Semantic Difference Detection in SNOMED CT

Yizheng Zhao
National Key Laboratory for Novel Software Technology, Nanjing University
China
zhaoyz@nju.edu.cn

Renate A. Schmidt
Department of Computer Science, The University of Manchester
United Kingdom
renate.schmidt@manchester.ac.uk

Yongsheng Gao
SNOMED International, London
United Kingdom
yga@snomed.org

Abstract

SNOMED CT is a comprehensive clinical terminology that underpins numerous web-based healthcare information systems worldwide, enabling consistent data capture, analytics, and decision support across the World Wide Web. The SNOMED CT browser and its web services,[1] serving over 5 million healthcare professionals globally, require frequent updates to accommodate domain changes and user requirements. These updates, delivered through web-based distribution platforms, need careful semantic validation to maintain data and service quality.

This paper presents a novel automated approach for detecting semantic differences between SNOMED CT releases, implemented as a Quality Assurance solution at SNOMED International since April 2023. Our solution serves terminologists, debuggers, and developers who maintain SNOMED CT and its applications. Unlike existing approaches that generate large, unwieldy difference sets, our method produces concise, focused semantic differences for user-specified signatures using an important ontology engineering technique called uniform interpolation, revealing inferred differences that existing alternatives are unable to reveal. Deployed as part of SNOMED's content management platform, it has processed over 100,000 semantic difference queries across 20 major releases, reducing validation time by 60%. Real-world deployment demonstrates the approach's substantial effectiveness and impact on maintaining semantic consistency across their web services ecosystem.

CCS Concepts

• **Theory of computation → Automated reasoning**; **Description logics**; • **Computing methodologies → Ontology engineering**.

Keywords

SNOMED CT, Ontologies; Description Logic; Semantic Difference; Uniform Interpolation; Forgetting

[1] https://www.implementation.snomed.org/browsers

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

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Yizheng Zhao, Renate A. Schmidt, and Yongsheng Gao. 2018. Semantic Difference Detection in SNOMED CT. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

1.1 Background & Motivation

Ontologies [38] fix a vocabulary of *terms* (known as the *signature* of the ontologies) specific to a subject domain, and define the meaning of these terms by logical statements. These terms comprise *classes* (sets of instances sharing common properties) and *relationships* (binary relations between classes and their instances). Through Tarski-style set-theoretic interpretations [35], ontologies gain formal semantics that enable machine processing and automated reasoning. Modern ontologies are expressed in the *Web Ontology Language* (OWL) [1], which is based on *Description Logics* (DLs)[2] — a family of knowledge representation formalisms where classes and relationships are termed *concepts* and *roles*, respectively.

SNOMED CT [37], owned and maintained by SNOMED International,[2] serves as a comprehensive clinical terminology supporting healthcare delivery worldwide. Implemented in OWL 2[3], it comprises over 350,000 clinical concepts, along with over 350,000 logical axioms defining their meanings. Several other major healthcare ontologies complement SNOMED CT: *NCIt* [36] provides reference terminology for cancer-related diseases and findings; the *Gene Ontology* (GO) [39] enables consistent description and querying of gene products across databases; and *GALEN* [33] offers comprehensive terminology for surgical procedures. These ontologies have gained widespread adoption by healthcare organizations, facilitating consistent coding of concepts and reproducible transmission of data between healthcare information systems.

SNOMED CT sustains its currency through regular updates to the International edition, with monthly releases incorporating additions of new concepts, retirement of obsolete terms, and refinements to existing definitions. The scope of these updates is substantial: between January 2015 and January 2016, SNOMED CT underwent approximately 192,000 content changes [46]. Even minor definitional modifications can have far-reaching implications for concept hierarchies and relationships, making their semantic impact challenging to fully assess. Therefore, maintaining their quality becomes crucial. Quality Assurance (QA) in ontology evolution focuses on two key aspects:

[2] https://www.snomed.org/
[3] Recent releases of SNOMED CT are expressed in $\mathcal{ELIH}$, which extends its previous expressivity by role inverse $\mathcal{I}$.

(1) Structural changes: These involve modifications to the ontology's structure, captured through the notion of *structural difference* [31]. This aspect treats ontologies as structured objects and characterizes changes through basic operations such as concept addition or axiom deletion.
(2) Semantic changes: These reflect differences in the meaning of terms between ontology versions, including both explicit and implicit information.

While tools like Protégé [11, 29], SWOOP [12], OntoView [13], PROMPTDIFF [31], and OBO-Edit [6] effectively track structural changes, the field lacks robust methodologies and tools for detecting semantic changes. Two major challenges complicate this task:

(1) The difficulty of identifying implicit information: Semantic changes include not just explicit modifications but also their logical implications, which may not be immediately visible.
(2) Signature inconsistency: When new releases of ontology introduce new terms or remove existing ones, comparing their semantic content becomes complicated. Consider:

$$\text{Version 1} = \{\text{Father} \equiv \text{Human} \sqcap \exists\text{hasSex.Male} \sqcap \exists\text{hasChild.}\top\}$$
$$\text{Version 2} = \{\text{Father} \equiv \text{Parent} \sqcap \exists\text{hasSex.Male}, \quad \text{Parent} \equiv \text{Human} \sqcap \exists\text{hasChild.}\top\},$$

While both axioms in Version 2 appear as semantic changes, they merely represent a reorganization using the new term 'Parent' rather than a genuine change in meaning.

To address these two challenges, we formally define the notion of semantic changes between ontologies O_1 and O_2 as axioms entailed by one but not the other [14]. Computing these differences requires:

(1) Identifying all entailed axioms (both explicit and implicit)
(2) Unifying the signatures of both versions while preserving their meaning
(3) Isolating the genuine semantic differences between versions

1.2 Related Work

Konev et al. [14] proposed using *Uniform Interpolation* (UI) [5, 9, 27, 32, 40] to unify ontology signatures and compute finite representations of semantic differences. UI is a non-standard reasoning service that aims to project an ontology down to its sub-signature — given an ontology using a specific signature, and a subset Σ of 'relevant names' of that signature, compute an ontology, namely a *uniform interpolant*, that uses only the relevant names while preserving the meanings (the semantics) of the relevant names in the absence of the other names. This process often requires deriving a large number of new axioms to maintain semantic preservation, as original axioms may lose meaning when definitional terms are excluded from the specified signature. Moreover, computing uniform interpolants is computationally expensive and may not terminate [27]. Despite these challenges, UI enables the computation of finite representations of the logical difference by focusing on strongest entailments rather than exhaustively computing all logical entailments of one version not entailed by another version. The complete set of logical entailments can in principle be derived from the deductive closure of these strongest entailments.[4]

[4] In the context of logical statements, a set O of statements is *deductively closed* if it contains every statement ϕ that can be logically deduced from O, i.e., $O \models \phi$ implies $\phi \in O$. The deductive closure of O is its smallest superset that is deductively closed.

UI poses significant computational challenges: Uniform interpolants may not exist for the DL $\mathcal{ELH}$ or $\mathcal{ALC}$ [15], with existence checking being ExpTime-complete for $\mathcal{ELH}$ [25] and 2ExpTime-complete for $\mathcal{ALC}$ [27]. Size complexity is also substantial: uniform interpolants can grow exponentially in DL-Lite [17] and triple exponentially in $\mathcal{ALC}$ [27]. As shown by Nikitina and Rudolph [30], finite uniform interpolants may not exist, and when they do, their minimal size can be triple exponential relative to the input ontology. Even for the less expressive propositional logic, polynomial-size uniform interpolants are not guaranteed [22].

Several methods have been developed to implement UI. Nui [16] uses a recursive algorithm for computing instance Σ-interpolants, but its effectiveness diminishes with larger interpolation signatures. When tested on SNOMED CT (January 2005), success rates dropped from 93% for $|\Sigma| = 2000$ to 59.5% for $|\Sigma| = 5000$, making it impractical for modern ontologies where $|\Sigma|$ often exceeds 150,000. Other tools like Lethe [20], UI-Fame [42, 46], and the method developed by [24, 43, 44] employ forgetting-based procedures. However, these methods face at least one of the following key limitations:

- They are designed for more expressive DLs than the $\mathcal{ELIH}$ commonly used in large-scale ontologies, producing interpolants with constructs outside $\mathcal{ELIH}$'s language.
- They struggle with computational efficiency when processing industrial-scale ontologies [4, 19].

1.3 Our Contribution

In this paper, we present an enhanced method for computing uniform interpolants of $\mathcal{ELIH}$-ontologies, extending previous work of [23] to accommodate SNOMED CT's recent adoption of inverse roles and addressing the need for efficient semantic difference detection. Our method is based on an optimized forgetting procedure that systematically eliminates non-specified names from ontology O (the names in $\text{sig}(O) \setminus \Sigma$) while preserving the semantic integrity within the interpolation signature Σ. This method is terminating and sound, offering a principled solution to tackling the computational challenges inherent in UI and forgetting.

Extensive empirical evaluation using our prototype implementation demonstrates superb performance, achieving 100% average success rates across two comprehensive ontology corpora created from Oxford ISG and NCBO BioPortal — the industry benchmark standards. When compared to Lethe, the current state-of-the-art UI system, our method shows marked superiority in success rates and substantial improvements in both computation time and memory consumption. A pivotal case study analyzing distinct releases of SNOMED CT, currently the largest healthcare ontology, demonstrates our method's ability to overcome UI's traditional sensitivity to ontology size. This breakthrough enables the identification of modeling changes across massive ontological structures, providing QA teams with a powerful tool for monitoring ontology evolution and streamlining data and service migration processes.

A **long version** of this paper, which includes all omitted proofs, additional illustrative examples, comprehensive experimental results, and the source code for our prototype implementation along with the test datasets, is available for review at https://github.com/anonymous-ai-researcher/www2025-industry.

2 Preliminaries

In this section, we introduce the underlying language $\mathcal{ELIH}$ and illustrate some basic notions used throughout this paper. Based on them, we further formalize the notions of *Semantic Difference* and *Uniform Interpolation.*

2.1 The Description Logic $\mathcal{ELIH}$

Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be pairwise disjoint, countably infinite sets of *concept names* and *role names*, respectively. *Roles* in $\mathcal{ELIH}$ can be a role name $\mathsf{r} \in \mathsf{N_R}$ or the inverse r^- of r. *Concepts* in $\mathcal{ELIH}$ are inductively defined by the following syntax rule:

$$C, D \longrightarrow \top \mid A \mid C \sqcap D \mid \exists R.C$$

where $A \in \mathsf{N_C}$, C, D range over concepts, and R over roles. If R is a role, we define the inverse $\mathsf{Inv}(R)$ of R by $\mathsf{Inv}(r) = r^-$ and $\mathsf{Inv}(r^-) = r$, for all $r \in \mathsf{N_R}$. We distinguish between *atomic* concepts ($A \in \mathsf{N_C}$) and *complex* concepts (all others). An *$\mathcal{ELIH}$-ontology $\mathcal{O}$* comprises a finite set of axioms in the following forms;

- *General concept inclusions* (GCIs): $C \sqsubseteq D$
- *Role inclusions*: $R \sqsubseteq S$

where C, D are general concepts, and R, S are general roles.

We denote concept equivalence $C \equiv D$ as shorthand for the pair of GCIs $C \sqsubseteq D$ and $D \sqsubseteq C$ (analogously for role equivalence $R \equiv S$). Throughout this paper, we assume w.l.o.g. that an $\mathcal{ELIH}$-ontology contains only GCIs and role inclusions. We define an occurrence of a (concept/role) name as *positive* in an axiom α if it occurs at the right-hand side of α and as *negative* if on the left-hand side.

