Strong Role Forgetting for $\mathcal{ALCQ}$-Ontologies

Author Name
Affiliation
email@example.com

Abstract

This paper presents the first method for strong forgetting in the description logic $\mathcal{ALCQ}$, specifically designed to address the forgetting of role names. The method is terminating and sound, overcoming the inherent unsolvability of the problem that limits its completeness. An empirical evaluation across a wide range of real-world ontologies has shown that this partial incompleteness does not noticeably affect the method's practicality; the method has demonstrated remarkable success rates over two prominent benchmark datasets, Oxford-ISG and BioPortal, delivering results within seconds. The key to this efficiency lies in our novel linear strategy for introducing definers, a stark contrast to the SOTA LETHE's strategy, which induces an exponential increase in definers and consequent computational inefficiency.

1 Introduction

Forgetting refers to the process of eliminating specific symbols (like constants, functions, or predicates) from a logical theory and adjusting the theory so that, for any sentences not involving these symbols, the modified theory preserves the same models and logical consequences as the original. This process, which can be seen as a formal, rigorous approach to what in everyday terms might be thought of as "ignoring" or "abstracting away from" specific details in a body of knowledge, results in a new theory that is simpler and more focused but still maintains certain essential properties of the original, larger theory. While the term "forgetting" was first coined in the paper by Lin and Reiter (Lin and Reiter 1994), its conceptual foundation can be traced back to the early work of Boole on propositional variable elimination (Boole 1854) and the seminal work of Ackermann (Ackermann 1935), who recognized the problem as a process of eliminating existential second-order quantifiers.

In logic, forgetting has been explored as a problem analogous to, or potentially the dual of, uniform interpolation (Visser 1996; D'Agostino and Hollenberg 2000; Herzig and Mengin 2008), a notion closely related to Craig interpolation (Craig 1957), yet with more stringent constraints. In AI, the significance of forgetting has been widely acknowledged spanning various subdomains, including *knowledge representation* (Lang, Liberatore, and Marquis 2003; Baral and Zhang 2005; Lang and Marquis 2010), *logic programming* (Lifschitz, Pearce, and Valverde 2001; Zhang and Foo 2006; Eiter and Wang 2008; Wang, Wang, and Zhang 2013; Delgrande and Wang 2015), *belief revision* (Alchourrón, Gärdenfors, and Makinson 1985; Liberatore and Schaerf 1998), *circumscription* (Doherty, Lukaszewicz, and Szalas 1997; Wernhard 2012), *action planning* (Nebel, Dimopoulos, and Koehler 1997; Erdem and Ferraris 2007), and *specification refinement* (Bicarregui et al. 2001). Recent strides in the study of forgetting have been primarily observed within the domain of Description Logic (DL) (Baader et al. 2017), which have profoundly influenced research areas such as ontology engineering and ontology-based knowledge management. This emerging trend harnesses the inherent capabilities of forgetting to customize ontological knowledge for specific application needs, or to create decomposed views of ontologies, crucial for both their reuse and detailed analysis (Xiang et al. 2022; Zhao 2023). This approach proves instrumental in a range of tasks where it is essential to preserve the broad functionality of an ontology, while the accessibility must be restricted to a specific subset of symbols. These tasks include *ontology merging and alignment* (Wang et al. 2005; Lambrix and Tan 2008), *debugging and repair* (Schlobach and Cornet 2003; Ribeiro and Wassermann 2009; Troquard et al. 2018), *versioning* (Klein and Fensel 2001; Plessers and Troyer 2005), *abduction and explanation generation* (Del-Pinto and Schmidt 2019; Koopmann et al. 2020; Koopmann 2021), *logical difference* (Konev et al. 2012; Zhao et al. 2019; Liu et al. 2021), and *interactive ontology revision* (Nikitina, Rudolph, and Glimm 2012).

Forgetting can be formalized in two closely related ways. Initially, Lin and Reiter (Lin and Reiter 1994) formalized forgetting within first-order logic from a *model-theoretic* perspective, where forgetting a predicate P in a logical theory $\mathcal{T}$ yields a new theory $\mathcal{T}'$ characterized by models that concur with those of $\mathcal{T}$, except possibly in the interpretation of P. They further identified that for finite theories, the problem of forgetting a predicate amounts to the elimination of existential second-order quantifiers over predicates (Gabbay, Schmidt, and Szalas 2008). This alignment implies that the results of forgetting extend beyond first-order definability and they can be computed using second-order quantifier elimination methods such as SCAN (Gabbay and Ohlbach 1992), DLS (Szalas 1993), SQEMA (Conradie, Goranko, and Vakarelov 2006), MSQEL (Schmidt 2012), etc. Lin and Reiter's notion of forgetting was later termed "strong forgetting” by Zhang and Zhou (Zhang and Zhou 2010), who then proposed from a *deductive* perspective an alternative “weak forgetting” for first-order logic. Here, the process of weakly forgetting a predicate P in a logical theory $\mathcal{T}$ yields the set $\mathcal{T}'$ comprising all first-order logical consequences of $\mathcal{T}$ irrelevant to P.[1] They further distinguished that weak forgetting yields results $\mathcal{T}_1$ that are generally weaker than those of strong forgetting $\mathcal{T}_2$, i.e., $\mathcal{T}_2 \models \mathcal{T}_1$, but these two notions coincide, i.e, $\mathcal{T}_2 \equiv \mathcal{T}_1$, when $\mathcal{T}_2$ is first-order definable. While always first-order definable, $\mathcal{T}_1$ may feature an infinite set of first-order formulas.

The definitions of strong and weak forgetting have subsequently been generalized to various DLs. In these contexts, they are characterized in terms of (model-theoretic or deductive) *inseparability* and *conservative extension* (Ghilardi, Lutz, and Wolter 2006; Grau et al. 2008; Lutz and Wolter 2010; Konev et al. 2013). Research in this domain has been focused on the following problems:

(i) Determining whether a DL $\mathcal{L}$ is closed under forgetting;

(ii) Analyzing the computational complexity of deciding if a forgetting result exists for $\mathcal{L}$; when it does, characterizing the dimensional attributes of such results;

(iii) Investigating the computational complexity of deriving forgetting results for $\mathcal{L}$;

(iv) Developing and optimizing practical methodologies to compute results of forgetting for $\mathcal{L}$.

Significant theoretical findings include:

(i) Very few DLs are known to be closed under forgetting, whether it be under the strong or weak notion, even when the source language is $\mathcal{EL}$ having very limited expressivity. This means that for a forgetting problem with a source language of $\mathcal{EL}$, a forgetting result within the expressivity of $\mathcal{EL}$ does not necessarily exist (Lutz and Wolter 2010). This also applies to $\mathcal{ALC}$ (Ghilardi, Lutz, and Wolter 2006);

(ii) Determining whether a result exists for strong forgetting is undecidable on $\mathcal{EL}$ and $\mathcal{ALC}$ (Konev et al. 2013);

(iii) Determining whether a result exists for weak forgetting is ExpTime-complete on $\mathcal{EL}$ (Lutz, Seylan, and Wolter 2012; Nikitina and Rudolph 2014) and 2ExpTime-complete on $\mathcal{ALC}$ (Lutz and Wolter 2011);

(iv) For $\mathcal{EL}$ and $\mathcal{ALC}$, the result of weak forgetting can be triple exponential in size compared to the source ontology in the worst-case scenario (Lutz, Seylan, and Wolter 2012; Nikitina and Rudolph 2014; Lutz and Wolter 2011).

These results definitively establish that the problem of forgetting constitutes a significant computational challenge.

The focus of this paper is on developing practical methods for strong forgetting in DLs with qualified number restrictions. Historically, strong forgetting has primarily focused on concept names, likely due to the inherent difficulties associated with role forgetting. An exception emerged in 2016 with the Ackermann-based Fame method (Zhao and Schmidt 2016) that enabled the forgetting of both concept and role names in $\mathcal{ALCOIH}$. Endeavors to extend this approach to incorporate $\mathcal{Q}$ produces only weak forgetting results (Zhao and Schmidt 2017; Zhao and Schmidt 2018b). As of now, there exists a void in strong forgetting algorithms for DLs incorporating $\mathcal{Q}$. In the realm of weak forgetting, Lethe (Koopmann 2020) stands out, offering the capability to compute weak forgetting results across a spectrum of languages from $\mathcal{ALC}$ to $\mathcal{SHQ}$, inclusive of role names.

This paper presents the first method for strong forgetting in DLs with qualified number restrictions $\mathcal{Q}$, specifically addressing the forgetting of role names. This specific focus is driven by the absence of strong forgetting approaches for role names in such DLs. We do not extend this approach to concept names because strong forgetting results for concept names in DLs with $\mathcal{Q}$ are generally not expressible within existing DLs. Our method processes an $\mathcal{ALCQ}$-ontology as input, with the output potentially remaining within $\mathcal{ALCQ}$ or extending to $\mathcal{ALCQ}(\nabla)$ by further including the universal role ∇ to avoid information loss. For example, strongly forgetting the role name r from the $\mathcal{ALCQ}$-ontology $\mathcal{O} = \{A \sqsubseteq {\geq}1r.B, {\geq}1r.B \sqsubseteq B\}$ yields the $\mathcal{ALCQ}(\nabla)$-ontology $\mathcal{V} = \{A \sqsubseteq {\geq}1\nabla.B, A \sqsubseteq B\}$; whereas in a DL without the universal role, the result would be $\mathcal{V}' = \{A \sqsubseteq B\}$, which essentially equates to weakly forgetting r. The method has been rigorously validated for termination and soundness, overcoming the inherent difficulty of the problem that limits its completeness. Comprehensive empirical evaluations across a range of real-world ontologies reveal that this partial incompleteness does not markedly affect the method’s efficiency; our approach has demonstrated remarkable success rates over two major benchmark datasets, Oxford-ISG and BioPortal, delivering results within seconds. The key to this efficiency lies in our novel linear strategy for introducing definers, a stark contrast to the SOTA Lethe’s strategy, which induces an exponential increase in definers and consequent computational inefficiency.

A long version of this paper including all missing proofs and additional empirical results, the source code for the prototype implementation alongside the datasets can be found at https://github.com/anonymous-ai-researcher/kr2024.

2 $\mathcal{ALCQ}(\nabla)$-Ontologies & Forgetting

Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be *countably infinite*, *pairwise disjoint* sets of *concept* and *role* names, respectively. *Roles* in $\mathcal{ALCQ}$ can be any role name $r \in \mathsf{N_R}$, while in $\mathcal{ALCQ}(\nabla)$ can additionally be the universal role ∇. *Concept descriptions* (or *concepts* for short) have one of the following forms:

$$\top \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid {\geq}mR.C \mid {\leq}nR.C,$$

where $A \in \mathsf{N_C}$, C and D range over concepts, R over roles, and $m \geq 1$ and $n \geq 0$ are natural numbers. Additional concepts and roles are defined as abbreviations: $\bot = \neg\top$, $\exists R.C = {\geq}1R.C$, $\forall R.C = {\leq}0R.\neg C$, $\neg{\geq}mR.C = {\leq}nR.C$ and $\neg{\leq}nR.C = {\geq}mR.C$ with $n = m - 1$. Concepts of the form ${\geq}mR.C$ and ${\leq}nR.C$ are referred to as *qualified number restrictions*, which allow for specifying cardinality constraints on roles. We assume without loss of generality that concepts and roles are equivalent relative to associativ-

[1]A sentence is deemed irrelevant to a predicate P if it is logically equivalent to another sentence that does not contain P.

ity and commutativity of $\sqcap$ and $\sqcup$, $\top$ and ∇ are units w.r.t. $\sqcap$, and $\neg$ is an involution.

An $\mathcal{ALCQ}(\nabla)$-ontology $\mathcal{O}$ is a finite set of *axioms* of the form $C \sqsubseteq D$, known as *general concept inclusion* (or *GCI*), where C and D are concepts. We use $C \equiv D$ as an abbreviation for the GCIs $C \sqsubseteq D$ and $D \sqsubseteq C$.

The semantics of $\mathcal{ALCQ}(\nabla)$ is defined in terms of an *interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a non-empty set of elements (the *domain of the interpretation*), and $\cdot^{\mathcal{I}}$ is the *interpretation function* that maps every concept name $A \in \mathsf{N_C}$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and every role name $r \in \mathsf{N_R}$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function $\cdot^{\mathcal{I}}$ is inductively extended to concepts and roles as follows:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad \nabla^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$$

$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$

$$({\geq}mR.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{(x,y) \in R^{\mathcal{I}} \mid y \in C^{\mathcal{I}}\} \geq m\}$$

$$({\leq}nR.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{(x,y) \in R^{\mathcal{I}} \mid y \in C^{\mathcal{I}}\} \leq n\},$$

where $\#\{\cdot\}$ denotes the cardinality of the set $\{\cdot\}$.

Let $\mathcal{I}$ be an interpretation. A GCI $C \sqsubseteq D$ is *true* in $\mathcal{I}$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$, written $\mathcal{I} \models \mathcal{O}$, iff every GCI in $\mathcal{O}$ is *true* in $\mathcal{I}$. A GCI $C \sqsubseteq D$ is a *logical consequence* of $\mathcal{O}$, written $\mathcal{O} \models C \sqsubseteq D$, iff $C \sqsubseteq D$ is true in every model $\mathcal{I}$ of $\mathcal{O}$.

Our forgetting method operates on GCIs in clausal normal form. We define a *literal* as a concept in one of the following forms: A, $\neg A$, ${\geq}mR.C$ or ${\leq}nR.C$, where $A \in \mathsf{N_C}$, $m \geq 1$ and $n \geq 0$ are natural numbers, C a concept, and R a role. A *clause* is a disjunction of finitely many literals.

Let $\mathcal{S} \in \mathsf{N_C} \cup \mathsf{N_R}$ be a designated name. A clause containing $\mathcal{S}$ is termed an *$\mathcal{S}$-clause.* An occurrence of $\mathcal{S}$ is said to be *positive* in an $\mathcal{S}$-clause if it occurs under an even number of (explicit and implicit) negations, and *negative* if under an odd number. To clarify these distinctions, transforming DL clauses into first-order logic uncovers all negation symbols. For example, in the clause ${\geq}2r.A$, r is positive in its first-order form $\forall x \exists y, z(r(x,y) \wedge r(x,z) \wedge A(y) \wedge A(z) \wedge y \neq z)$. Conversely, in the clause ${\leq}1r.A$, r is negative, as shown in $\forall x, y, z(\neg r(x,y) \vee \neg r(x,z) \vee \neg A(y) \vee \neg A(z) \vee y = z)$. An ontology of clauses is said to be *positive* (or *negative*) w.r.t. $\mathcal{S}$ if all occurrences of $\mathcal{S}$ in it are positive (or negative).

A *signature* Σ is a finite subset of concept and role names from $\mathsf{N_C} \cup \mathsf{N_R}$. For any syntactic object X—which may include concepts, roles, GCIs, or ontologies—the sets $\mathsf{sig_C}(X)$ and $\mathsf{sig_R}(X)$ represent the concept and role names in X, respectively. We further let $\mathsf{sig}(X) = \mathsf{sig_C}(X) \cup \mathsf{sig_R}(X)$.

