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

Anonymous Author(s)

ABSTRACT

Forgetting is a non-standard reasoning procedure used to refine an ontology into a sub-signature by eliminating symbols not included in this subset, addressing fundamental challenges in knowledge management where ontology refinement and reuse are crucial for efficient information processing. It has two forms: weak forgetting (aka uniform interpolation), which preserves entailments within the source language, and strong forgetting, which additionally ensures model preservation modulo the eliminated symbols. This makes the latter significantly more challenging to compute.

In this paper, we present the first method for strong role forgetting in description logics with qualified number restrictions ($\mathcal{Q}$). In particular, the method takes $\mathcal{ALCQ}$-ontologies as input, yielding output either in $\mathcal{ALCQ}$ or in $\mathcal{ALCQ}(\nabla)$ by further incorporating the universal role ∇ to avoid information loss. This preserves model-theoretic properties crucial for applications such as modal correspondence theory and second-order quantifier elimination. While the method guarantees termination and soundness, its completeness is inherently constrained by the undecidability of strong forgetting. However, empirical evaluations on the Oxford-ISG and BioPortal benchmarks show that this theoretical limitation barely impedes practical utility, with experiment results demonstrating superb success rates and remarkably high efficiency.

CCS CONCEPTS

• **Information systems → Web Ontology Language (OWL)**; • **Computing methodologies → Description logics**; **Ontology engineering**.

KEYWORDS

Ontology, Description Logic, Strong Forgetting, Knowledge Reuse

ACM Reference Format:

Anonymous Author(s). 2018. Strong Forgetting for $\mathcal{ALCQ}$-Ontologies. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 14 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Forgetting [33] 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 or logical consequences as the original. This process, which can be seen as a formal, rigorous approach to what in everyday terms may 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 properties of the original, larger theory. While the term 'forgetting' was first coined by Lin and Reiter [33], its conceptual foundation can be traced back to the early work of Boole on propositional variable elimination [5] and the seminal work of Ackermann [1], who recognized the problem as that of eliminating second-order quantifiers.

1.1 Related Work

Forgetting in logical theories (e.g., ontologies) can be formalized in two closely related ways. The first formalization, introduced by Lin and Reiter [33], adopts a *model-theoretic* perspective — when forgetting a predicate symbol P from a logical theory $\mathcal{T}$, it produces a new theory $\mathcal{T}'$ whose models agree on those of $\mathcal{T}$, differing only possibly in their interpretations of P. The work further showed that for finite theories, the problem of predicate forgetting is equivalent to eliminating second-order quantifiers over predicates [18]. This equivalence entails that forgetting results transcend first-order definability and can be computed using second-order quantifier elimination methods such as Scan [17], Dls [46], Sqema [7], and Msqel [45]. Zhang and Zhou [56] later termed Lin and Reiter's approach 'strong forgetting' and introduced an alternative formalization known as 'weak forgetting' from a *deductive* perspective. In weak forgetting, forgetting a predicate symbol P from a logical theory $\mathcal{T}$ yields a set $\mathcal{T}'$ containing all first-order logical consequences of $\mathcal{T}$ that are irrelevant to P.[1] Notably, weak forgetting typically produces results $\mathcal{T}_1$ that are logically weaker than those of strong forgetting $\mathcal{T}_2$ (i.e., $\mathcal{T}_2 \models \mathcal{T}_1$). However, when $\mathcal{T}_2$ is first-order definable, both notions yield equivalent results (i.e., $\mathcal{T}_2 \equiv \mathcal{T}_1$). While weak forgetting always yields first-order definable results, the resulting set $\mathcal{T}_1$ may contain infinitely many first-order formulas.

In logic, forgetting has been studied as a dual problem to uniform interpolation [9, 21, 49], a notion that extends the well-known Craig interpolation [8], yet with stronger semantic guarantees. In AI, the fundamental importance of forgetting has been recognized across multiple domain. These include *circumscription* [13], *planning* [15, 38], *specification refinement* [4], *knowledge representation* [3, 31], *logic programming* [12, 14, 51, 55], and *belief revision* [32]

Recent advances in forgetting have primarily occurred within the domain of Description Logics (DLs) [2], a knowledge representation formalism that serves as the logical underpinning of modern ontology languages. DL-based forgetting proves essential in tasks where ontological functionality must be preserved while restricting access to specific symbols. These tasks include *ontology merging and alignment* [30, 50], *debugging and repair* [42, 44, 48], *versioning* [22, 41], *abduction and explanation generation* [11, 27], *logical difference* [23, 57], and *interactive ontology revision* [40].

[1] A sentence is considered irrelevant to a predicate P if it is logically equivalent to another sentence not containing P.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

Forgetting in DLs is typically characterized in terms of (model-theoretic and deductive) *inseparability* and *conservative extension* relations [19, 20, 24, 35]. 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 a forgetting result for $\mathcal{L}$;
(iv) Developing and optimizing practical methods to compute a forgetting result for $\mathcal{L}$.

Significant theoretical results include:

(i) Closure under forgetting, whether it be under the strong or weak notion, is a rare property among DLs, even for those with limited expressivity, such as $\mathcal{EL}$ — a forgetting result for an ontology expressed in $\mathcal{EL}$ may not necessarily exist within the expressivity of $\mathcal{EL}$ itself [35]. This result also extends to $\mathcal{ALC}$ [19].
(ii) Determining the existence of a strong forgetting result is undecidable for both $\mathcal{EL}$ and $\mathcal{ALC}$ [24];
(iii) Determining the existence of a weak forgetting result is ExpTime-complete for $\mathcal{EL}$ [34, 39] and 2ExpTime-complete for $\mathcal{ALC}$ [36];
(iv) The size of weak forgetting result can be triple exponential compared to the size of the input ontology in the worst-case scenario for both $\mathcal{EL}$ and $\mathcal{ALC}$ [36, 39]. Strong forgetting only makes this problem even more challenging.

These results demonstrate the substantial computational challenges inherent to forgetting, motivating the urgent demand for efficient methods and heuristics to make forgetting more useful in practice.

