Whereof One Cannot Explain, Thereof One Must Invent: Definitorial Abduction in $\mathcal{ALCI}$ via Strong Forgetting

Jinghan Wu[1], Renate A. Schmidt[2] and Yizheng Zhao[1,*]

[1]State Key Laboratory of Novel Software Technology & School of Artificial Intelligence, Nanjing University, Nanjing, China
[2]Department of Computer Science, The University of Manchester, Manchester, UK

Abstract

Signature-based abduction in description logics (DLs) seeks the weakest sufficient hypothesis, formulated over a designated signature, that together with a background ontology explains a given observation. When no informative hypothesis is expressible within the allowed signature, existing methods based on weak forgetting inevitably produce rationally unacceptable results such as $A \sqsubseteq \bot$. We formalize *definitorial abduction*, which returns a pair $\langle \mathcal{H}, \mathcal{X} \rangle$ of a hypothesis $\mathcal{H}$ respecting the signature restriction, and a definitorial extension $\mathcal{X}$ introducing fresh concept names anchored to the original ontology through conservative definitorial axioms. Standard signature-based abduction is the special case when $\mathcal{X} = \emptyset$.

To realize definitorial abduction, we develop a sound, complete, and terminating Ackermann-style strong forgetting calculus for acyclic $\mathcal{ALCI}$ ontologies. The key technical contribution is the $\exists$-*surfacing rule*, which resolves a fundamental incompleteness of the Ackermann framework when a concept name to be eliminated occurs under existential restrictions in both polarities. The unnamed concept names that this rule introduces are precisely the fresh symbols that definitorial abduction exploits. An evaluation of the approach on 547 real-world ontologies shows: 20–29% of instances require unnamed concept names, confirming the practical prevalence of cases where standard abduction degenerates; our prototype is 36% faster on average than the only existing baseline.

Keywords

Description logics, Abduction, Strong forgetting

1. Introduction

Ontologies formalized in description logics (DLs) [1] provide the formal backbone for automated reasoning over structured domain knowledge. In practice, however, real-world ontologies are not always complete, and newly observed facts may not be entailed by the existing axioms. Abductive reasoning addresses this gap: given a background ontology $\mathcal{O}$ and an observation Φ with $\mathcal{O} \not\models \Phi$, it seeks a hypothesis $\mathcal{H}$ such that $\mathcal{O} \cup \mathcal{H} \models \Phi$ [2, 3, 4, 5, 6]. Without further constraints, however, the hypothesis may merely restate the observation or make unnecessarily strong claims. *Signature-based abduction* [7, 8] addresses both issues: the hypothesis $\mathcal{H}$ must be formulated using only concept and role names from a designated signature Σ of explanatory terms, and among all such hypotheses one prefers the weakest sufficient one [9]. This constraint forces explanations to be framed in an abstract signature rather than merely echoing the observation.

Depending on the form of the observation and the hypothesis, abduction in DLs takes several different forms [4, 10], including concept abduction [11], ABox abduction [12], and TBox abduction [13]; this work considers the latter.

The central computational question is how to derive such a hypothesis systematically. The key is to reframe the problem. The core condition of abduction, $\mathcal{O} \cup \mathcal{H} \models \Phi$, is logically equivalent to $\mathcal{O} \cup \neg\Phi \models \neg\mathcal{H}$ through contraposition [14]. This turns the creative task of finding a hypothesis $\mathcal{H}$ for the observation into the mechanical task of deriving consequences when the negation of the observation is assumed. The negation of the hypothesis ($\neg\mathcal{H}$) must be a logical consequence of the counterfactual ontology $\mathcal{O} \cup \neg\Phi$. This insight yields a three-step pipeline:

CI-BD-SOQE 2026: Workshop on Craig Interpolation, Beth Definability, and Second-Order Quantifier Elimination, at FLoC 2026: The 9th Federated Logic Conference, July 24–25, 2026, Lisbon, Portugal

*Corresponding author.
jinghan.wu@smail.nju.edu.cn (J. Wu); renate.schmidt@manchester.ac.uk (R. A. Schmidt); zhaoyz@nju.edu.cn (Y. Zhao)

1. Construct a temporary ontology by combining the background knowledge with the negated observation: $\mathcal{O} \cup \neg\Phi$.
2. From this temporary ontology, derive all logical consequences that can be expressed using only the terms in Σ. This step of principled symbol elimination is formally known as *forgetting* [15]: it removes every concept name outside Σ, yielding $\neg\mathcal{H}$.
3. Negate the result to obtain the hypothesis $\mathcal{H}$.

Forgetting is thus the computational engine for signature-based abduction, and the semantic quality of the resulting hypothesis depends entirely on the power of the forgetting mechanism employed.

Existing approaches [7, 8] instantiate this pipeline using *weak forgetting* [16], which preserves the deductive property of inseparability [17]: the result entails exactly the same consequences over Σ as the original ontology. Weak forgetting is well-suited for standard signature-based abduction, as it produces the weakest sufficient hypothesis expressible within Σ. Its output is confined to the original signature and can never introduce concept names beyond those already present. This means that when no non-degenerate hypothesis can be expressed using Σ alone, the pipeline necessarily degenerates into inquiry-blocking axioms such as $A \sqsubseteq \bot$ that are logically sufficient but explanatorily vacuous.

Example 1. *The 1918–1919 influenza pandemic exhibited anomalously high fatality among young adults. We model this with background knowledge:*

$$\mathcal{O} = \{\, \texttt{ReturnSoldiers} \sqsubseteq \exists\, \texttt{diagWith}.\texttt{Influenza},\ \texttt{CiviliansNoExp} \sqsubseteq \forall\, \texttt{diagWith}.\neg\texttt{Influenza} \,\}$$

and the observation:

$$\Phi = \{\texttt{DeadYoungAdults} \sqsubseteq \forall\, \texttt{diagWith}.\texttt{Influenza}\}$$

Setting $\Sigma = sig(\mathcal{O}) \setminus \{\texttt{Influenza}\}$, weak forgetting yields:

$$\mathcal{H} = \{\texttt{DeadYoungAdults} \sqcap \exists\, \texttt{diagWith}.\top \sqsubseteq \bot\}$$

This asserts that dead young adults cannot have any diagnosis, closing off inquiry rather than guiding it. The root cause is that the concept needed to bridge the explanatory gap—an as-yet-unnamed subtype of influenza linking soldiers and civilians—simply has no name in Σ. To recover an acceptable explanation, we introduce a fresh concept name $\mathrm{U}_0 \notin sig(\mathcal{O}) \cup sig(\Phi)$, representing this unidentified category, and return a hypothesis $\mathcal{H}' = \{\texttt{DeadYoungAdults} \sqsubseteq \forall\, \texttt{diagWith}.\mathrm{U}_0\}$. Of course, a bare fresh symbol carries no meaning on its own. We therefore pair it with definitorial axioms*—axioms that anchor U_0 to the existing signature and thereby define its intended interpretation:*

$$\mathcal{X} = \{\, \mathrm{U}_0 \sqsubseteq \texttt{Influenza}, \texttt{ReturnSoldiers} \sqsubseteq \exists\, \texttt{diagWith}.\mathrm{U}_0, \mathrm{U}_0 \sqsubseteq \exists\, \texttt{diagWith}^{-}.\texttt{ReturnSoldiers} \,\}.$$

The first axiom says U_0 is a subtype of influenza; the second says every returning soldier has been diagnosed with some U_0 case; and the third says every U_0 case is a diagnosis of at least one returning soldier. Crucially, $\mathcal{O} \cup \mathcal{X}$ is a conservative extension *[18] of $\mathcal{O}$: the definitorial axioms give U_0 a precise meaning without altering any entailment over the original signature. Together, $\mathcal{H}'$ and $\mathcal{X}$ assert that dead young adults were all diagnosed with a latent influenza subtype U_0 tightly linked to returning soldiers, yielding an investigation-guiding explanation satisfying $\mathcal{O} \cup \mathcal{X} \cup \mathcal{H}' \models \Phi$ rather than a dead end.*