The semantics of $\mathcal{ELIH}$ is defined in terms of an *interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a nonempty set of elements, known as the *domain of the interpretation*, and $\cdot^{\mathcal{I}}$ denotes the *interpretation function* that maps every concept name $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, 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 general concepts as follows:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad (C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y.(x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$$

$$(R^-)^{\mathcal{I}} = \{(y,x) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (x,y) \in R^{\mathcal{I}}\}$$

and to axioms as follows:

- A GCI $C \sqsubseteq D$ is *true* in $\mathcal{I}$ (written $\mathcal{I} \models C \sqsubseteq D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- A role inclusion $R \sqsubseteq S$ is *true* in $\mathcal{I}$ (written $\mathcal{I} \models R \sqsubseteq S$) iff $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$

An interpretation $\mathcal{I}$ is a *model* of ontology $\mathcal{O}$ (written $\mathcal{I} \models \mathcal{O}$) iff every axiom in $\mathcal{O}$ is true in $\mathcal{I}$. We say that:

- An axiom α is *entailed* by $\mathcal{O}$ (written $\mathcal{O} \models \alpha$) iff α is true in every model of $\mathcal{O}$
- An ontology $\mathcal{O}_1$ is *entailed* by $\mathcal{O}_2$ (written $\mathcal{O}_2 \models \mathcal{O}_1$) iff every model of $\mathcal{O}_2$ is also a model of $\mathcal{O}_1$

A *signature* $\Sigma \subseteq \mathsf{N_C} \cup \mathsf{N_R}$ is a finite set of concept and role names. For any syntactic object X (concept, GCI, axiom, or ontology), we define:

- $\mathsf{sig_C}(X)$: the set of concept names in X
- $\mathsf{sig_R}(X)$: the set of role names in X
- $\mathsf{sig}(X)$: $\mathsf{sig_C}(X) \cup \mathsf{sig_R}(X)$

For any signature Σ, we say that a syntactic object X is a Σ-object if $\mathsf{sig}(X) \subseteq \Sigma$ (e.g., Σ-concept, Σ-GCI, Σ-axiom).

2.2 UI-Based Semantic Difference

The notion of semantic difference requires a unified signature basis for comparison between ontology versions. This unification can be achieved through Uniform Interpolation (UI), which we formally define here for $\mathcal{ELIH}$-ontologies.

Definition 1 (**Uniform Interpolation for $\mathcal{ELIH}$**). *Given an $\mathcal{ELIH}$-ontology $\mathcal{O}$ and an* interpolation signature $\Sigma \subseteq sig(\mathcal{O})$, *an $\mathcal{ELIH}$-ontology $\mathcal{V}$ is a* uniform Σ-interpolant *of $\mathcal{O}$ if it satisfies:*

(1) Signature containment: $sig(\mathcal{V}) \subseteq \Sigma$
(2) Logical equivalence: For every $\mathcal{ELIH}$-axiom α with $sig(\alpha) \subseteq \Sigma$, $\mathcal{V} \models \alpha$ iff $\mathcal{O} \models \alpha$

Definition 1 characterizes uniform interpolants through two key conditions:

(1) Condition (1) constrains the signature to $\mathsf{sig}(\mathcal{V}) \subseteq \Sigma$
(2) Condition (2) ensures semantic integrity — $\mathcal{V}$ and $\mathcal{O}$ maintain logical equivalence w.r.t. Σ

This makes $\mathcal{V}$ the *strongest* Σ*-entailment of* $\mathcal{O}$, meaning:

(1) $\mathsf{sig}(\mathcal{V}) \subseteq \Sigma$
(2) For any ontology $\mathcal{V}'$ with $\mathcal{O} \models \mathcal{V}'$ and $\mathsf{sig}(\mathcal{V}') \subseteq \Sigma$, we have $\mathcal{V} \models \mathcal{V}'$

A key property of uniform interpolants is their uniqueness up to equivalence: any two uniform Σ-interpolants $\mathcal{V}_1$ and $\mathcal{V}_2$ of $\mathcal{O}$ are semantically equivalent.

Let $\mathsf{Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$ denote the semantic difference between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ w.r.t. the signature $\Sigma \subseteq \mathsf{sig}(\mathcal{O}_1) \cap \mathsf{sig}(\mathcal{O}_2)$. Each axiom in this set constitutes a *witness* of $\mathsf{Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$, representing a Σ-entailment of $\mathcal{O}_2$ not entailed by $\mathcal{O}_1$. While an empty difference ($\mathsf{Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2) = \emptyset$) indicates semantic equivalence over Σ, a non-empty difference typically yields an infinite witness set [24]. To address this challenge, we introduce the *UI-based semantic difference*, which provides a finite representation of a witness set while generalizing the standard notion.

Definition 2 (**UI-based Semantic Difference for $\mathcal{ELIH}$**). *For $\mathcal{ELIH}$-ontologies $\mathcal{O}_1, \mathcal{O}_2$ and a signature $\Sigma \subseteq sig(\mathcal{O}_1) \cap sig(\mathcal{O}_2)$ that intersects both, the* UI-based semantic difference *UI-Diff$_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$ comprises all $\mathcal{ELIH}$-axioms α satisfying:*

(1) $sig(\alpha) \subseteq \Sigma$
(2) $\alpha \in \mathcal{V}_2$, where $\mathcal{V}_2$ is a uniform Σ-interpolant of $\mathcal{O}_2$
(3) $\mathcal{O}_1 \not\models \alpha$

Each such axiom α is termed a UI-witness.

This definition establishes two key properties:

(1) Every UI-witness is also a witness, i.e., $\mathsf{UI\text{-}Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2) \subseteq \mathsf{Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$, as $\mathcal{V}_2$ contains only entailments of $\mathcal{O}_2$;
(2) $\mathsf{UI\text{-}Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$ is a finite representation of $\mathsf{Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$, as all witnesses can in principle be derived from the deductive closure of $\mathcal{V}_2$.

For any $\mathcal{ELIH}$-ontologies $\mathcal{O}_1, \mathcal{O}2$, the computation of $\mathsf{UI\text{-}Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$ follows a two-step process:

Step 1: Compute uniform Σ-interpolant $\mathcal{V}_2$ of $\mathcal{O}_2$ using a UI tool

Step 2: Identify axioms $\alpha \in \mathcal{V}_2$ that are not entailed by $\mathcal{O}_1$ using a DL reasoner

We notice that $\text{UI-Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$ captures both *information gain* from $\mathcal{O}_1$ to $\mathcal{O}_2$ and *information loss* from $\mathcal{O}_2$ to $\mathcal{O}_1$ over Σ.

2.3 Duality of UI and Forgetting

Our method computes uniform interpolants via forgetting — a process that eliminates non-Σ names from an ontology while preserving semantic integrity over the remaining names [19, 21, 24, 41].

Definition 3 (**Forgetting for** $\mathcal{ELIH}$). *Given an* $\mathcal{ELIH}$ *ontology* $\mathcal{O}$ *and a* forgetting signature $\mathcal{F} \subseteq sig(\mathcal{O})$, *an* $\mathcal{ELIH}$*-ontology* $\mathcal{V}$ *is a* result of forgetting $\mathcal{F}$ from $\mathcal{O}$ *if:*

(1) $sig(\mathcal{V}) \subseteq sig(\mathcal{O}) \setminus \mathcal{F}$

(2) For any $\mathcal{ELIH}$*-axiom* α *with* $sig(\alpha) \subseteq sig(\mathcal{O}) \setminus \mathcal{F}$, $\mathcal{V} \models \alpha$ *iff* $\mathcal{O} \models \alpha$

UI and forgetting exhibit a duality: $\mathcal{V}$ is a uniform Σ-interpolant of $\mathcal{O}$ precisely when $\mathcal{V}$ results from forgetting $\mathcal{F} = \text{sig}(\mathcal{O}) \setminus \Sigma$ from $\mathcal{O}$. For $\text{UI-Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$, we have $\mathcal{F} = \text{sig}(\mathcal{O}_2) \setminus \text{sig}(\mathcal{O}_1)$.

3 Normalization of $\mathcal{ELIH}$-Ontologies

Given an $\mathcal{ELIH}$-ontology $\mathcal{O}$ and a forgetting signature $\mathcal{F}$,[5] our method computes the forgetting result through iterative elimination of individual names from $\mathcal{F}$. This elimination process employs two independent calculi working on specialized normal forms tailored for $\mathcal{ELIH}$-ontologies.

3.1 A-Normal Form (A-NF)

Definition 4 (A-**Normal Form**). *We say that a GCI is in* A-normal form *(or* A-NF*) if it has one of the following forms, where (i)* R, S *are general roles, and (ii)* C, D, E, F, *and* G *are general concepts not containing* A. *An* $\mathcal{ELIH}$*-ontology* $\mathcal{O}$ *is in* A-NF *if every* A*-GCI in* $\mathcal{O}$ *is in A-NF.*

	A-NF		***A-NF***
I	$C \sqsubseteq \mathsf{A}$	*III*	$\mathsf{A} \sqcap E \sqsubseteq F$
II	$C \sqsubseteq \exists R.(\mathsf{A} \sqcap D)$	*IV*	$\exists S.(\mathsf{A} \sqcap E) \sqcap F \sqsubseteq G$

An $\mathcal{ELIH}$-ontology $\mathcal{O}$ can be transformed into A-NF through exhaustive application of normalization rules (NR1 – NR5) to each A-GCI in $\mathcal{O}$ not already in A-NF. Let X, Y, Y_1, and Y_2 denote $\mathcal{ELIH}$-concepts, and Z a 'definer' — a newly introduced concept serving as an abbreviation for complex concepts during normalization.

- NR1 For each instance of $X \sqsubseteq Y_1 \sqcap Y_2$, if either Y_1 or Y_2 contains A, replace with $X \sqsubseteq Y_1$ and $X \sqsubseteq Y_2$;
- NR2 For each instance of $X \sqsubseteq Y$ with multiple occurrences of A, if X contains a concept of the form $\exists R.C$ at surface level with $\mathsf{A} \in \text{sig}(C)$, replace C with a new definer $Z \in \mathsf{N_C}$, and add $C \sqsubseteq Z$ to $\mathcal{O}$;
- NR3 For each instance of $X \sqsubseteq Y$ with multiple occurrences of A, if Y contains a concept of the form $\exists R.C$ at surface level with $\mathsf{A} \in \text{sig}(C)$, replace C with a new definer $Z \in \mathsf{N_C}$, and add $Z \sqsubseteq C$ to $\mathcal{O}$;
- NR4 For each instance of $X \sqsubseteq Y$ with a single occurrence of A, if X contains a concept of the form $\exists R.C$ at surface level with $\mathsf{A} \in \text{sig}(C)$ but $C \neq \mathsf{A} \sqcap E$ (per A-NF V or VI), replace C with a new definer $Z \in \mathsf{N_C}$, and add $C \sqsubseteq Z$ to $\mathcal{O}$;
- NR5 For each instance of $X \sqsubseteq Y$ with a single occurrence of A, if Y contains a concept of the form $\exists R.C$ at surface level with $\mathsf{A} \in \text{sig}(C)$ but $C \neq \mathsf{A} \sqcap D$ (per A-NF II or III), replace C with a new definer $Z \in \mathsf{N_C}$, and add $Z \sqsubseteq C$ to $\mathcal{O}$.

A normalization method must satisfy three essential properties to ensure its usability:

- *Soundness*: The normalized ontology $\mathcal{O}'$ must preserve all logical consequences of the original ontology $\mathcal{O}$ within their shared signature $\text{sig}(\mathcal{O})$, satisfying Condition (ii) of Definition 1.
- *Completeness*: The method must successfully transform any $\mathcal{ELIH}$-ontology into A-NF.
- *Efficiency*: The normalization process must terminate and complete within polynomial time.

Lemmas 1 and 2 establish the method's soundness, while Lemma 3 confirms its termination and completeness.