Consider a role name $r \in \mathsf{N_C}$ and two interpretations $\mathcal{I}$ and $\mathcal{I}'$. They are defined as *r-equivalent*, denoted $\mathcal{I} \sim_r \mathcal{I}'$, if they are identical except possibly in how they interpret r. Extending this, $\mathcal{I}$ and $\mathcal{I}'$ are *$\mathcal{F}$-equivalent*, denoted $\mathcal{I} \sim_{\mathcal{F}} \mathcal{I}'$, if they only potentially differ in the interpretations of the names in a role set $\mathcal{F}$. This equivalence establishes the following conditions:

(i) $\mathcal{I}$ and $\mathcal{I}'$ have the same domain ($\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$) and interpret each concept name $A \in \mathsf{N_C}$ identically ($A^{\mathcal{I}} = A^{\mathcal{I}'}$);

(ii) for each role name $r \in \mathsf{N_R}$ not in $\mathcal{F}$, its interpretation is the same in both $\mathcal{I}$ and $\mathcal{I}'$ ($r^{\mathcal{I}} = r^{\mathcal{I}'}$).

Definition 1 (Forgetting). *Let $\mathcal{O}$ be an $\mathcal{ALCQ}$-ontology and $r \in sig_R(\mathcal{O})$ be a role name. An ontology $\mathcal{V}$ is a* result of forgetting $\{r\}$ from $\mathcal{O}$ *if the following conditions hold:*

(i) *$sig(\mathcal{V}) \subseteq sig(\mathcal{O}) \backslash \{r\}$, and*

(ii) *for any interpretation $\mathcal{I}'$, $\mathcal{I}' \models \mathcal{V}$ iff there is an interpretation $\mathcal{I} \sim_r \mathcal{I}'$ such that $\mathcal{I} \models \mathcal{O}$.*

More generally, let $\mathcal{F} \subseteq sig_R(\mathcal{O})$ be a finite set of role names, referred to as the forgetting signature. *An ontology $\mathcal{V}$ is a* result of forgetting $\mathcal{F}$ from $\mathcal{O}$ *if the following conditions hold:*

(i) *$sig(\mathcal{V}) \subseteq sig(\mathcal{O}) \backslash \mathcal{F}$, and*

(ii) *for any interpretation $\mathcal{I}'$, $\mathcal{I}' \models \mathcal{V}$ iff there is an interpretation $\mathcal{I} \sim_{\mathcal{F}} \mathcal{I}'$ such that $\mathcal{I} \models \mathcal{O}$.*

It follows from this definition that: (i) the result of forgetting $\mathcal{F}$ from $\mathcal{O}$ can be computed through the elimination of single names in $\mathcal{F}$, irrespective of their order of elimination, and (ii) forgetting is unique up to logical equivalence, that is, any two results, such as $\mathcal{V}$ and $\mathcal{V}'$, derived from the same forgetting action, are logically equivalent, despite potential differences in their explicit representations.

3 Normalization of $\mathcal{ALCQ}(\nabla)$-Ontologies

3.1 r-Normal Form

Given an $\mathcal{ALCQ}(\nabla)$-ontology $\mathcal{O}$ and a forgetting signature $\mathcal{F}$,[2] our method computes the result of forgetting through a process of iterative single name elimination in $\mathcal{F}$. This process is underpinned by a calculus specifically designed for single role name elimination, which operates on specialized normal forms tailored to $\mathcal{ALCQ}(\nabla)$-ontologies.

Definition 2 (r-Normal Form). *An r-clause is in r-*normal form *(or r-*NF*) if it has one of the following forms,*

$$C \sqcup {\geq}mr.D \text{ or } C \sqcup {\leq}nr.D$$

*where $r \in N_R$, and C, D are concepts (not necessarily concept names) not containing r. An $\mathcal{ALCQ}(\nabla)$-ontology $\mathcal{O}$ is in r-*NF *if every r-clause in $\mathcal{O}$ is in r-NF.*

Normal form provides a unified representation of clauses, allowing inference systems to process a consistent input format. For example, consider the $\mathcal{ALCQ}(\nabla)$-clauses $C \sqcup {\geq}mr.D$ and $C \sqcup {\geq}m\nabla.{\leq}nr.D$. If we forget the role name r from these clauses, we would need to develop distinct inference rules for $\mathcal{ALCQ}(\nabla)$-clauses with different syntactic structures. Given that $\mathcal{ALCQ}(\nabla)$ can accommodate an infinite number of concepts and clauses conforming to its syntax, developing specific inference rules for every possible form is impractical. A practical approach is to design a standardized normal form alongside a method to transform any $\mathcal{ALCQ}(\nabla)$-ontology into this normal form. This allows for a finite number of inference rules applicable to this unified normal form. However, in designing such a normal form, it is crucial to ensure that it encompasses all basic structures of $\mathcal{ALCQ}(\nabla)$-clauses. A key feature of r-NF is its ability

[2]Note that $\mathcal{F}$ is not necessarily confined to a subset of $\mathsf{sig}(\mathcal{O})$, as Definition 1 indicates that forgetting a name not present in $\mathcal{O}$ leads to no change to $\mathcal{O}$. Consequently, forgetting a name not present in $\mathcal{O}$ yields $\mathcal{O}$ itself as the result.

to uniformly capture the essence of $\mathcal{ALCQ}(\nabla)$-clauses by ensuring a single occurrence of r, precisely placed either under a $\geq$-restriction or a $\leq$-restriction.

3.2 Transformation to r-Normal Form

Any given $\mathcal{ALCQ}(\nabla)$-ontology $\mathcal{O}$ can be transformed into r-NF by exhaustively applying a set of normalization rules (NR1 − NR4) to the r-clauses in $\mathcal{O}$ that have yet to be in r-NF (C and D are general concepts). These rules involve the use of newly-introduced concept names referred to as *definers* (Koopmann and Schmidt 2013). Definers, such as "Z" in the rules, are uniquely created to act as placeholders or abbreviations for compound concepts during normalization. Analogous to "auxiliary lines" in high school geometry, definers are instrumental in simplifying the structure but are not intended to appear in the final results. Essentially, the introduction of Z enables syntactic adaptation without altering the inherent logical structures within $\mathcal{O}$.

NR1 For each instance of $C \sqcup {\geq}mr.D$ or $C \sqcup {\leq}nr.D$, if $r \in \mathsf{sig}(C)$, replace C by a fresh definer $Z \in \mathsf{N_C}$ and add $\neg Z \sqcup C$ to $\mathcal{O}$;

NR2 For each instance of $C \sqcup {\geq}mr.D$ or $C \sqcup {\leq}nr.D$, if $r \in \mathsf{sig}(D)$, replace D by a fresh definer $Z \in \mathsf{N_C}$ and add $\neg Z \sqcup D$ to $\mathcal{O}$;

NR3 For each instance of $C \sqcup {\geq}ms.D$ or $C \sqcup {\leq}ns.D$ ($s \neq r$), if $r \in \mathsf{sig}(C)$, replace C by a fresh definer $Z \in \mathsf{N_C}$ and add $\neg Z \sqcup C$ to $\mathcal{O}$;

NR4 For each instance of $C \sqcup {\geq}ms.D$ or $C \sqcup {\leq}ns.D$ ($s \neq r$), if $r \in \mathsf{sig}(D)$, replace D by a fresh definer $Z \in \mathsf{N_C}$ and add $\neg Z \sqcup D$ to $\mathcal{O}$.

Example 1. *Consider the following $\mathcal{ALCQ}$-ontology $\mathcal{O}$:*

$$\{1.\ {\leq}1r.{\geq}2r.A_1 \sqcup {\geq}2s.{\leq}1t.{\leq}3r.A_2 \sqcup {\leq}0t.A_3\}$$

Let $\mathcal{F} = \{r\}$. r-NF of $\mathcal{O}$ is computed by exhaustively applying the normalization rules NR1 − NR4 as described above. Consider the literals in a given clause: if the literal ${\leq}1r.{\geq}2r.A_1$ corresponds to the concept ${\leq}nr.D$ in these rules, and the literal ${\geq}2s.{\leq}1t.{\leq}3r.A_2 \sqcup {\leq}0t.A_3$ to the C in these rules, then, based on the side conditions, we can apply either NR1 or NR2. Similarly, if ${\geq}2s.{\leq}1t.{\leq}3r.A_2$ corresponds to the ${\geq}mr.D$ in the rules, and ${\leq}1r.{\geq}2r.A_1 \sqcup {\leq}0t.A_3$ to the C in the rules, then we can apply either NR3 or NR4. This indicates that in the same situation, different normalization rules are available for use. At this point, our method executes NR1 − NR4 in sequence (see below). The sequence of applying these rules does not affect the correctness of the result.

Applying NR1 to Clause 1 gives ($Z_1 \in N_C$ is a fresh definer):

$$\{2.\ {\leq}1r.{\geq}2r.A_1 \sqcup Z_1 \quad 3.\ \neg Z_1 \sqcup {\geq}2s.{\leq}1t.{\leq}3r.A_2 \sqcup {\leq}0t.A_3\}$$

Applying NR2 to Clause 2 gives ($Z_2 \in N_C$ is a fresh definer):

$$\{4.\ {\leq}1r.Z_2 \sqcup Z_1 \quad 5.\ \neg Z_2 \sqcup {\geq}2r.A_1 \quad 3.\ \neg Z_1 \sqcup {\geq}2s.{\leq}1t.{\leq}3r.A_2 \sqcup {\leq}0t.A_3\}$$

Applying NR3 to Clause 3 gives ($Z_3 \in N_C$ is a fresh definer):

$$\{4.\ {\leq}1r.Z_2 \sqcup Z_1 \quad 5.\ \neg Z_2 \sqcup {\geq}2r.A_1 \quad 3.\ \neg Z_1 \sqcup {\geq}2s.Z_3 \sqcup {\leq}0t.A_3 \quad 6.\ \neg Z_3 \sqcup {\leq}1t.{\leq}3r.A_2\}$$

Applying NR4 to Clause 6 gives ($Z_4 \in N_C$ is a fresh definer):

$$\{4.\ {\leq}1r.Z_2 \sqcup Z_1 \quad 5.\ \neg Z_2 \sqcup {\geq}2r.A_1 \quad 7.\ \neg Z_4 \sqcup {\leq}3r.A_2 \quad 3.\ \neg Z_1 \sqcup {\geq}2s.Z_3 \sqcup {\leq}0t.A_3 \quad 6.\ \neg Z_3 \sqcup {\leq}1t.Z_4\}$$

Now the resulting $\mathcal{O}' = \{3, 4, 5, 6, 7\}$ is in r-NF.

When developing such a normalization method, we must ensure the following properties:

- For "*soundness*" of the method, we must ensure that the refined $\mathcal{O}'$ computed by the method is equivalent to the original ontology $\mathcal{O}$ up to the definers introduced;
- For "*completeness*" of the method, we must ensure that any $\mathcal{ALCQ}(\nabla)$-ontology can be transformed into r-NF using this method;
- For "*efficiency*" of the method, we must ensure that the transformation process is terminating and is preferably finished in polynomial time.

Lemma 1. *Let $\mathcal{O}$ be an $\mathcal{ALCQ}(\nabla)$-ontology. Then $\mathcal{O}$ can be transformed into r-NF $\mathcal{O}'$ by a linear number of applications of the above normalization rules NR1 − NR4. In addition, the size of the resulting ontology $\mathcal{O}'$ is linear in the size of $\mathcal{O}$.*

Lemma 2. *Let $\mathcal{O}$ be an $\mathcal{ALCQ}(\nabla)$-ontology, and $\mathcal{O}'$ the normalized one obtained from $\mathcal{O}$ using the normalization rules NR1 − NR4. Let $\mathsf{sig}_D(\mathcal{O}')$ denote the set of definers introduced in $\mathcal{O}'$. Then, for any interpretation $\mathcal{I}'$, the following holds:*

$$\mathcal{I}' \models \mathcal{O}' \text{ iff } \mathcal{I} \models \mathcal{O},$$

for some interpretation $\mathcal{I}$ such that $\mathcal{I} \sim_{\mathsf{sig}_D(\mathcal{O}')} \mathcal{I}'$.

Lemma 1 states the termination and completeness of the transformation and Lemma 2 states its soundness.

4 Different Definer Introduction Strategies

Compared to the SOTA methods Lethe and Fame, our forgetting method adopts a unique normal form specification coupled with a non-traditional, yet notably efficient strategy for definer introduction during the normalization. This strategy markedly enhances the efficiency of our method.

In closely examining Lethe, we analyze its introduction and usage of definers in the normalization process, a critical preliminary step for applying its inference rules. Lethe operates on clauses of the form $L_1 \sqcup \ldots \sqcup L_k$, where each L_i ($1 \leq i \leq k$) is a TBox literal, defined as:

$$A \mid \neg A \mid \exists r.Z \mid \exists r^-.Z \mid \forall r.Z \mid \forall r^-.Z,$$

with $r \in \mathsf{N_R}$ and $A, Z \in \mathsf{N_C}$. Lethe mandates every "Z" — any subconcept immediately below an $\exists$- or a $\forall$-restriction — as a definer throughout the forgetting process. In contrast, our method provides more flexibility for Z, allowing it to be any concept not containing r. These subtle differences have, however, greatly affected the scope of the inference rules these methods employ and, consequently, their efficiency in inference.

To gain a more nuanced understanding of Lethe's strategy for introducing definers, we first fix several notations. The set of definers introduced in an ontology $\mathcal{O}$ is denoted by $\mathsf{sig}_\mathsf{D}(\mathcal{O})$, while $\mathsf{sub}^{\forall}_{\exists}(\mathcal{O})$ denote the set of all subconcepts in $\mathcal{O}$ taking the form $\exists r^{(-)}.X$ or $\forall r^{(-)}.X$, where $r \in \mathsf{N_R}$ and X is an arbitrary concept. Additionally, $\mathsf{sub}_X(\mathcal{O})$ denotes the set of all subconcepts X in $\mathcal{O}$ with $\exists r^{(-)}.X \in \mathsf{sub}^{\forall}_{\exists}(\mathcal{O})$ or $\forall r^{(-)}.X \in \mathsf{sub}^{\forall}_{\exists}(\mathcal{O})$.

Within the LETHE framework, which implements a strategy of reusing definers (i.e., applying a consistent definer for identical subconcepts), an injective function f can be defined over $\mathsf{sig}_\mathsf{D}(\mathcal{O})$, namely $f : \mathsf{sig}_\mathsf{D}(\mathcal{O}) \to \mathsf{sub}_X(\mathcal{O})$. f also exhibits surjectivity, given LETHE's exhaustive manner to introduce defines — requiring every subconcept immediately below an $\exists$- or a $\forall$-restriction to be a definer. On the other hand, within our framework, f is defined as non-surjective. However, for both methods, the number of definers, denoted as $|\mathsf{sig}_\mathsf{D}(\mathcal{O})|$, is bounded by $\mathsf{O}(n)$, with n representing the number of $\exists$- and $\forall$-restrictions in $\mathcal{O}$. This implies a linear growth in the introduction of definers.

LETHE employs a saturation-based reasoning approach to facilitate the elimination of single concept and role names from an ontology $\mathcal{O}$. This process involves the generation of new consequences, which are subsequently added to $\mathcal{O}$, using a generalized resolution calculus, referred to as Res, as described in (Fermüller et al. 2001). LETHE transforms $\mathcal{O}$ into normal form through a bifurcated methodology. The initial phase, termed as the *pre-resolution phase*, witnesses LETHE's inaugural computation of $\mathcal{O}$'s normal form to fire up Res. As previously examined, this stage features a *linear* and *static* introduction of definers.