The need for fresh concept names in abductive hypotheses has deep roots. Peirce [2, 19] characterized abduction as the only form of reasoning that introduces genuinely new ideas, yet signature-based abduction, by confining hypotheses to a fixed Σ, explicitly forecloses this capacity. This tension echoes Hempel's [20] Theoretician's Dilemma: theoretical terms can be logically eliminated, yet the resulting theory loses its power of inductive systematization—just as in the pipeline above; when confined to the original signature, the result collapses to $A \sqsubseteq \bot$. Schurz [21] formalized this distinction as *creative* versus *selective* abduction. In machine learning, the same barrier manifests as the predicate invention problem: when the target hypothesis lies outside the available vocabulary, the learner must coin new predicates [22, 23]. Within DL reasoning, Koopmann [24] has observed that ABox abduction may require fresh individuals to obtain any solution; we lift this principle to the concept-name level. Our experiments confirm the practical prevalence: approximately 30% of abduction instances on real-world ontologies require unnamed concept names to avoid degenerate results (Section 5).

We formalize this as "*Definitorial Abduction*". In mathematical logic, an extension by definitions introduces fresh symbols via defining axioms and is inherently conservative [25]; this notion has been studied extensively for DL ontologies [18, 26]. Definitorial abduction returns a solution $\langle \mathcal{H}, \mathcal{X} \rangle$ where $\mathcal{X}$ introduces fresh concept names via definitorial axioms such that $\mathcal{O} \cup \mathcal{X}$ is a conservative extension of $\mathcal{O}$, and $\mathcal{O} \cup \mathcal{X} \cup \mathcal{H} \models \Phi$. Standard signature-based abduction is the special case where $\mathcal{X} = \emptyset$.

Realizing definitorial abduction computationally requires a forgetting paradigm that can go beyond the original signature and expressivity. *Strong forgetting* [27, 28, 29] meets this requirement: it preserves the model-theoretic structure projected onto Σ, and to faithfully represent this projection it may introduce fresh concept names as existential witnesses accompanied by definitorial axioms. Strong forgetting thus serves as the natural computational engine for definitorial abduction. Where weak forgetting is confined to Σ and degenerates when Σ is insufficient, strong forgetting extends the signature only as needed.

However, the only existing practical method for strong forgetting in expressive DLs, the Ackermann-based Fame framework [30], is incomplete: Ackermann's Lemma [31] cannot be applied when a concept name occurs under existential restrictions [32] in both positive and negative polarity. We resolve this by introducing the $\exists$-*surfacing rule*, which transforms such axioms into a form where the target concept is explicitly defined, introducing a fresh unnamed concept as an existential witness. Completeness for acyclic $\mathcal{ALCI}$ ontologies is achieved precisely because the procedure admits controlled extensions of the target language: inverse roles extend expressivity beyond $\mathcal{ALC}$, and fresh concept names, introduced via definitorial axioms, extend the signature. Without admitting such extensions, finite strong forgetting solutions need not exist even for acyclic inputs [27]. The unnamed concept names that the $\exists$-surfacing rule introduces for completeness are exactly the auxiliary concepts that definitorial abduction exploits for richer hypotheses. Furthermore, unlike existing abduction approaches that encode negated observations via nominals, our method operates directly on $\mathcal{O}$ and Φ at the TBox level.

Contributions. Our main contributions are:

1. We introduce the $\exists$-*surfacing rule*, a novel inference rule that resolves a key incompleteness of strong forgetting: when a concept name to be eliminated occurs under existential restrictions in both positive and negative polarity, Ackermann's Lemma does not apply, and our rule transforms such axioms into an eliminable form.
2. Based on this new rule, we develop an Ackermann-style strong forgetting procedure for acyclic $\mathcal{ALCI}$ ontologies that is sound, complete, and terminating. Completeness is achieved by allowing fresh unnamed concept names, introduced via definitorial axioms, to persist in the result.
3. Based on this procedure, we present the first application of complete strong forgetting to signature-based abduction and formalize *definitorial abduction*, which creates a result $\langle \mathcal{H}, \mathcal{X} \rangle$ where $\mathcal{X}$ introduces fresh concept names via conservative definitorial extensions.
4. We implement our approach and evaluate it on real-world ontologies, showing that definitorial abduction produces investigation-guiding hypotheses where standard methods yield only degenerate and rationally unacceptable results.

The source code and supplementary materials needed to reproduce our results are available at https://anonymous.4open.science/r/abduction-0F8E/.

2. Preliminaries

2.1. The Description Logic $\mathcal{ALCI}$

Our approach builds upon the DL $\mathcal{ALCI}$, which extends the basic $\mathcal{ALC}$ with inverse roles [1]. The signature of $\mathcal{ALCI}$ comprises two countably infinite, disjoint sets: concept names $\mathsf{N_C}$ and role names $\mathsf{N_R}$. A *role* R in $\mathcal{ALCI}$ is either a role name $R \in \mathsf{N_R}$, or an inverse role R^- of a role name R. *Concept descriptions* (or *concepts* for short) in $\mathcal{ALCI}$ are constructed inductively as:

$$C, D ::= \top \mid \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C$$

where $A \in \mathsf{N_C}$, C and D are concepts, and R is a role.

An $\mathcal{ALCI}$-*ontology* $\mathcal{O}$ is defined as a finite set of *axioms* of the form $C \sqsubseteq D$ (*General Concept Inclusion*, GCI for short), where C and D are concepts. We use $C \equiv D$ (*Concept Equivalence*) as an abbreviation for the pair of GCIs $C \sqsubseteq D$ and $D \sqsubseteq C$.

The semantics of $\mathcal{ALCI}$ is defined in terms of an *interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a non-empty set, known as the *domain of the interpretation*, and $\cdot^{\mathcal{I}}$ is the *interpretation function* mapping 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 concepts as follows:

$$\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \qquad \bot^{\mathcal{I}} = \emptyset \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \exists y.(x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \forall y.(x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\} \\
(R^{-})^{\mathcal{I}} &= \{(y,x) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (x,y) \in R^{\mathcal{I}}\}
\end{aligned}$$

Let $\mathcal{I}$ be an interpretation. A GCI $C \sqsubseteq D$ holds in $\mathcal{I}$, written $\mathcal{I} \models C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$, denoted $\mathcal{I} \models \mathcal{O}$, if every GCI in $\mathcal{O}$ holds in $\mathcal{I}$. A GCI $C \sqsubseteq D$ is a *logical entailment* of $\mathcal{O}$, written $\mathcal{O} \models C \sqsubseteq D$, if it holds in every model of $\mathcal{O}$. We write $\mathsf{sig}(X)$ for the set of concept and role names occurring in a concept, axiom, or ontology X. An ontology $\mathcal{O}'$ is a *conservative extension* of $\mathcal{O}$ with respect to $\mathsf{sig}(\mathcal{O})$ if, for every axiom α with $\mathsf{sig}(\alpha) \subseteq \mathsf{sig}(\mathcal{O})$, we have $\mathcal{O} \models \alpha$ iff $\mathcal{O}' \models \alpha$.

Our forgetting method operates on GCIs in clausal normal form. A *literal* is a concept of the form A, $\neg A$, $\exists R.C$, or $\forall R.C$, where $A \in \mathsf{N_C}$, C is a concept, and R is a role. An $\mathcal{ALCI}$ *clause* is a GCI of the form

$$\top \sqsubseteq L_1 \sqcup \cdots \sqcup L_n,$$

where each L_i is a literal. We omit the leading $\top \sqsubseteq$ and identify a clause with the finite set or disjunction of its literals. An ontology is in *clausal normal form* if all its GCIs are clauses. Thus, throughout the calculus, a clause is a GCI in this syntactic form.