1.2 Motivation & Contributions

Research on practical forgetting in DLs has predominantly centered on the weaker notion, with very limited works exploring the stronger notion [47, 52]. Moreover, these works have largely concentrated on forgetting concept names, presumably due to the inherent difficulties of role forgetting [59]. The Ackermann's Lemma-based Fame approach [58] emerged as an exception, enabling the elimination of both concept and role names in the DL $\mathcal{ALCOIH}$, but attempts to extend this approach to incorporate qualified number restrictions ($\mathcal{Q}$) yields only weak forgetting results when Ackermann's Lemma is not applicable [59, 61]. Qualified number restrictions represent one of the most widely used constructors in DLs, enabling fine-grained cardinality constraints on roles. This expressive power is crucial for accurately modeling real-world domain knowledge with ontologies. For example, in medical ontologies, $\mathcal{Q}$ allows one to specify that 'A human heart has exactly four chambers', or in automotive ontologies, that 'A car has at least four wheels'. Given the fundamental role of $\mathcal{Q}$ in modern knowledge modeling, the current lack of strong forgetting approaches for DLs with $\mathcal{Q}$ poses a significant limitation in the aforementioned tasks.

In this paper, we present the first approach for strong role forgetting in DLs with $\mathcal{Q}$. We focus only on role forgetting because it subsumes concept forgetting through a simple reduction: any concept name A can be eliminated by introducing a fresh role name r and replacing A with '$\exists r.\top$', and finally eliminating r from the modified ontology. Models of the resulting ontology can be straightforwardly translated to the original ontology and vice versa [61]. Our approach takes $\mathcal{ALCQ}$-ontologies as input, yielding output either in $\mathcal{ALCQ}$ or $\mathcal{ALCQ}(\nabla)$ by further incorporating the universal role ∇ to preserve semantic equivalence and avoid information loss. Consider strongly forgetting role r from an $\mathcal{ALCQ}$ ontology $\mathcal{O} = \{A \sqsubseteq {\geq}2r.B, {\geq}2r.B \sqsubseteq B\}$: the result is an $\mathcal{ALCQ}(\nabla)$ ontology $\mathcal{V} = \{A \sqsubseteq {\geq}2\nabla.B, A \sqsubseteq B\}$, while no result exists without ∇. The method guarantees termination and soundness, though completeness is inherently limited by the undecidability of the forgetting problem. Evaluations on the Oxford-ISG and BioPortal benchmarks show that this theoretical limitation barely impedes practical performance, with experiment results demonstrating superb success rates and remarkably high efficiency.

An extended version of this paper containing complete proofs and additional experimental results, along with the source code for our prototype implementation and corresponding test datasets, is available for anonymous review and reproduction at https://github.com/anonymous-ai-researcher/cikm2025.

2 PRELIMINARIES

Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be countably infinite, disjoint sets of concept and role names, respectively. A role in $\mathcal{ALCQ}$ is defined as any role name $r \in \mathsf{N_R}$, while $\mathcal{ALCQ}(\nabla)$ extends this definition to include the universal role ∇. Concept descriptions in $\mathcal{ALCQ}(\nabla)$ (or simply concepts) are constructed according to the following syntax:

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

where $A \in \mathsf{N_C}$, C and D range over concepts, R over roles, and $m \geq 1$ and $n \geq 0$ over 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 cardinality constraints on roles.

An $\mathcal{ALCQ}(\nabla)$-ontology $\mathcal{O}$ is defined as 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 using 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. We define a *clause* as a GCI 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. Traditionally, we omit the prefix '$\top \sqsubseteq$' and treat clauses as sets, meaning that they contain no duplicates and their order is not important. Hence, when stating that $C^{\mathcal{I}} \cup (\geq mr.D)^{\mathcal{I}}$ is true, we imply that $\top \sqsubseteq C \sqcup (\geq mr.D)$ is true in $\mathcal{I}$. GCIs can be converted into clauses using standard transformations.

Let $\mathcal{S} \in \mathsf{N_C} \cup \mathsf{N_R}$ be a designated name. A clause that contains $\mathcal{S}$ is termed an $\mathcal{S}$*-clause*. Within an $\mathcal{S}$-clause, an occurrence of $\mathcal{S}$ is said to be *positive* if it occurs under an even number of (explicit and implicit) negations, and *negative* if under an odd number. Translating DL clauses to first-order logic reveals all implicit negations. For instance, in the clause $\geq 2r.A$, r occurs positively in its first-order representation $\forall x \exists y, z(r(x,y) \wedge r(x,z) \wedge A(y) \wedge A(z) \wedge y \neq z)$. In contrast, in the clause $\leq 1r.A$, r occurs negatively, as evidenced by its first-order representation $\forall x, y, z(\neg r(x,y) \vee \neg r(x,z) \vee \neg A(y) \vee \neg A(z) \vee y = z)$.

Consider a name $\mathcal{S} \in \mathsf{N_C} \cup \mathsf{N_R}$ and two interpretations $\mathcal{I}$ and $\mathcal{I}'$. We say that $\mathcal{I}$ and $\mathcal{I}'$ are $\mathcal{S}$-equivalent (denoted $\mathcal{I} \sim_{\mathcal{S}} \mathcal{I}'$) if they are identical except in how they interpret $\mathcal{S}$. This notion extends to $\mathcal{F}$-equivalence (denoted $\mathcal{I} \sim_{\mathcal{F}} \mathcal{I}'$) for a set $\mathcal{F}$ of names, where interpretations may differ only in how they interpret names in $\mathcal{F}$.

(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 (**Strong Forgetting for** $\mathcal{ALCQ}$). *Let $\mathcal{O}$ be an $\mathcal{ALCQ}$-ontology and $\mathcal{F} \subseteq sig_R(\mathcal{O})$ be a set of role names to be forgotten, called 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 two key properties hold:

(i) A result of forgetting $\mathcal{F}$ from $\mathcal{O}$ can be obtained through stepwise elimination of individual names in $\mathcal{F}$, irrespective of their order of elimination.

(ii) Any two results $\mathcal{V}$ and $\mathcal{V}'$ derived through the same forgetting process are logically equivalent, despite potential differences in their syntactic representations. This uniqueness property allows us to speak definitively of 'the result of forgetting' rather than indefinitely of 'a result of forgetting'.

3 NORMALIZATION OF ONTOLOGIES

Although our method takes an $\mathcal{ALCQ}$-ontology as input, the intermediate computations may generate $\mathcal{ALCQ}$-clauses extended with the universal role ∇. Consequently, the method must be capable of handling $\mathcal{ALCQ}(\nabla)$ to process these intermediate results. Specifically, for any $\mathcal{ALCQ}(\nabla)$-ontology $\mathcal{O}$ and forgetting signature $\mathcal{F}$, our method successively eliminates individual names in $\mathcal{F}$ using an inference calculus developed for single role elimination. The calculus operates on a dedicated normal form designed for $\mathcal{ALCQ}(\nabla)$-ontologies.