Transitioning to the second *intra-resolution stage*, LETHE exhaustively applies the inference rules of Res to $\mathcal{O}$ until reaching a saturation state $\mathsf{Res}(\mathcal{O})$. In this phase, a number of new consequences are deduced from existing ones. For example, the application of the $\forall\exists$-propagation rule of Res to the clauses $C_1 \sqcup \forall r.D_1$ and $C_2 \sqcup \exists r.D_2$ yields a new clause $C_1 \sqcup C_2 \sqcup \exists r.(D_1 \sqcap D_2)$. To normalize this clause, LETHE introduces a fresh definer $D_{12} \in \mathsf{N_C}$ to replace the subconcept $D_1 \sqcap D_2$, and subsequently adds $\neg D_{12} \sqcup (D_1 \sqcap D_2)$ to $\mathcal{O}$. Within this stage, definers are *dynamically* introduced as Res iterates over $\mathcal{O}$. An additional injective yet non-surjective function f' can be formulated over $\mathsf{sig}_\mathsf{D}(\mathsf{Res}(\mathcal{O}))$, namely $f : \mathsf{sig}_\mathsf{D}(\mathsf{Res}(\mathcal{O})) \to \mathsf{sub}_X(\mathsf{Res}(\mathcal{O}))$, and the size of its codomain $|\mathsf{sub}_X(\mathsf{Res}(\mathcal{O}))|$ equals $2^{|\mathsf{sub}_\mathsf{D}(\mathcal{O})|}$. The number of definers necessary for the normalization of $\mathsf{Res}(O)$ is thus bounded by $\mathsf{O}(2^n)$, where n denotes the number of $\exists$- and $\forall$-restrictions in $\mathcal{O}$. Therefore, during the intra-resolution stage, LETHE introduces definers at an exponential rate. In contrast, our method restricts its normalization activities within the pre-resolution stage, indicating a linear trajectory in the introduction of definers during its entire forgetting lifespan.

Definers are extraneous to the desired signature and thus should be excluded from the result of forgetting $\mathcal{F}$ from $\mathcal{O}$. Consequently, in the most demanding scenarios,LETHE will be tasked with discarding as many as $2^n + |\mathcal{F}|$ names and executing Res for $2^n + |\mathcal{F}|$ iterations to compute the forgetting result. In contrast, our method introduces a maximum of n definers, and in the worst case, only needs to activate the forgetting calculus (described next) $n + |\mathcal{F}|$ times.

5 Single Name Elimination

Once the given ontology $\mathcal{O}$ has been transformed into r-NF $\mathcal{O}'$, our method applies the inference rule shown in Figure 1 to eliminate r from $\mathcal{O}'$. This rule is bifurcated into two parts: the *premises* above the line and the *conclusion* below it. It operates as a replacement rule, where all premises are replaced by the conclusion, which is the result of forgetting r from $\mathcal{O}'$. $\mathcal{O}'$ typically features three types of clauses: those without r, denoted as $\mathcal{O}^{-r}$, and two types of r-clauses conforming to $C \sqcup {\geq}mr.D$ or $C \sqcup {\leq}nr.D$ r-NFs, denoted as $\mathcal{P}^+(r)$ and $\mathcal{P}^-(r)$, respectively. $\mathcal{P}^+(r)$ contains all positive occurrences of r, while $\mathcal{P}^-(r)$ contains all negative ones.

Addressing a forgetting problem, such as the elimination of r from $\mathcal{O}'$, our method is streamlined as follows: maintain $\mathcal{O}^{-r}$ as is, while combining all r-clauses in $\mathcal{P}^+(r)$ with all in $\mathcal{P}^-(r)$, following the classic binary resolution principle (Robinson 1965). This principle, applied to pairs of clauses containing complementary literals (e.g., $A \vee B$ and $\neg B \vee C$), derives a new clause (e.g., $A \vee C$) by combining all the literals of both clauses. The resulting clause, in this case, $\{A \vee C\}$, maintains logical equivalence to its premises, in this case, $\{A \vee B, \neg B \vee C\}$, up to the excluded literal B.

While unlike propositional resolution, which resolves directly upon propositional variables, in $\mathcal{ALCQ}$, the resolution targets a role name r between two literals ${\geq}mr.D$ and ${\leq}nr.D$ (r is associated with $\geq$- or $\leq$-restrictions). The complementarity between these literals becomes evident via first-order translations; see (Zhao and Schmidt 2018b) for an example. Our inference rule directly resolves the literals ${\geq}mr.D$ and ${\leq}nr.D$ on r, yielding a forgetting result in DL, but the target language mandates an extension of $\mathcal{ALCQ}$ with the universal role (∇) to achieve such representation.

The result of the combination (i.e., the conclusion of the inference rule) is a finite set of clauses devoid of r, denoted as $\mathbf{Block}(\mathcal{P}^+(r), \mathcal{P}^-(r))$. The conclusion of the rule, derived as the union of $\mathcal{O}^{-r}$ and $\mathbf{Block}(\mathcal{P}^+(r), \mathcal{P}^-(r))$, constitutes the result $\mathcal{V}$ of forgetting r from $\mathcal{O}'$. We further note that $\mathbf{Block}(\mathcal{P}^+(r), \mathcal{P}^-(r)) = \bigcup \mathbf{Block}(\mathcal{P}^+(r), X)$ for every clause $X \in \mathcal{P}^-(r)$. Hence, for easier presentation and a better understanding of the rule's results, we present it by breaking it down in Figure 1.

The inference rule produces multiple clause sets, with each representing different tiers of logical consequences. The first set captures 0th-tier implicit logical consequences, which involve reasoning solely from positive r-clauses without negative r-clauses. Utilizing the monotonicity of r in positive r-clauses and the tautology $r \sqsubseteq \nabla$, for a positive r-clause $C_i \sqcup {\geq}x_i r.D_i$, it is possible to infer $C_i \sqcup {\geq}x_i \nabla.D_i$. The original clause means that any element in the domain either belongs to the extension of C_i or has x_i r-successors that belong to the extension of D_i; the inferred result means that for any element in the domain, it either belongs to the extension of C_i or has x_i elements belonging to the extension of D_i. The influence of r is eliminated in the inference result. The second set captures 1st-tier implicit logical consequences, involving monotonic reasoning for one negative r-clause in conjunction with a positive r-clause. The third set captures 2nd-tier implicit logical consequences, which involve the interactions of any two negative r-clauses and a

$$\frac{\mathcal{O}^{-r}, \overbrace{C_1 \sqcup {\geq}x_1 r.D_1, \cdots, C_m \sqcup {\geq}x_m r.D_m}^{\mathcal{P}^+(r)\text{(containing } m \text{ clauses)}} \quad \overbrace{E_1 \sqcup {\leq}y_1 r.F_1, \cdots, E_n \sqcup {\leq}y_n r.F_n}^{\mathcal{P}^-(r)\text{(containing } n \text{ clauses)}}}{\mathcal{O}^{-r}, \textbf{\textsc{Block}}(\mathcal{P}^-(r), C_1 \sqcup {\geq}x_1 r.D_1), \cdots, \textbf{\textsc{Block}}(\mathcal{P}^-(r), C_m \sqcup {\geq}x_m r.D_m)}$$

$$\textbf{\textsc{Block}}(\mathcal{P}^+(r), E_1 \sqcup {\leq}y_1 r.F_1), \cdots, \textbf{\textsc{Block}}(\mathcal{P}^+(r), E_n \sqcup {\leq}y_n r.F_n)$$

1. Block$(\mathcal{P}^-(r), C_i \sqcup {\geq}x_i r.D_i)(1 \leq i \leq m)$ denotes the union of the following sets:

0th-tier: $\{C_i \sqcup {\geq}x_i \nabla.D_i\}$

1st-tier: $\bigcup_{1\leq j \leq n} \{C_i \sqcup E_j \sqcup {\geq}(x_i - y_j)\nabla.(D_i \sqcap \neg F_j)\}$, if $x_i > y_j$;

2nd-tier: $\bigcup_{1\leq j_1 < j_2 \leq n} \{C_i \sqcup E_{j_1} \sqcup E_{j_2} \sqcup \geq (x_i - y_{j_1} - y_{j_2})\nabla.(D_i \sqcap \neg F_{j_1} \sqcap \neg F_{j_2})\}$, if $x_i > y_{j_1} + y_{j_2}$;

...

nth-tier: $C_i \sqcup E_1 \sqcup \ldots \sqcup E_n \sqcup \geq (x_i - y_{j_1} - \cdots - y_{j_n})\nabla.(D_i \sqcap \neg F_1 \sqcap \ldots \sqcap \neg F_n)$, if $x_i > y_{j_1} + \ldots + y_{j_n}$;

2. Block$(\mathcal{P}^+(r),\ E_j \sqcup \leq y_j r.F_j)(1 \leq j \leq n)$ denotes the union of the following sets:

when k is even and $2 \leq k \leq m$, kth-tier:

$$\bigcup_{1\leq i_1 < \cdots < i_k \leq m} \Big\{E_j \sqcup C_{i_1} \sqcup \cdots \sqcup C_{i_k} \sqcup \bigsqcup_{1\leq j_1<j_2\leq k} \geq z_{i_{j_1},i_{j_2}} \nabla.(D_{i_{j_1}} \sqcap D_{i_{j_2}}) \sqcup \bigsqcup_{1\leq j_1<j_2<j_3\leq k} \leq z_{i_{j_1},i_{j_2},i_{j_3}}$$
$$\nabla.(D_{i_{j_1}} \sqcap D_{i_{j_2}} \sqcap D_{i_{j_3}}) \sqcup \cdots \sqcup \geq z_{i_1,\cdots i_k} \nabla.(D_{i_1} \sqcap \cdots \sqcap D_{i_k}) \sqcup \geq \big(x_{i_1} + \cdots + x_{i_k} - \sum_{1\leq j_1<j_2\leq k} (z_{i_{j_1},i_{j_2}} - 1)$$
$$+ \sum_{1\leq j_1<j_2<j_3\leq k} (z_{i_{j_1},i_{j_2},i_{j_3}} + 1) + \cdots + (-z_{i_1,\cdots i_k} + 1) - y_j\big)\nabla.\big((D_{i_1} \sqcup \cdots \sqcup D_{i_k}) \sqcap \neg F_j\big) \;\big|\; z_{i_{j_1},i_{j_2}} \in$$
$$\big[\max\{x_{i_{j_1}}, x_{i_{j_2}}\} + 1\big] \text{ for } 1 \leq j_1 < j_2 \leq k, z_{i_{j_1},i_{j_2},i_{j_3}} \in \big[\min\{x_{i_{j_1}}, x_{i_{j_2}}, x_{i_{j_3}}\} - 1\big] \text{ for } 1 \leq j_1 < j_2 < j_3 \leq k,$$
$$\cdots,\ z_{i_1,\cdots i_k} \in \big[\max\{x_{i_1},\ \cdots,\ x_{i_k}\} + 1\big] \Big\}$$

when k is odd and $2 \leq k \leq m$, kth-tier:

$$\bigcup_{1\leq i_1 < \cdots < i_k \leq m} \Big\{E_j \sqcup C_{i_1} \sqcup \cdots \sqcup C_{i_k} \sqcup \bigsqcup_{1\leq j_1<j_2\leq k} \geq z_{i_{j_1},i_{j_2}} \nabla.(D_{i_{j_1}} \sqcap D_{i_{j_2}}) \sqcup \bigsqcup_{1\leq j_1<j_2<j_3\leq k} \leq z_{i_{j_1},i_{j_2},i_{j_3}}$$
$$\nabla.(D_{i_{j_1}} \sqcap D_{i_{j_2}} \sqcap D_{i_{j_3}}) \sqcup \cdots \sqcup \leq z_{i_1,\cdots i_k} \nabla.(D_{i_1} \sqcap \cdots \sqcap D_{i_k}) \sqcup \geq \big(x_{i_1} + \cdots + x_{i_k} - \sum_{1\leq j_1<j_2\leq k} (z_{i_{j_1},i_{j_2}} - 1)$$
$$+ \sum_{1\leq j_1<j_2<j_3\leq k} (z_{i_{j_1},i_{j_2},i_{j_3}} + 1) + \cdots + (z_{i_1,\cdots i_k} + 1) - y_j\big)\nabla.\big((D_{i_1} \sqcup \cdots \sqcup D_{i_k}) \sqcap \neg F_j\big) \;\big|\; z_{i_{j_1},i_{j_2}} \in$$
$$\big[\max\{x_{i_{j_1}}, x_{i_{j_2}}\} + 1\big] \text{ for } 1 \leq j_1 < j_2 \leq k, z_{i_{j_1},i_{j_2},i_{j_3}} \in \big[\min\{x_{i_{j_1}}, x_{i_{j_2}}, x_{i_{j_3}}\} - 1\big] \text{ for } 1 \leq j_1 < j_2 < j_3 \leq k,$$
$$\cdots,\ z_{i_1,\cdots i_k} \in \big[\min\{x_{i_1},\ \cdots,\ x_{i_k}\} - 1\big] \Big\}$$

Figure 1: Inference rule for eliminating $r \in \mathsf{N_R}$ from an $\mathcal{ALCQ}(\nabla)$-ontology in r-NF

positive r-clause. Likewise, the mth set captures $(m-1)$th-tier implicit logical consequences, involving the interactions of all negative r-clauses with a positive r-clause. This tiered approach allows for increasingly complex levels of logical analysis as more tiers are considered. Each tier incrementally incorporates additional layers of reasoning that build upon the foundational results established in the preceding tiers.

Example 2. *Consider the ontology $\mathcal{O}'$ obtained from Example 1 as the input for Example 2:*

$$\{1.\ {\leq}1r.Z_2 \sqcup Z_1 \quad 2.\ \neg Z_2 \sqcup {\geq}2r.A_1 \quad 3.\ \neg Z_4 \sqcup {\leq}3r.A_2$$
$$4.\ \neg Z_1 \sqcup {\geq}2s.Z_3 \sqcup {\leq}0t.A_3 \quad 5.\ \neg Z_3 \sqcup {\leq}1t.Z_4\}$$

Applying the inference rule to resolve Clauses 1 with Clauses 2 and 3 on r yields:

$$\{6.\ \neg Z_2 \sqcup {\geq}2\nabla.A_1 \quad 7.\ Z_1 \sqcup \neg Z_2 \sqcup {\geq}1\nabla.(A_1 \sqcap \neg Z_2)$$
$$4.\ \neg Z_1 \sqcup {\geq}2s.Z_3 \sqcup {\leq}0t.A_3 \quad 5.\ \neg Z_3 \sqcup {\leq}1t.Z_4\}$$

The set $\mathcal{V} = \{4, 5, 6, 7\}$ represents the result of excluding r from $\mathcal{O}'$. However, this does not equate to the result of forgetting r from the original ontology $\mathcal{O}$, as $\mathcal{V}$ retains the definers introduced during the process. These definers, not intended for inclusion in the final forgetting output, necessitate further elimination from $\mathcal{V}$ to align with the desired result.

Lemma 3. *Let $\mathcal{O}'$ be an $\mathcal{ALCQ}(\nabla)$-ontology in r-NF, serving as the premises of the rule in Figure 1, and $\mathcal{V}$ be its conclusion. Then, for any interpretation $\mathcal{I}'$, the following holds:*

$$\mathcal{I}' \models \mathcal{O}' \text{ iff } \mathcal{I} \models \mathcal{V},$$

for some interpretation $\mathcal{I}$ such that $\mathcal{I} \sim_r \mathcal{I}'$.