A concept, clause, or ontology X is A-*free* if $A \notin \mathsf{sig}(X)$. An occurrence of a concept name A is *positive* if it falls under an even number of negations, and *negative* otherwise. A clause (set) is *positive* (*negative*) w.r.t. A if all occurrences of A in it are positive (negative). An A-*clause* is a clause containing the concept name A. For a clause set $\mathcal{N}$ and concepts C, E, we write $\mathcal{N}^C_E$ for the clause set obtained from $\mathcal{N}$ by replacing every occurrence of C by E.

Let $\Gamma \subseteq \mathsf{N_C}$ be a set of fresh concept names. An interpretation $\mathcal{I}'$ is an *expansion* of an interpretation $\mathcal{I}$ by Γ if $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$ and $\mathcal{I}'$ agrees with $\mathcal{I}$ on every concept and role name outside Γ. Interpretations $\mathcal{I}$ and $\mathcal{I}'$ are $\mathcal{F}$-*equivalent*, written $\mathcal{I} \sim_{\mathcal{F}} \mathcal{I}'$, if (i) $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$, (ii) $B^{\mathcal{I}} = B^{\mathcal{I}'}$ for every $B \in \mathsf{N_C} \setminus \mathcal{F}$, and (iii) $r^{\mathcal{I}} = r^{\mathcal{I}'}$ for every $r \in \mathsf{N_R}$. Since $\mathcal{F} \subseteq \mathsf{N_C}$, $\mathcal{F}$-equivalent interpretations may differ only on concept names in $\mathcal{F}$.

Definition 1 (Acyclicity). *Let $\mathcal{N}$ be a clause set and let $A, B \in \mathsf{N_C}$ be concept names occurring in $\mathcal{N}$. We say that A* directly depends on *B in $\mathcal{N}$ ($A \prec B$) iff $\mathcal{N}$ contains a clause in which A occurs positively and B negatively, or A negatively and B positively. A* depends on *B iff (A, B) is in the transitive closure of $\prec$. The clause set $\mathcal{N}$ is* acyclic *if no concept name occurring in $\mathcal{N}$ depends on itself; otherwise it is* cyclic. *An ontology is acyclic if its clausal form is acyclic.*

2.2. Two Paradigms of Forgetting

Forgetting eliminates designated concept names from an ontology [15]. We denote by $\mathcal{F} \subseteq \mathsf{N_C}$ the *forgetting signature*: the set of concept names to be eliminated. A concept name being eliminated is called a *pivot*. The *remaining signature* is $\Sigma = \mathsf{sig}(\mathcal{O}) \setminus \mathcal{F}$: the set of concept and role names in which the forgetting result is to be expressed. Since $\mathcal{F}$ consists only of concept names, Σ retains all role names of $\mathcal{O}$ and partitions its concept names with $\mathcal{F}$. In the context of signature-based abduction (Section 2.3), Σ coincides with the abducible signature introduced in Section 1.

Two paradigms of forgetting exist, differentiated by what they preserve.

Definition 2 (Weak Forgetting)**.** *An ontology $\mathcal{O}'$ is a solution of* weakly forgetting *$\mathcal{F}$ from $\mathcal{O}$ if:*

1. *$sig(\mathcal{O}') \subseteq sig(\mathcal{O}) \setminus \mathcal{F}$.*
2. *For any axiom α with $sig(\alpha) \subseteq sig(\mathcal{O}) \setminus \mathcal{F}$, it holds that $\mathcal{O} \models \alpha$ iff $\mathcal{O}' \models \alpha$.*

Weak forgetting preserves logical entailments over the remaining signature Σ: $\mathcal{O}$ and $\mathcal{O}'$ are Σ-*inseparable*, meaning that they agree on all entailments expressible over Σ [17].

Definition 3 (Strong Forgetting)**.** *Let $\Gamma = sig(\mathcal{O}') \setminus sig(\mathcal{O})$ be the fresh concept names used in the output. An ontology $\mathcal{O}'$ is a solution of* strongly forgetting *$\mathcal{F}$ from $\mathcal{O}$ if:*

1. *$sig(\mathcal{O}') \cap \mathcal{F} = \emptyset$.*
2. *For every model $\mathcal{I}'$ of $\mathcal{O}'$, there exists an interpretation $\mathcal{I}$ with $\mathcal{I}' \sim_{\mathcal{F} \cup \Gamma} \mathcal{I}$ and $\mathcal{I} \models \mathcal{O}$.*
3. *For every model $\mathcal{I}$ of $\mathcal{O}$, there exists an interpretation $\mathcal{I}'$ with $\mathcal{I}' \sim_{\mathcal{F} \cup \Gamma} \mathcal{I}$ and $\mathcal{I}' \models \mathcal{O}'$.*

We write $\mathsf{Forget}_{\mathcal{F}}(\mathcal{O})$ for a result satisfying Definition 3. Conditions 2 and 3 say that $\mathcal{O}$ and $\mathcal{O}'$ determine the same models after the hidden names are ignored. When $\Gamma = \emptyset$, this is the standard projection condition: an interpretation satisfies $\mathcal{O}'$ exactly when its restriction to the remaining signature can be expanded on the forgotten names $\mathcal{F}$ to a model of $\mathcal{O}$. When $\Gamma \neq \emptyset$, the fresh names in Γ are auxiliary presentation symbols introduced by the calculus.

Strong forgetting preserves the model-theoretic structure projected onto the remaining signature Σ: $\mathcal{O}$ and $\mathcal{O}'$ have exactly the same models once the names in $\mathcal{F}$ and the fresh presentation symbols Γ are disregarded. As shown by Lin and Reiter [15], this is equivalent to eliminating second-order quantifiers over the symbols in $\mathcal{F}$, which means that strong forgetting results can transcend the expressivity of the base language. Zhang and Zhou [16] later termed this *strong forgetting* and introduced the deductive notion of *weak forgetting* described above. A strong forgetting solution $\mathcal{O}'_s$ always entails a weak forgetting solution $\mathcal{O}'_w$ (i.e., $\mathcal{O}'_s \models \mathcal{O}'_w$); when $\mathcal{O}'_s$ is expressible in the base language without fresh names, the two coincide (i.e., $\mathcal{O}'_s \equiv \mathcal{O}'_w$).

The two definitions differ in their relationship to the output language. Weak forgetting is language-dependent: the result is defined relative to a chosen language L, and different choices of L yield different solutions. Hence condition 1 must explicitly fix the output signature to $\mathrm{sig}(\mathcal{O}) \setminus \mathcal{F}$. Strong forgetting, by contrast, is model-theoretic: Conditions 2 and 3 characterize $\mathcal{O}'$ through projected interpretations, so the target model set is determined independently of a particular syntactic output language. However, this model set may not be finitely expressible in the base language; when it is not, a strong forgetting solution in the base language simply does not exist. Extending the signature with fresh concept names or the expressivity of the language [28] can make a finite representation possible. Condition 1 of strong forgetting is accordingly stated as $\mathrm{sig}(\mathcal{O}') \cap \mathcal{F} = \emptyset$ rather than $\mathrm{sig}(\mathcal{O}') \subseteq \mathrm{sig}(\mathcal{O}) \setminus \mathcal{F}$, leaving room for such extensions. Our calculus exploits this by introducing unnamed concept names when needed.

2.3. Signature-based Abduction

Abduction seeks a hypothesis $\mathcal{H}$ to explain an observation Φ not entailed by a background ontology $\mathcal{O}$. *Signature-based abduction* further restricts the signature of the hypothesis to a designated set Σ of concept and role names.