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

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

where $r \in N_R$, and C, D, E, and F are concepts (but 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.

The normal form establishes a standard clause structure, enabling subsequent inference systems to operate on syntactically uniform inputs. A key property of r-NF is its canonical representation of $\mathcal{ALCQ}(\nabla)$-clauses through unique placement of role name r within either a $\geq$-restriction or $\leq$-restriction.

Any $\mathcal{ALCQ}(\nabla)$-ontology $\mathcal{O}$ can be transformed into r-NF by exhaustively applying the normalization rules NR1–NR3 to those r-clauses in $\mathcal{O}$ that are not yet in r-NF (where R is an arbitrary role name). This transformation process introduces fresh concept names, termed *definers* [28], with each definer 'Z' uniquely introduced as an abbreviation for complex concepts during normalization.

- NR1 For each clause instance of $C \sqcup \geq mR.D$ or $E \sqcup \leq nR.F$ not in r-NF, if $r \in \mathrm{sig}(C)$ (or if $r \in \mathrm{sig}(E)$), replace C (or E) with a fresh definer $Z \in \mathsf{N_C}$ and add $\neg Z \sqcup C$ (or $\neg Z \sqcup E$) to $\mathcal{O}$;
- NR2 For each clause instance of $C \sqcup \geq mR.D$ not in r-NF, if $r \in \mathrm{sig}(D)$, replace D by a fresh definer $Z \in \mathsf{N_C}$ and add $\neg Z \sqcup D$ to $\mathcal{O}$.
- NR3 For each clause instance of $E \sqcup \leq nR.F$ not in r-NF, if $r \in \mathrm{sig}(F)$, replace F by a fresh definer $Z \in \mathsf{N_C}$ and add $Z \sqcup \neg F$ 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 $\mathcal{F} = \{r\}$. Consider the following rule applicability: if we correspond the literal $\leq 1r.\geq 2r.A_1$ to $\leq nR.D$, and $\geq 2s.\leq 1t.\leq 3r.A_2 \sqcup \leq 0t.A_3$ to C from NR1 or NR3, the side conditions allow the application of either rule, indicating that multiple normalization rules may be applicable in the same situation. Our method addresses this by applying NR1–NR3 sequentially, as illustrated below. Importantly, the order of rule application does not affect the correctness of the final 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 NR3 to Clause 2 gives ($Z_2 \in N_C$ is a fresh definer):

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

Applying NR2 to Clause 3 gives ($Z_3 \in N_C$ is a fresh definer):

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

Applying NR3 to Clause 7 gives ($Z_4 \in N_C$ is a fresh definer):

$$\{4. \leq 1r.Z_2 \sqcup Z_1 \quad 5. \neg Z_2 \sqcup \geq 2r.A_1 \quad 9. Z_4 \sqcup \geq 4r.\neg A_2$$
$$6. \neg Z_1 \sqcup \geq 2s.Z_3 \sqcup \leq 0t.A_3 \quad 8. \neg Z_3 \sqcup \leq 1t.Z_4\}$$

Now the resulting $\mathcal{O}' = \{4, 5, 6, 8, 9\}$ is in r-NF.

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 normalization rules NR1–NR3. 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 $\mathcal{O}$ by applying the rules NR1–NR3. Let $sig_D(\mathcal{O}')$ 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 $\mathcal{I} \sim_{sig_D(\mathcal{O}')} \mathcal{I}'$.

Lemma 1 establishes the termination and completeness of the normalization process and Lemma 2 states its soundness.

The definers introduced during the forgetting process are auxiliary symbols that fall outside the desired signature; they must be removed from the final result. The approach for definer elimination is described in Section 6.

4 LINEAR DEFINER INTRODUCTION

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 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 $A, Z \in \mathsf{N_C}$. A key restriction in Lethe is that it mandates every 'Z' — any subconcept immediately below an ∃- or a ∀-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.

Our analysis of Lethe's strategy is based on the following notations. The set of definers introduced in an ontology $\mathcal{O}$ is denoted by $\mathrm{sig_D}(\mathcal{O})$, while $\mathrm{sub}_{\exists}^{\forall}(\mathcal{O})$ 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, $\mathrm{sub}_X(\mathcal{O})$ denotes the set of all subconcepts X in $\mathcal{O}$ with $\exists r^{(-)}.X \in \mathrm{sub}_{\exists}^{\forall}(\mathcal{O})$ or $\forall r^{(-)}.X \in \mathrm{sub}_{\exists}^{\forall}(\mathcal{O})$.

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 $\mathrm{sig_D}(\mathcal{O})$: $f : \mathrm{sig_D}(\mathcal{O}) \to \mathrm{sub}_X(\mathcal{O})$. Note that f also exhibits surjectivity, given Lethe's exhaustive manner to introduce defines — requiring every subconcept immediately below an ∃- or a ∀-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 $|\,\mathrm{sig_D}(\mathcal{O})|$, is bounded by $\mathrm{O}(n)$, with n representing the number of ∃- and ∀-restrictions in $\mathcal{O}$. This implies a linear growth in the introduction of definers.

Lethe uses a saturation-based approach to facilitate the elimination of a single concept or role name 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 [16]. 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 $\mathrm{Res}(\mathcal{O})$. In this phase, a number of new consequences are deduced from existing ones. For example, the application of the ∀∃-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 $\mathrm{sig_D}(\mathrm{Res}(\mathcal{O}))$, namely $f : \mathrm{sig_D}(\mathrm{Res}(\mathcal{O})) \to \mathrm{sub}_X(\mathrm{Res}(\mathcal{O}))$, and the size of its codomain $|\mathrm{sub}_X(\mathrm{Res}(\mathcal{O}))|$ equals $2^{|\mathrm{sub_D}(\mathcal{O})|}$. The number of definers necessary for the normalization of $\mathrm{Res}(\mathcal{O})$ is thus bounded by $\mathrm{O}(2^n)$, where n denotes the number of ∃- and ∀-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 therefore should be excluded from the forgetting results. 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

Given an ontology $\mathcal{O}'$ in r-NF, our method eliminates a role name r from $\mathcal{O}'$ by applying the inference rule IR shown in Figure 1. This rule follows a premise-conclusion schema: the premises (clauses above the line) are replaced by the conclusion (below the line) to produce the result of forgetting r from $\mathcal{O}'$. The input $\mathcal{O}'$ typically contains three types of clauses: r-free clauses, denoted as $\mathcal{O}^{-r}$, clauses with positive occurrences of r in the form $C \sqcup {\geq}mr.D$, collectively denoted as $\mathcal{P}^{+}(r)$, and clauses with negative occurrences of r in the form $E \sqcup {\leq}nr.F$, collectively denoted as $\mathcal{P}^{-}(r)$.