Lemma 4. *Let x and y be the respective counts of r-clauses of the forms $C \sqcup {\geq}mr.D$ and $C \sqcup {\leq}nr.D$. Let z represent the maximum qualified number present in $C \sqcup {\geq}mr.D$ clauses. Then, the total count of the derived clauses resulting from these clauses is bounded by $O(y \cdot 2^x \cdot z^{2^x})$.*

Lemma 3 states soundness of the inference rule; Lemma 4 shows its termination and computational complexity.

6 The Forgetting Iteration

Our method takes as input an $\mathcal{ALCQ}$-ontology $\mathcal{O}$ and a forgetting signature $\mathcal{F}$ including the role names to be eliminated from $\mathcal{O}$. Notably, the elimination sequence is flexible, accommodating any order specified by the user. Initially, $\mathcal{O}$ undergoes clausification, with GCIs being converted into clauses. This step involves several standard simplifications such as removing redundancies, eliminating double negations, and reducing duplicate literals. The standard simplifications preserve logical equivalence between the input and output, ensuring that the simplified ontology faithfully represents the original. Such simplifications are integral to the entire forgetting process, eagerly applied to ensure that the ontology is in its simplest form at any stage.

Entering the core phase of forgetting, let us assume r is the name in $\mathcal{F}$ targeted for elimination. Our method starts by transforming $\mathcal{O}$ into r-NF $\mathcal{O}'$ using normalization rules NR1 $-$ NR4, and then applies the inference rule in Figure 1 to eliminate r from $\mathcal{O}'$. This procedure is iteratively applied to each name in $\mathcal{F}$, with each round yielding an intermediate result. Throughout this process, definers may be introduced. According to Definition 1, these definers should not be present in the final result of forgetting. Therefore, once all names in $\mathcal{F}$ have been addressed, any introduced definers must also be forgotten from the intermediate result.

The final phase of the forgetting process aims to eliminate the introduced definers. In most cases, this can be achieved using a generalized version of *Ackermann's Lemma* (Koopmann and Schmidt 2014). This lemma allows a single concept name to be eliminated from an ontology while ensuring the original and resulting ontologies are equivalent, except for the interpretations of the eliminated name (i.e., satisfying Conditions (i) and (ii) of Definition1). Soundness of this lemma is a straightforward adaptation of Ackermann's original result (Ackermann 1935).

Definition 3 (Ackermann's Lemma)**.** *Let $\mathcal{O}$ be an ontology containing the clauses $C_1 \sqcup A, ..., C_n \sqcup A$, where $A \in \mathsf{N_C}$, and the C_i ($1 \leq i \leq n$) are concepts not containing A. If the remaining part $\mathcal{O}\backslash\{C_1 \sqcup A, ..., C_n \sqcup A\}$ (i.e., $\mathcal{O}$ excluding these clauses) is negative w.r.t. A, then $\mathcal{O}^A_{\neg C_1 \sqcup \ldots \sqcup \neg C_n}$ is a* result of strongly forgetting A from $\mathcal{O}$. *Here, $\mathcal{O}^A_{\neg C_1 \sqcup \ldots \sqcup \neg C_n}$ denotes the ontology obtained by replacing every occurrence of A in $\mathcal{O}$ with the concept $\neg C_1 \sqcup \ldots \sqcup \neg C_n$. A dual version is as follows: let $\mathcal{O}$ be an ontology containing the clauses $C_1 \sqcup \neg A, ..., C_n \sqcup \neg A$, where $A \in \mathsf{N_C}$, and the C_i ($1 \leq i \leq n$) are concepts not containing A. If $\mathcal{O}\backslash\{C_1 \sqcup \neg A, ..., C_n \sqcup \neg A\}$ is positive w.r.t. A, then $\mathcal{O}^A_{C_1 \sqcup \ldots \sqcup C_n}$ is a* result of strongly forgetting A from $\mathcal{O}$.

This approach often successfully eliminates definers, typically because one polarity of a definer (usually the negative) appears at the surface level of a clause. However, there is no guarantee of completely removing all definers, even with the generalized Ackermann's Lemma that employs fixpoint operators (Zhao and Schmidt 2016). In practice, most real-world ontologies are already in normal form due to their flat structure, meaning that for such ontologies definer introduction and elimination are obsolete.

Theorem 1. *Given any $\mathcal{ALCQ}$-ontology $\mathcal{O}$ and any forgetting signature $\mathcal{F}$ of role names, our forgetting method always terminates and returns an $\mathcal{ALCQ}(\nabla)$-ontology $\mathcal{V}$.*

(i) *If $\mathcal{V}$ does not contain any definers, then it is a result of forgetting $\mathcal{F}$ from $\mathcal{O}$, i.e., for any interpretation $\mathcal{I}'$, $\mathcal{I}' \models \mathcal{V}$ iff there is an interpretation $\mathcal{I} \sim_{\mathcal{F}} \mathcal{I}'$ such that $\mathcal{I} \models \mathcal{O}$.*

(ii) *If $\mathcal{V}$ does contain definers, then for any interpretation $\mathcal{I}'$, $\mathcal{I}' \models \mathcal{V}$ iff there is an interpretation $\mathcal{I} \sim_{\mathcal{F} \cup \mathsf{sig}_D(\mathcal{V})} \mathcal{I}'$ such that $\mathcal{I} \models \mathcal{O}$.*

Theorem 1 states termination and soundness of our forgetting method, yet its completeness cannot be established due to the inherent undecidability of the problem (Konev et al. 2013). Although our method can reliably forget any role names from any $\mathcal{ALCQ}(\nabla)$-ontology, it does not guarantee the elimination of all introduced definers. A prominent example arises with cyclic dependencies among the names targeted for forgetting, as identified in (Konev, Walther, and Wolter 2009). In such scenarios, the inference process risks entering an endless loop, potentially causing the forgetting process to never terminate. Take an example where we aim to forget r from an $\mathcal{ALCQ}$-ontology $\mathcal{O} = \{{\geq}1r.{\leq}0r.\top\}$, which exhibits implicit cyclic behavior over r. Transforming $\mathcal{O}$ into r-NF $\mathcal{O}'$ introduces a definer Z, yielding $\mathcal{O}' = \{{\geq}1r.Z, \neg Z \sqcup {\leq}0r.\top\}$. Applying the inference rule to $\mathcal{O}'$ yields an intermediate result $\{{\geq}1\nabla.Z, \neg Z \sqcup {\geq}1\nabla.Z\}$ exhibiting explicit cyclic behavior over Z. Further eliminating the definer Z from this result produces an infinite set $\mathcal{V} = \{\overbrace{{\geq}1\nabla. \cdots {\geq}1\nabla.}^{n}\top \mid n \geq 1\}$. This result cannot be finitely axiomatized in DLs without additional expressivity. Lethe and Fame exploit fixpoints (Calvanese, Giacomo, and Lenzerini 1999) to find finite representation of the result. However, since fixpoints are not supported by mainstream DL systems, nor by the OWL API,[3] our method does not adopt the extension of fixpoints as a solution. Instead, it

[3]The universal role ∇ is supported by the OWL API.

terminates the forgetting process with Z remaining in the result, prioritizing method termination over the pursuit of completeness.

7 Empirical Evaluation

We have developed a prototype of our forgetting method in Java using OWL API Version 5.1.7[4]. To evaluate its practicality, we compared this prototype against two SOTA systems — LETHE[5] and FAME[6] — based on two large corpora of real-world ontologies. The first corpus, taken from the Oxford ISG Library[7], includes a wide range of ontologies from various sources. The second corpus, a March 2017 snapshot from NCBO BioPortal[8] (Matentzoglu and Parsia 2017), features specifically biomedical ontologies.

From the Oxford ISG snapshot, we initially cherry-picked 488 ontologies, each with a GCI count not exceeding 10,000. Upon excluding those lacking the $\geq$- or $\leq$-restrictions, we found that all 488 ontologies still qualified. For further refinement, we distilled these ontologies to their $\mathcal{ALCQ}$ fragments by excluding the GCIs not expressible in $\mathcal{ALCQ}$. This led to a 4.3% reduction in the average GCI count.

Table 1 sums up statistical information about the refined ontologies, where the notation $|\mathsf{N_C}|$, $|\mathsf{N_R}|$, and |Onto| represents the average counts of concept names, role names, and GCIs, respectively, within these ontologies.

Table 1: Statistical information about Oxford-ISG & BioPortal

Oxford ISG		**min**	**max**	**median**	**mean**	**upper decile**
I	$\lvert\mathsf{N_C}\rvert$	0	1582	86	191	545
	$\lvert\mathsf{N_R}\rvert$	0	332	10	29	80
	|Onto|	10	990	162	262	658
II	$\lvert\mathsf{N_C}\rvert$	200	5877	1665	1769	2801
	$\lvert\mathsf{N_R}\rvert$	0	887	11	34	61
	|Onto|	1008	4976	2282	2416	3937
III	$\lvert\mathsf{N_C}\rvert$	1162	9809	4042	5067	8758
	$\lvert\mathsf{N_R}\rvert$	1	158	4	23	158
	|Onto|	5112	9783	7277	7195	9179
BioPortal		**min**	**max**	**median**	**mean**	**upper decile**
I	$\lvert\mathsf{N_C}\rvert$	0	784	127	192	214
	$\lvert\mathsf{N_R}\rvert$	0	122	5	15	17
	|Onto|	10	794	283	312	346
II	$\lvert\mathsf{N_C}\rvert$	5	4530	1185	1459	1591
	$\lvert\mathsf{N_R}\rvert$	0	131	12	30	33
	|Onto|	1023	4880	2401	2619	2782
III	$\lvert\mathsf{N_C}\rvert$	432	8340	4363	4387	4806
	$\lvert\mathsf{N_R}\rvert$	0	135	17	30	34
	|Onto|	5457	8339	6934	6912	7109

To gain granular insights into our method's performance across variably sized Oxford-ISG ontologies, we partitioned these selections into three categories:

- Part I: 355 ontologies with $10 \leq |\text{Onto}| < 1000$
- Part II: 108 ontologies with $1000 \leq |\text{Onto}| < 4999$
- Part III: 25 ontologies with $5000 \leq |\text{Onto}| < 10000$

Implementing the same strategy, we assembled a corpus of 326 BioPortal ontologies and categorized them as follows:

- Part I: 202 ontologies with $10 \leq |\text{Onto}| < 1000$
- Part II: 104 ontologies with $1000 \leq |\text{Onto}| < 4999$
- Part III: 20 ontologies with $5000 \leq |\text{Onto}| < 10000$

The composition of the signature $\mathcal{F}$ targeted for forgetting varies according to the specific requirements of tasks or applications. To address this variability, we designed three evaluation configurations to forget 10%, 30%, and 50% of the role names in the signature of each test ontology. These configurations were consistent with common practices in evaluating forgetting approaches, as documented in literature such as (Zhao and Schmidt 2018a; Koopmann 2020; Wu et al. 2020; Yang et al. 2023). We utilized a shuffling algorithm to ensure randomized selection of $\mathcal{F}$. Our experiments were conducted on a laptop equipped with an Intel Core i7-9750H processor with 6 cores, capable of reaching up to 2.70 GHz, and 12 GB of DDR4-1600 MHz RAM. For consistent performance assessment, we set a maximum runtime of 300 seconds and a heap space limit of 9GB. An experiment was deemed *successful* if it met the following criteria: (i) all names specified in $\mathcal{F}$ were successfully eliminated; (ii) no definers were present in the output, if introduced during the process; (iii) completion within the 300-second limit; and (iv) operation within the set 9GB space limit. We repeated the experiments 100 times for each test case and took the average to validate our findings.

Table 2: Our prototype's results over Oxford-ISG

$\mathcal{F}$%	**Part**	**Time**	**Mem**	**SR**	**TR**	**MR**	**DR**	**DC**
10%	I	2.84	687.43	92.96	0.28	1.41	5.35	1.09
	II	4.10	891.37	85.19	0.93	3.70	10.19	4.36
	III	5.71	1046.50	80.00	0.00	16.00	4.00	6.57
	Avg	**3.23**	**746.13**	**90.57**	**0.41**	**2.66**	**6.35**	**2.09**
30%	I	4.19	974.70	89.86	0.56	3.94	5.63	3.41
	II	7.33	1248.24	73.15	3.70	13.89	9.26	12.49
	III	11.42	1576.18	64.00	8.00	20.00	8.00	16.31
	Avg	**5.07**	**1050.14**	**84.84**	**1.64**	**6.97**	**6.56**	**6.08**
50%	I	9.72	1662.82	84.79	1.69	5.63	7.89	5.71
	II	12.16	2040.73	50.93	4.63	18.52	25.93	22.47
	III	17.27	2516.79	44.00	12.00	28.00	16.00	34.67
	Avg	**10.31**	**1745.05**	**75.20**	**2.87**	**9.63**	**12.30**	**10.90**

The experiment results are shown in Tables 2 and 3. Notably, the SR (success rate) column stands out as the most critical data point. Across both Oxford-ISG and BioPortal, our prototype consistently achieved average success rates of approximately 90%, 85%, and 75% for the 10%, 30%, and 50% forgetting targets, respectively. We observed that the success rates decreased as the size of the forgetting signature $\mathcal{F}$ increased. Failures were due to three reasons: timeout (TR), memory overflow (MR), and the inability to eliminate all introduced definers (DR), all of which became more

[4]http://owlcs.github.io/owlapi/

[5]https://lat.inf.tu-dresden.de/ koopmann/LETHE/

[6]https://www.cs.man.ac.uk/ schmidt/sf-fame/

[7]http://krr-nas.cs.ox.ac.uk/ontologies/lib/

[8]https://bioportal.bioontology.org/

Table 3: Our prototype's results over BioPortal

$\mathcal{F}$%	Part	Time	Mem	SR	TR	MR	DR	DC
10%	I	1.67	553.79	98.02	0.00	0.50	1.49	0.34
	II	2.96	772.41	88.46	0.96	3.85	6.73	1.83
	III	4.80	920.54	50.00	10.00	25.00	15.00	24.50
	Avg	**2.17**	**633.06**	**92.02**	**0.92**	**3.07**	**3.99**	**1.60**
30%	I	2.98	843.31	93.07	0.00	1.98	4.95	1.39
	II	5.74	1076.45	84.62	0.96	4.81	9.62	4.05
	III	10.13	1473.66	50.00	10.00	25.00	15.00	28.00
	Avg	**4.08**	**937.09**	**87.73**	**0.92**	**4.29**	**7.06**	**3.14**
50%	I	6.17	1146.81	80.20	0.99	3.96	14.85	0.75
	II	8.68	1783.45	76.92	1.92	6.73	14.42	6.15
	III	14.53	2218.46	40.00	10.00	25.00	25.00	37.50
	Avg	**7.24**	**1384.83**	**76.69**	**1.84**	**6.13**	**15.34**	**3.65**

Table 4: Lethe's results over Oxford-ISG

$\mathcal{F}$%	Part	Time	Mem	SR	TR	MR	DR	DC
10%	I	9.70	990.3	89.29	3.10	2.82	4.79	35.96
	II	10.20	1317.4	81.48	2.78	6.48	9.26	48.57
	III	48.10	1966.1	60.00	4.00	32.00	4.00	66.36
	Avg	**11.77**	**1112.68**	**86.06**	**3.08**	**5.12**	**5.74**	**40.31**
30%	I	21.70	1304.01	84.51	4.79	5.35	5.35	135.41
	II	41.40	1799.34	67.59	4.63	20.37	7.41	152.67
	III	69.70	2003.47	48.00	8.00	36.00	8.00	170.36
	Avg	**28.52**	**1449.47**	**78.90**	**4.92**	**10.24**	**5.94**	**141.02**
50%	I	35.57	2367.98	76.61	7.61	8.17	7.61	259.30
	II	76.53	2904.35	49.08	8.33	23.15	19.44	273.13
	III	137.07	3511.66	32.00	12.00	44.00	12.00	270.20
	Avg	**49.83**	**2545.28**	**68.85**	**7.99**	**13.32**	**9.84**	**262.92**

Table 5: Fame's results over Oxford-ISG

$\mathcal{F}$%	Part	Time	Mem	SR	TR	MR	DR	DC
10%	I	5.85	749.55	91.83	1.97	1.41	4.79	9.57
	II	9.23	980.10	82.41	3.70	4.63	9.26	13.19
	III	23.15	1621.51	60.00	4.00	28.00	4.00	32.11
	Avg	**5.99**	**876.59**	**88.32**	**2.46**	**3.48**	**5.74**	**11.53**
30%	I	10.44	916.45	86.76	3.10	4.79	5.35	25.56
	II	21.78	1235.13	70.37	3.70	18.52	7.41	39.57
	III	48.56	1999.58	48.00	12.00	32.00	8.00	43.66
	Avg	**11.37**	**1084.56**	**81.22**	**3.62**	**9.22**	**5.94**	**29.08**
50%	I	21.46	1316.43	79.15	6.20	7.04	7.61	58.22
	II	33.75	1934.25	54.63	7.41	18.52	19.44	79.46
	III	53.93	2522.57	44.00	8.00	36.00	12.00	84.32
	Avg	**23.37**	**1587.52**	**72.54**	**6.56**	**11.06**	**9.84**	**54.77**

frequent with larger $\mathcal{F}$. The definer retention rate (DR) indicates the proportion of definers that remained in the ontology after the forgetting process had been applied. Our observations revealed that timeouts and memory overflows were primarily triggered by a recurrent pattern: the excessive presence of certain role names under $\geq$-restrictions, leading to exponential growth in the number of resultant clauses. For example, in the SDO ontology from BioPortal, the "hasPart" role appeared in more than 50 $\geq$-restrictions. Selecting this role for forgetting would result in substantial memory demand, generating upwards of 2^{50} clauses during the forgetting process.