Definition 4 (Signature-based Abduction)**.** *Let $\mathcal{O}$ be the background ontology and Φ the observation, with $\mathcal{O} \cup \Phi \not\models \bot$ and $\mathcal{O} \not\models \Phi$. Let $\Sigma \subseteq sig(\mathcal{O}) \cup sig(\Phi)$ be the abducible signature: the set of names permitted in the hypothesis. Define the forgetting signature $\mathcal{F} = (sig(\mathcal{O}) \cup sig(\Phi)) \setminus \Sigma$. The* signature-based abduction problem *$\langle \mathcal{O}, \Phi, \Sigma \rangle$ is the task of finding a hypothesis $\mathcal{H}$ that satisfies:*

1. ***Consistency:*** *$\mathcal{O} \cup \mathcal{H} \not\models \bot$.*

2. ***Explanation:*** $\mathcal{O} \cup \mathcal{H} \models \Phi$.

3. ***Signature Restriction:*** $sig(\mathcal{H}) \cap \mathcal{F} = \emptyset$.

A solution $\mathcal{H}$ is optimal *if, for any other hypothesis $\mathcal{H}'$ satisfying (1)–(3), it holds that:*

4. ***Optimality:*** $\mathcal{O} \cup \mathcal{H}' \models \mathcal{H}$.

In some cases, however, the signature restriction (Condition 3) forces explanations to collapse into degenerate hypotheses (e.g., $A \sqsubseteq \bot$) that are logically adequate but explanatorily vacuous. To obtain more informative explanations, we permit fresh concept names not in $\text{sig}(\mathcal{O}) \cup \text{sig}(\Phi)$, introduced through a finite set $\mathcal{X}$ of definitorial axioms that form a conservative extension of $\mathcal{O}$. The resulting explanation is a pair $\langle \mathcal{H}, \mathcal{X} \rangle$ with $\mathcal{O} \cup \mathcal{X} \cup \mathcal{H} \models \Phi$, where the signature restriction applies only to $\mathcal{H}$.

Definition 5 (Definitorial Abduction)**.** *Let $\mathcal{O}$ be the background ontology and Φ the observation, with $\mathcal{O} \cup \Phi \not\models \bot$ and $\mathcal{O} \not\models \Phi$. Let $\mathcal{F} \subseteq sig(\mathcal{O}) \cup sig(\Phi)$. The* definitorial abduction problem *$\langle \mathcal{O}, \Phi, \mathcal{F} \rangle$ is the task of finding a pair $\langle \mathcal{H}, \mathcal{X} \rangle$ such that:*

1. ***Signature Restriction (on $\mathcal{H}$):*** $sig(\mathcal{H}) \cap \mathcal{F} = \emptyset$.

2. ***Freshness (of $\mathcal{X}$):*** $sig(\mathcal{X}) \subseteq sig(\mathcal{O}) \cup sig(\Phi) \cup \Gamma$ *for some set* Γ *with* $\Gamma \cap (sig(\mathcal{O}) \cup sig(\Phi)) = \emptyset$.

3. ***Conservativity:*** $\mathcal{O} \cup \mathcal{X}$ *is a conservative extension of* $\mathcal{O}$ *w.r.t.* $sig(\mathcal{O})$.

4. ***Non-Definability:*** *For every $U \in \Gamma$, the concept name U is not implicitly definable from $sig(\mathcal{O}) \cup sig(\Phi)$ under $\mathcal{O} \cup \mathcal{X}$, i.e., there exist models $\mathcal{I}_1, \mathcal{I}_2$ of $\mathcal{O} \cup \mathcal{X}$ that agree on $sig(\mathcal{O}) \cup sig(\Phi)$ but $U^{\mathcal{I}_1} \neq U^{\mathcal{I}_2}$.*

5. ***Consistency:*** $\mathcal{O} \cup \mathcal{X} \cup \mathcal{H} \not\models \bot$.

6. ***Explanation:*** $\mathcal{O} \cup \mathcal{X} \cup \mathcal{H} \models \Phi$.

Given a fixed $\mathcal{X}$, a solution $\langle \mathcal{H}, \mathcal{X} \rangle$ is $\mathcal{X}$-optimal *if, for any other hypothesis $\mathcal{H}'$ such that $\langle \mathcal{H}', \mathcal{X} \rangle$ also satisfies (1)–(6), it holds that:*

7. ***$\mathcal{X}$-Optimality:*** $\mathcal{O} \cup \mathcal{X} \cup \mathcal{H}' \models \mathcal{H}$.

To exclude trivial solutions that merely rephrase the observation using fresh names, we require that every fresh concept name introduced by $\mathcal{X}$ carries genuinely new information. We formalize this via implicit definability in the sense of Beth [33]: a concept name U is *implicitly definable* from a signature Σ under a theory T if every two models of T that agree on Σ necessarily agree on U. Beth's Definability Theorem shows that implicit and explicit definability coincide in first-order logic; analogous results hold for expressive DLs [33]. Condition 4 requires that no fresh concept name is implicitly definable, ensuring that the definitorial extension introduces concepts whose interpretation is not already determined by the original signature.

Signature-based abduction is the special case where $\mathcal{X} = \emptyset$: the conditions of Definition 5 then reduce to those of Definition 4. As the following example shows, definitorial abduction can succeed where signature-based abduction has an unacceptable solution.

Example 2. *Consider $\langle \mathcal{O}, \Phi, \mathcal{F} \rangle$ with*

$$\mathcal{O} = \{\neg A \sqcup \exists s.B\},\ \Phi = \{\exists r.B\},\ \mathcal{F} = \{B\}$$

Eliminating B from the observation via weak forgetting yields $\mathcal{H} = \{\top \sqsubseteq \bot\}$. Although $\mathcal{O} \cup \mathcal{H} \models \Phi$ holds trivially, $\mathcal{H}$ violates the consistency requirement of Definition 4.

By contrast, definitorial abduction introduces a fresh concept name $\mathrm{U}_0 \notin sig(\mathcal{O}) \cup sig(\Phi)$ together with definitorial axioms that form a conservative extension of $\mathcal{O}$. Let $\mathcal{H} = \{\exists r.\mathrm{U}_0\}$ and $\mathcal{X} = \{\neg \mathrm{U}_0 \sqcup B,\ \neg A \sqcup \exists s.\mathrm{U}_0,\ \neg \mathrm{U}_0 \sqcup \exists s^{-}.A\}$. Then $sig(\mathcal{H}) \cap \mathcal{F} = \emptyset$, $\mathcal{O} \cup \mathcal{X}$ is a conservative extension of $\mathcal{O}$, and $\mathcal{O} \cup \mathcal{X} \cup \mathcal{H} \models \exists r.B$ follows from $\mathrm{U}_0 \sqsubseteq B$. This illustrates that definitorial abduction can strictly enlarge the class of solvable instances.

3. A Strong Forgetting Calculus for $\mathcal{ALCI}$

In this section, we present our strong forgetting calculus for concept names in $\mathcal{ALCI}$ ontologies. The calculus extends the Ackermann-based FAME framework [30] with a novel inference rule, the $\exists$-*surfacing rule*, that lifts pivots out of existential restrictions. Combined with the existing $\forall$-surfacing rule, the resulting calculus is *sound*, *complete*, and *terminating* for acyclic ontologies.

3.1. Normalization to A-Reduced Form

Before a pivot $A \in \mathcal{F}$ can be eliminated from a clause set, every A-clause must be brought into a normal form that exposes A to the elimination and surfacing rules. This preprocessing step transforms clauses into *A-reduced form*.