The fundamental idea of IR is to derive inferences on the name to be eliminated (in this case, r), inferring new consequences (clauses) that are not relevant to r from all r-clauses; this enables one to safely remove all r-clauses — so does r itself — while preserving the original semantics of the remaining names. The key challenge lies in ensuring (internal) completeness of the inference rule — that is, its ability to derive all relevant consequences. For DLs with $\mathcal{Q}$, it is particularly challenging to find such rules and show that they really derive all relevant consequences.

IR derives consequences following a generalization of the binary resolution principle. This principle, originally developed for propositional logic [10] and then lifted to first-order logic [43], operates

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

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

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

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

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

$\cdots$

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. $\textsc{Resolve}(\mathcal{P}^+(r),\ E_j \sqcup \leq y_j r.F_j)$ $(1 \leq j \leq n)$ 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:

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

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

by generating a resolvent clause from two parent clauses containing complementary literals.

In $\mathcal{ALCQ}$, however, resolution operates rather differently from its propositional counterpart. Instead of resolving upon propositional variables, it targets role name r that appears within qualified number restrictions. Specifically, it resolves between literals of the form $\geq mr.D$ and $\leq nr.F$. The complementary nature of these literals becomes apparent through their first-order translations, as demonstrated in [61]. IR directly resolves qualified number restrictions upon r, producing a forgetting result within the DL framework. However, to achieve such representation, the target language has to mandate an extension of $\mathcal{ALCQ}$ with the universal role.

Let $|\mathcal{P}^+(r)| = m$ and $|\mathcal{P}^-(r)| = n$. To generate all relevant consequences, IR must, in principle, resolve all possible combinations of clauses from $\mathcal{P}^+(r)$ with all possible combinations from $\mathcal{P}^-(r)$ upon r, yielding the resolvent clause set $\textsc{Resolve}(\mathcal{P}^+(r), \mathcal{P}^-(r))$. This would require resolving $2^m \times 2^n$ combinations in total. While this represents a many-to-many relationship between the two sets, we further notice that it can be reduced to two simpler one-to-many relationships:

- Between each individual clause from $\mathcal{P}^+(r)$ and every possible combination of clauses from $\mathcal{P}^-(r)$, yielding the result $\textsc{Resolve}(\mathcal{P}^-(r), C_i \sqcup {\geq} x_i r.D_i)$ for $1 \leq i \leq m$;
- Between each individual clause from $\mathcal{P}^-(r)$ and every possible combination of clauses from $\mathcal{P}^+(r)$, yielding the result $\textsc{Resolve}(\mathcal{P}^+(r), E_j \sqcup {\leq} y_j r.F_j)$ for $1 \leq j \leq n$.

Therefore, the conclusion of IR can be represented as the union of these two sets. Delving deeper into $\textsc{Resolve}(\mathcal{P}^-(r), C_i \sqcup {\geq} x_i r.D_i)$, we can organize the resolutions into tiers based on the number of clauses involved from $\mathcal{P}^-(r)$:

- 0th-tier represents the result of resolving $C_i \sqcup {\geq} x_i r.D_i$ with the empty set $\emptyset$ upon r;
- 1st-tier represents the result of resolving $C_i \sqcup {\geq} x_i r.D_i$ with individual clauses from $\mathcal{P}^-(r)$;
- 2nd-tier represents the result of resolving $C_i \sqcup {\geq} x_i r.D_i$ with all possible pairs of clauses from $\mathcal{P}^-(r)$;
- This continues up to nth-tier, which represents the result of resolving $C_i \sqcup {\geq} x_i r.D_i$ with all n clauses from $\mathcal{P}^-(r)$.

A similar tier structure exists for $\textsc{Resolve}(\mathcal{P}^+(r), E_j \sqcup {\leq} y_j r.F_j)$ too, with tiers ranging from 0 to m based on the number of clauses involved in $\mathcal{P}^+(r)$. The forgetting result is then given by $\mathcal{O}^{-r} \cup \textsc{Resolve}(\mathcal{P}^-(r), C_i \sqcup {\geq} x_i r.D_i) \cup \textsc{Resolve}(\mathcal{P}^+(r), E_j \sqcup {\leq} y_j r.F_j)$.

Example 2. *Consider an $\mathcal{ALCQ}$-ontology $\mathcal{O}$ with $\mathcal{F} = \{r\}$:*

$$\{1.\ A_1 \sqcup {\leq} 2s.{\leq} 2r.\neg B_1 \quad 2.\ A_2 \sqcup {\geq} 1r.B_2 \quad 3.\ A_3 \sqcup {\leq} 2r.B_3\}$$

Applying NR3 to Clause 1 gives ($Z_1 \in N_C$ is a fresh definer):

$$\begin{gathered}\{2.\ A_2 \sqcup {\geq} 1r.B_2 \quad 3.\ A_3 \sqcup {\leq} 2r.B_3 \\ 4.\ A_1 \sqcup {\leq} 2s.Z_1 \quad 5.\ Z_1 \sqcup {\geq} 3r.B_1\},\end{gathered}$$

Applying IR to Clauses 3, 4, 5 gives:

$$\begin{aligned}
0th\text{-}tier\text{:}&\ \{6.\ A_2 \sqcup {\geq} 1\nabla.B_2 \quad 7.\ Z_1 \sqcup {\geq} 3\nabla.B_1\} \\
1st\text{-}tier\text{:}&\ \{8.\ Z_1 \sqcup A_3 \sqcup {\geq} 1\nabla.(B_1 \sqcap \neg B_3)\} \\
2nd\text{-}tier\text{:}&\ \{9.\ Z_1 \sqcup A_2 \sqcup A_3 \sqcup {\geq} 1\nabla(B_1 \sqcap B_2) \sqcup {\geq} 2\nabla.(B_1 \sqcup B_2) \sqcap \neg B_3 \\
&\qquad 10.\ Z_1 \sqcup A_2 \sqcup A_3 \sqcup {\geq} 2\nabla(B_1 \sqcap B_2) \sqcup {\geq} 1\nabla.(B_1 \sqcup B_2) \sqcap \neg B_3\}
\end{aligned}$$

Now $\mathcal{O}' = \{6, 7, 8, 9, 10\}$ constitutes the intermediate result of forgetting r from $\mathcal{O}$, but the definer 'Z' remains in it and must be eliminated to obtain the final result.