Establishing benchmarks is essential to contextualize our method's success rate, time consumption (sec.), and memory usage (MB). Therefore, we additionally conducted comparative experiments with Lethe and Fame, although both performed weak forgetting. While Lethe features an approach to forgetting role names in DLs with $\mathcal{Q}$, it remains unimplemented, as confirmed by Lethe's official page. This prevented direct comparisons using our specifically crafted datasets. As a workaround, we conducted these comparisons on the $\mathcal{ALC}$-fragments of the test ontologies. Our primary objective was to compare time consumption (the Time column) and memory usage (the Mem column), with success rate comparisons being a secondary focus, though our method consistently showcased better success rates. Thus, using the $\mathcal{ALC}$-ontologies as test data did not detract from the precision and value of the comparisons. However, due to the page limit, we only present the results of two methods on the Oxford-ISG dataset here; the results for the BioPortal dataset can be found in the long version of this paper.

Although our prototype was tested on $\mathcal{ALCQ}$ ontologies and Lethe and Fame on their $\mathcal{ALC}$ fragments, our method consistently achieved a success rate 5% to 10% higher than the two weak forgetting tools. The DR column indicates the number of ontologies containing cycles. Despite our test dataset having more cycles, this highlights the inherent superiority of our method in terms of time and space savings. This computational efficiency is closely linked to the strategy used for introducing definers. The DC column presents the average number of definers introduced during the forgetting process. This number was significantly lower — by an order of magnitude — compared to the rest. Our analysis reveals that Lethe and Fame tended to add definers exponentially, in direct proportion to the number of role restrictions in the ontology. In contrast, our method introduced definers linearly, demonstrating a more efficient approach. This is detailed statistically in the long version of this paper.

8 Conclusion and Future Work

This paper has introduced the first strong forgetting method capable of handling role names in DLs with qualified number restrictions. Demonstrating both soundness and termination, our method's incompleteness — due to the inherent unsolvability of the problem — does not markedly impact its success rates in real-world scenarios. A distinctive feature of our approach is the linear strategy for introducing definers, which significantly boosts computational efficiency and distinguishes our method from others in the field.

We believe that this method provides a useful tool to the DL and KR communities because it computes results using a stronger definition of forgetting. The ability to compute strong forgetting results means that our approach can be applied to any fragments of the $\mathcal{ALCQ}$ syntax, ensuring consistent results irrespective of source language variations. This versatility is particularly beneficial in problems such as modal correspondence theory (Szalas 1993; Szalas 1993) and second-order quantifier elimination (Schmidt 2012; Gabbay, Schmidt, and Szalas 2008). Since strong forgetting yields more informative results, this method's application in ontology engineering, particularly in the abduction task, offers results richer in information than those obtained using weak forgetting procedures like (Del-Pinto and Schmidt 2019; Koopmann et al. 2020; Koopmann 2021).

This makes it more effective in identifying true causes or explanations in abduction tasks.

Moving forward, our immediate objective is to refine our method to seamlessly integrate ABoxes. In addition, we aim to adapt our method to accommodate more expressive DLs, further expanding its scope and applicability.

A Missing Proofs

To prove Lemma 1, we first define some key notions. Consider X as a syntactic object including concepts, roles, and clauses. Given a role name $r \in \mathsf{N_R}$, the *frequency* $\mathsf{fq}(r, X)$ of r in X is defined inductively as follows:

- $\mathsf{fq}(r, s) = \begin{cases} 1, & \text{if } s = r, \\ 0, & \text{if otherwise;} \end{cases}$
- $\mathsf{fq}(r, A) = 0$, for any $A \in \mathsf{N_C} \cup \{\top\}$;
- $\mathsf{fq}(r, \neg D) = \mathsf{fq}(r, D)$,
- $\mathsf{fq}(r, \geq ms.D) = \mathsf{fq}(r, s) + \mathsf{fq}(r, D)$;
- $\mathsf{fq}(r, \leq ns.D) = \mathsf{fq}(r, s) + \mathsf{fq}(r, D)$;
- $\mathsf{fq}(r, E * F) = \mathsf{fq}(r, E) + \mathsf{fq}(r, F)$, for $* \in \{\sqcap, \sqcup\}$,

and its *role depth* in X, denoted as $\mathsf{rd}(r, X)$, as follows:

- $\mathsf{rd}(r, s) = 0$, for any $s \in \mathsf{N_R} \cup \{\nabla\}$;
- $\mathsf{rd}(r, A) = 0$, for any $A \in \mathsf{N_C} \cup \{\top\}$;
- $\mathsf{rd}(r, \neg D) = \mathsf{rd}(r, D)$,
- $\mathsf{rd}(r, \geq ms.D) = 1 + \mathsf{rd}(r, D)$;
- $\mathsf{rd}(r, \leq ns.D) = 1 + \mathsf{rd}(r, D)$;
- $\mathsf{rd}(r, E * F) = max(\mathsf{rd}(r, E), \mathsf{rd}(r, F))$, for $* \in \{\sqcap, \sqcup\}$.

Proposition 1. *A clause X is in r-NF iff $\mathsf{fq}(r, X) = 1$ and $\mathsf{rd}(r, X) = 1$.*

Lemma 1. *Let $\mathcal{O}$ be an $\mathcal{ALCQ}(\nabla)$-ontology. Then $\mathcal{O}$ can be transformed into r-NF $\mathcal{O}'$ by a linear number of applications of the above normalization rules NR1 $-$ NR4. In addition, the size of the resulting ontology $\mathcal{O}'$ is linear in the size of $\mathcal{O}$.*

Proof. Proposition 1 specifies the conditions for an r-clause to be in r-NF. For any r-clause X, applying one of the normalization rules NR1 $-$ NR4 to X either reduces $\mathsf{fq}(r, X)$ by at least 1 or reduces $\mathsf{rd}(r, X)$ by at least 1. This ensures that the r-NF of X can be computed in finite steps. Each introduction of a definer corresponds to the addition of a new clause, which is associated with a role restriction. Consequently, the upper limit on the number of definers and new clauses introduced is bounded by $O(n)$, where n is the number of role restrictions in the given ontology $\mathcal{O}$. □

Lemma 2. *Let $\mathcal{O}$ be an $\mathcal{ALCQ}(\nabla)$-ontology, and $\mathcal{O}'$ the normalized one obtained from $\mathcal{O}$ using the normalization rules NR1 $-$ NR4. Let $\mathsf{sig}_{\mathsf{D}}(\mathcal{O}')$ denote the set of definers introduced in $\mathcal{O}'$. Then, for any interpretation $\mathcal{I}'$, the following holds:*

$$\mathcal{I}' \models \mathcal{O}' \text{ iff } \mathcal{I} \models \mathcal{O},$$

for some interpretation $\mathcal{I}$ such that $\mathcal{I} \sim_{\mathsf{sig}_{\mathsf{D}}(\mathcal{O}')} \mathcal{I}'$.

Proof. Observe that the application of any of the normalization rules results in the introduction of a single fresh definer $Z \in \mathsf{N_C}$. We show that for any new ontology $\mathcal{O}'$ obtained from the ontology $\mathcal{O}$ by applying any of the normalization rules, the following holds: for any interpretation $\mathcal{I}'$,

$$\mathcal{I}' \models \mathcal{O}' \text{ iff } \mathcal{I} \models \mathcal{O},$$

for some interpretation $\mathcal{I}$ such that $\mathcal{I} \sim_Z \mathcal{I}'$. We treat NR1 in detail. NR2, NR3 and NR4 can be proved similarly.

In examining NR1, let us consider that $\mathcal{O}'$ is derived from $\mathcal{O}$ by substituting the clause $C \sqcup \geq mr.D$ (with $r \in \mathsf{sig}(C)$) with two new clauses: $Z \sqcup \geq mr.D$ and $\neg Z \sqcup C$, where $Z \in \mathsf{N_C}$ is a fresh definer. Evidently, $\mathsf{sig}(\mathcal{O}')$ extends $\mathsf{sig}(\mathcal{O})$ by including Z.

Consider $\mathcal{I}'$ as a model of $\mathcal{O}'$, then $Z^{\mathcal{I}'} \cup (\geq mr.D)^{\mathcal{I}'}$ and $(\neg Z)^{\mathcal{I}'} \cup C^{\mathcal{I}'}$ hold.[9] This suggests that $C^{\mathcal{I}'} \cup (\geq mr.D)^{\mathcal{I}'}$ holds. Since $\mathcal{I} \sim_Z \mathcal{I}'$ and $Z \notin \mathsf{sig}(C \cup \geq mr.D)$, we have $C^{\mathcal{I}} \cup (\geq mr.D)^{\mathcal{I}}$, confirming that $\mathcal{I}$ is a model of $\mathcal{O}$.

Conversely, if $\mathcal{I}$ is a model of $\mathcal{O}$, let $\mathcal{I}'$ be an interpretation identical to $\mathcal{I}$ on all names except for Z, where $Z^{\mathcal{I}'}$ is defined as $C^{\mathcal{I}}$. Since $\mathcal{I}$ is a model of $\mathcal{O}$, $C^{\mathcal{I}} \cup (\geq mr.D)^{\mathcal{I}}$ holds. Since $Z \notin \mathsf{sig}(C \sqcup \geq mr.D)$, we have $C^{\mathcal{I}} = C^{\mathcal{I}'}$ and $(\geq mr.D)^{\mathcal{I}} = (\geq mr.D)^{\mathcal{I}'}$. This further yields $Z^{\mathcal{I}'} = C^{\mathcal{I}'}$, which leads to the satisfaction of $Z^{\mathcal{I}'} \cup (\geq mr.D)^{\mathcal{I}'}$ and $(\neg Z)^{\mathcal{I}'} \cup C^{\mathcal{I}'}$. Therefore, $\mathcal{I}'$ is as a model of $\mathcal{O}'$. The proof for the clause $C \sqcup \leq nr.D$ follows a similar reasoning. □

Lemma 3. *Let $\mathcal{O}'$ be an $\mathcal{ALCQ}(\nabla)$-ontology in r-NF, serving as the premises of the rule in Figure 1, and $\mathcal{V}$ be its conclusion. Then, for any interpretation $\mathcal{I}'$, the following holds:*

$$\mathcal{I}' \models \mathcal{O}' \text{ iff } \mathcal{I} \models \mathcal{V},$$

for some interpretation $\mathcal{I}$ such that $\mathcal{I} \sim_r \mathcal{I}'$.

Proof. We denote all r-clauses in the premises of this rule as $\mathcal{P}$ and the conclusion, excluding $\mathcal{O}^{-r}$, as $\mathcal{C}$. Lemma 3 states r-equivalence of $\mathcal{O}'$ and $\mathcal{V}$. Given that $\mathcal{O}^{-r}$ is not involved in the inference process, proving r-equivalence of $\mathcal{P}$ and $\mathcal{C}$ would suffice, that is, for any interpretation $\mathcal{I}'$,

$$\mathcal{I}' \models \mathcal{P} \text{ iff } \mathcal{I} \models \mathcal{C},$$

for some interpretation $\mathcal{I}$ such that $\mathcal{I} \sim_r \mathcal{I}'$.

The "only if" direction: Given that $\mathcal{I} \sim_r \mathcal{I}'$, it suffices to show that if any model $\mathcal{I}' \models \mathcal{P}$, then $\mathcal{I}' \models \mathcal{C}$. This ensures that $\mathcal{I} \models \mathcal{C}$ as well, as $r \notin \mathsf{sig}(\mathcal{C})$.
We first examine the block $\textbf{Block}(\mathcal{P}^{-}(r), C_i \sqcup \geq x_i r.D_i)$.
Clearly, $C_i \sqcup \geq x_i r.D_i \models C_i \sqcup \geq x_i \nabla.D_i$ (0th-tier), then we have $\mathcal{I}' \models C_i \sqcup \geq x_i \nabla.D_i$.
Consider the case $C_i \sqcup E_{j_1} \sqcup \cdots \sqcup E_{j_k} \sqcup \geq (x_i - y_{j_1} - \cdots - y_{j_k})\nabla.(D_i \sqcap \neg F_{j_1} \sqcap \cdots \neg F_{j_k})$ (1st-tier $-$ nth-tier). Suppose there exists a domain element $a \in \Delta^{\mathcal{I}'}$ such that

[9] Note that we define a clause as a concept inclusion in the form $\top \sqsubseteq L_1 \sqcup \ldots \sqcup L_n$, where each L_i (for $1 \leq i \leq n$) is a literal. Typically, we omit the prefix "$\top \sqsubseteq$" and treat clauses as sets, meaning that they contain no duplicates and their order is not important. Therefore, when stating that $Z^{\mathcal{I}'} \cup (\geq mr.D)^{\mathcal{I}'}$ holds (is true), we imply that $\top \sqsubseteq Z \sqcup (\geq mr.D)$ holds (is true) in $\mathcal{I}'$.