Lemma 1. *Let* $\mathcal{O}$ *be an* $\mathcal{ELIH}$*-ontology and* $\mathcal{O}'$ *the normalized one obtained from* $\mathcal{O}$ *using NR1 – NR5. Then we have*

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

for any $\mathcal{ELIH}$*-axiom* α *with* $sig(\alpha) \subseteq sig(\mathcal{O})$.

3.2 r-Normal Form (r-NF)

Definition 5 (r-**Normal Form**). *We say that a GCI or a role inclusion is in* r-normal form *(or* r-NF*) if it has one of the following forms, where (i)* $r \in \mathsf{N_R}$, *(ii)* S, T *are general roles, and (iii)* C, D, E, F, *and* G *are general concepts not containing* r. *An* $\mathcal{ELIH}$*-ontology* $\mathcal{O}$ *is in* r-NF *if every* r*-axiom in* $\mathcal{O}$ *is in r-NF.*

	r-NF		***r-NF***
I	$C \sqsubseteq \exists r.D$	*V*	$E \sqcap \exists r.F \sqsubseteq G$
II	$C \sqsubseteq \exists r^-.D$	*VI*	$E \sqcap \exists r^-.F \sqsubseteq G$
III	$S \sqsubseteq r$	*VII*	$r \sqsubseteq T$
IV	$S \sqsubseteq r^-$	*VIII*	$r^- \sqsubseteq T$

Transformating an $\mathcal{ELIH}$-ontology $\mathcal{O}$ into r-NF can be achieved through an adaptation of the A-NF transformation procedure.

Lemma 2. *Let* $\mathcal{O}$ *be an* $\mathcal{ELIH}$*-ontology and* $\mathcal{O}'$ *the normalized one obtained from* $\mathcal{O}$ *using the slightly adjusted normalization rules for r-NF transformation. Then we have*

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

for any $\mathcal{ELIH}$*-axiom* α *with* $sig(\alpha) \subseteq sig(\mathcal{O})$.

Lemma 3. *Let* $\mathcal{O}$ *be an* $\mathcal{ELIH}$*-ontology. The transformation of* $\mathcal{O}$ *into A-NF or r-NF* $\mathcal{O}'$ *exhibits the following complexity properties:*

(1) The transformation requires a linear number of normalization rule applications;

(2) The size of the resulting ontology $\mathcal{O}'$ *is linear in* $|\mathcal{O}|$*;*

(3) The number $|sig_D(\mathcal{O})|$ *of introduced definers is bounded by* $O(n)$*, where* n *represents the number of* $\exists$*-restrictions in* $\mathcal{O}$*.*

[5] Note that $\mathcal{F}$ need not be confined to $\text{sig}(\mathcal{O})$, as Definition 3 shows that forgetting a name absent from $\mathcal{O}$ leaves the ontology unchanged, yielding $\mathcal{O}$ itself as a result.

Our forgetting method introduces a linear definer-based normalization strategy that fundamentally differs from existing approaches such as LETHE and UI-FAME, which require exponential definer introduction. This optimization achieves significant computational gains by maintaining linear space complexity during normalization, thereby enhancing the method's overall efficiency.

4 Single Name Elimination

4.1 Calculus for Eliminating A

We describe a two-phase calculus for eliminating a single concept name A from an $\mathcal{ELIH}$-ontology $\mathcal{O}$:

Phase I: Transform $\mathcal{O}$ into A-NF using the normalization rules;
Phase II: Eliminate A through exhaustive application of inference rules (Figure 1) to the normalized ontology.

In contrast to LETHE and UI-FAME, which operate on clausal representations, our method works directly with GCIs and role inclusions, enabling more efficient structural inference.

IR1. $C \sqsubseteq \mathsf{A}, \mathsf{A} \sqcap E \sqsubseteq G \implies C \sqcap E \sqsubseteq G$

IR2. $C \sqsubseteq \mathsf{A}, \exists R.(\mathsf{A} \sqcap E) \sqcap F \sqsubseteq G \implies \exists R.(C \sqcap E) \sqcap F \sqsubseteq G$

IR3. $C \sqsubseteq \exists R.(\mathsf{A} \sqcap D), \mathsf{A} \sqcap E_1 \sqsubseteq G_1, \ldots, \mathsf{A} \sqcap E_n \sqsubseteq G_n$
$\implies C \sqsubseteq \exists R.(E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n \sqcap D)$
provided that: $\mathcal{O} \models \mathsf{A} \sqcap D \sqsubseteq E_1 \sqcap \cdots \sqcap E_n$
$C \sqsubseteq \exists R.(\mathsf{A} \sqcap D), \mathsf{A} \sqcap E \sqsubseteq G \implies C \sqsubseteq \exists R.D$
provided that: $\mathcal{O} \not\models \mathsf{A} \sqcap D \sqsubseteq E$

IR4. $C \sqsubseteq \exists R.(\mathsf{A} \sqcap D), \exists S.(\mathsf{A} \sqcap E) \sqcap F \sqsubseteq G$
$\implies C \sqsubseteq \exists R.D, C \sqcap F \sqsubseteq G$
provided that: $\mathcal{O} \models \mathsf{A} \sqcap D \sqsubseteq E$ and $\mathcal{O} \models R \sqsubseteq S$
$C \sqsubseteq \exists R.(\mathsf{A} \sqcap D), \exists S.(\mathsf{A} \sqcap E) \sqcap F \sqsubseteq G \implies C \sqsubseteq \exists R.D$
provided that: $\mathcal{O} \not\models \mathsf{A} \sqcap D \sqsubseteq E$ or $\mathcal{O} \not\models R \sqsubseteq S$

Figure 1: Inference rules for eliminating A

Given an $\mathcal{ELIH}$-ontology in A-NF, the elimination procedure operates on A-GCIs using the inference rules shown in Figure 1. These rules are schemata rather than concrete rules: meta-variables A, C, D, E, F, G represent concepts, while R, S represent roles. Each rule consists of *premises* (left of $\implies$) and a *conclusion* (right).

The inference process systematically reveals implicit logical entailments over $\mathsf{sig}(\mathcal{O}) \setminus \{\mathsf{A}\}$ by combining positive occurrences of A (forms A-NF I or II) with negative occurrences (forms A-NF III or IV). This generates four distinct combination cases (IR1–IR4) in Figure 1. The process continues until reaching saturation — when no inference rule applications yield new GCIs.

Our combination strategy extends the binary resolution principle [34], which traditionally combines clauses with complementary literals (e.g., deriving conclusion $A \vee C$ from premises $A \vee B$ and $\neg B \vee C$). However, $\mathcal{ELIH}$ resolution poses unique challenges: A may appear under $\exists$-restrictions or within $\sqcap$-conjunctions, with complementarity only visible through first-order representation [45]. Our inference rules address this by directly resolving $\exists$ and $\sqcap$ literals on r. The final result of forgetting, $\mathcal{O}^{-\mathsf{A}}$, is obtained by removing all A-GCIs from the saturated ontology, yielding an ontology that preserves all relevant entailments while eliminating A.

LEMMA 4. *For any $\mathcal{ELIH}$-ontology $\mathcal{O}$ in A-NF, the elimination procedure preserves entailments:*

$$\mathcal{O}^{-A} \models \alpha \text{ iff } \mathcal{O} \models \alpha,$$

for any $\mathcal{ELIH}$-axiom α with $sig(\alpha) \subseteq sig(\mathcal{O}) \setminus \{A\}$.

Lemma 4 establishes the semantic preservation property of our calculus: $\mathcal{O}^{-\mathsf{A}}$ satisfies the entailment preservation condition required for forgetting A from $\mathcal{O}$ (Condition (2) of Definition 3). However, the signature containment condition (Condition (1)) may be violated due to the presence of auxiliary definers in $\mathcal{O}^{-\mathsf{A}}$ that extend beyond $\mathsf{sig}(\mathcal{O}) \setminus \{\mathsf{A}\}$. We address this signature expansion challenge in a later subsection by a definer elimination procedure.

4.2 Calculus for Eliminating r

Role name elimination follows a parallel structure to concept name elimination, employing a two-phase calculus:

Phase I: Transform $\mathcal{O}$ into r-NF using the normalization rules;
Phase II: Eliminate r through exhaustive application of inference rules (Figure 2) to the normalized ontology.

The inference process combines positive occurrences of r (forms r-NF I, II, III, IV) with negative occurrences (forms r-NF V, VI, VII, VIII), resulting in a total of $4 \times 4 = 16$ resolution cases. By introducing an involution mapping Inv where $\mathsf{Inv}(R) = r^-$ if $R = r$ and $\mathsf{Inv}(R) = r$ if $R = r^-$, we can express these 16 cases with 6 generalized cases (IR5—IR10). Upon achieving saturation and removing all r-axioms, we obtain $\mathcal{O}^{-\mathsf{r}}$ — an ontology free of r.

IR5. $C \sqsubseteq \exists R.D, F \sqcap \exists R.E \sqsubseteq G \implies F \sqcap C \sqsubseteq G$
provided that: $\mathcal{O} \models \exists R.D \sqsubseteq \exists R.E$

IR6. $C \sqsubseteq \exists R.D, F \sqcap \exists R^-.E \sqsubseteq G \implies F \sqcap C \sqsubseteq G$
provided that: $\mathcal{O} \models \exists R.D \sqsubseteq \exists R^-.E$

IR7. $C \sqsubseteq \exists R.D, R \sqsubseteq T \implies C \sqsubseteq \exists T.D$

IR8. $C \sqsubseteq \exists R.D, R^- \sqsubseteq T \implies C \sqsubseteq \exists T^-.D$

IR9. $S \sqsubseteq R, F \sqcap \exists R.E \sqsubseteq G \implies F \sqcap \exists S.E \sqsubseteq G$

IR10. $S \sqsubseteq R, F \sqcap \exists R^-.E \sqsubseteq G \implies F \sqcap \exists S^-.E \sqsubseteq G$

Figure 2: Inference rules for forgetting role name r

LEMMA 5. *For any $\mathcal{ELIH}$-ontology $\mathcal{O}$ in r-NF, the elimination procedure preserves entailments:*

$$\mathcal{O}^{-r} \models \alpha \text{ iff } \mathcal{O} \models \alpha$$

for any $\mathcal{ELIH}$-axiom α with $sig(\alpha) \subseteq sig(\mathcal{O}) \setminus \{r\}$.

The side conditions of the inference rules are verified using an external DL reasoner — HermiT [8], with subsumption checking in $\mathcal{ELIH}$ being ExpTime-complete [2].

4.3 Property of the Method

Our method takes an $\mathcal{ELIH}$-ontology $\mathcal{O}$ and a forgetting signature $\mathcal{F}$ comprising concept and role names to be eliminated. The elimination proceeds iteratively, processing one name at a time in any user-specified order. While each iteration may introduce definers requiring elimination from $\mathcal{O}^{-\mathcal{F}}$ (using the concept elimination calculus as described above), complete definer elimination is not always achievable, as illustrated by the following example:

- Initial ontology: $\mathsf{A} \sqsubseteq \exists r^{-}.\mathsf{A}$
- After forgetting A: $\mathsf{D}_1 \sqsubseteq \exists r^{-}.\mathsf{D}_1$, where D_1 is a fresh definer
- Attempting to forget D_1 yields: $\mathsf{D}_2 \sqsubseteq \exists r^{-}.\mathsf{D}_2$, where D_1 is a another fresh definer

This cyclic pattern would generate an infinite chain of definers. While fixpoint solutions exist [3], their implementation faces practical constraints due to limited support in mainstream reasoning tools and the OWL API. Instead of adopting fixpoints, our method ensures termination by preserving irreducible definers and explicitly marking such cases as partial forgetting results.