Definition 6 (A-Reduced Form)**.** *Let $A \in \mathsf{N_C}$. An $\mathcal{ALCI}$ clause is in* A-reduced form *if it has one of the following shapes: $C \sqcup A$, $C \sqcup \neg A$, $C \sqcup \exists R.(A \sqcup D)$, $C \sqcup \exists R.(\neg A \sqcup D)$, $C \sqcup \forall R.(A \sqcup D)$, or $C \sqcup \forall R.(\neg A \sqcup D)$, where R is a role and C, D are A-free concepts. A clause set $\mathcal{N}$ is in A-reduced form if every A-clause in $\mathcal{N}$ is in A-reduced form.*

In A-reduced form, the pivot A (or $\neg A$) occurs either at the top level of a clause, where it is directly accessible to Ackermann-style elimination (Section 3.2), or immediately inside the scope of exactly one role restriction, where it is accessible to the surfacing rules (Section 3.3). In the latter case, the scope may contain an A-free residual concept D alongside A; this disjunction $A \sqcup D$ (or $\neg A \sqcup D$) is the form that the surfacing rules will operate on.

If an A-clause is not in A-reduced form, it is rewritten by introducing fresh concept names called *definers* [34]. The procedure distinguishes two cases. If A occurs once and is nested too deeply, we identify a surface-level literal of the form $\exists R.E$ or $\forall R.E$ whose scope E contains A, replace E by a fresh definer $X \in \mathsf{N_C}$, and add the one-way defining clause $\neg X \sqcup E$ (i.e., $X \sqsubseteq E$). The reverse direction is not required for Lemma 1. This replacement reduces the nesting depth of A by one.

If A occurs multiple times, the procedure first selects a disjunct containing one occurrence of A. If the remainder of the clause still contains A, the selected disjunct is replaced by a fresh definer and the one-way defining clause for that disjunct is added; otherwise the filler of the selected role restriction is replaced. For example, the clause $A \sqcup \exists r.A$ is handled by selecting the top-level disjunct A, replacing it by a fresh definer D, and adding $\neg D \sqcup A$, yielding $D \sqcup \exists r.A$ and $\neg D \sqcup A$, both in A-reduced form. The process is repeated until every occurrence of A lies at the required depth. Each definer is itself treated as a pivot and added to $\mathcal{F}$ for elimination in subsequent iterations.

Lemma 1. *Let $\mathcal{N}'$ be obtained from an $\mathcal{ALCI}$ clause set $\mathcal{N}$ by normalization into A-reduced form. Then:*

1. *$\mathcal{N}'$ is a conservative extension of $\mathcal{N}$ w.r.t. sig($\mathcal{N}$).*
2. *For every model $\mathcal{I}$ of $\mathcal{N}$, there exists an expansion $\mathcal{I}'$ of $\mathcal{I}$ interpreting the introduced definers such that $\mathcal{I}' \models \mathcal{N}'$.*

Moreover, the transformation runs in polynomial time.

3.2. Ackermann-Style Elimination

After normalization, A-clauses fall into two categories. In the first, A ($\neg A$) occurs as a *top-level disjunct*, i.e., a literal that is not in the scope of any role restriction; such clauses have the form $C \sqcup A$ or $C \sqcup \neg A$. In the second, A remains inside the scope of a role restriction, yielding clauses of the form $C \sqcup \exists R.(A \sqcup D)$ or $C \sqcup \forall R.(A \sqcup D)$. The Generalized Ackermann's Lemma [31, 28] eliminates pivots from clauses of the first kind.

Lemma 2 (Generalized Ackermann's Lemma)**.** *Let $\mathcal{N}$ be a clause set containing the clauses $\neg C_1 \sqcup A, \ldots, \neg C_n \sqcup A$ or $\neg A \sqcup D_1, \ldots, \neg A \sqcup D_m$, where $A \in \mathcal{F}$ is a pivot and each C_i, D_j is A-free. If $\mathcal{N} \setminus \{\neg C_1 \sqcup A, \ldots, \neg C_n \sqcup A\}$ is negative w.r.t. A, then a solution of strongly forgetting A from $\mathcal{N}$ is:*

$$\frac{\mathcal{N} \setminus \{\neg C_1 \sqcup A, \ldots, \neg C_n \sqcup A\},\ \neg C_1 \sqcup A, \ldots, \neg C_n \sqcup A}{\mathcal{N} \setminus \{\neg C_1 \sqcup A, \ldots, \neg C_n \sqcup A\}\ {}^{\neg A}_{\neg C_1 \sqcap \cdots \sqcap \neg C_n}}$$

Similarly, if $\mathcal{N} \setminus \{\neg A \sqcup D_1, \ldots, \neg A \sqcup D_m\}$ is positive w.r.t. A, then a solution of strongly forgetting A from $\mathcal{N}$ is:

$$\frac{\mathcal{N} \setminus \{\neg A \sqcup D_1, \ldots, \neg A \sqcup D_m\},\ \neg A \sqcup D_1, \ldots, \neg A \sqcup D_m}{\mathcal{N} \setminus \{\neg A \sqcup D_1, \ldots, \neg A \sqcup D_m\}\ {}^{A}_{D_1 \sqcap \cdots \sqcap D_m}}$$

The fraction bar in Lemma 2 denotes a derivation: the clause set above the bar is the input and the clause set below the bar is the output. The "solution" is the output clause set, i.e., the resulting ontology after the indicated substitution.

Theorem 1. *The Generalized Ackermann's Lemma produces a strong forgetting solution: the resulting clause set is $\mathcal{F}$-equivalent to the input (Definition 3).*

Ackermann's Lemma requires the pivot to occur as a top-level disjunct in every A-clause. For A-clauses of the second category, where A remains inside a role restriction, the next subsection introduces rules that lift A to the top level so that Ackermann's Lemma becomes applicable.

3.3. Lifting Pivots from Role Restrictions

When A remains inside the scope of a role restriction after normalization, it must be *lifted* to the top level before Ackermann's Lemma can be applied. Universal and existential restrictions require fundamentally different treatments.

The $\forall$-surfacing rule. If the pivot A occurs inside a universal restriction, the $\forall$-surfacing rule (Figure 1) lifts A to the top level by exploiting the duality between $\forall$ and $\exists$ together with the semantics of inverse roles. A now appears as a top-level disjunct, ready for Ackermann-style elimination. Note that this is the same rule as in [28], written in a pivot-oriented form.

Lemma 3. *The $\forall$-surfacing rule preserves equivalence.*

The $\exists$-surfacing rule. If A occurs inside an existential restriction $\exists r.(A \sqcup D)$, the $\forall$-surfacing strategy does not apply: rewriting $\exists r.(A \sqcup D)$ as $\neg\forall r.(\neg A \sqcap \neg D)$ merely moves A from inside $\exists$ to inside $\forall$ under negation, without bringing it to the top level. When the pivot occurs under existential restrictions in both positive and negative polarity, no existing method in the Ackermann framework can eliminate it.

In logics with nominals (e.g., $\mathcal{ALCO}\mathcal{I}$), this situation can be handled via Skolemization [28]: a fresh nominal b is introduced to name an individual existential witness, yielding clauses $\neg a \sqcup \exists R.b$ and $\neg b \sqcup \neg E$. Because a nominal is interpreted as a singleton, it precisely identifies one witness and no further constraint is needed to tie the witness back to its source. However, $\mathcal{ALCI}$ does not have nominals, so Skolemization is not available.

We introduce the $\exists$-surfacing rule (Figure 1) to handle this case within $\mathcal{ALCI}$. Instead of naming an individual witness via a nominal, the rule introduces a fresh concept name U that denotes the *set* of existential witnesses. The three conclusion clauses play the following roles:

1. $C \sqcup \exists r.\mathrm{U}$: the original existential witness is now provided by an element of U.
2. $\neg\mathrm{U} \sqcup A \sqcup D$: every element of U satisfies $A \sqcup D$, inheriting the content constraint from the original filler.
3. $\neg\mathrm{U} \sqcup \exists r^{-}.\neg C$: every element of U is an r-successor of some element satisfying $\neg C$, tying each witness back to its source.