$a \notin C_i^{\mathcal{I}'}, a \notin E_{j_1}^{\mathcal{I}'}, \cdots, a \notin E_{j_k}^{\mathcal{I}'}$, then we have $a \in (\geq x_i r.D_i)^{\mathcal{I}'}, a \in (\leq y_{j_1} r.F_{j_1})^{\mathcal{I}'}, \cdots, a \in (\leq y_{j_k} r.F_{j_k})^{\mathcal{I}'}$. If the number of r-successors of a in $(D_i \sqcap \neg F_{j_1} \sqcap \cdots \neg F_{j_k})^{\mathcal{I}'}$ is less than $x_i - y_{j_1} - \cdots - y_{j_k}$, then there must be at least $y_{j_1} + \cdots + y_{j_k} + 1$ r-successors of a in $(F_{j_1} \sqcup \cdots \sqcup F_{j_k})^{\mathcal{I}'}$, which contradicts $a \in (\leq y_{j_1} r.F_{j_1})^{\mathcal{I}'}, \cdots, a \in (\leq y_{j_k} r.F_{j_k})^{\mathcal{I}'}$. Therefore we have $\mathcal{I}' \models C_i \sqcup E_{j_1} \sqcup \cdots \sqcup E_{j_k} \sqcup \geq (x_i - y_{j_1} - \cdots - y_{j_k}) \nabla.(D_i \sqcap \neg F_{j_1} \sqcap \cdots \neg F_{j_k})$

We then examine the block $\textbf{Block}(\mathcal{P}^+(r), E_j \sqcup \leq y_j r.F_j)$. For convenience, we refer to this block as $\mathcal{C}'$

For any element $a \in \Delta^{\mathcal{I}'}$, let us examine its relation to $E_j^{\mathcal{I}'}$:

- If $a \in E_j^{\mathcal{I}'}$, then for any clause $\alpha \in \mathcal{C}'$, we have $a \in \alpha^{\mathcal{I}'}$;
- If $a \notin E_j^{\mathcal{I}'}$, then suppose that $a \notin C_1^{\mathcal{I}'}, \cdots, a \notin C_k^{\mathcal{I}'}$ and $a \in C_{k+1}^{\mathcal{I}'}, \cdots, a \in C_m^{\mathcal{I}'}$. For any clause $\alpha \in \mathcal{C}'$, if α contains C_t, for $t > k$, then $a \in \alpha^{\mathcal{I}'}$. Thus, our focus narrows to those clauses in $\mathcal{C}'$ that exclude C_t. In a more generalized context, consider a clause β containing $C_1, \cdots, C_k$. For a literal $l_{\geq}$ of the form $\geq Z \nabla.D$, if $a \notin l_{\geq}^{\mathcal{I}'}$, it indicates $Z > |D^{\mathcal{I}'}|$; similarly, for a literal $l_{\leq}$ of the form $\leq Z \nabla.D$, if $a \notin l_{\leq}^{\mathcal{I}'}$, it indicates $Z < |D^{\mathcal{I}'}|$. Considering the extreme case where, for all literals $l_{\geq}$ and $l_{\leq}$ in β, $a \notin l_{\geq}^{\mathcal{I}'}$ and $a \notin l_{\leq}^{\mathcal{I}'}$, the requirement for $a \in \beta^{\mathcal{I}'}$ to hold requires that:

$$a \in \Big(\geq \big(\sum_{i=1}^{k} x_i - \sum_{1 \leq j_1 < j_2 \leq k} (Z_{j_1,j_2} - 1) + \sum_{1 \leq j_1 < j_2 < j_3 \leq k} (Z_{j_1,j_2,j_3} + 1) + \cdots + (-1)^{k+1} Z_{1,\cdots,k} + 1 - y_j\big) \nabla.((D_{i_1} \sqcup \cdots \sqcup D_{i_k}) \sqcap \neg F_j)\Big)^{\mathcal{I}'}$$

When examining the above literal, which involves the $\geq$-restrictions, it suffices to demonstrate that a satisfies the belonging condition when the cardinality of $\geq$ takes its maximum value. This is because $Z_{j1,j2} > |(D_{j1} \sqcap D_{j2})^{\mathcal{I}'}|$ and $Z_{j1,j2,j3} < |(D_{j1} \sqcap D_{j2} \sqcap D_{j3})^{\mathcal{I}'}|$. Hence, the upper limit S_k for this cardinality is:

$$S_k = \sum_{i=1}^{k} x_i - \sum_{1 \leq j1 < j2 \leq k} |(D_{j1} \sqcap D_{j2})^{\mathcal{I}'}| + \sum_{1 \leq j_1 < j_2 < j_3 \leq k} |(D_{j1} \sqcap D_{j2} \sqcap D_{j3})^{\mathcal{I}'}| + \cdots + (-1)^{k+1} |(D_1 \sqcap \cdots \sqcap D_k)^{\mathcal{I}'}| - y_j$$

Building on our initial assumption, where a is a domain element in $(\geq x_i \mathsf{r}.D_i)^{\mathcal{I}'}$, it follows that a has at least x_i r-successors in each $D_i^{\mathcal{I}'}$ for $1 \geq i \geq k$. To understand how S_k is derived, let us consider the r-successors in the combined set $D_1^{\mathcal{I}'} \cup D_2^{\mathcal{I}'}$. While $D_1^{\mathcal{I}'}$ and $D_2^{\mathcal{I}'}$ each have at x_1 and x_2 r-successors, respectively, the union $(D_1 \sqcup D_2)^{\mathcal{I}'}$ may not have exactly $x_1 + x_2$ r-successors due to overlaps. In fact, by the inclusion-exclusion principle, $(D_1 \sqcup D_2)^{\mathcal{I}'}$ contains at least $x_1 + x_2 - |(D_1 \sqcap D_2)^{\mathcal{I}'}|$ r-successors. Extending this idea, $(D_1 \sqcup D_2 \sqcup D_3)^{\mathcal{I}'}$ has at least $x_1 + x_2 + x_3 - |(D_1 \sqcap D_2)^{\mathcal{I}'}| - |(D_1 \sqcap D_3)^{\mathcal{I}'}| - |(D_2 \sqcap D_3)^{\mathcal{I}'}| + |(D_1 \sqcap D_2 \sqcap D_3)^{\mathcal{I}'}|$ r-successors. Thus, $S_k + y_j$ denotes the minimum count of r-successors in $(D_1 \sqcup \cdots \sqcup D_k)^{\mathcal{I}'}$.

To prove that $a \in (\geq S_k \nabla.((D_1 \sqcup \cdots \sqcup D_k) \sqcap \neg F_j))^{\mathcal{I}'}$, we use a proof by contradiction. Suppose $|((D_1 \sqcup \cdots \sqcup D_k) \sqcap \neg F_j)^{\mathcal{I}'}| < S_k$. This implies that $((D_1 \sqcup \cdots \sqcup D_k) \sqcap F_j)^{\mathcal{I}'}$ contains at least $y_j + 1$ r-successors, contradicting $a \in (\leq y_j \mathsf{r}.F_j)^{\mathcal{I}'}$. Hence, $|((D_1 \sqcup \cdots \sqcup D_k) \sqcap \neg F_j)^{\mathcal{I}'}| \geq S_k$, and $a \in (\geq S_k \nabla.((D_1 \sqcup \cdots \sqcup D_k) \sqcap \neg F_j))^{\mathcal{I}'}$.

The "if" direction: For any model $\mathcal{I}$ of $\mathcal{C}$, we can always extend this interpretation w.r.t. the role name r to construct a new interpretation $\mathcal{I}'$ such that $\mathcal{I}' \models \mathcal{P}$. $\mathcal{I}$ and $\mathcal{I}'$ have the same domain, i.e., $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$, and differ only possibly in how they interpret r, while their interpretations of all other names remain identical. For an element $a \in \Delta^{\mathcal{I}'}$, the construction of $r^{\mathcal{I}'}$ depends on whether a is an element of $E_j^{\mathcal{I}'}$:

- If $a \notin E_j^{\mathcal{I}'}$ and suppose that $a \notin C_1^{\mathcal{I}'}, \cdots, a \notin C_k^{\mathcal{I}'}$ and $a \in C_{k+1}^{\mathcal{I}'}, \cdots, a \in C_m^{\mathcal{I}'}$, it is clear that $a \in (C_{k+1} \sqcup \geq x_{k+1} \mathsf{r}.D_{k+1})^{\mathcal{I}'}, \cdots, a \in (C_m \sqcup \geq x_m \mathsf{r}.D_m)^{\mathcal{I}'}$. Given that $\mathcal{I} \models \mathcal{C}$, this implies $a \in (\geq x_1 \nabla.D_1)^{\mathcal{I}'}, \cdots, a \in (\geq x_k \nabla.D_k)^{\mathcal{I}'}$, which further implies that for any $1 \leq i \leq k$, $|D_i^{\mathcal{I}'}| \geq x_i$. Our goal is to designate x_i elements from each $D_i^{\mathcal{I}'}$ $(1 \leq i \leq k)$ as r-successors. This ensures that no more than y_i among these belong to $F_j^{\mathcal{I}'}$. Since $|D_i^{\mathcal{I}'}| \geq x_i$, the first part of this goal is fulfilled. The following discusses how to achieve the second part which concerns the restriction on elements belonging to $F_j^{\mathcal{I}'}$.

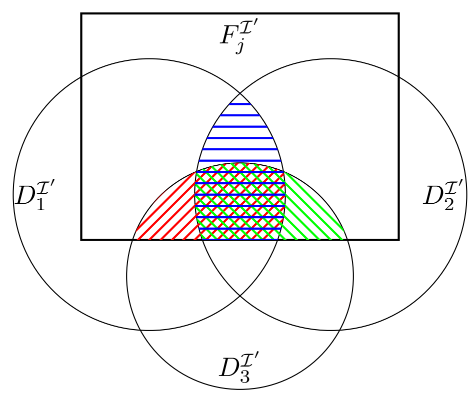

Figure 2: Relations among $D_1^{\mathcal{I}'}$, $D_2^{\mathcal{I}'}$, $D_3^{\mathcal{I}'}$, and $F_j^{\mathcal{I}'}$

To provide clarity and enhance comprehension, we exemplify our proof strategy with the specific case where $k = 3$. In this case, the relationships among $D_1^{\mathcal{I}'}$, $D_2^{\mathcal{I}'}$, $D_3^{\mathcal{I}'}$, and $F_j^{\mathcal{I}'}$ can be visually conceptualized by a Venn diagram, as depicted in Figure 2. Here, the rectangular area at the top represents $F_j^{\mathcal{I}'}$, while three circles positioned on the left, right, and bottom represent $D_1^{\mathcal{I}'}$, $D_2^{\mathcal{I}'}$, and $D_3^{\mathcal{I}'}$, respectively. Recall the meaning of S_k; specifically, when $k = 3$, S_3 is defined as:

$$S_3 = x_1 + x_2 + x_3 - |D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'}| - |D_1^{\mathcal{I}'} \cap D_3^{\mathcal{I}'}| - \\ |D_2^{\mathcal{I}'} \cap D_3^{\mathcal{I}'}| + |D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'} \cap D_3^{\mathcal{I}'}| - y_j$$

Building on the analysis from the preceding section of the proof, we deduce that $a \in (\geq S_3 \nabla.((D_1 \sqcup D_2 \sqcup D_3) \sqcap \neg F))^{\mathcal{I}'}$, indicating that $|((D_1 \sqcup D_2 \sqcup D_3) \sqcap \neg F)^{\mathcal{I}'}| \geq S_3$. Applying the inclusion-exclusion principle, we can derive the following conclusion:

$$\begin{aligned} &|((D_1 \sqcup D_2 \sqcup D_3) \sqcap \neg F)^{\mathcal{I}'}| \\ =&|D_1^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}| + |D_2^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}| + |D_3^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}| \\ &- |D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}| - |D_1^{\mathcal{I}'} \cap D_3^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}| \\ &- |D_2^{\mathcal{I}'} \cap D_3^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}| \\ &+ |D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'} \cap D_3^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}| \geq S_3 \end{aligned} \tag{1}$$

In Figure 2, the region marked with horizontal lines represents the intersection $D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'} \cap F_j^{\mathcal{I}'}$, while the regions with left and right diagonal lines represent $D_1^{\mathcal{I}'} \cap D_3^{\mathcal{I}'} \cap F_j^{\mathcal{I}'}$ and $D_2^{\mathcal{I}'} \cap D_3^{\mathcal{I}'} \cap F_j^{\mathcal{I}'}$, respectively. Finally, the region where all three line styles converge represents $D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'} \cap D_3^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}$.

In selecting elements from $D_i^{\mathcal{I}'}$, our first step is to choose all elements from $(D_i \sqcap \neg F_j)^{\mathcal{I}'}$. Consequently, we need to select only $x_i - |(D_i \sqcap \neg F_j)^{\mathcal{I}'}|$ elements from $(D_i \sqcap F_j)^{\mathcal{I}'}$. The total count of elements to be chosen from $F_j^{\mathcal{I}'}$, not accounting for other conditions, is $x_1 - |(D_1 \sqcap \neg F_j)^{\mathcal{I}'}| + x_2 - |(D_2 \sqcap \neg F_j)^{\mathcal{I}'}| + x_3 - |(D_3 \sqcap \neg F_j)^{\mathcal{I}'}|$, which we denote as M. As depicted in Figure 2, overlaps may exist between $(D_1 \sqcap F_j)^{\mathcal{I}'}$, $(D_2 \sqcap F_j)^{\mathcal{I}'}$, and $(D_3 \sqcap F_j)^{\mathcal{I}'}$. Therefore, in selecting elements from $(D_1 \sqcap F_j)^{\mathcal{I}'}$, prioritizing the overlapping regions can reduce the total number of needed elements. Elements in these overlapping regions need to be selected only once, necessitating a deduction of the duplicated portions from M. By the inclusion-exclusion principle, the adjusted count of elements chosen from $F_j^{\mathcal{I}'}$ is $M - |(D_1 \sqcap D_2 \sqcap F)^{\mathcal{I}'}| - |(D_1 \sqcap D_3 \sqcap F)^{\mathcal{I}'}| - |(D_2 \sqcap D_3 \sqcap F)^{\mathcal{I}'}| + |(D_1 \sqcap D_2 \sqcap D_3 \sqcap F)^{\mathcal{I}'}|$. Rewriting Equation 1, we obtain the following formula:

$$\begin{aligned} &x_1 - |(D_1 \sqcap \neg F_j)^{\mathcal{I}'}| + x_2 - |(D_2 \sqcap \neg F_j)^{\mathcal{I}'}| + x_3 \\ &- |(D_3 \sqcap \neg F_j)^{\mathcal{I}'}| - |D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'}| - |D_1^{\mathcal{I}'} \cap D_3^{\mathcal{I}'}| \\ &- |D_2^{\mathcal{I}'} \cap D_3^{\mathcal{I}'}| + |D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'} \cap D_3^{\mathcal{I}'}| + |D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}| \\ &+ |D_1^{\mathcal{I}'} \cap D_3^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}| + |D_2^{\mathcal{I}'} \cap D_3^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}| \\ &- |D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'} \cap D_3^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}| \leq y_j \end{aligned} \tag{2}$$

Upon examining Figure 1, we notice that the intersection $D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'}$ comprise two distinct segments. The first segment falls within $F_j^{\mathcal{I}'}$, represented as $D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'} \cap F_j^{\mathcal{I}'}$, while the second falls outside $F_j^{\mathcal{I}'}$, represented as $D_1^{\mathcal{I}'} \cap D_2^{\mathcal{I}'} \cap \neg F_j^{\mathcal{I}'}$. Consequently, Equation 2 can be further expressed as follows:

$$\begin{aligned} &x_1 - |(D_1 \sqcap \neg F_j)^{\mathcal{I}'}| + x_2 - |(D_2 \sqcap \neg F_j)^{\mathcal{I}'}| \\ &+ x_3 - |(D_3 \sqcap \neg F_j)^{\mathcal{I}'}| - |(D_1 \sqcap D_2 \sqcap F)^{\mathcal{I}'}| \\ &- |(D_1 \sqcap D_3 \sqcap F)^{\mathcal{I}'}| - |(D_2 \sqcap D_3 \sqcap F)^{\mathcal{I}'}| \\ &+ |(D_1 \sqcap D_2 \sqcap D_3 \sqcap F)^{\mathcal{I}'}| \leq y_j \end{aligned} \tag{3}$$