Theorem 1. *For any $\mathcal{ELIH}$-ontology $\mathcal{O}$ and forgetting signature $\mathcal{F} \subseteq sig(\mathcal{O})$, our method:*

- *Always terminates and produces an $\mathcal{ELIH}$-ontology $\mathcal{V}$*
- *If $\mathcal{V}$ is definer-free, it constitutes:*
  - *A result of forgetting $\mathcal{F}$ from $\mathcal{O}$*
  - *A uniform Σ-interpolant of $\mathcal{O}$ where $\Sigma = sig(\mathcal{O}) \setminus \mathcal{F}$*

This formulation admits the inherent limitations of forgetting while providing guaranteed termination and clear success criteria.

5 Empirical Evaluation

We implemented our UI method as a Java prototype using OWL API 3.5.7[6] and conducted comparative evaluations against Lethe, the current state-of-the-art system.[7] The evaluation utilized two industry benchmark standards: (1) Oxford ISG Library[8] featuring 797 ontologies from diverse sources and (2) NCBO BioPortal (March 2017 snapshot) [28] featuring 438 biomedical ontologies.

For systematic analysis, we selected ontologies with axiom count $|\text{Onto}| < 10000$ and categorized them by size: **Oxford ISG Corpus** (488 ontologies after acyclic $\mathcal{ELIH}$-fragment refinement, leading to 8.9% axiom reduction):

- PART I: 355 ontologies ($10 \leq |\text{Onto}| < 1000$)
- PART II: 108 ontologies ($1000 \leq |\text{Onto}| < 5000$)
- PART III: 25 ontologies ($5000 \leq |\text{Onto}| \leq 10000$)

BioPortal Corpus (326 ontologies after refinement):

- PART I: 202 ontologies ($10 \leq |\text{Onto}| < 1000$)
- PART II: 104 ontologies ($1000 \leq |\text{Onto}| < 5000$)
- PART III: 20 ontologies ($5000 \leq |\text{Onto}| \leq 10000$)

Table 1 provides statistical analysis of the refined ontologies.

To simulate real-world forgetting scenarios, we conducted two series of experiments varying the size of the signature $\mathcal{F}$:

- Small-size: forgetting 10% of concept and role names;
- Medium-size: forgetting 30% of concept and role names;
- Large-size: forgetting 50% of concept and role names.

The forgetting signatures were generated using a randomized selection algorithm to ensure unbiased testing. All experiments were conducted on an Intel Core i7-9750H processor (6 cores, 2.70 GHz) with 12 GB DDR4-1600 MHz RAM and 9 GB heap allocation. **Success Criteria:** A forgetting attempt was deemed successful if it:

- Eliminated all names in $\mathcal{F}$;
- Produced a definer-free result;
- Completed within 300 seconds (timeout threshold);
- Operated within allocated memory constraints.

The experiment results (Table 2) demonstrate significant performance differentials between our prototype and Lethe. Specifically, our prototype maintained consistent success rates of 100% across all test configurations. In contrast, Lethe exhibited performance degradation patterns correlated with ontology size and forgetting percentage. On the Oxford-ISG corpus, Lethe's success rate dropped from 92% for small ontologies with 10% forgetting to 65% for large ontologies with 50% forgetting, with timeout rates reaching 35% for large-scale forgetting operations. Similarly, on the BioPortal corpus, performance declined from 92% success rate for small ontologies with 10% forgetting to 61% for large ontologies with 50% forgetting, with timeout rates peaking at 39% for large-scale operations. Additionally, Lethe encountered consistent runtime errors across both corpora, most likely because of compatibility issues coming up between the OWL API version Lethe employed (Version 5.1.7) and the test ontologies.

Regarding computational efficiency, our prototype demonstrated substantial advantages in both time and memory usage. Time-wise, our implementation proved significantly faster, operating 52 times faster on Oxford-ISG and 37 times faster on BioPortal datasets. This performance gap became more pronounced with larger forgetting signatures. For instance, with PART III ontologies and 50% forgetting, our prototype completed tasks in 1.54 seconds compared to

Table 1: Statistic analysis of Oxford-ISG & BioPortal

Oxford-ISG		Min	Max	Medium	Mean	90th %ile
I	$\lvert N_C \rvert$	0	1582	86	191	545
	$\lvert N_R \rvert$	0	332	10	29	80
	$\lvert\text{Onto}\rvert$	0	990	162	262	658
II	$\lvert N_C \rvert$	200	5877	1665	1769	2801
	$\lvert N_R \rvert$	0	887	11	34	61
	$\lvert\text{Onto}\rvert$	1008	4976	2282	2416	3937
III	$\lvert N_C \rvert$	1162	9809	4042	5067	8758
	$\lvert N_R \rvert$	1	158	4	23	158
	$\lvert\text{Onto}\rvert$	5112	9783	7277	7195	9179
BioPortal		**Min**	**Max**	**Medium**	**Mean**	**90 Ptl**
I	$\lvert N_C \rvert$	0	784	127	192	214
	$\lvert N_R \rvert$	0	122	5	15	17
	$\lvert\text{Onto}\rvert$	0	982	283	312	346
II	$\lvert N_C \rvert$	5	4530	1185	1459	1591
	$\lvert N_R \rvert$	0	131	12	30	33
	$\lvert\text{Onto}\rvert$	1023	4977	2401	2619	2782
III	$\lvert N_C \rvert$	432	8340	4363	4387	4806
	$\lvert N_R \rvert$	0	135	17	30	34
	$\lvert\text{Onto}\rvert$	5084	8836	6934	6912	7109

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

[7] https://lat.inf.tu-dresden.de/~koopmann/LETHE/; Comparisons with Nui and UI-Fame were not possible due to their unavailability (as of November 11, 2024).

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

Table 2: Experiment results over Oxford-ISG and BioPortal (Time: Time Consumption, Mem: Memory Consumption, SR: Success Rate, TR: Timeout Rate, RER: Runtime Error Rate)

Oxford	%	PART	Time (sec.)	Mem (MB)	SR	TR	RER
LETHE	10%	I	4.23	37.06	92.42	4.10	3.48
		II	9.72	58.76	86.68	11.07	2.25
		III	14.58	88.54	77.86	22.14	0.00
	30%	I	12.46	52.24	87.09	9.45	3.48
		II	29.23	75.16	75.41	22.35	2.24
		III	41.36	123.11	67.82	32.18	0.00
	50%	I	14.63	71.36	79.91	16.61	3.48
		II	43.16	134.65	71.31	26.45	2.24
		III	74.02	189.13	64.55	35.45	0.00
Proto	10%	I	0.17	24.33	100	0.00	0.00
		II	0.46	38.54	100	0.00	0.00
		III	0.85	59.21	100	0.00	0.00
	30%	I	0.27	35.66	100	0.00	0.00
		II	0.74	51.02	100	0.00	0.00
		III	1.05	86.77	100	0.00	0.00
	50%	I	0.80	48.13	100	0.00	0.00
		II	1.33	91.21	100	0.00	0.00
		III	1.54	130.32	100	0.00	0.00
BioPortal	**%**	**PART**	**Time (sec.)**	**Mem (MB)**	**SR**	**TR**	**RER**
LETHE	10%	I	5.03	39.96	92.33	5.04	2.63
		II	11.16	59.04	85.58	10.58	3.57
		III	14.89	95.83	75.46	24.54	0.00
	30%	I	14.07	53.26	83.29	14.08	2.63
		II	32.11	88.33	73.01	23.42	3.57
		III	46.06	133.20	65.50	34.50	0.00
	50%	I	14.23	76.48	77.24	20.13	2.63
		II	45.73	140.11	69.00	27.43	3.57
		III	81.53	187.93	60.60	39.40	0.00
Proto	10%	I	0.16	21.34	100	0.00	0.00
		II	0.43	34.01	100	0.00	0.00
		III	0.83	52.03	100	0.00	0.00
	30%	I	0.31	31.23	100	0.00	0.00
		II	0.67	47.54	100	0.00	0.00
		III	1.04	78.27	100	0.00	0.00
	50%	I	0.76	44.09	100	0.00	0.00
		II	1.34	88.55	100	0.00	0.00
		III	1.53	120.81	100	0.00	0.00

LETHE's 74.02 seconds, while maintaining consistent scaling factors across different ontology sizes.

Memory utilization patterns also favored our prototype, showing consistently lower memory footprints across all test cases. Memory usage scaled more gracefully with ontology size: for PART III ontologies with 50% forgetting, our prototype required 130.50MB compared to LETHE's 189.13MB, with the gap widening proportionally as ontology size and forgetting percentage increased.

A key architectural distinction between the systems lies in their definer introduction strategies. LETHE employs a clause-flattening algorithm with exponential complexity, introducing $O(2^n)$ definers in the worst case, where n denotes the number of input clauses [18]. In contrast, our approach implements an optimized strategy that maintains linear growth w.r.t. input size by introducing definers only when structurally necessary.

To validate this hypothesis, we conducted a comparative analysis (Figure 3).[9] The analysis plots ontology size (the number of axioms)

[9] Direct verification of LETHE's internal operations was not possible as its source code is not publicly available.

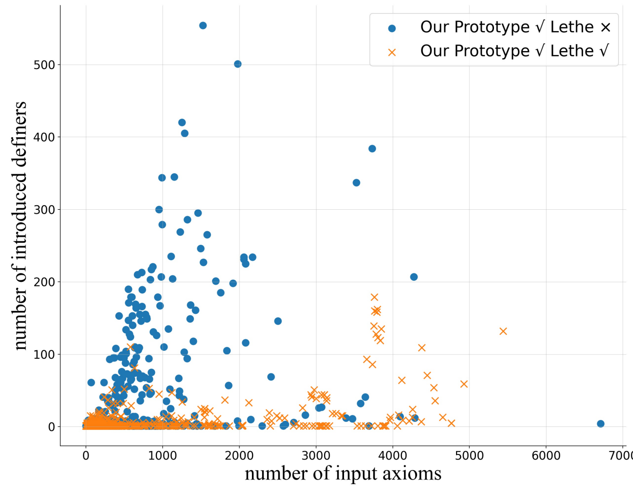

Figure 3: Distributions of successful and failure cases of our prototype and LETHE on Oxford ISG

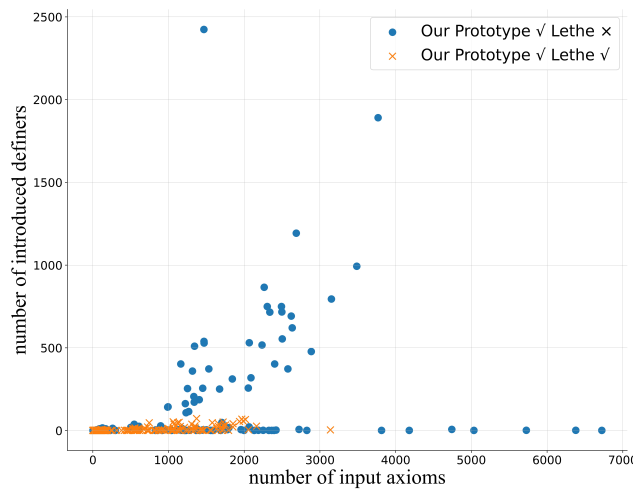

Figure 4: Distributions of successful and failure cases of our prototype and LETHE on BioPortal

against the number of introduced definers — a metric reflecting ontological complexity. Blue points indicate cases where our prototype succeeded but LETHE failed, while orange points represent mutual successes. The results reveal that LETHE's performance degradation correlates not with ontology size but with structural complexity requiring extensive definer introduction. However, LETHE's comprehensive definer introduction strategy does yield flatter clausal forms, potentially simplifying certain inference operations.