$\forall$-Surfacing Rule

$$\frac{\mathcal{N},\ C \sqcup \forall r.(A \sqcup D)}{\mathcal{N},\ A \sqcup D \sqcup \forall r^{-}.C}$$

$\exists$-Surfacing Rule

$$\frac{\mathcal{N},\ C \sqcup \exists r.(A \sqcup D)}{\mathcal{N},\ C \sqcup \exists r.\mathrm{U},\ \neg\mathrm{U} \sqcup A \sqcup D,\ \neg\mathrm{U} \sqcup \exists r^{-}.\neg C}$$

Provisos:

(i) $A \in \mathcal{F}$ is the pivot being eliminated,
(ii) A does not occur in C,
(iii) U is a fresh concept name not occurring in $\mathcal{N}$.

Figure 1: The quantifier surfacing rules for lifting a pivot A out of a role restriction.

The third clause, which we call the *backward clause*, is not required for the conservativity and expandability properties below: the two-clause variant without it is also a valid projected representation under Definition 3. We include the backward clause because it yields more informative unnamed concepts. Unlike a nominal, which is interpreted as a singleton and therefore automatically identifies its source, a concept name denotes an arbitrary subset of the domain. The backward clause constrains U to elements that serve as existential witnesses for elements of $\neg C$, adding a differentia to the broader constraint $\mathrm{U} \sqsubseteq A \sqcup D$. The pivot A now appears at the top level in the second conclusion clause.

Unnamed concept names vs. definers. The fresh U introduced by the $\exists$-surfacing rule must be distinguished from a definer introduced during normalization (Section 3.1). A definer is a syntactic abbreviation that replaces a complex subconcept; it is added to $\mathcal{F}$ and eliminated in subsequent iterations. An unnamed concept U, by contrast, carries semantic constraints imposed by the backward clause $\neg\mathrm{U} \sqcup \exists r^{-}.\neg C$. It is *not* added to $\mathcal{F}$; it persists in the forgetting result and, in the context of abduction (Section 2.3), becomes part of the definitorial extension $\mathcal{X}$. Such witness-style names for existential structure also appear in other DL settings, e.g., in category-theoretic semantics [35].

Semantic properties. The $\forall$-surfacing rule preserves logical equivalence (Lemma 3). The $\exists$-surfacing rule does not preserve equivalence, but it yields a conservative extension with the expandability property required by strong forgetting. The following lemma makes this precise.

Lemma 4. *Let $\mathcal{N}'$ be obtained from a clause set $\mathcal{N}$ by applying the $\exists$-surfacing rule, introducing a fresh concept name U. Then:*

1. *$\mathcal{N}'$ is a conservative extension of $\mathcal{N}$ w.r.t. sig($\mathcal{N}$).*
2. *For every model $\mathcal{I}$ of $\mathcal{N}$, there exists an expansion $\mathcal{I}'$ of $\mathcal{I}$ interpreting U such that $\mathcal{I}' \models \mathcal{N}'$.*

Proposition 1 (Non-definability of retained surfacing names)**.** *Every unnamed concept name U retained from a non-vacuous application of the $\exists$-surfacing rule, i.e., one whose source concept $\neg C$ is satisfiable in the current clause set, satisfies the non-definability condition of Definition 5: it is not implicitly definable from sig($\mathcal{O}$) $\cup$ sig(Φ) under $\mathcal{O} \cup \mathcal{X}$.*

Proof sketch. The rule never introduces a synonym of the form $\mathrm{U} \equiv B$ for an old concept name B. Instead, it constrains U through witness conditions, in particular $\mathrm{U} \sqsubseteq A \sqcup D$ and $\mathrm{U} \sqsubseteq \exists r^{-}.\neg C$, while leaving freedom in which eligible witnesses are included. Hence two expansions can agree on the original signature while assigning different admissible subsets to U. □

Proposition 2 (Genus-differentia structure)**.** *Every unnamed concept name U introduced by the $\exists$-surfacing rule has concepts G_{U} and D_{U} over the original signature such that $\mathcal{O} \cup \mathcal{X} \models \mathrm{U} \sqsubseteq G_{\mathrm{U}} \sqcap D_{\mathrm{U}}$.*

Proof sketch. For the rule in Figure 1, take $G_{\mathrm{U}} = A \sqcup D$ as the genus and $D_{\mathrm{U}} = \exists r^{-}.\neg C$ as the differentia. The two clauses $\neg \mathrm{U} \sqcup A \sqcup D$ and $\neg \mathrm{U} \sqcup \exists r^{-}.\neg C$ entail $\mathrm{U} \sqsubseteq (A \sqcup D) \sqcap \exists r^{-}.\neg C$. This genus-differentia pattern is a property of Algorithm 1's chosen representation, not an additional requirement in the definition of definitorial abduction. □

By repeatedly applying the $\forall$- and $\exists$-surfacing rules, all occurrences of A can be lifted out of role restrictions until A appears only at the top level of clauses, at which point Ackermann's Lemma (Lemma 2) becomes applicable to eliminate A. The complete procedure for strongly forgetting concept names from an $\mathcal{ALCI}$ ontology thus processes each pivot $A \in \mathcal{F}$ in turn: (1) normalize the current clause set into A-reduced form, introducing definers as needed (Section 3.1); (2) apply the surfacing rules to lift A from inside role restrictions to the top level; and (3) apply Ackermann's Lemma to eliminate A. Definers introduced during normalization are added to $\mathcal{F}$ and eliminated in subsequent iterations. Unnamed concept names introduced by the $\exists$-surfacing rule are not added to $\mathcal{F}$; they persist in the result.

Theorem 2 (Soundness, Completeness, and Termination)**.** *The forgetting calculus consisting of the normalization, the quantifier surfacing rules, and the Generalized Ackermann's Lemma is sound, complete, and terminating for strong forgetting of concept names in acyclic $\mathcal{ALCI}$ ontologies.*

4. A Constructive Procedure for Definitorial Abduction in $\mathcal{ALCI}$

With the forgetting calculus in place, we now connect strong forgetting to abduction. A standard forgetting-based construction follows the *negate–forget–negate* scheme: negate the observation Φ, forget the forgetting signature from $\mathcal{O} \cup \neg\Phi$, and negate the resulting view (ontology) to obtain the hypothesis [8]. In DLs, this scheme is not directly available at the TBox level, since there is no native negation of a GCI. It can be implemented by encoding $\neg\Phi$ at the ABox level, e.g., $\neg(A \sqsubseteq B)$ is represented as $(\exists \nabla.(A \sqcap \neg B))(a)$ using a fresh individual a and the universal role ∇. Our method avoids this encoding entirely: it stays at the TBox level and works directly with $\mathcal{O}$ and Φ.

The key idea is to eliminate each pivot $A \in \mathcal{F}$ from the observation Φ by substituting it with definable concepts derived from $\mathcal{O}$, guided by the polarity of its occurrences. The forgetting calculus supplies the necessary bounds: after reducing $\mathcal{O}$ into A-reduced form and applying the surfacing rules, all occurrences of A become top-level, and the derived clause set contains only clauses of the form $\neg X \sqcup A$ and $\neg A \sqcup Y$, where X and Y are A-free. A clause $\neg X \sqcup A$ is equivalent to the GCI $X \sqsubseteq A$, so X is a *definable sub-concept* (lower bound) of A; a clause $\neg A \sqcup Y$ is equivalent to $A \sqsubseteq Y$, so Y is a *definable super-concept* (upper bound) of A. The disjunction of all lower bounds, $Def_A^{+} = X_1 \sqcup \cdots \sqcup X_n$, is the maximal definable sub-concept of A w.r.t. $\mathcal{O}$; the conjunction of all upper bounds, $Def_A^{-} = Y_1 \sqcap \cdots \sqcap Y_m$, is the minimal definable super-concept. Because abduction seeks the weakest sufficient hypothesis, a positive occurrence of A in Φ (where A appears as a consequent that the hypothesis must entail) is replaced by the strongest available approximation from below, while a negative occurrence (where A appears as a precondition) is replaced by the weakest approximation from above.