In Equation 3, the left-hand side quantifies the number of elements required to be chosen from $F_j^{\mathcal{I}'}$. The analysis, illustrated with $k = 3$ as an example, establishes that selecting no more than y_j elements from $F_j^{\mathcal{I}'}$ suffices to fulfill the condition $a \in (\geq x_1 \mathsf{r}.D_1)^{\mathcal{I}'}, \cdots, a \in (\geq x_m \mathsf{r}.D_m)^{\mathcal{I}'}$. This finding is applicable to scenarios where k assumes any value. For any k, we deduce that $|((D_1 \sqcup \cdots \sqcup D_k) \sqcap \neg F)^{\mathcal{I}'}| \geq S_k$. Applying the inclusion-exclusion principle, we can conclude the following:

$$\begin{aligned} &|((D_1 \sqcup \cdots \sqcup D_k) \sqcap \neg F)^{\mathcal{I}'}| \\ &= \sum_{1 \leq i \leq k} |(D_i \sqcap \neg F_j)^{\mathcal{I}'}| \\ &- \sum_{1 \leq i_1 < i_2 \leq k} |(D_{i_1} \sqcap D_{i_2} \sqcap \neg F_j)^{\mathcal{I}'}| + \\ &\sum_{1 \leq i_1 < i_2 < i_3 \leq k} |(D_{i_1} \sqcap D_{i_2} \sqcap D_{i_3} \sqcap F_j)^{\mathcal{I}'}| + \cdots \\ &+ (-1)^{k+1} |(D_1 \sqcap D_2 \sqcap \cdots \sqcap D_k \sqcap \neg F_j)^{\mathcal{I}'}| \end{aligned} \tag{4}$$

Disregarding other conditions, the total number of elements required to be selected from $F_j^{\mathcal{I}'}$ is given by:

$$M = \sum_{1 \leq i \leq k} (x_i - |(D_i \sqcap \neg F_j)^{\mathcal{I}'}|).$$

but when accounting for potential overlaps among the sets, the inclusion-exclusion principle modifies this total. Hence, the actual number of elements selected from $F_j^{\mathcal{I}'}$ is determined as follows:

$$\begin{aligned} &M - \sum_{1 \leq i_1 < i_2 \leq k} |(D_{i_1} \sqcap D_{i_2} \sqcap F_j)^{\mathcal{I}'}| \\ &+ \sum_{1 \leq i_1 < i_2 < i_3 \leq k} |(D_{i_1} \sqcap D_{i_2} \sqcap D_{i_3} \sqcap F_j)^{\mathcal{I}'}| \\ &+ \cdots + (-1)^{k+1} |(D_1 \sqcap D_2 \sqcap \cdots \sqcap D_k \sqcap F_j)^{\mathcal{I}'}| \end{aligned} \tag{5}$$

Considering that Eq.4 is greater than or equal to S_k, we deduce that Eq. 5 is less than or equal to y_j. This implies that the number of elements selected from $F_j^{\mathcal{I}'}$ in constructing the interpretation of r does not exceed y_j. Therefore, the condition $a \in (\leq y_j r.F_j)^{\mathcal{I}'}$ always hold.

- If $a \in E_j^{\mathcal{I}'}$, it follows that $a \in (E_j \sqcup {\leq} y_j \mathsf{r}.F_j)^{\mathcal{I}'}$. Consider any C_i, either $a \in C_i^{\mathcal{I}'}$ or $a \notin C_i^{\mathcal{I}'}$. Assuming $a \notin C_1^{\mathcal{I}'}, \cdots, a \notin C_k^{\mathcal{I}'}$ but $a \in C_{k+1}^{\mathcal{I}'}, \cdots, a \in C_m^{\mathcal{I}'}$, we can deduce that $a \in (C_{k+1} \sqcup {\geq} x_{k+1} \mathsf{r}.D_{k+1})^{\mathcal{I}'}, \cdots, a \in (C_m \sqcup {\geq} x_m \mathsf{r}.D_m)^{\mathcal{I}'}$. Since $a \notin C_1^{\mathcal{I}'}, \cdots, \notin C_k^{\mathcal{I}'}$, it implies that $a \in ({\geq} x_1 \nabla.D_1)^{\mathcal{I}'}, \cdots, a \in ({\geq} x_k \nabla.D_k)^{\mathcal{I}'}$, leading to that $|D_1^{\mathcal{I}'}| \geq x_1, \cdots, |D_k^{\mathcal{I}'}| \geq x_k$. In the case where only the negative r-clause $E_j \sqcup {\leq} y_j r.F_j$ is considered, all elements in $D_1^{\mathcal{I}'} \cup \cdots \cup D_k^{\mathcal{I}'}$ can be taken as r-successors of a. Since $|D_1^{\mathcal{I}'}| \geq x_1, \cdots, |D_k^{\mathcal{I}'}| \geq x_k$, it follows that $a \in ({\geq} x_1 r.D_1)^{\mathcal{I}'}, \cdots, a \in ({\geq} x_k r.D_k)^{\mathcal{I}'}$, and hence $a \in (C_1 \sqcup {\geq} x_1 r.D_1)^{\mathcal{I}'}, \cdots, a \in (C_k \sqcup {\geq} x_k r.D_k)^{\mathcal{I}'}$. However, for other negative r-clauses $E_i \sqcup \leq y_i r.F_i$ ($i \neq j$), this construction may result in the number of r-successors of a in $F_i^{\mathcal{I}'}$ exceeding y_i. Consequently, if $a \notin E_i^{\mathcal{I}'}$, it follows that $a \notin (E_i \sqcup \leq y_i r.F_i)^{\mathcal{I}'}$. To avoid the above situation when constructing the interpretation of r, it is necessary to fully consider all negative r-clauses. For $E_i \sqcup \leq y_i r.F_i$ ($i \neq j$), whenever $a \in E_i^{\mathcal{I}'}$, it holds that $a \in (E_i \sqcup \leq y_i r.F_i)^{\mathcal{I}'}$ regardless of the method used to construct the interpretation of r. Therefore, we only need to consider the case where $a \notin E_i^{\mathcal{I}'}$. For convenience, assume $a \notin E_{i_1}^{\mathcal{I}'}, \cdots, a \notin E_{i_t}^{\mathcal{I}'}$. For positive r-clauses $C_i \sqcup \geq x_i r.D_i$ ($1 \leq i \leq k$), since $a \in (C_i \sqcup E_{i_1} \sqcup \cdots \sqcup E_{i_t} \sqcup \geq (x_i - y_{i_1} - \cdots - y_{i_t})\nabla.(D_i \sqcap \neg F_1 \sqcap \cdots \sqcap \neg F_{i_t}))^{\mathcal{I}'}$, and $a \notin C_i^{\mathcal{I}'}$, it follows that $|(D_i \sqcap \neg F_{i_1} \sqcap \cdots \sqcap \neg F_{i_t})^{\mathcal{I}'}| \geq x_i - y_{i_1} - \cdots - y_{i_t}$. When constructing the interpretation of r, first select $x_i - y_{i_1} - \cdots - y_{i_t}$ elements from $(D_i \sqcap \neg F_{i_1} \sqcap \cdots \sqcap \neg F_{i_t})^{\mathcal{I}'}$ as the successors of a with respect to r. Then, as discussed earlier for the case $a \notin E^{\mathcal{I}'}$, for $i_1 \leq s \leq i_t$, select y_s elements from the intersection of F_s and D_i as the successors of a with respect to r.

□

Lemma 4. *Let x and y be the respective counts of r-clauses of the forms $C \sqcup {\geq} mr.D$ and $C \sqcup {\leq} nr.D$. Let z represent the maximum qualified number present in $C \sqcup {\geq} mr.D$ clauses. Then, the total count of the derived clauses resulting from these clauses is bounded by $O\left(y \cdot 2^x \cdot z^{2^x}\right)$.*

Proof. We define literals $Qt_1 r.D$ and $Qt_2 r.D$ as structurally identical but with distinct cardinality restrictions, a concept which we refer to as *isomorphism*, where $Q \in \{\geq, \leq\}$. When the corresponding literals of two clauses are isomorphic, the clauses themselves are deemed isomorphic.

Within the k-th tier clauses, the range of the number restriction $z_{i_{j_1}, i_{j_2}}$ is confined to $[z]$, indicating a total of z distinct potential values. Given $1 \leq j_1 < j_2 \leq k$, there exists a total of $\binom{k}{2}$ permutations for $z_{i_{j_1}, i_{j_2}}$ across different combinations of j_1 and j_2. Thus, we observe $z^{\binom{k}{2}}$ potential literals for $\bigsqcup_{1 \leq j_1 < j_2 \leq k} \geq z_{i_{j_1}, i_{j_2}} \nabla.(D_{i_{j_1}} \sqcap D_{i_{j_2}})$. Employing a similar methodology allows for the enumeration of other number restriction scenarios, enabling an analysis of the number of isomorphic clauses within the k-th tier. The mathematical expression for such an enumeration is given by:

$$z^{\binom{k}{2}} \cdot z^{\binom{k}{3}} \cdots z^{\binom{k}{k}} < z^{2^k}$$

For the k-th tier, the total number of non-isomorphic clauses equals $\binom{x}{k}$. Consequently, the number of clauses contained within a single Block is bounded above by:

$$\sum_{k=0}^{x} \binom{x}{k} \cdot z^{2^k} \leq \sum_{k=0}^{x} \binom{x}{k} \cdot z^{2^x} = 2^x \cdot z^{2^x}$$

The count of Blocks produced by the inference rule matches that of negative r-clauses. Therefore, the maximal number of clauses generated during the forgetting process is $y \cdot 2^x \cdot z^{2^x}$, which results in a growth pattern that is double exponential relative to x.

□

Theorem 1. *Given as inputs any $\mathcal{ALCQ}$-ontology $\mathcal{O}$ and any forgetting signature $\mathcal{F}$ of role names, our forgetting method always terminates and returns an $\mathcal{ALCQ}(\nabla)$-ontology $\mathcal{V}$.*

(i) *If $\mathcal{V}$ does not contain any definers, it is a result of forgetting $\mathcal{F}$ from $\mathcal{O}$, i.e., for any interpretation $\mathcal{I}$:*

$$\mathcal{I} \models \mathcal{V} \text{ iff } \mathcal{I}' \models \mathcal{O},$$

for some interpretation $\mathcal{I}'$ such that $\mathcal{I}' \sim_{\mathcal{F}} \mathcal{I}$;

(ii) *If $\mathcal{V}$ does contain definers, then for any interpretation $\mathcal{I}$:*

$$\mathcal{I} \models \mathcal{V} \text{ iff } \mathcal{I}' \models \mathcal{O},$$

for some interpretation $\mathcal{I}'$ such that $\mathcal{I}' \sim_{\mathcal{F} \cup sig_D(\mathcal{O}')} \mathcal{I}$.

Proof. This follows from Lemmas 1, 2, 3 and 4. □

B Additional Empirical Results

Tables 6 and 7 provide the additional experiment results for Lethe and Fame on the BioPortal dataset. Overall, the three methods demonstrated slightly better results on BioPortal than on the Oxford ISG dataset, with modest gains in success rate, time consumption, and memory usage. The variation in performance largely arose from the simpleness of the BioPortal ontologies, which have fewer axioms and role names and generally flatter structures than those in the Oxford ISG dataset.

These results mirror the performance observed with the Oxford ISG dataset, where our prototype achieved higher success rates while streamlining the computation process, completing the forgetting of all role names and definers (if introduced) more quickly and with reduced memory usage. To further elucidate the factors driving this efficiency, we compare the number of definers Lethe and our prototype introduced during the forgetting process.

The two tables below detail the number of definers introduced by Lethe and our prototype while forgetting 10%, 30%, and 50% of role names across a randomly selected subset of test ontologies. This was intended to demonstrate that an effective definer introduction strategy can significantly

Table 6: Lethe's results over BioPortal

$\mathcal{F}$%	Part	Time	Mem	SR	TR	MR	DR	DC
10%	I	7.79	711.62	92.57	1.49	4.45	1.49	21.21
	II	11.21	935.51	85.58	2.88	4.81	6.73	42.96
	III	31.76	1146.50	50.00	10.00	25.00	15.00	61.76
	Avg	**10.35**	**809.72**	**87.73**	**2.45**	**5.83**	**3.99**	**30.64**
30%	I	18.44	1106.63	87.62	3.96	3.47	4.95	65.19
	II	34.63	1499.14	79.81	4.81	5.77	9.62	116.32
	III	56.36	1935.11	50.00	10.00	25.00	15.00	193.23
	Avg	**25.93**	**1282.67**	**82.82**	**4.60**	**5.52**	**7.06**	**89.25**
50%	I	32.18	1662.82	76.24	4.95	7.42	11.39	166.21
	II	65.94	2040.73	71.15	7.69	8.65	12.50	205.85
	III	129.37	2516.79	40.00	10.00	25.00	25.00	286.12
	Avg	**48.91**	**1745.05**	**72.39**	**6.13**	**8.90**	**12.58**	**186.21**

Table 7: Fame's results over BioPortal

$\mathcal{F}$%	Part	Time	Mem	SR	TR	MR	DR	DC
10%	I	2.31	634.47	93.07	2.48	2.97	1.49	5.64
	II	8.47	853.98	87.50	2.88	2.88	6.73	12.07
	III	19.56	974.28	50.00	10.00	25.00	15.00	16.20
	Avg	**5.33**	**718.73**	**88.65**	**2.45**	**4.91**	**3.99**	**8.34**
30%	I	7.54	993.26	89.11	3.47	2.48	4.95	16.41
	II	15.28	1203.44	81.73	4.81	3.85	9.62	27.05
	III	27.84	1593.19	50.00	10.00	25.00	15.00	48.65
	Avg	**11.25**	**1097.12**	**84.35**	**4.29**	**4.29**	**7.06**	**21.78**
50%	I	12.98	1269.38	79.21	5.45	3.96	11.39	42.50
	II	19.47	1427.58	73.08	8.65	5.77	12.50	50.40
	III	31.67	1632.04	40.00	10.00	25.00	25.00	64.25
	Avg	**18.59**	**1342.10**	**74.85**	**6.75**	**5.83**	**12.58**	**46.36**

enhance the efficiency of the forgetting process. Since each experimental run was repeated 100 times, to manage the statistical workload, only the results from a single run were displayed. Entries marked with -1 indicate that the forgetting was unsuccessful due to problems like timeouts, memory overflow, or definer retention.

References

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

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.* 50(2):510–530.

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

Baral, C., and Zhang, Y. 2005. Knowledge updates: Semantics and complexity issues. *Artif. Intell.* 164(1-2):209–243.

Bicarregui, J.; Dimitrakos, T.; Gabbay, D. M.; and Maibaum, T. S. E. 2001. Interpolation in practical formal development. *Log. J. IGPL* 9(2):231–244.

Boole, G. 1854. *An Investigation of the Laws of Thought.* Walton & Maberly.