6 Semantic Change Detection in SNOMED CT

We conducted an extensive evaluation of our UI method's practical efficacy through analyzing semantic evolutionary changes in SNOMED CT, examining:

- 15 consecutive international releases

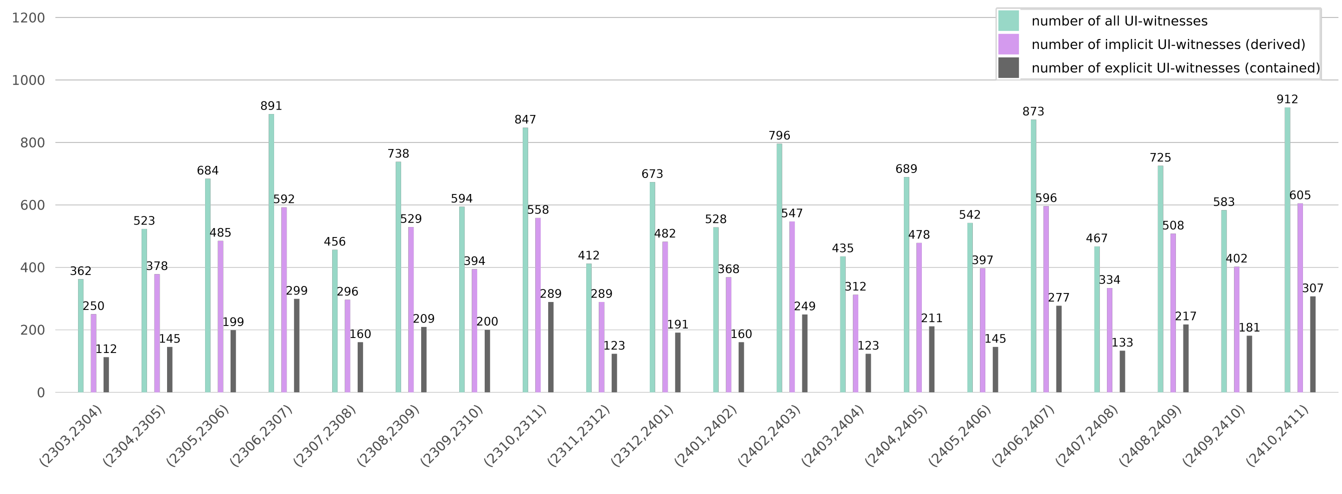

Figure 5: Semantic difference across continuous releases of SNOMED CT

Table 3: Metrics of the forgetting tasks on SNOMED CT

Tasks	$\lvert\mathcal{F}_C\rvert$	$\lvert\mathcal{F}_R\rvert$	$\lvert\sqsubseteq\rvert$	T (s)	Tasks	$\lvert\mathcal{F}_C\rvert$	$\lvert\mathcal{F}_R\rvert$	$\lvert\sqsubseteq\rvert$	T (s)
(2303,2304)	834	2	771K	75	(2401,2402)	876	1	771K	80
(2304,2305)	1127	4	772K	95	(2402,2403)	523	1	772K	55
(2305,2306)	1718	35	772K	125	(2403,2404)	128	0	773K	9
(2306,2307)	962	3	773K	85	(2404,2405)	287	2	773K	35
(2307,2308)	1743	19	773K	130	(2405,2406)	916	3	774K	82
(2308,2309)	1495	28	774K	115	(2406,2407)	451	0	774K	50
(2309,2310)	1236	12	774K	98	(2407,2408)	871	1	774K	78
(2310,2311)	1367	2	775K	105	(2408,2409)	928	2	775K	83
(2311,2312)	473	0	775K	48	(2409,2410)	1382	0	776K	110
(2312,2401)	937	0	776K	85	(2410,2411)	715	0	776K	65

- 5 non-consecutive international releases

This analysis aimed to validate our method's potential as a back-end technology for SNOMED's knowledge base management system, with particular emphasis on QA in healthcare data. We developed a UI-Diff framework that integrates:

- Our prototype for uniform interpolant computation (**Step 1**)
- HermiT [8] for semantic difference detection (**Step 2**)

This integration enables precise identification of semantic changes across SNOMED CT releases.

The performance analysis (Table 3) across SNOMED CT releases (formatted as YYMM, e.g., 1601 for January 2016) reveals several key insights about our method's effectiveness and efficiency. Our method successfully handled a wide range of forgetting signature sizes, from minimal changes ($|\mathcal{F}_C| = 128$, $|\mathcal{F}_R| = 0$) to substantial modifications ($|\mathcal{F}_C| = 1718$, $|\mathcal{F}_R| = 35$). Processing time scaled efficiently with signature size: negligible time for minor changes (2403, 2404), up to 150 seconds for major changes (2308, 2309), with most tasks completing within 100 seconds despite the substantial terminology scales. This consistent performance showcases robust handling of both incremental updates and major modeling changes throughout the evolution of SNOMED CT.

Figure 5 shows evolutionary changes across SNOMED CT releases, with version pairs on the x-axis (e.g., 2401 for January 2024) and UI-witness counts on the y-axis. These witnesses include both explicit ones (verified through HermiT's entailment checking) and implicit ones (computed via both UI and entailment checking). This visualization captures the dynamic evolution of the SNOMED CT International edition over time. Notably, the number of UI witnesses, which were derived using our UI prototype, averages 69.1% of the total derived witnesses, indicating a substantial portion of implicit knowledge capture. This underscores the practical utility and robustness of the UI approach in identifying and preserving essential semantic elements through successive ontology versions.

Our UI prototype has been operational within SNOMED International's back-end management system since April 2023, providing comprehensive semantic change detection capabilities. The system serves three critical functions: to identify semantic changes between ontology versions, to ensure that changes are safe in the sense of being conservative extensions [7, 26], and to identify unexpected semantic consequences in new releases. This implementation has strengthened SNOMED's quality control framework by enabling systematic validation of ontological evolution and early detection of potential semantic inconsistencies.

7 Conclusion and Future Work

We have developed a highly optimized UI method tailored for semantic change detection in large-scale ontologies. Our approach has overcome major limitations of existing UI methods, demonstrating superior performance on standard benchmark datasets. The method's practical impact is evidenced by its successful handling of SNOMED CT's monthly releases — since its deployment at SNOMED International in April 2023, our UI prototype has become an integral component of their QA pipeline, enabling systematic tracking of semantic evolution across releases, early detection of unintended consequences, and efficient validation of change safety.

Future work will focus on enhancing the system's analytical capabilities by developing fine-grained UI-witness classification and implementing change justification generation [10]. These enhancements aim to further strengthen SNOMED International's ontology management capabilities and support their mission of maintaining high-quality, semantically consistent healthcare terminology.

A APPENDIX

A.1 Missing Proofs

To prove Lemma 1, we first introduce the following definition:

Definition 6 (Conservative Extension). *Given two ontologies $\mathcal{O}$ and $\mathcal{O}'$, we say that $\mathcal{O}'$ is a* conservative extension *of $\mathcal{O}$ if*

(1) $sig(\mathcal{O}) \subseteq sig(\mathcal{O}')$,
(2) every model of $\mathcal{O}'$ is a model of $\mathcal{O}$, and
(3) for every model $\mathcal{I}$ of $\mathcal{O}$, there exists a model $\mathcal{I}'$ of $\mathcal{O}'$ such that the interpretations of the concept and role names from $sig(\mathcal{O})$ coincide in $\mathcal{I}$ and $\mathcal{I}'$, i.e.,
  - *$A^{\mathcal{I}} = A^{\mathcal{I}'}$ for all concept names $A \in sig(\mathcal{O})$, and*
  - *$r^{\mathcal{I}} = r^{\mathcal{I}'}$ for all role names $r \in sig(\mathcal{O})$.*

Conservative extension has two key properties:

- *Transitivity*: For ontologies $\mathcal{O}$, $\mathcal{O}'$, and $\mathcal{O}''$, if $\mathcal{O}''$ conservatively extends $\mathcal{O}'$ and $\mathcal{O}'$ conservatively extends $\mathcal{O}$, then $\mathcal{O}''$ conservatively extends $\mathcal{O}$; this is quite obvious.
- *Subsumption Preservation*: If $\mathcal{O}'$ conservatively extends $\mathcal{O}$, then for all clauses constructed using only the names from $\mathrm{sig}(\mathcal{O})$, subsumption w.r.t. $\mathcal{O}$ and $\mathcal{O}'$ coincide; see Prop. 1.

Proposition 1. *Let $\mathcal{O}$ and $\mathcal{O}'$ be general $\mathcal{ELIH}$ ontologies such that $\mathcal{O}$ is a conservative extension of $\mathcal{O}'$, and C, D are $\mathcal{ELIH}$ concepts containing only concept and role names from $\mathcal{O}'$. Then we have*

$$\mathcal{O}' \models C \sqsubseteq D \text{ iff } \mathcal{O} \models C \sqsubseteq D.$$

Proof. We prove this by contraposition. First, assume that $\mathcal{O} \not\models C \sqsubseteq D$. Then there is a model $\mathcal{I}$ of $\mathcal{O}$ such that $C^{\mathcal{I}} \not\subseteq D^{\mathcal{I}}$. Since $\mathcal{I}$ is also a model of $\mathcal{O}'$, this implies $\mathcal{O}' \not\models C \sqsubseteq D$. Next, assume that $\mathcal{O}' \not\models C \sqsubseteq D$. Then there is a model $\mathcal{I}'$ of $\mathcal{O}'$ such that $C^{\mathcal{I}'} \not\subseteq D^{\mathcal{I}'}$. Let $\mathcal{I}$ be a model of $\mathcal{O}$ such that the extensions of concept and role names from $\mathrm{sig}(\mathcal{O}')$ coincide in $\mathcal{I}$ and $\mathcal{I}'$. Since C and D contain only concept and role names from $\mathrm{sig}(\mathcal{O}')$, we have $C^{\mathcal{I}} = C^{\mathcal{I}'} \not\subseteq D^{\mathcal{I}'} = D^{\mathcal{I}}$, and $\mathcal{O} \not\models C \sqsubseteq D$. ◻

Lemma 1. *Let $\mathcal{O}$ be an $\mathcal{ELIH}$-ontology and $\mathcal{O}'$ the normalized one obtained from $\mathcal{O}$ using the normalization rules NR1 – NR5. Then we have*

$$\mathcal{O} \models C \sqsubseteq D \text{ iff } \mathcal{O}' \models C \sqsubseteq D,$$

for any $\mathcal{ELIH}$-GCI $C \sqsubseteq D$ with $sig(C \sqsubseteq D) \subseteq sig(\mathcal{O})$.

Proof. Our proof relies on the notion of conservative extension defined above. Specifically, we show that the ontology $\mathcal{O}'$ obtained from $\mathcal{O}$ by applying one of the normalization rules is a conservative extension of $\mathcal{O}$. We treat NR2 in detail. The rules NR3, NR4 and NR5 can be proved similarly. This statement holds trivially for NR1 since in that case $\mathcal{O}$ and $\mathcal{O}'$ have the same signature and are obviously equivalent.