Lemma 3. *Let $\mathcal{O}'$ be an $\mathcal{ALCQ}(\nabla)$-ontology in r-NF, serving as the premises of the IR rule, 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 denote the numbers of r-clauses of the form $C \sqcup {\geq} mr.D$ and $C \sqcup {\leq} nr.D$, respectively, and let z be the maximum qualified number in clauses of the form $C \sqcup {\geq} mr.D$. The number of derived clauses is bounded by $O(y \cdot 2^x \cdot z^{2^x})$.*

Lemma 3 states soundness of IR; Lemma 4 shows its termination and upper bound of the size of the output.

Algorithm 1 The Forgetting Process

Require: An $\mathcal{ALCQ}$-ontology $\mathcal{O}$, a forgetting signature $\mathcal{F}$
Ensure: An $\mathcal{ALCQ}(\nabla)$-ontology $\mathcal{V}$ with $\text{sig}(\mathcal{V}) \cap \mathcal{F} = \emptyset$

```
function Forget(O, F)
    O_c ← Clausify(O)                 ▷ Convert GCIs to clauses
    V ← Simplify(O_c)                 ▷ Apply simplifications
    D ← ∅                             ▷ Track introduced definers
    for each role name r ∈ F do
        O_n ← Normalize(V, r)         ▷ Apply NR1–NR3
        O_f, D_r ← Forget(O_n, r)     ▷ Apply IR
        D ← D ∪ D_r                   ▷ Collect new definers
        V ← Simplify(O_f)
    end for
    if D ≠ ∅ then
        for each definer Z ∈ D do
            V ← Ackermann(V, Z)
            V ← Simplify(V)
        end for
    end if
     return V
end function
```

6 THE FORGETTING PROCESS

Algorithm 1 presents the entire forgetting iteration process. It takes as input an $\mathcal{ALCQ}$-ontology $\mathcal{O}$ and a forgetting signature $\mathcal{F}$ comprising the names to be forgotten. A notable feature is the flexibility of the elimination sequence, which accommodates any order specified by the users. Throughout the forgetting process, equivalence-preserving standard simplifications are eagerly applied to ensure that the ontology remains in its simplest form at all times.

The final phase of the forgetting process aims to eliminate the introduced definers. In most cases, this can be achieved using a generalization of *Ackermann's Lemma* [29, 58]. This lemma allows a single concept name to be eliminated from an ontology while preserving logical equivalence between the original and resulting ontologies, modulo the interpretations of the eliminated name (i.e., satisfying Conditions (i) and (ii) of Definition 1). Soundness of this generalized lemma follows from [58].

Example 3. *Consider the result $\mathcal{O}'$ obtained from Example 2. Applying Ackermann's Lemma to $\mathcal{O}'$ gives:*

$$\begin{aligned}
\{11.\ & A_2 \sqcup {\geq} 1\nabla.B_2 \\
12.\ & A_1 \sqcup {\leq} 2s.\Big({\geq} 3\nabla.B_1 \sqcap \big(A_3 \sqcup {\geq} 1\nabla.(B_1 \sqcap \neg B_3)\big), \\
&\quad \sqcap \big(A_2 \sqcup A_3 \sqcup {\geq} 1\nabla(B_1 \sqcap B_2) \sqcup {\geq} 2\nabla.(B_1 \sqcup B_2) \sqcap \neg B_3\big) \\
&\quad \sqcap \big(A_2 \sqcup A_3 \sqcup {\geq} 2\nabla(B_1 \sqcap B_2) \sqcup {\geq} 1\nabla.(B_1 \sqcup B_2) \sqcap \neg B_3\big)\Big)
\end{aligned}$$

$\mathcal{V} = \{11, 12\}$ constitutes the result of forgetting r from $\mathcal{O}$.

This approach often successfully eliminates the introduced definers, particularly when one polarity of a definer appears at the surface level of a clause. However, there is no guarantee of completely removing all definers, often in cases involving cyclic dependencies [25]. In such cases, the algorithm risks entering an endless loop, potentially causing the forgetting procedure to never terminate. Consider forgetting r from $\mathcal{O} = \{\geq 1r.\leq 0r.\bot\}$, which exhibits implicit cyclic behavior. Transforming $\mathcal{O}$ into r-NF introduces a fresh definer Z, yielding $\mathcal{O}' = \{\geq 1r.Z, \neg Z \sqcup \leq 0r.\bot\}$. Applying IR produces $\{\geq 1\nabla.Z, \neg Z \sqcup \geq 1\nabla.Z\}$ with explicit cyclic behavior over Z. Attempting to eliminate Z would produce an infinite set $\mathcal{V} = \{\overbrace{\geq 1\nabla.\cdots\geq 1\nabla}^{n}.\top \mid n \geq 1\}$, which cannot be finitely axiomatized in standard DLs.

Lethe and Fame exploit fixpoints [6] to attain finite representation of the forgetting result. However, since fixpoints are not supported by DL reasoners, nor by the OWL API,[2] our method does not adopt the extension of fixpoints as a solution. Instead, it terminates the forgetting process with Z remaining in the result, prioritizing method termination over the pursuit of completeness.

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

Theorem 1 states termination and soundness of our forgetting method, though completeness cannot be achieved due to the inherent undecidability of the forgetting problem [24]. Nevertheless, our empirical evidence shows that this theoretical limitation barely impedes practical performance.

7 EMPIRICAL EVALUATION

We have developed a prototype of our method in Java using OWL API Version 5.1.7.[3] To evaluate its practicality, we compared this prototype against two SOTA systems — Lethe[4] and Fame[5] — based on two large corpora of real-world ontologies. The first corpus, taken from the Oxford ISG Library[6], includes a range of ontologies from various sources. The second corpus, a March 2017 snapshot from BioPortal[7] [37], 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 the statistics of the refined ontologies, where $|N_C|$, $|N_R|$, and $|\text{Onto}|$ denote the average counts of concept names, role names, and GCIs, respectively, within these ontologies.

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 $5000 \leq |\text{Onto}| < 10000$

Table 1: Statistics of Oxford-ISG & BioPortal

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

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 [26, 53, 54, 60]. 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.

The results are shown in Tables 2 and 5. 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 frequent with larger $\mathcal{F}$.