Lemma 5 (Polarity-Guided Substitution)**.** *Let $A \in \mathcal{F}$ be a pivot to be eliminated from Φ. Suppose the forgetting calculus derives from $\mathcal{O}$ the clauses $\neg C_1 \sqcup A, \ldots, \neg C_n \sqcup A$ (so that $\mathcal{O} \models C_i \sqsubseteq A$ for each i) and $\neg A \sqcup D_1, \ldots, \neg A \sqcup D_m$ (so that $\mathcal{O} \models A \sqsubseteq D_j$ for each j). Define:*

$$Def_A^{+} = C_1 \sqcup \cdots \sqcup C_n, \qquad Def_A^{-} = D_1 \sqcap \cdots \sqcap D_m.$$

Then an optimal hypothesis is obtained by replacing every positive occurrence of A in Φ with Def_A^{+} and every negative occurrence with Def_A^{-}.

When the forgetting calculus introduces unnamed concept names via the $\exists$-surfacing rule during the derivation of these bounds, the corresponding definitorial axioms are collected into $\mathcal{X}$, ensuring that every fresh concept name is accompanied by an explicit defining theory.

Algorithm 1 Constructive Definitorial Abduction

Require: Background ontology $\mathcal{O}$, observation Φ, forgetting signature $\mathcal{F}$
Ensure: A pair $\langle \mathcal{H}, \mathcal{X} \rangle$

if $\mathcal{O} \models \Phi$ **then**
  return "Nothing to explain"
else if $\mathcal{O} \cup \Phi \models \bot$ **then**
  return "Inconsistent"
else
  $\mathcal{O} \leftarrow \mathsf{Simplify}(\mathcal{O})$ and $\Phi \leftarrow \mathsf{Simplify}(\Phi)$
  Initialize $\mathcal{H} \leftarrow \Phi$, $\mathcal{X} \leftarrow \emptyset$
  Initialize $\mathcal{D} \leftarrow \emptyset$
  $\mathcal{F}_{\text{obs}} \leftarrow \{A \in \mathcal{F} \cup \mathcal{D} \mid A \text{ occurs in } \mathcal{H}\}$
  while $\mathcal{F}_{\text{obs}} \neq \emptyset$ **do**
    Pick some $A \in \mathcal{F}_{\text{obs}}$
    if $A \notin \mathsf{sig}(\mathcal{O})$ **then**
      Replace positive occurrences of A in $\mathcal{H}$ by $\bot$
      Replace negative occurrences of A in $\mathcal{H}$ by $\top$
      $\mathcal{H} \leftarrow \mathsf{Simplify}(\mathcal{H})$
    else
      $(\mathcal{O}_A, \mathcal{D}_A) \leftarrow \mathsf{Reduce}(\mathcal{O}, A)$
      Apply surfacing to $\mathcal{O}_A$; collect unnamed-concept axioms $\Delta\mathcal{X}$
      Collect A^+-clauses $\neg C_i \sqcup A$ and A^--clauses $\neg A \sqcup D_j$
      $Def_A^+ \leftarrow C_1 \sqcup \cdots \sqcup C_n$ and $Def_A^- \leftarrow D_1 \sqcap \cdots \sqcap D_m$
      $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_A$ and $\mathcal{X} \leftarrow \mathcal{X} \cup \Delta\mathcal{X}$
      Replace positive occurrences of A in $\mathcal{H}$ by Def_A^+
      Replace negative occurrences of A in $\mathcal{H}$ by Def_A^-
      $\mathcal{H} \leftarrow \mathsf{Simplify}(\mathcal{H})$
      $\mathcal{O} \leftarrow \mathsf{StrongForget}(A, \mathcal{O}_A)$
    end if
    $\mathcal{F}_{\text{obs}} \leftarrow \{B \in \mathcal{F} \cup \mathcal{D} \mid B \text{ occurs in } \mathcal{H}\}$
  end while
  if $\mathcal{O} \cup \mathcal{X} \cup \mathcal{H} \models \bot$ **then**
    return "Inconsistent explanation"
  end if
  return $\langle \mathcal{H}, \mathcal{X} \rangle$
end if

We note that this polarity-guided substitution is not an ad hoc construction but a TBox-level counterpart of the classical negate–forget–negate scheme. In that scheme, a view $\mathcal{V}$ is computed from $\mathcal{O} \cup \neg\Phi$ after eliminating the forgetting signature, and the hypothesis is taken as $\neg\mathcal{V}$. Polarity-guided substitution achieves the same effect: definable lower bounds in $\mathcal{O}$ replace positive occurrences of a pivot, while definable upper bounds replace negative occurrences.

Lemma 6. *Let $\langle \mathcal{O}, \Phi, \mathcal{F} \rangle$ be an abduction instance. Let $\mathcal{V} := \mathsf{Forget}_{\mathcal{F}}(\mathcal{O} \cup \neg\Phi)$ and $\mathcal{H}_{\text{nfn}} := \neg\mathcal{V}$. Let $\mathcal{H}_{\text{sub}}$ be obtained from Φ by polarity-guided substitution using definable lower/upper bounds induced by $\mathcal{O}$. Then $\mathcal{H}_{\text{sub}}$ coincides with $\mathcal{H}_{\text{nfn}}$.*

The entire procedure is summarized in Algorithm 1. Here Simplify denotes clausal simplification only: removing tautological clauses and simplifying literals involving $\top$ and $\bot$. Definer introduction is performed by $\mathsf{Reduce}(\mathcal{O}, A)$ inside the loop, because the required A-reduced form is pivot-specific. After the initial entailment and consistency checks, the algorithm simplifies $\mathcal{O}$ and Φ and then processes pivots iteratively. For each $A \in \mathcal{F} \cup \mathcal{D}$ that still occurs in $\mathcal{H}$, two cases arise. If $A \notin \mathsf{sig}(\mathcal{O})$, then $\mathcal{O}$ provides no constraint for deriving definable bounds; the algorithm falls back to *purification*, replacing positive occurrences of A in $\mathcal{H}$ by $\bot$ and negative occurrences of A by $\top$. This is the degenerate case of Lemma 5: with no lower-bound clauses, Def_A^+ reduces to $\bot$ (the empty disjunction), and with no upper-bound clauses, Def_A^- reduces to $\top$ (the empty conjunction). We note that this direction

is opposite to the purification used in forgetting, where one weakens the ontology (positive $\to \top$, negative $\to \bot$); here the goal is to produce the weakest sufficient hypothesis from Φ. If $A \in \mathrm{sig}(\mathcal{O})$, the algorithm first reduces $\mathcal{O}$ to A-reduced form, collecting any definers into $\mathcal{D}$. It then applies surfacing, collects the A^+-clauses $\neg C_i \sqcup A$ and A^--clauses $\neg A \sqcup D_j$, and computes $Def_A^+ = C_1 \sqcup \cdots \sqcup C_n$ and $Def_A^- = D_1 \sqcap \cdots \sqcap D_m$. These bounds drive the polarity-guided substitution of Lemma 5. When the surfacing step introduces unnamed concept names, the corresponding definitorial axioms are accumulated in $\mathcal{X}$.

After each substitution step, A is also strongly forgotten from $\mathcal{O}$. This prevents A from being reintroduced into $\mathcal{H}$ in later iterations through subsequent substitutions. The loop terminates when no symbol from $\mathcal{F} \cup \mathcal{D}$ occurs in $\mathcal{H}$. A final consistency check ensures that the combined ontology $\mathcal{O} \cup \mathcal{X} \cup \mathcal{H}$ is consistent; if not, the procedure reports inconsistency. Otherwise, it returns the pair $\langle \mathcal{H}, \mathcal{X} \rangle$ satisfying $\mathrm{sig}(\mathcal{H}) \cap \mathcal{F} = \emptyset$ and $\mathcal{O} \cup \mathcal{X} \cup \mathcal{H} \models \Phi$.

