$ are query inseparable with respect to $\mathcal{F}(\mathcal{O})\backslash\mathcal{F}$. Let $\mathcal{N}$ be the set of definers satisfying the conditions in 2. These would be the definers not occurring in the clauses of $\mathcal{V}_D$. Assume there are $n$ definers in $\mathcal{N}$. Let $N_t$ be a set of $n$ concept names such that $N_t \cap N_c = \emptyset$, $N_t \cap N_d = \emptyset$, and $N_t \cap N_r = \emptyset$. Consider the ontology $\mathcal{V}_2 = \mathcal{V}_1 \cup \Delta_t$, where $\Delta_t$ is defined as follow.

$$\Delta_t = \{\neg D_i \sqcup S_i \mid D_i \in \mathcal{N}, S_i \in N_t \text{ and } 1 \leq i \leq n\} \tag{94}$$

We show that $\mathcal{V}_1$ and $\mathcal{V}_2$ are model inseparable with respect to $sig(\mathcal{V}_1)$. To prove the statement we show that the following are true.

1. Every model of $\mathcal{V}_2$ is a model of $\mathcal{V}_1$.
2. Every model $\mathcal{I}$ of $\mathcal{V}_1$ can be extended to a model $\mathcal{J}$ of $\mathcal{V}_2$ such that $\mathcal{I}$ and $\mathcal{J}$ $sig(\mathcal{V}_1)$-coincide.

Let $\mathcal{I}$ be a model of $\mathcal{V}_2$, then $\mathcal{I}$ is a model of $\mathcal{V}_1$ because removing axioms does not invalidate models.

Let $\mathcal{I}$ be a model of $\mathcal{V}_1$, and $\mathcal{J}$ the model extended from $\mathcal{I}$ such that $S_i^{\mathcal{J}} = \Delta^{\mathcal{J}}$ for every $S_i \in N_t$ with $1 \leq i \leq n$. We have $\mathcal{J} \models \mathcal{V}_2$. Moreover, we have that $\mathcal{I}$ and $\mathcal{J}$ $\{sig(\mathcal{O})\backslash\mathcal{F}\}$-coincide, because the interpretations of the symbols in $\mathcal{V}_1$ did not change in $\mathcal{J}$.

Since $\mathcal{V}_1$ and $\mathcal{V}_2$ are model inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$, we get from Lemma **??** that the two ontologies are query inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.

A difference between $\mathcal{V}_2$ and $\mathcal{V}_1$ is that LB inferences can be performed on the clauses of $\mathcal{V}_2$. Let $G_i \sqcup \forall r_1.D_1 \sqcup \cdots \sqcup \forall r_n.D_n$ be a clause in $\mathcal{V}_\forall$. A set of $n$ LB inferences would be performed to obtain conclusions of the form $D_i \sqcup E$, where $E$ is a concept of the form $\forall r^-.C$, and $\forall r.S$ is a disjunct in $C$, where $r \in \{r_1, \ldots, r_n\}$ and $S \in N_t$. With the LB conclusions obtained, we resolve the obtained conclusions and the clauses in $\Delta^u$. When resolution has been exhaustively performed, the premises of the resolution inferences are removed. This will remove the clauses in the two sets $\mathcal{V}_D$ and $\mathcal{V}_D^2$.

Each computed resolvent would have a disjunct of the form of $\forall r_i^-.C$ as described above. Let $\mathcal{V}_3$ be the ontology obtained, and denote by $\mathcal{V}_q$ the set of the clauses computed by resolution. $\mathcal{V}_3$ can be viewed as follows.

$$\mathcal{V}_3 = \mathcal{V}_0 \cup \mathcal{V}_\forall \cup \mathcal{V}_q \tag{95}$$

As we only perform definer elimination, the two ontologies $\mathcal{V}_2$ and $\mathcal{V}_3$ are query inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.

Next, we forget the symbols $S \in N_t$ from $\mathcal{V}_3$, and purify the definers occurring in $\mathcal{V}_\forall$. The definers are eliminated using definer purification because they occur only with positive polarity in $\mathcal{V}_\forall$. When the definers are purified the clauses of $\mathcal{V}_\forall$ would be removed as they become tautologies. The symbols $S \in N_t$ are eliminated by purification because they occur only positively in $\mathcal{V}_q$. The purification of the symbols $S$ makes the clauses of $\mathcal{V}_q$ tautologies. To see this consider the clauses obtained by resolution. As discussed above each resolvent would use a disjunct of the form $\forall r_i^-.C$ where $\forall r.S$ is a disjunct in $C$. If $\top$ is substituted for $S$ by purification, the concept $\forall r.S$ would be equivalent to $\top$. Furthermore, the disjunct $\forall r_i^-.C$ would be equivalent to $\top$, and the resolvent would be a tautology. Let $\mathcal{V}_4$ be the ontology obtained. We have $\mathcal{V}_4 = \mathcal{V}_0$ because $\mathcal{V}_\forall$ and $\mathcal{V}_q$ are removed by the purification inferences, and $\mathcal{V}_0$ is not impacted because definers and symbols from $N_t$ do not occur in $\mathcal{V}_0$. Since purification preserves model inseparability, we get from Lemma **??** that $\mathcal{V}_3$ and $\mathcal{V}_4$ are query inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.

We now have the following.

1. $\mathcal{V}_1$ and $\mathcal{V}_2$ are query inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.
2. $\mathcal{V}_2$ and $\mathcal{V}_3$ are query inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.
3. $\mathcal{V}_3$ and $\mathcal{V}_4$ are query inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.

From Lemma **??** we get that $\mathcal{V}_1$ and $\mathcal{V}_4$ are query inseparable with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.