Regarding NR2, assume that $\mathcal{O}'$ is obtained from $\mathcal{O}$ by replacing the GCI $X \sqsubseteq Y$ (NR2 generalizes the cases where $X = \exists R.C \sqcap D$) with two GCIs $\exists R.Z \sqcap D \sqsubseteq Y$ and $C \sqsubseteq Z$, where $Z \in \mathsf{N_C}$ is a fresh definer, i.e., $Z \notin \mathrm{sig}(\mathcal{O})$. Obviously, $\mathrm{sig}(\mathcal{O}') = \mathrm{sig}(\mathcal{O}) \cup \{Z\}$, and therefore $\mathrm{sig}(\mathcal{O}) \subseteq \mathrm{sig}(\mathcal{O}')$, satisfying Condition (1) of Definition 6. Next, assume that $\mathcal{I}'$ is a model of $\mathcal{O}'$. Then we have $(\exists R.Z)^{\mathcal{I}'} \cap D^{\mathcal{I}'} \subseteq Y^{\mathcal{I}'}$ and $C^{\mathcal{I}'} \subseteq Z^{\mathcal{I}'}$. This implies $(\exists R.C)^{\mathcal{I}'} \cap D^{\mathcal{I}'} \subseteq (\exists R.Z)^{\mathcal{I}'} \cap D^{\mathcal{I}'} \subseteq Y^{\mathcal{I}'}$, and thus $\mathcal{I}'$ is a model of $\mathcal{O}$. Finally, assume that $\mathcal{I}$ is a model of $\mathcal{O}$. Let $\mathcal{I}'$ be the interpretation that coincides with $\mathcal{I}$ on all concept and role names with the exception of Z. For Z, we define the extension in $\mathcal{I}'$ as $Z^{\mathcal{I}'} = C^{\mathcal{I}}$. Since $\mathcal{I}$ is a model of $\mathcal{O}$, we have $(\exists R.C)^{\mathcal{I}} \cap D^{\mathcal{I}} \subseteq Y^{\mathcal{I}}$. In addition, since Z does not occur in $\exists R.C$, D, or Y, we have $C^{\mathcal{I}} = C^{\mathcal{I}'}$, $(\exists R.C)^{\mathcal{I}} = (\exists R.C)^{\mathcal{I}'}$, $D^{\mathcal{I}} = D^{\mathcal{I}'}$ and $Y^{\mathcal{I}} = Y^{\mathcal{I}'}$. This yields $C^{\mathcal{I}'} = C^{\mathcal{I}} = Z^{\mathcal{I}'}$ and $(\exists R.Z)^{\mathcal{I}'} \cap D^{\mathcal{I}'} = (\exists R.C)^{\mathcal{I}} \cap D^{\mathcal{I}} \subseteq Y^{\mathcal{I}} = Y^{\mathcal{I}'}$, showing that $\mathcal{I}'$ is a model of $\mathcal{O}'$. Because of transitivity, Lemma 1 is an immediate consequence of Proposition 1. ◻

Lemma 2. *Let $\mathcal{O}$ be an $\mathcal{ELIH}$-ontology and $\mathcal{O}'$ the normalized one obtained from $\mathcal{O}$ using the slightly adjusted normalization rules for r-NF transformation. Then we have*

$$\mathcal{O} \models C \sqsubseteq D \text{ iff } \mathcal{O}' \models C \sqsubseteq D,$$

for any $\mathcal{ELIH}$-GCI $C \sqsubseteq D$ with $sig(C \sqsubseteq D) \subseteq sig(\mathcal{O})$.

Proof. Lemma 2 can be established through a proof methodology analogous to that employed in the demonstration of Lemma 1. ◻

To prove Lemma 3, we first introduce the following notions:

We define the *frequency* $\mathrm{fq}(\mathsf{A}, C)$ of A in an A-concept C inductively as follows:

- $\mathrm{fq}(\mathsf{A}, X) = \begin{cases} 1, & \text{if } X = \mathsf{A}; \\ 0, & \text{if } X \in \mathsf{N_C} \text{ and } X \neq \mathsf{A}, \end{cases}$
- $\mathrm{fq}(\mathsf{A}, \exists R.E) = \mathrm{fq}(\mathsf{A}, E)$,
- $\mathrm{fq}(\mathsf{A}, E \sqcap F) = \mathrm{fq}(\mathsf{A}, E) + \mathrm{fq}(\mathsf{A}, F)$.

We define the *frequency* $\mathrm{fq}(\mathsf{A}, X \sqsubseteq Y)$ of A in a GCI $X \sqsubseteq Y$:

- $\mathrm{fq}(\mathsf{A}, X \sqsubseteq Y) = \mathrm{fq}(\mathsf{A}, X) + \mathrm{fq}(\mathsf{A}, Y)$.

Let A^* be a designated occurrence of A in C. We define the *role depth* $\mathrm{dp}(\mathsf{A}^*, C)$ of A^* in C inductively as follows:

- $\mathrm{dp}(\mathsf{A}^*, C) = 0$, if C is of the form $\mathsf{A}^* \sqcap D$, where D is an arbitrary concept,
- $\mathrm{dp}(\mathsf{A}^*, C) = \mathrm{dp}(\mathsf{A}^*, E) + 1$, if C is of the form $D \sqcap \exists R.E$, where $r \in \mathsf{N_R}$, E is a concept that contains A^*, and D is an arbitrary concept,

i.e., $\mathrm{dp}(\mathsf{A}^*, C)$ counts the number of $\exists$-restrictions guarding A^* in C. The *role depth* $\mathrm{dp}(\mathsf{A}, C)$ of A in C is defined as the sum of the *role depth* of all occurrences of A in C. The *role depth* $\mathrm{dp}(\mathsf{A}, X \sqsubseteq Y)$ of A in a GCI $X \sqsubseteq Y$ is defined as:

- $\mathrm{dp}(\mathsf{A}, X \sqsubseteq Y) = \mathrm{dp}(\mathsf{A}, X) + \mathrm{dp}(\mathsf{A}, Y)$.

Proposition 2. *Any* A-*GCI* $X \sqsubseteq Y$ *with* $\mathrm{dp}(\mathsf{A}, X \sqsubseteq Y) = 0$ *is in* A-*normal form.*

Proof. $\mathrm{dp}(\mathsf{A}, X \sqsubseteq Y) = 0$ indicates that A is not guarded by any $\exists$-restriction in $X \sqsubseteq Y$, ensuring that A appears only at the surface level of the concepts X and Y. This leads to three possible cases (a simpler representation of $X \sqsubseteq Y$ is assumed):

- $\mathsf{A} \notin \mathrm{sig_C}(X)$ and $\mathsf{A} \in \mathrm{sig_C}(Y)$ (Form I)
- $\mathsf{A} \in \mathrm{sig_C}(X)$ and $\mathsf{A} \notin \mathrm{sig_C}(Y)$ (Form III)
- $\mathsf{A} \in \mathrm{sig_C}(X)$ and $\mathsf{A} \in \mathrm{sig_C}(Y)$

The last case can be generalized as $\mathsf{A} \sqcap C \sqsubseteq \mathsf{A} \sqcap D$, with C, D being A-free concepts. This is equivalent to $\mathsf{A} \sqcap C \sqsubseteq \mathsf{A}$ and $\mathsf{A} \sqcap C \sqsubseteq D$; the former is a tautology and the latter aligns with Form III. ◻

Proposition 3. *Any* A*-GCI* $X \sqsubseteq Y$ *with* $\mathrm{fq}(\mathsf{A}, X \sqsubseteq Y) = 1$ *and* $\mathrm{dp}(\mathsf{A}, X \sqsubseteq Y) = 1$ *is in* A*-normal form.*

Proof. $\mathrm{fq}(\mathsf{A}, X \sqsubseteq Y) = 1$ ensures a single occurrence of A in $X \sqsubseteq Y$ and $\mathrm{dp}(\mathsf{A}, X \sqsubseteq Y) = 1$ indicates that A is guarded by a single $\exists$-restriction. This leads to two possible cases:

- $\mathsf{A} \notin \mathrm{sig}_{\mathsf{C}}(X)$ and $\mathsf{A} \in \mathrm{sig}_{\mathsf{C}}(Y)$ (Form II)
- $\mathsf{A} \in \mathrm{sig}_{\mathsf{C}}(X)$ and $\mathsf{A} \notin \mathrm{sig}_{\mathsf{C}}(Y)$ (Form IV)

Both cases are already in A-normal form. □

Proposition 4. *For any* A*-GCI* $X \sqsubseteq Y$, $\mathrm{fq}(\mathsf{A}, X \sqsubseteq Y) - \mathrm{dp}(\mathsf{A}, X \sqsubseteq Y) \leq 1$.

Proof. For any A-GCI $X \sqsubseteq Y$ with a depth $\mathrm{dp}(\mathsf{A}, X \sqsubseteq Y) = m$ ($m \geq 0$), A appears in $X \sqsubseteq Y$ at most m+1 times. As per Proposition 2, A can occur unguarded (not within an $\exists$-restriction) only once in $X \sqsubseteq Y$. Given A's depth of m in $X \sqsubseteq Y$, there can be at most m guarded occurrences of A. Thus, A appears in $X \sqsubseteq Y$ a maximum of $m + 1$ times. □

We define $\mathsf{NLZ}_1(O)$ as the set derived from O by applying one of the rules NR1 – NR5 to O, indicating one round of normalization for O. Similarly, $\mathsf{NLZ}_k(O)$ is defined for any $k \geq 0$, with $\mathsf{NLZ}_0(O)$ representing the original O. We further define $\mathrm{dp}(\mathsf{A}, O)$ as the sum of $\mathrm{dp}(\mathsf{A}, X \sqsubseteq Y)$, for every A-GCI $X \sqsubseteq Y$ in O.

Proposition 5. $\mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_{k-1}(O)) - \mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_k(O)) = 1$*, where (i)* $k \geq 1$ *and (ii)* $\mathsf{NLZ}_{k-1}(O)$ *is not in* A*-normal form.*

Proof. Condition (ii) ensures that $\mathsf{NLZ}_{k-1}(O) \neq \mathsf{NLZ}_k(O)$ and activates the applicability of NR2 – NR5. We prove that applying any of these rules decreases the depth of A in the resulting O by 1. We treat NR2 in detail. The rules NR3 – NR5 can be proved similarly. Consider $\mathsf{NLZ}_{k-1}(O)$ as a set with an A-GCI $X \sqsubseteq Y$ not in A-normal form, where $\mathrm{dp}(\mathsf{A}, X \sqsubseteq Y) = m$ and $\mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_{k-1}(O)) = m + x$ for $x \geq 0$. Rule NR2 implies that $m \geq 1$. Assume $\mathrm{dp}(\mathsf{A}, C) = n$, and Rule NR2 indicates $n \geq 1$. Consequently, $\mathrm{dp}(\mathsf{A}, \exists r.C) = n + 1$ and $m = n + 1 + y$ ($y \geq 0$). Replacing C with a new definer Z eliminates $X \sqsubseteq Y$ from $\mathsf{NLZ}_k(O)$ and introduces two GCIs: $(X \sqsubseteq Y)^C_Z$ and $C \sqsubseteq Z$. Here, $(X \sqsubseteq Y)^C_Z$ denotes the GCI obtained from $X \sqsubseteq Y$ by replacing C with Z.

Since $\mathrm{dp}(\mathsf{A}, (X \sqsubseteq Y)^C_Z) = \mathrm{dp}(\mathsf{A}, (X \sqsubseteq Y)) - \mathrm{dp}(\mathsf{A}, \exists r.C) = n + 1 + y - (n + 1) = y$ and $\mathrm{dp}(\mathsf{A}, C \sqsubseteq Z) = \mathrm{dp}(\mathsf{A}, C) + \mathrm{dp}(\mathsf{A}, Z) = n$, $\mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_k(O)) = \mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_{k-1}(O)) - \mathrm{dp}(\mathsf{A}, X \sqsubseteq Y) + y + n = x + y + n$. Given that $\mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_{k-1}(O)) = m + x = x + y + n + 1$, $\mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_{k-1}(O)) - \mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_k(O)) = 1$. □

Lemma 3. *Let O be an $\mathcal{ELIH}$-ontology. The transformation of O into A-NF or r-NF O' exhibits the following complexity properties:*

1. *The transformation requires a linear number of normalization rule applications;*
2. *The size of the resulting ontology O' is linear in $|O|$;*
3. *The number $|sig_D(O)|$ of introduced definers is bounded by $O(n)$, where n represents the number of $\exists$-restrictions in O.*

Proof. The A-normal form ensures that every A-GCI contains exactly one occurrence of A; it covers all elementary structures of an A-GCI and specifies where the A appears in these structures. Specifically, an A-concept is either the atomic concept A (the base case) or is constructed with $\exists$ and $\sqcap$ (the induction cases). In each case, A appears as a conjunct in an explicit or implicit conjunction. For example, the atomic concept A is a simpler representation of $\mathsf{A} \sqcap D$, with $D \equiv \top$. Since $\mathsf{A} \sqcap D$ appears under $\exists$-restrictions or without any, we identify two basic forms of an A-concept: $\mathsf{A} \sqcap D$ and $\exists \mathsf{r}.(\mathsf{A} \sqcap D) \sqcap F$. These can be on either side of a GCI. Form I and III generalize the scenarios where $\mathsf{A} \sqcap D$ is on the right and left sides of a GCI, respectively, while Form II (IV) covers cases with $\exists \mathsf{r}.(\mathsf{A} \sqcap D) \sqcap F$ on the right and left sides.