[2] The universal role ∇ is supported by the OWL API.
[3] http://owlcs.github.io/owlapi/
[4] https://lat.inf.tu-dresden.de/ koopmann/LETHE/
[5] https://www.cs.man.ac.uk/ schmidt/sf-fame/
[6] http://krr-nas.cs.ox.ac.uk/ontologies/lib/
[7] https://bioportal.bioontology.org/

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**

Table 3: 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 4: 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**

Table 5: 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 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**

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 ≥-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 ≥-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 particularly essential to contextualize our method's success rate, time consumption (sec.), and memory usage (MB). Thus, we 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 performed 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.

The results are shown in Tables 3, 4, 6, and 7. Although our prototype was tested on $\mathcal{ALCQ}$ ontologies with Lethe and Fame on their $\mathcal{ALC}$ fragments, it consistently achieved a success rate 5% to 10% higher than the two weak forgetting tools. Overall, the three

Table 8: Definers introduced during forgetting (Oxford)

Ontology Code	Lethe (10%)	Proto (10%)	Lethe (30%)	Proto (30%)	Lethe (50%)	Proto (50%)
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

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.

Specifically, the DR column indicates the proportion of ontologies exhibiting cyclic dependencies. Despite our $\mathcal{ALCQ}$ dataset containing a higher incidence of such cycles, our method demonstrates inherent computational advantages in terms of time and memory 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 tended to add definers exponentially, in direct proportion to the number of role restrictions in the ontology. In contrast, our method introduced definers linearly, showcasing a more efficient approach.

Table 8 details 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 Oxford-ISG ontologies. 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 issues like timeouts, memory overflow, or definer retention.

8 CONCLUSION AND FUTURE WORK

In this paper, we have developed the first method for strong forgetting of role names (also concept names via a standard reduction) in a DL with qualified number restrictions. This ability to compute strong forgetting results means that our approach can be applied to any syntactic fragments of $\mathcal{ALCQ}(\nabla)$, ensuring consistent results regardless of source language variations. This versatility is particularly beneficial in problems such as modal correspondence theory [46] and second-order quantifier elimination [18, 45], while addressing contemporary challenges in knowledge management and data integration where efficient ontology processing is essential for scalable AI systems. Since strong forgetting yields stronger results, this approach's application in ontology-based knowledge management tasks, particularly in abduction tasks, offers results richer in information than those obtained using weak forgetting approaches like [11, 27]. This makes it more powerful in identifying true causes or explanations in abduction tasks.

Moving forward, our immediate goal is to extend the method to seamlessly integrate ABoxes. Additionally, we plan to adapt our approach to accommodate several decidable extensions of $\mathcal{ALCQ}$ [2], further expanding its theoretical scope and practical applicability.

GENAI USAGE DISCLOSURE

The authors acknowledge the use of GenAI tools for light editing of author-written text, including automated grammar checks, spell-checking, and slight stylistic refinements. No text was produced entirely by GenAI tools. No GenAI tools were used in the research design, data collection, analysis, code development, or generation of research findings. All technical content and scientific contributions are the original work of the human authors.

REFERENCES

[1] Wilhelm Ackermann. 1935. Untersuchungen úber das Eliminationsproblem der mathematischen Logik. *Math. Ann.* 110, 1 (1935), 390–413.

[2] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. 2017. *An Introduction to Description Logic.* Cambridge University Press.

[3] Chitta Baral and Yan Zhang. 2005. Knowledge updates: Semantics and complexity issues. *Artif. Intell.* 164, 1-2 (2005), 209–243.

[4] Juan Bicarregui, Theodosis Dimitrakos, Dov M. Gabbay, and T. S. E. Maibaum. 2001. Interpolation in Practical Formal Development. *Log. J. IGPL* 9, 2 (2001), 231–244.

[5] George Boole. 1854. *An Investigation of the Laws of Thought.* Walton & Maberly.

[6] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. 1999. Reasoning in Expressive Description Logics with Fixpoints based on Automata on Infinite Trees. In *Proc. IJCAI'99.* Morgan Kaufmann, 84–89.

[7] Willem Conradie, Valentin Goranko, and Dimiter Vakarelov. 2006. Algorithmic correspondence and completeness in modal logic. I. The core algorithm SQEMA. *Log. Methods Comput. Sci.* 2, 1 (2006).

[8] William Craig. 1957. Three Uses of the Herbrand-Gentzen Theorem in Relating Model Theory and Proof Theory. *J. Symb. Log.* 22, 3 (1957), 269–285.

[9] Giovanna D'Agostino and Marco Hollenberg. 2000. Logical Questions Concerning The μ-Calculus: Interpolation, Lyndon and Los-Tarski. *J. Symb. Log.* 65, 1 (2000), 310–332.

[10] Martin Davis and Hilary Putnam. 1960. A Computing Procedure for Quantification Theory. *J. ACM* 7, 3 (1960), 201–215.

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

[12] James P. Delgrande and Kewen Wang. 2015. A Syntax-Independent Approach to Forgetting in Disjunctive Logic Programs. In *Proc. AAAI'15'.* AAAI Press, 1482–1488.

[13] Patrick Doherty, Witold Lukaszewicz, and Andrzej Szalas. 1997. Computing Circumscription Revisited: A Reduction Algorithm. *J. Autom. Reason.* 18, 3 (1997), 297–336.

[14] Thomas Eiter and Kewen Wang. 2008. Semantic forgetting in answer set programming. *Artif. Intell.* 172, 14 (2008), 1644–1672.

[15] Esra Erdem and Paolo Ferraris. 2007. Forgetting Actions in Domain Descriptions. In *Proc. AAAI'07.* AAAI Press, 409–414.

[16] Christian G. Fermüller, Alexander Leitsch, Ullrich Hustadt, and Tanel Tammet. 2001. Resolution Decision Procedures. In *Handbook of Automated Reasoning (in 2 volumes)*, John Alan Robinson and Andrei Voronkov (Eds.). Elsevier and MIT Press, 1791–1849.

[17] Dov M. Gabbay and Hans Jürgen Ohlbach. 1992. Quantifier Elimination in Second-Order Predicate Logic. *South African Computer Journal* 7 (1992), 35–43.

[18] Dov M. Gabbay, Renate A. Schmidt, and Andrzej Szalas. 2008. *Second-Order Quantifier Elimination - Foundations, Computational Aspects and Applications.* Studies in logic : Mathematical logic and foundations, Vol. 12. College Publications.