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

Conradie, W.; Goranko, V.; and Vakarelov, D. 2006. Algo-

Table 8: Definers introduced during forgetting (BioPortal)

Onto	Lethe (0.1)	Proto (0.1)	Lethe (0.3)	Proto (0.3)	Lethe (0.5)	Proto (0.5)
neomark3	45	0	237	0	515	0
qudt	32	0	272	0	446	0
ogi	88	0	-1	-1	-1	-1
gro	10	0	198	16	-1	-1
nifsubcell	52	0	1672	36	-1	-1
ciinteado	8	0	17	0	43	0
aao	6	0	45	0	78	0
vico	124	26	-1	-1	-1	-1
abd	5	0	15	0	32	0
xco	1	0	82	6	-1	51
cbo	13	0	125	0	212	0
fb	55	7	-1	-1	-1	-1
chmo	13	0	99	0	-1	58
cabro	11	0	29	0	65	0
sse	15	0	37	0	86	0
opb	105	0	-1	0	-1	0
fbbi	29	1	-1	0	-1	-1
oborel	23	0	59	0	112	0
hupson	-1	-1	-1	-1	-1	-1
bim	21	0	57	0	132	0
idoden	11	0	42	0	134	0
provo	21	0	182	0	378	0
suicideo	24	0	71	0	164	0
shr	32	0	119	0	357	0
cisaviado	7	0	17	0	51	0
pav	9	0	47	0	142	0
sp	11	0	52	0	105	0
nihss	3	0	31	0	94	0
ico	77	2	-1	-1	-1	-1
medeon	8	0	32	0	-1	-1
ctcae	-1	-1	-1	-1	-1	-1
ecp	2	0	28	0	63	0
canco	36	0	156	8	-1	-1
adar	27	0	209	15	563	27
bof	-1	-1	-1	-1	-1	-1
pdo	29	0	145	0	327	0
bspo	19	0	41	0	104	0
hpio	12	0	39	0	-1	-1
obi	63	9	385	41	-1	-1
gene	0	0	0	0	119	24
vt	5	0	14	0	29	0
triage	37	0	194	0	-1	-1
bt	-1	-1	-1	-1	-1	-1
ancestro	6	0	17	0	51	0
cheminf	-1	-1	-1	-1	-1	-1
omrse	62	4	-1	-1	-1	-1
moocciado	7	0	18	0	46	0
xeo	4	0	10	0	17	0
dseo	13	0	46	0	94	0
envo	36	0	133	0	-1	0
ontopneumo	11	0	38	0	73	0
aeo	21	0	69	0	-1	-1
onlira	7	0	15	0	35	0
obiws	-1	-1	-1	-1	-1	-1
sao	37	0	62	0	113	0
nemo	49	3	199	37	-1	-1
allergydetector	46	7	-1	-1	-1	-1
piero	45	0	126	0	382	0
cogpo	57	10	-1	-1	-1	-1
adalab	104	0	-1	0	-1	-1
ooevv	41	0	73	0	131	0
bmt	7	0	35	0	82	0
neomark4	11	0	32	0	64	0
rnao	144	0	391	0	673	0
opl	12	0	25	0	42	0
jerm	19	0	42	0	-1	-1
carre	43	0	117	0	217	0
bp	57	0	184	13	-1	-1
psds	7	1	24	5	-1	-1
cno	10	0	24	0	54	0

Table 9: Definers introduced during forgetting (Oxford)

Onto Code	Lethe (0.1)	Proto (0.1)	Lethe (0.3)	Proto (0.3)	Lethe (0.5)	Proto (0.5)
00006	36	0	86	0	183	0
00356	301	0	303	0	232	0
00357	59	0	315	0	323	0
00358	7	0	16	0	15	0
00359	17	0	54	0	73	0
00366	1	0	3	0	6	0
00367	4	0	4	0	6	0
00402	49	0	248	0	215	0
00403	83	0	210	0	439	21
00411	7	0	24	0	30	0
00412	34	0	160	0	163	14
00413	34	0	96	0	138	0
00423	24	0	59	0	82	0
00433	8	0	24	0	40	0
00445	5	0	21	0	16	0
00451	84	0	208	0	416	0
00452	125	0	554	0	439	0
00457	4	0	6	0	15	0
00458	2	0	9	0	14	0
00464	22	0	20	0	41	14
00468	0	0	0	0	1	0
00469	1	0	2	0	8	0
00494	80	0	208	0	427	0
00495	74	0	213	0	324	0
00497	284	0	716	0	1137	0
00498	300	0	1410	0	1537	0
00505	2	0	1	0	2	0
00513	2	0	7	0	5	0
00514	5	0	6	0	7	0
00515	30	0	113	0	148	0
00519	1	0	8	0	14	0
00520	7	0	13	0	12	0
00522	161	0	398	0	786	27
00523	734	0	406	0	590	0
00527	18	0	90	0	69	0
00544	245	0	1054	0	1095	0
00545	199	0	1102	0	1154	0
00546	63	0	352	0	278	0
00547	63	0	352	0	296	0
00548	4	0	10	0	13	0
00562	2	0	6	0	7	0
00563	2	0	7	0	14	0
00570	6	0	4	0	6	0
00571	1	0	4	0	6	0
00578	28	0	108	0	90	0
00589	7	0	20	0	20	0
00591	2	0	12	0	12	0
00592	7	0	13	0	19	0
00593	5	0	21	0	28	0
00594	4	0	31	0	24	0
00596	17	0	32	0	32	0
00600	3	0	31	0	19	0
00605	6	0	14	0	13	0
00606	6	0	9	0	16	0
00627	18	0	69	0	95	0
00629	20	0	75	0	88	0
00639	13	0	28	0	49	0
00640	14	0	39	0	57	0
00645	304	0	177	0	312	0
00646	373	0	235	0	383	0
00649	20	0	65	0	124	0
00650	20	0	63	0	124	0
00667	36	0	211	0	167	0
00669	473	0	529	0	491	0
00689	42	0	153	0	187	0
00690	56	0	149	0	241	0
00694	148	0	814	0	981	19
00696	216	0	532	0	1173	16
00770	90	0	402	0	372	0
00771	8	0	34	0	78	0

rithmic correspondence and completeness in modal logic. i. the core algorithm SQEMA. *Log. Methods Comput. Sci.* 2(1).

Craig, W. 1957. Three uses of the herbrand-gentzen theorem in relating model theory and proof theory. *J. Symb. Log.* 22(3):269–285.

D'Agostino, G., and Hollenberg, M. 2000. Logical questions concerning the μ-calculus: Interpolation, lyndon and lostarski. *J. Symb. Log.* 65(1):310–332.

Del-Pinto, W., and Schmidt, R. A. 2019. ABox Abduction via Forgetting in $\mathcal{ALC}$. In *Proc. AAAI'19*, 2768–2775. AAAI Press.

Delgrande, J. P., and Wang, K. 2015. A syntax-independent approach to forgetting in disjunctive logic programs. In *Proc. AAAI'15'*, 1482–1488. AAAI Press.

Doherty, P.; Lukaszewicz, W.; and Szalas, A. 1997. Computing circumscription revisited: A reduction algorithm. *J. Autom. Reason.* 18(3):297–336.

Eiter, T., and Wang, K. 2008. Semantic forgetting in answer set programming. *Artif. Intell.* 172(14):1644–1672.

Erdem, E., and Ferraris, P. 2007. Forgetting actions in domain descriptions. In *Proc. AAAI'07*, 409–414. AAAI Press.

Fermüller, C. G.; Leitsch, A.; Hustadt, U.; and Tammet, T. 2001. Resolution decision procedures. In Robinson, J. A., and Voronkov, A., eds., *Handbook of Automated Reasoning (in 2 volumes)*. Elsevier and MIT Press. 1791–1849.

Gabbay, D. M., and Ohlbach, H. J. 1992. Quantifier elimination in second-order predicate logic. *South African Computer Journal* 7:35–43.

Gabbay, D. M.; Schmidt, R. A.; and Szalas, A. 2008. *Second-Order Quantifier Elimination - Foundations, Computational Aspects and Applications*, volume 12 of *Studies in logic : Mathematical logic and foundations*. College Publications.

Ghilardi, S.; Lutz, C.; and Wolter, F. 2006. Did I damage my ontology? A case for conservative extensions in description logics. In *Proc. KR'06*, 187–197. AAAI Press.

Grau, B. C.; Horrocks, I.; Kazakov, Y.; and Sattler, U. 2008. Modular reuse of ontologies: Theory and practice. *J. Artif. Intell. Res.* 31:273–318.

Herzig, A., and Mengin, J. 2008. Uniform interpolation by resolution in modal logic. In *Proc. JELIA'08*, volume 5293 of *LNCS*, 219–231. Springer.

Klein, M. C. A., and Fensel, D. 2001. Ontology versioning on the Semantic Web. In *Proc. SWWS'01*, 75–91.

Konev, B.; Ludwig, M.; Walther, D.; and Wolter, F. 2012. The logical difference for the lightweight description logic $\mathcal{EL}$. *J. Artif. Intell. Res.* 44:633–708.

Konev, B.; Lutz, C.; Walther, D.; and Wolter, F. 2013. Model-theoretic inseparability and modularity of description logic ontologies. *Artif. Intell.* 203:66–103.

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

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

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

Koopmann, P.; Del-Pinto, W.; Tourret, S.; and Schmidt, R. A. 2020. Signature-based abduction for expressive description logics. In *Proc. KR'20*, 592–602.

Koopmann, P. 2020. LETHE: forgetting and uniform interpolation for expressive description logics. *Künstliche Intell.* 34(3):381–387.

Koopmann, P. 2021. Signature-based abduction with fresh individuals and complex concepts for description logics. In *Proc. IJCAI'21*, 1929–1935. ijcai.org.

Lambrix, P., and Tan, H. 2008. Ontology Alignment and Merging. In *Anatomy Ontologies for Bioinformatics, Principles and Practice*, volume 6 of *Computational Biology*. Springer. 133–149.

Lang, J., and Marquis, P. 2010. Reasoning under inconsistency: A forgetting-based approach. *Artif. Intell.* 174(12-13):799–823.

Lang, J.; Liberatore, P.; and Marquis, P. 2003. Propositional independence: Formula-variable independence and forgetting. *J. Artif. Intell. Res.* 18:391–443.

Liberatore, P., and Schaerf, M. 1998. Arbitration (or how to merge knowledge bases). *IEEE Trans. Knowl. Data Eng.* 10(1):76–90.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly equivalent logic programs. *ACM Trans. Comput. Log.* 2(4):526–541.

Lin, F., and Reiter, R. 1994. Forget It! In *Proc. AAAI Fall Symposium on Relevance*, 154–159. AAAI Press.

Liu, Z.; Lu, C.; Alghamdi, G.; Schmidt, R. A.; and Zhao, Y. 2021. Tracking semantic evolutionary changes in large-scale ontological knowledge bases. In *Proc. CIKM'21*, 1130–1139. ACM.

Lutz, C., and Wolter, F. 2010. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *J. Symb. Comput.* 45(2):194–228.

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

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

Matentzoglu, N., and Parsia, B. 2017. BioPortal Snapshot 30.03.2017.

Nebel, B.; Dimopoulos, Y.; and Koehler, J. 1997. Ignoring irrelevant facts and operators in plan generation. In *Proc. ECP'97*, volume 1348 of *LNCS*, 338–350. Springer.

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

Nikitina, N.; Rudolph, S.; and Glimm, B. 2012. Interactive ontology revision. *J. Web Semant.* 12:118–130.

Plessers, P., and Troyer, O. D. 2005. Ontology change detection using a version log. In *Proc. ISWC'05*, volume 3729, 578–592. Springer.

Ribeiro, M. M., and Wassermann, R. 2009. Base revision for ontology debugging. *J. Log. Comput.* 19(5):721–743.

Robinson, J. A. 1965. A machine-oriented logic based on the resolution principle. *J. ACM* 12(1):23–41.

Schlobach, S., and Cornet, R. 2003. Non-standard reasoning services for the debugging of description logic terminologies. In *Proc. IJCAI'03*, 355–362. Morgan Kaufmann.

Schmidt, R. A. 2012. The Ackermann approach for modal logic, correspondence theory and second-order reduction. *J. Appl. Log.* 10(1):52–74.

Szalas, A. 1993. On the correspondence between modal and classical logic: An automated approach. *J. Log. Comput.* 3(6):605–620.

Troquard, N.; Confalonieri, R.; Galliani, P.; Peñaloza, R.; Porello, D.; and Kutz, O. 2018. Repairing ontologies via axiom weakening. In *Proc. AAAI'18*, 1981–1988. AAAI Press.

Visser, A. 1996. *Bisimulations, Model Descriptions and Propositional Quantifiers.* Logic Group Preprint Series. Utrecht University.

Wang, K.; Antoniou, G.; Topor, R.; and Sattar, A. 2005. Merging and aligning ontologies in dl-programs. In *Proc. RuleML'05*, volume 3791 of *LNCS*, 160–171. Springer.

Wang, Y.; Wang, K.; and Zhang, M. 2013. Forgetting for answer set programs revisited. In *Proc. IJCAI'13'*, 1162–1168. IJCAI/AAAI.

Wernhard, C. 2012. Projection and scope-determined circumscription. *J. Symb. Comput.* 47(9):1089–1108.

Wu, X.; Deng, W.; Lu, C.; Feng, H.; and Zhao, Y. 2020. UI-FAME: A high-performance forgetting system for creating views of ontologies. In *Proc. CIKM'20*, 3473–3476. ACM.

Xiang, Y.; Wu, X.; Lu, C.; and Zhao, Y. 2022. Creating signature-based views for description logic ontologies with transitivity and qualified number restrictions. In *Proc. WWW'22*, 808–817. ACM.

Yang, H.; Koopmann, P.; Ma, Y.; and Bidoit, N. 2023. Efficient computation of general modules for $\mathcal{ALC}$ ontologies. In *Proc. IJCAI'23*, 3356–3364. ijcai.org.

Zhang, Y., and Foo, N. Y. 2006. Solving logic program conflict through strong and weak forgettings. *Artif. Intell.* 170(8-9):739–778.

Zhang, Y., and Zhou, Y. 2010. Forgetting revisited. In *Proc. KR'10.* AAAI Press.

Zhao, Y., and Schmidt, R. A. 2016. Forgetting concept and role symbols in $\mathcal{ALCOIH}\mu^+(\triangledown, \sqcap)$-ontologies. In *Proc. IJCAI'16*, 1345–1353. IJCAI/AAAI Press.

Zhao, Y., and Schmidt, R. A. 2017. Role forgetting for $\mathcal{ALCOQH}(\triangledown)$-ontologies using an Ackermann-based approach. In *Proc.IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 1354–1361. ijcai.org.

Zhao, Y., and Schmidt, R. A. 2018a. FAME: An automated tool for semantic forgetting in expressive description logics. In *Proc. IJCAR'18*, volume 10900 of *LNCS*, 19–27. Springer.

Zhao, Y., and Schmidt, R. A. 2018b. On concept forgetting in description logics with qualified number restrictions. In *Proc. IJCAI'18*, 1984–1990. ijcai.org.

Zhao, Y.; Alghamdi, G.; Schmidt, R. A.; Feng, H.; Stoilos, G.; Juric, D.; and Khodadadi, M. 2019. Tracking logical difference in large-scale ontologies: A forgetting-based approach. In *Proc. AAAI'19*, 3116–3124. AAAI Press.

Zhao, Y. 2023. Highly-optimized forgetting for creating signature-based views of ontologies. In *Proc. CIKM'23*, 3444–3452. ACM.