Proposition 4 states that any unnormalized A-GCI $X \sqsubseteq Y$ can be reduced to a set $\mathsf{NLZ}(X \sqsubseteq Y)$ of GCIs in finite steps with each GCI α in $\mathsf{NLZ}(X \sqsubseteq Y)$ having $\mathrm{dp}(\mathsf{A}, \alpha) \leq 1$. If $\mathrm{dp}(\mathsf{A}, \alpha) = 0$, α is either a non-A-GCI, or in A-normal form (as per Proposition 1). If $\mathrm{dp}(\mathsf{A}, \alpha) = 1$, α may not be in A-normal form because it might contain two occurrences of A (as per Proposition 3); for instance, $\mathrm{dp}(\mathsf{A}, \mathsf{A} \sqcap \exists r.\mathsf{A} \sqsubseteq Y) = 1$ but $\mathrm{fq}(\mathsf{A}, \mathsf{A} \sqcap \exists r.\mathsf{A} \sqsubseteq Y) = 2$. These cases, with an unguarded A and an A guarded by a single $\exists$-restriction, meet NR2 – NR3's syntactic requirements, and can be normalized in one step using NR2–NR3. Since every definer replaces a subconcept immediately under an $\exists$-restriction, the number of the definers for O's normalization and the newly-added GCIs is bounded by $\mathrm{O}(n)$, where n is the count of $\exists$-restrictions in O. This proves termination and completeness of the normalization procedure. □

Lemma 4. *For any $\mathcal{ELIH}$-ontology O in A-NF, the elimination procedure preserves entailments:*

$$O^{-A} \models C \sqsubseteq D \textit{ iff } O \models C \sqsubseteq D,$$

for any $\mathcal{ELIH}$-GCI $C \sqsubseteq D$ with $sig(C \sqsubseteq D) \subseteq sig(O) \backslash \{A\}$.

Proof. The rules IR1 and IR2 essentially reverse the normalization process, thereby preserving subsumption directly. We then focus on proving Rule IR3. Rule IR4 can be proved similarly. We show that the GCIs on the left side (the *premises*, denoted by O) of the $\Longrightarrow$ symbol are a conservative extension of those on the right side (the *conclusion*, denoted by O'). Obviously, $\mathrm{sig}(O) = \mathrm{sig}(O') \cup \{\mathsf{A}\}$, and therefore $\mathrm{sig}(O') \subseteq \mathrm{sig}(O)$, satisfying Condition (1) of Definition 3.

Case I (i.e., provided that: $O \models \mathsf{A} \sqcap D \sqsubseteq E_1 \sqcap \ldots \sqcap E_n$):

Assume that $\mathcal{I}$ is a model of O. Then we have the following:

$$C^{\mathcal{I}} \subseteq (\exists R.(\mathsf{A} \sqcap D))^{\mathcal{I}} \quad (1)$$

$$(\mathsf{A} \sqcap E_1)^{\mathcal{I}} \subseteq {G_1}^{\mathcal{I}}, \ldots, (\mathsf{A} \sqcap E_n)^{\mathcal{I}} \subseteq {G_n}^{\mathcal{I}} \quad (2)$$

$$(\mathsf{A} \sqcap D)^{\mathcal{I}} \subseteq (E_1 \sqcap \ldots \sqcap E_n)^{\mathcal{I}} \quad (3)$$

As Inclusion (2) implies $(\mathsf{A} \sqcap E_1)^{\mathcal{I}} \cap \ldots \cap (\mathsf{A} \sqcap E_n)^{\mathcal{I}} \subseteq {G_1}^{\mathcal{I}} \cap \ldots \cap {G_n}^{\mathcal{I}}$, we have $(\mathsf{A} \sqcap E_1 \sqcap \ldots \sqcap E_n)^{\mathcal{I}} \subseteq (G_1 \sqcap \ldots \sqcap G_n)^{\mathcal{I}}$ and further $(\mathsf{A} \sqcap E_1 \sqcap \ldots \sqcap E_n)^{\mathcal{I}} \subseteq (E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n)^{\mathcal{I}}$. As Inclusion (3) implies $(\mathsf{A} \sqcap D)^{\mathcal{I}} \subseteq (\mathsf{A} \sqcap E_1 \sqcap \ldots \sqcap E_n)^{\mathcal{I}}$, we have $(\mathsf{A} \sqcap D)^{\mathcal{I}} \subseteq (E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n)^{\mathcal{I}}$ and further $(\mathsf{A} \sqcap D)^{\mathcal{I}} \subseteq (E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n \sqcap D)^{\mathcal{I}}$. Together with Inclusion (1), we have the following:

$$C^{\mathcal{I}} \subseteq (\exists R.(E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n \sqcap D))^{\mathcal{I}} \quad (4)$$

This satisfies Condition (2) of Definition 3.

Assume that $\mathcal{I}'$ is a model of O'. Let $\mathcal{I}$ be the interpretation that coincides with $\mathcal{I}'$ on all concept and role names with the exception of A. For A, we define the extension in $\mathcal{I}$ as $\mathsf{A}^{\mathcal{I}} = (E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n)^{\mathcal{I}'}$. Since $\mathcal{I}'$ is a model of O', we have $C^{\mathcal{I}'} \subseteq (\exists R.(E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n \sqcap D))^{\mathcal{I}'}$. In addition, since

A does not occur in C or $\exists r.(E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n \sqcap D)$, we have $C^{\mathcal{I}'} = C^{\mathcal{I}}$, $(E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n)^{\mathcal{I}'} = (E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n)^{\mathcal{I}} = \mathsf{A}^{\mathcal{I}}$, $(\exists R.(E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n \sqcap D))^{\mathcal{I}'} = (\exists R.(E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n \sqcap D))^{\mathcal{I}} = (\exists R.(\mathsf{A} \sqcap D))^{\mathcal{I}}$, and thus, we have

$$C^{\mathcal{I}} \subseteq (\exists R.(\mathsf{A} \sqcap D))^{\mathcal{I}} \tag{5}$$

Since $(E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n \sqcap E_1)^{\mathcal{I}} \subseteq {G_1}^{\mathcal{I}}$, we have

$$(\mathsf{A} \sqcap E_1)^{\mathcal{I}} \subseteq {G_1}^{\mathcal{I}} \tag{6}$$

In the same way, we can prove the other GCIs in (2). Since $(E_1 \sqcap \ldots \sqcap E_n \sqcap G_1 \sqcap \ldots \sqcap G_n \sqcap D)^{\mathcal{I}} \subseteq (E_1 \sqcap \ldots \sqcap E_n)^{\mathcal{I}}$, we have

$$(\mathsf{A} \sqcap D)^{\mathcal{I}} \subseteq (E_1 \sqcap \ldots \sqcap E_n)^{\mathcal{I}} \tag{7}$$

Inclusions (5), (6), and (7) collectively satisfy Condition (3) of Definition 3.

Case II (i.e., provided that: $\mathcal{O} \not\models \mathsf{A} \sqcap D \sqsubseteq E$):

Assume that $\mathcal{I}$ is a model of $\mathcal{O}$. Then we have the following:

$$C^{\mathcal{I}} \subseteq (\exists R.(\mathsf{A} \sqcap D))^{\mathcal{I}} \tag{8}$$

$$(\mathsf{A} \sqcap E)^{\mathcal{I}} \subseteq G^{\mathcal{I}} \tag{9}$$

Directly, we have

$$C^{\mathcal{I}} \subseteq (\exists R.D)^{\mathcal{I}} \tag{10}$$

Assume that $\mathcal{I}'$ is a model of $\mathcal{O}'$. Let $\mathcal{I}$ be the interpretation that coincides with $\mathcal{I}'$ on all concept and role names with the exception of A, E and G. For A, we define the extension in $\mathcal{I}$ as $\mathsf{A}^{\mathcal{I}} = \top^{\mathcal{I}'}$. Since $\mathcal{I}'$ is a model of $\mathcal{O}'$, we have $C^{\mathcal{I}'} \subseteq (\exists R.D)^{\mathcal{I}'}$. In addition, since A does not occur in C or $\exists R.D$, we have $C^{\mathcal{I}'} = C^{\mathcal{I}}$, $\top^{\mathcal{I}'} = \top^{\mathcal{I}} = \mathsf{A}^{\mathcal{I}}$, $(\exists r.D)^{\mathcal{I}'} = (\exists R.(\top \sqcap D))^{\mathcal{I}'} = (\exists R.(\top \sqcap D))^{\mathcal{I}} = (\exists R.(\mathsf{A} \sqcap D))^{\mathcal{I}}$, and thus, we have

$$C^{\mathcal{I}} \subseteq (\mathsf{A} \sqcap \exists R.D)^{\mathcal{I}} \tag{11}$$

For E and G, we define their extensions in $\mathcal{I}$ as $E^{\mathcal{I}} = G^{\mathcal{I}} = \top^{\mathcal{I}'}$. Then we have

$$(\mathsf{A} \sqcap E)^{\mathcal{I}} \subseteq G^{\mathcal{I}} \tag{12}$$

Thus, $\mathcal{O}$ is a conservative extension of $\mathcal{O}'$. Because of the transitivity of conservative extension, $\mathcal{O}$ in Lemma 4 is a conservative extension of $\mathcal{O}^{-\mathsf{A}}$. According to Proposition 1, Lemma 4 holds. □

Likewise, Lemma 5 establishes the partial soundness of the calculus. Specifically, the derived ontology $\mathcal{O}^{-\mathsf{r}}$ fulfills the second condition necessary for it to be the result of forgetting $\{\mathsf{r}\}$ from $\mathcal{O}$. However, $\mathcal{O}^{-\mathsf{r}}$ may include definers which fall outside the scope of $\mathrm{sig}(\mathcal{O})\backslash\{\mathsf{r}\}$, potentially failing to fulfill the first condition.

Lemma 5. *For any $\mathcal{ELIH}$-ontology $\mathcal{O}$ in r-NF, the elimination procedure preserves entailments:*

$$\mathcal{O}^{-r} \models C \sqsubseteq D \textit{ iff } \mathcal{O} \models C \sqsubseteq D,$$

for any $\mathcal{ELIH}$-GCI $C \sqsubseteq D$ with $sig(C \sqsubseteq D) \subseteq sig(\mathcal{O})\backslash\{r\}$.

Proof. We prove Rules IR5 – IR10, with a detailed examination of Rule IR5. The other rules can be proved similarly. Regarding IR5, we show that the GCIs on the left side (the *premises*, denoted by $\mathcal{O}$) of the $\Longrightarrow$ symbol are a conservative extension of those on the right side (the *conclusion*, denoted by $\mathcal{O}'$). Obviously, $\mathrm{sig}(\mathcal{O}) = \mathrm{sig}(\mathcal{O}') \cup \{\mathsf{r}\}$, and therefore $\mathrm{sig}(\mathcal{O}') \subseteq \mathrm{sig}(\mathcal{O})$, satisfying Condition (1) of Definition 3.