[19] Silvio Ghilardi, Carsten Lutz, and Frank Wolter. 2006. Did I Damage My Ontology? A Case for Conservative Extensions in Description Logics. In *Proc. KR'06.* AAAI Press, 187–197.

[20] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. 2008. Modular Reuse of Ontologies: Theory and Practice. *J. Artif. Intell. Res.* 31 (2008), 273–318.

[21] A. Herzig and J. Mengin. 2008. Uniform Interpolation by Resolution in Modal Logic. In *Proc. JELIA'08 (LNCS, Vol. 5293).* Springer, 219–231.

[22] Michel C. A. Klein and Dieter Fensel. 2001. Ontology versioning on the Semantic Web. In *Proc. SWWS'01.* 75–91.

[23] Boris Konev, Michel Ludwig, Dirk Walther, and Frank Wolter. 2012. The Logical Difference for the Lightweight Description Logic $\mathcal{EL}$. *J. Artif. Intell. Res.* 44 (2012), 633–708.

[24] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. 2013. Model-theoretic inseparability and modularity of description logic ontologies. *Artif. Intell.* 203 (2013), 66–103.

[25] Boris Konev, Dirk Walther, and Frank Wolter. 2009. Forgetting and Uniform Interpolation in Large-Scale Description Logic Terminologies. In *Proc. IJCAI'09.* IJCAI/AAAI Press, 830–835.

[26] Patrick Koopmann. 2020. LETHE: Forgetting and Uniform Interpolation for Expressive Description Logics. *Künstliche Intell.* 34, 3 (2020), 381–387.

[27] Patrick Koopmann, Warren Del-Pinto, Sophie Tourret, and Renate A. Schmidt. 2020. Signature-Based Abduction for Expressive Description Logics. In *Proc. KR'20.* 592–602.

[28] Patrick Koopmann and Renate A. Schmidt. 2013. Uniform Interpolation of $\mathcal{ALC}$-Ontologies Using Fixpoints. In *Proc. FroCoS'13 (LNCS, Vol. 8152).* Springer, 87–102.

[29] Patrick Koopmann and Renate A. Schmidt. 2014. Count and Forget: Uniform Interpolation of $\mathcal{SHQ}$-Ontologies. In *Proc. IJCAR'14 (LNCS, Vol. 8562).* Springer, 434–448.

[30] Patrick Lambrix and He Tan. 2008. Ontology Alignment and Merging. In *Anatomy Ontologies for Bioinformatics, Principles and Practice.* Computational Biology, Vol. 6. Springer, 133–149.

[31] Jérôme Lang and Pierre Marquis. 2010. Reasoning under inconsistency: A forgetting-based approach. *Artif. Intell.* 174, 12-13 (2010), 799–823.

[32] Paolo Liberatore and Marco Schaerf. 1998. Arbitration (or How to Merge Knowledge Bases). *IEEE Trans. Knowl. Data Eng.* 10, 1 (1998), 76–90.

[33] Fangzhen Lin and Ray Reiter. 1994. Forget It!. In *Proc. AAAI Fall Symposium on Relevance.* AAAI Press, 154–159.

[34] Carsten Lutz, Inanç Seylan, and Frank Wolter. 2012. An Automata-Theoretic Approach to Uniform Interpolation and Approximation in the Description Logic $\mathcal{EL}$. In *Proc. KR'12.* AAAI Press, 286–296.

[35] Carsten Lutz and Frank Wolter. 2010. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *J. Symb. Comput.* 45, 2 (2010), 194–228.

[36] Carsten Lutz and Frank Wolter. 2011. Foundations for Uniform Interpolation and Forgetting in Expressive Description Logics. In *Proc. IJCAI'11.* IJCAI/AAAI Press, 989–995.

[37] Nicolas Matentzoglu and Bijan Parsia. 2017. BioPortal Snapshot 30.03.2017. https://doi.org/10.5281/zenodo.439510

[38] Bernhard Nebel, Yannis Dimopoulos, and Jana Koehler. 1997. Ignoring Irrelevant Facts and Operators in Plan Generation. In *Proc. ECP'97 (LNCS, Vol. 1348).* Springer, 338–350.

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

[40] Nadeschda Nikitina, Sebastian Rudolph, and Birte Glimm. 2012. Interactive ontology revision. *J. Web Semant.* 12 (2012), 118–130.

[41] Peter Plessers and Olga De Troyer. 2005. Ontology Change Detection Using a Version Log. In *Proc. ISWC'05*, Vol. 3729. Springer, 578–592.

[42] Márcio Moretto Ribeiro and Renata Wassermann. 2009. Base Revision for Ontology Debugging. *J. Log. Comput.* 19, 5 (2009), 721–743.

[43] John Alan Robinson. 1965. A Machine-Oriented Logic Based on the Resolution Principle. *J. ACM* 12, 1 (1965), 23–41.

[44] Stefan Schlobach and Ronald Cornet. 2003. Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. In *Proc. IJCAI'03.* Morgan Kaufmann, 355–362.

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

[46] Andrzej Szalas. 1993. On the Correspondence between Modal and Classical Logic: An Automated Approach. *J. Log. Comput.* 3, 6 (1993), 605–620.

[47] Andrzej Szalas. 2006. Second-order reasoning in description logics. *J. Appl. Non Class. Logics* 16, 3-4 (2006), 517–530.

[48] Nicolas Troquard, Roberto Confalonieri, Pietro Galliani, Rafael Peñaloza, Daniele Porello, and Oliver Kutz. 2018. Repairing Ontologies via Axiom Weakening. In *Proc. AAAI'18.* AAAI Press, 1981–1988.

[49] Albert Visser. 1996. *Bisimulations, Model Descriptions and Propositional Quantifiers.* Utrecht University.

[50] Kewen Wang, Grigoris Antoniou, Rodney Topor, and Abdul Sattar. 2005. Merging and Aligning Ontologies in dl-Programs. In *Proc. RuleML'05 (LNCS, Vol. 3791).* Springer, 160–171.

[51] Yisong Wang, Kewen Wang, and Mingyi Zhang. 2013. Forgetting for Answer Set Programs Revisited. In *Proc. IJCAI'13'.* IJCAI/AAAI, 1162–1168.

[52] Zhe Wang, Kewen Wang, Rodney W. Topor, and Jeff Z. Pan. 2010. Forgetting for knowledge bases in DL-Lite. *Ann. Math. Artif. Intell.* 58, 1-2 (2010), 117–151.

[53] Xuan Wu, Wenxing Deng, Chang Lu, Hao Feng, and Yizheng Zhao. 2020. UI-FAME: A High-Performance Forgetting System for Creating Views of Ontologies. In *Proc. CIKM'20.* ACM, 3473–3476.

[54] Hui Yang, Patrick Koopmann, Yue Ma, and Nicole Bidoit. 2023. Efficient Computation of General Modules for $\mathcal{ALC}$ Ontologies. In *Proc. IJCAI'23.* ijcai.org, 3356–3364.

[55] Yan Zhang and Norman Y. Foo. 2006. Solving logic program conflict through strong and weak forgettings. *Artif. Intell.* 170, 8-9 (2006), 739–778.

[56] Yan Zhang and Yi Zhou. 2010. Forgetting Revisited. In *Proc. KR'10.* AAAI Press.

[57] Yizheng Zhao, Ghadah Alghamdi, Renate A. Schmidt, Hao Feng, Giorgos Stoilos, Damir Juric, and Mohammad Khodadadi. 2019. Tracking Logical Difference in Large-Scale Ontologies: A Forgetting-Based Approach. In *Proc. AAAI'19.* AAAI Press, 3116–3124.

[58] Yizheng Zhao and Renate A. Schmidt. 2016. Forgetting Concept and Role Symbols in $\mathcal{ALCOIH}\mu^+(\triangledown,\sqcap)$-Ontologies. In *Proc. IJCAI'16.* IJCAI/AAAI Press, 1345–1353.

[59] Yizheng Zhao and Renate A. Schmidt. 2017. Role Forgetting for $\mathcal{ALCOQH}(\triangledown)$-Ontologies Using an Ackermann-Based Approach. In *Proc.IJCAI 2017, Melbourne, Australia, August 19-25, 2017.* ijcai.org, 1354–1361.

[60] Yizheng Zhao and Renate A. Schmidt. 2018. FAME: An Automated Tool for Semantic Forgetting in Expressive Description Logics. In *Proc. IJCAR'18 (LNCS, Vol. 10900).* Springer, 19–27.

[61] Yizheng Zhao and Renate A. Schmidt. 2018. On Concept Forgetting in Description Logics with Qualified Number Restrictions. In *Proc. IJCAI'18.* ijcai.org, 1984–1990.

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 normalization rules NR1–NR3. In addition, the size of the resulting ontology $\mathcal{O}'$ is linear in the size of $\mathcal{O}$.*

Proof. When applying rules NR1–NR3 to an r-clause X, either $\mathsf{fq}(r, X)$ or $\mathsf{rd}(r, X)$ decreases by at least 1. This ensures finite computation of the r-NF of X. Since each definer introduction corresponds to a new clause with a number restriction, the number of definers and new clauses is bounded by $O(n)$, where n is the number of number restrictions in $\mathcal{O}$. □

Lemma 2. *Let $\mathcal{O}$ be an $\mathcal{ALCQ}(\nabla)$-ontology, and $\mathcal{O}'$ the normalized one obtained from $\mathcal{O}$ by applying the rules NR1–NR3. Let $sig_D(\mathcal{O}')$ 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 $\mathcal{I} \sim_{sig_D(\mathcal{O}')} \mathcal{I}'$.

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 and NR3 can be proved in a similar way.

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.[8] 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 IR rule, 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 **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 $a \notin C_i^{\mathcal{I}'}, a \notin E_{j_1}^{\mathcal{I}'}, \cdots, a \notin E_{j_k}^{\mathcal{I}'}$, 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 **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

[8] 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 $\mathcal{I}'$.

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:

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

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:

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

Building on our initial assumption, where a is a domain element in $(\geq x_i \mathrm{r}.D_i)^{\mathcal{I}'}$, 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 $a \in (\leq y_j \mathrm{r}.F_j)^{\mathcal{I}'}$. 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 $a \in (C_{k+1} \sqcup \geq x_{k+1} \mathrm{r}.D_{k+1})^{\mathcal{I}'}, \cdots, a \in (C_m \sqcup \geq x_m \mathrm{r}.D_m)^{\mathcal{I}'}$. 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 $F_j^{\mathcal{I}'}$.

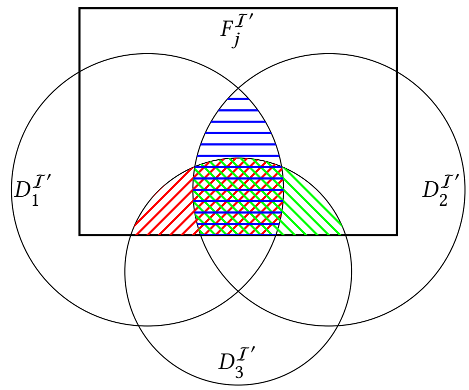

Figure 2: Relations among $D_1^{\mathcal{I}'}$, $D_2^{\mathcal{I}'}$, $D_3^{\mathcal{I}'}$, and $F_j^{\mathcal{I}'}$

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:

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

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 w.r.t. 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 w.r.t. r.

∎

Table 9: Definers introduced during forgetting (BioPortal)

Ontology	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

LEMMA 4. *Let x and y denote the numbers of r-clauses of the form $C \sqcup \geq mr.D$ and $C \sqcup \leq nr.D$, respectively, and let z be the maximum qualified number in clauses of the form $C \sqcup \geq mr.D$. The number of derived clauses is bounded by $O(y \cdot 2^x \cdot z^{2^x})$.*

PROOF. We define literals $Qt_1r.D$ and $Qt_2r.D$ 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 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 $\mathcal{I}'$, $\mathcal{I}' \models \mathcal{V}$ iff there is an interpretation $\mathcal{I} \sim_{\mathcal{F}} \mathcal{I}'$ such that $\mathcal{I} \models \mathcal{O}$.*
(ii) *If $\mathcal{V}$ contains definers, for any interpretation $\mathcal{I}'$, $\mathcal{I}' \models \mathcal{V}$ iff there is an interpretation $\mathcal{I} \sim_{\mathcal{F} \cup sig_D(\mathcal{V})} \mathcal{I}'$ such that $\mathcal{I} \models \mathcal{O}$.*

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

B ADDITIONAL EXPERIMENTAL RESULTS

Table 9 presents 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 BioPortal ontologies. Following the same experimental protocol as the Oxford-ISG evaluation, each run was repeated 100 times, and only results from a single representative run are shown to manage the statistical presentation. The results demonstrate consistent patterns with the Oxford-ISG findings, where our prototype consistently produces fewer definers across all forgetting percentages and ontology instances.