Assume that $\mathcal{I}$ is a model of $\mathcal{O}$. Then we have the following:

$$C^{\mathcal{I}} \subseteq (\exists R.D)^{\mathcal{I}} \tag{13}$$

$$(F \sqcap \exists R.E)^{\mathcal{I}} \subseteq G^{\mathcal{I}} \tag{14}$$

$$(\exists R.D)^{\mathcal{I}} \subseteq (\exists R.E)^{\mathcal{I}} \tag{15}$$

Inclusions (13) and (15) jointly imply $C^{\mathcal{I}} \subseteq (\exists R.E)^{\mathcal{I}}$, and when combined with (14), they further imply $(F \sqcap C)^{\mathcal{I}} \subseteq G^{\mathcal{I}}$.

Assume that $\mathcal{I}'$ is a model of $\mathcal{O}'$. Let $\mathcal{I}$ be the interpretation that coincides with $\mathcal{I}'$ on all concept and role names with the exception of R, D, and E. For the exceptions, we define their extensions in $\mathcal{I}$ as $(\exists R.D)^{\mathcal{I}} = (\exists R.E)^{\mathcal{I}} = C^{\mathcal{I}'}$. Then we have $(\exists R.D)^{\mathcal{I}} \subseteq (\exists R.E)^{\mathcal{I}}$ Since $\mathcal{I}'$ is a model of $\mathcal{O}'$, we have $(F \sqcap C)^{\mathcal{I}'} \subseteq G^{\mathcal{I}'}$. In addition, since R does not occur in C, D, E, F, and G, we have $C^{\mathcal{I}'} = C^{\mathcal{I}}$, $F^{\mathcal{I}'} = F^{\mathcal{I}}$ and $G^{\mathcal{I}'} = G^{\mathcal{I}}$. Thus we have $C^{\mathcal{I}'} = C^{\mathcal{I}} \subseteq (\exists R.D)^{\mathcal{I}}$ and $(F^{\mathcal{I}} \sqcap \exists R.E)^{\mathcal{I}} \subseteq G^{\mathcal{I}}$

Thus, $\mathcal{O}$ is a conservative extension of $\mathcal{O}'$. Because of the transitivity of conservative extension, $\mathcal{O}$ in Lemma 5 is a conservative extension of $\mathcal{O}^{-\mathsf{r}}$. According to Proposition 1, Lemma 5 holds. □

Theorem 1. *Given any $\mathcal{ELIH}$-ontology $\mathcal{O}$ and any forgetting signature $\mathcal{F} \subseteq sig(\mathcal{O})$ as input, our forgetting method always terminates and returns an $\mathcal{ELIH}$-ontology $\mathcal{V}$. If $\mathcal{V}$ does not contain any definers, then it is a result of forgetting $\mathcal{F}$ from $\mathcal{O}$.*

Proof. Note that the normalization and inference rules do not introduce new cycles. For cases where cyclic behavior originally exhibits over the names in $\mathcal{F}$, the method terminates upon detecting a cycle. In acyclic cases, termination of the method follows from Lemma 3 and the termination of the forgetting calculi. The method's soundness is ensured by Lemmas 1 and 2, and Lemmas 4 and 5. □

A.2 Illustrative Examples

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

$$\{1.\ \mathsf{E} \sqsubseteq \exists \mathsf{r}^-.(\mathsf{F} \sqcap \exists \mathsf{t.A}),\ 2.\ \exists \mathsf{t.A} \sqcap \exists \mathsf{t}^-.\mathsf{E} \sqsubseteq \mathsf{D}\}$$

Let $\mathcal{F} = \{\mathsf{A}\}$. The first step is to compute the A-NF of $\mathcal{O}$ by applying the normalization rules as described earlier, where $Z_1 \in \mathsf{N_D}$ is a fresh definer:

$$\{3.\ \mathsf{E} \sqsubseteq \exists \mathsf{r}^-.Z_1,\ 5.\ Z_1 \sqsubseteq \mathsf{F},\ 6.\ Z_1 \sqsubseteq \exists \mathsf{t.A},\ 2.\ \exists \mathsf{t.A} \sqcap \exists \mathsf{t}^-.\mathsf{E} \sqsubseteq \mathsf{D}\}$$

The above ontology is now in A-normal form. The second step is to apply the inference rules in Figure 1. Applying Rule IR5 to GCIs 2 and 6 gives:

$$\{3.\ \mathsf{E} \sqsubseteq \exists \mathsf{r}^-.Z_1,\ 5.\ Z_1 \sqsubseteq \mathsf{F},\ 7.\ Z_1 \sqcap \exists \mathsf{t}^-.\mathsf{E} \sqsubseteq \mathsf{D},\ 8.\ Z_1 \sqsubseteq \exists t.\top\}$$

Definers are treated as regular concept names, and are eliminated once all $\mathcal{F}$-names have been eliminated. Applying Rule IR7 to GCIs 3 and 7 gives $\{9.\ \mathsf{E} \sqsubseteq \exists r^-.\top\}$. Applying Rule IR7 to GCIs 3, 5 and 8 gives $\{10.\ E \sqsubseteq \exists r^-.(F \sqcap \exists t.\top)\}$. GCI 9 is redundant w.r.t. GCI 10 and thus removed. Our method implements a set of straightforward simplifications. In this case, $\{10.\ E \sqsubseteq \exists r^-.(F \sqcap \exists t.\top)\}$ is a uniform Σ-interpolant of $\mathcal{O}$, where $\Sigma = sig(\mathcal{O}) \setminus \mathcal{F}$.

References

[1] Grigoris Antoniou and Frank van Harmelen. 2004. *Web Ontology Language: OWL.* Springer Berlin Heidelberg, 67–92.

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

[3] 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.

[4] Jieying Chen, Ghadah Alghamdi, Renate A. Schmidt, Dirk Walther, and Yongsheng Gao. 2019. Ontology Extraction for Large Ontologies via Modularity and Forgetting. In *K-CAP'19.* ACM, 45–52.

[5] Giovanna D'Agostino and Giacomo Lenzi. 2015. Bisimulation quantifiers and uniform interpolation for guarded first order logic. *Theor. Comput. Sci.* 563 (2015), 75–85.

[6] John Day-Richter, Midori A. Harris, Melissa Haendel, The Gene Ontology OBO-Edit Working Group, and Suzanna Lewis. 2007. OBO-Edit—an ontology editor for biologists. *Bioinformatics* 23, 16 (2007), 2198–2200.

[7] 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.

[8] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. 2014. HermiT: An OWL 2 Reasoner. *J. Autom. Reasoning* 53, 3 (2014), 245–269.

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

[10] Matthew Horridge. 2011. *Justification based explanation in ontologies.* Ph. D. Dissertation. The University of Manchester, UK.

[11] Matthew Horridge, Rafael S. Gonçalves, Csongor I. Nyulas, Tania Tudorache, and Mark A. Musen. 2019. WebProtégé: A Cloud-Based Ontology Editor. In *Proc. WWW'19.* ACM, 686–689.

[12] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, Bernardo Cuenca Grau, and James A. Hendler. 2006. Swoop: A Web Ontology Editing Browser. *J. Web Semant.* 4, 2 (2006), 144–153.

[13] Michel Klein, Dieter Fensel, Atanas Kiryakov, and Damyan Ognyanov. 2002. OntoView: Comparing and Versioning Ontologies.

[14] 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.

[15] Boris Konev, Dirk Walther, and Frank Wolter. 2008. The Logical Difference Problem for Description Logic Terminologies. In *Proc. IJCAR'14 (Lecture Notes in Computer Science, Vol. 5195).* Springer, 259–274.

[16] 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.

[17] Roman Kontchakov, Frank Wolter, and Michael Zakharyaschev. 2010. Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artif. Intell.* 174, 15 (2010), 1093–1141.

[18] Patrick Koopmann. 2015. *Practical Uniform Interpolation for Expressive Description Logics.* Ph. D. Dissertation. The University of Manchester, UK.

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

[20] Patrick Koopmann and Renate A. Schmidt. 2015. LETHE: Saturation-Based Reasoning for Non-Standard Reasoning Tasks. In *Proc. DL'15 (CEUR Workshop Proceedings, Vol. 1387).* CEUR-WS.org, 23–30.

[21] Patrick Koopmann and Renate A. Schmidt. 2015. Saturated-Based Forgetting in the Description Logic $\mathcal{SIF}$. In *Proc. DL'15 (CEUR Workshop Proc., Vol. 1350).*

[22] Hongkai Liu, Carsten Lutz, Maja Milicic, and Frank Wolter. 2006. Updating Description Logic ABoxes. In *Proc. KR'06.* AAAI Press, 46–56.

[23] Zhao Liu, Chang Lu, Ghadah Alghamdi, Renate A. Schmidt, and Yizheng Zhao. 2021. Tracking Semantic Evolutionary Changes in Large-Scale Ontological Knowledge Bases. In *Proc. CIKM'21.* ACM, 1130–1139.

[24] Michel Ludwig and Boris Konev. 2014. Practical Uniform Interpolation and Forgetting for $\mathcal{ALC}$ TBoxes with Applications to Logical Difference. In *Proc. KR'14.* AAAI Press, 318–327.

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

[26] 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.

[27] 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.

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

[29] Mark A. Musen. 2015. The Protégé project: a look back and a look forward. *AI Matters* 1, 4 (2015), 4–12.

[30] 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.

[31] Natalya F. Noy and Mark A. Musen. 2002. PROMPTDIFF: A Fixed-Point Algorithm for Comparing Ontology Versions. In *Proc. AAAI/IAAI'02.* AAAI/MIT Press, 744–750.

[32] Andrew M. Pitts. 1992. On an Interpretation of Second Order Quantification in First Order Intuitionistic Propositional Logic. *J. Symb. Log.* 57, 1 (1992), 33–52.

[33] Alan L. Rector, Jeremy Rogers, Pieter E. Zanstra, and Egbert J. van der Haring. 2003. OpenGALEN: Open Source Medical Terminology and Tools. In *AMIA 2003, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 8-12, 2003.* AMIA.

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

[35] John Alan Robinson and Andrei Voronkov (Eds.). 2001. *Handbook of Automated Reasoning (in 2 volumes).* Elsevier and MIT Press.

[36] Nicholas Sioutos, Sherri de Coronado, Margaret W. Haber, Frank W. Hartel, Wen-Ling Shaiu, and Lawrence W. Wright. 2007. NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *J. Biomed. Informatics* 40, 1 (2007), 30–43.

[37] Kent A. Spackman. 2000. SNOMED RT and SNOMED CT. Promise of an international clinical ontology. *M.D. Computing 17* (2000).

[38] Steffen Staab and Rudi Studer (Eds.). 2009. *Handbook on Ontologies.* Springer.

[39] The Gene Ontology Consortium. 2021. The Gene Ontology resource: enriching a GOld mine. *Nucleic Acids Res.* 49, Database-Issue (2021), D325–D334.

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

[41] Kewen Wang, Zhe Wang, Rodney W. Topor, Jeff Z. Pan, and Grigoris Antoniou. 2014. Eliminating Concepts and Roles from Ontologies in Expressive Descriptive Logics. *Computational Intelligence* 30, 2 (2014), 205–232.

[42] 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.

[43] Zhihao Yang and Yizheng Zhao. 2024. What a Surprise! Computing Rewritten Modules Can Be as Efficient as Computing Subset Modules. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024.* ACM, 2940–2949.

[44] Yizheng Zhao. 2024. Efficient Computation of Signature-Restricted Views for Semantic Web Ontologies. In *Proc. of WWW'24.* ACM, 1945–1953.

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

[46] Yizheng Zhao, Ghadah Alghamdi, Schmidt Renate A., 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.