Choice of forgetting signature. The behavior of Algorithm 1 depends on the choice of $\mathcal{F}$. If a symbol occurs in both $\mathcal{O}$ and Φ, then $\mathcal{O}$ provides definable lower/upper bounds, and polarity-guided substitution eliminates it in an informative way. If a symbol occurs only in the observation ($A \in \mathrm{sig}(\Phi) \setminus \mathrm{sig}(\mathcal{O})$), then $\mathcal{O}$ offers no constraint, and elimination falls back to purification. Since purification removes concept names specific to Φ, the forgetting signature $\mathcal{F}$ is typically chosen among symbols present in $\mathcal{O}$, and symbols appearing only in Φ are excluded.

Auxiliary symbols. Two types of fresh concepts may arise during the procedure: definers and unnamed concept names. Definers are introduced during normalization (Section 3.1); they are added to $\mathcal{F} \cup \mathcal{D}$ and eliminated in subsequent iterations. Unnamed concept names are introduced by the $\exists$-surfacing rule (Section 3.3); they are *not* added to $\mathcal{F}$ and persist in both $\mathcal{H}$ and $\mathcal{X}$. The following example illustrates why unnamed concept names must not be treated as pivots.

Example 3. *Let A be the pivot; suppose the clause set is*

$$C \sqcup \exists r.A, \quad E \sqcup \exists r.\neg A.$$

After applying the $\exists$-surfacing rule to lift A, we obtain

$$C \sqcup \exists r.\mathrm{U}_0, \quad \neg \mathrm{U}_0 \sqcup A, \quad \neg \mathrm{U}_0 \sqcup \exists r^-.\neg C, \quad E \sqcup \exists r.\neg A.$$

Ackermann's Lemma (Lemma 2) then eliminates A, yielding

$$C \sqcup \exists r.\mathrm{U}_0, \quad \neg \mathrm{U}_0 \sqcup \exists r^-.\neg C, \quad E \sqcup \exists r.\neg \mathrm{U}_0.$$

If one were to treat U_0 as a pivot and attempt to eliminate it, the situation would reproduce: U_0 now occurs both positively and negatively under existential restrictions, just like the original pivot A, so eliminating U_0 would introduce yet another unnamed concept name, leading to infinite regress.

Theorem 3. *Algorithm 1 is sound, terminating, and $\mathcal{X}$-complete for acyclic $\mathcal{ALCI}$ ontologies: given the definitorial extension $\mathcal{X}$ computed by the forgetting calculus, it produces an $\mathcal{X}$-optimal hypothesis whenever a consistent one exists under $\mathcal{X}$, and reports failure when no consistent hypothesis exists under that $\mathcal{X}$.*

The qualifier "under $\mathcal{X}$" is important. The $\exists$-surfacing rule deterministically selects a definitorial extension, but strong forgetting may admit different finite presentations of the same projected model set. Global optimality across all possible choices of $\mathcal{X}$ is therefore a separate selection problem, left for future work.

5. Experimental Evaluation

5.1. Experimental Setup

The proposed method was implemented as a Java prototype built upon the OWLAPI.[1] Ontologies were selected from the "DL Instantiation" track of the OWL Reasoner Competition 2015 [36] to reflect real application settings.

For each ontology, 10 GCIs were randomly sampled from the original ontology to serve as the observation Φ, and removed from the background ontology $\mathcal{O}$. The forgetting signature $\mathcal{F}$ was randomly selected from $\text{sig}(\mathcal{O})$, subject to the constraint that $\geq$1 concept name in $\mathcal{F}$ appears in both $\mathcal{O}$ and Φ. This ensures that the computed hypothesis bridges the two rather than reduce to a trivial replacement.

All experiments were conducted on a MacBook Air laptop equipped with an Apple M3 chip and 24 GB of unified memory, running OpenJDK 23.0.1. The timeout for each abduction task was set to 500 seconds. Consistency checking was performed using the HermiT reasoner [37], with a separate timeout of 120 seconds.

5.2. Results and Analysis

To demonstrate general applicability, we evaluated the prototype on 547 ontologies of varying sizes (from 55 to 41 296 axioms), covering both lightweight and large-scale cases. Each ontology was restricted to its $\mathcal{ALCI}$ fragment by discarding axioms that exceed the logic's expressivity.

The experiments were run under three forgetting ratios, $|\mathcal{F}|/|\mathsf{N_C} \cap \text{sig}(\mathcal{O})|$: 10%, 30%, and 50%. The average observation size was kept consistent at $\approx$7.7 axioms across ratios, so that results primarily reflect the impact of the forgetting signature rather than input complexity. Tables 1 and 2 summarize the results; we discuss each group of metrics in turn.

Hypothesis size. The average hypothesis size (AvgSz) remains stable across forgetting ratios, indicating that the complexity of individual clauses does not grow with $|\mathcal{F}|$. The clause count, however, exhibits a heavy tail: while the 90th percentile stays below 20 at every ratio, a small number of structurally complex ontologies produce clause sets orders of magnitude larger. This suggests that hypothesis blowup is driven by ontology structure rather than by the forgetting ratio itself. The definer ratio is 0.00% at every ratio, confirming that the normalization-elimination loop successfully removes all auxiliary definers.

Unnamed concepts and definitorial extensions. Each unnamed concept name U_i introduced by the $\exists$-surfacing rule triggers exactly three definitorial axioms in $\mathcal{X}$ (Section 3.3), so the two ratios in Table 2 coincide by construction, and $\mathcal{X}$ scales linearly with the number of unnamed concept names. Notably, 20–29% of runs across all three forgetting ratios introduce at least one unnamed concept name. This is the empirical counterpart of the theoretical motivation in Section 1: in roughly a quarter of all instances, weak forgetting shows degeneration, and definitorial abduction is needed to produce an informative hypothesis.

Runtime. This analysis is restricted to runs that completed successfully under three ratios. Runtime grows roughly linearly with the forgetting ratio (Table 1), yet the median stays below 3s even at the 50% ratio, and 90% of instances finish within 80s. The heavy tail (max $\approx$500s) is attributable to a small number of large or structurally complex ontologies.

Failures. Failure rates increase with $|\mathcal{F}|$, driven primarily by timeouts and cyclicity. The failure categories have different status. *Contr* ($\mathcal{O} \cup \Phi \models \bot$) is inherent: no consistent explanation can exist because the observation conflicts with the background ontology. *Ent* ($\mathcal{O} \models \Phi$) is also inherent, but benign: there is simply nothing to explain; it is consistently 0%, as expected, since the observation is

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

Table 1
Runtime, failure modes, and success rate across forgetting ratios (547 ontologies). Runtime is in seconds. Failure columns report the percentage of runs that do not produce an explanation pair: Cyclic = acyclicity assumption violated; Contr = $\mathcal{O} \cup \Phi \models \bot$; TO = timeout (>500 s); OOM = out of memory; Ent = $\mathcal{O} \models \Phi$. SR = success rate (%).

Ratio	Runtime (s)				Failure (%)					SR (%)
	Avg	Med	P90	Max	Cyclic	Contr	TO	OOM	Ent	
10%	5.85	0.71	10.41	121.92	3.10	2.92	4.74	0.55	0.00	69.11
30%	13.24	0.97	27.82	432.08	8.21	2.92	13.14	1.64	0.00	44.94
50%	31.09	2.27	79.90	497.55	8.94	2.37	21.35	3.65	0.00	23.75

Table 2
Output characterization across forgetting ratios (547 ontologies). For the hypothesis $\mathcal{H}$: AvgSz = average number of symbols per clause; clause-count statistics (Avg, Med, P90, Max). For unnamed concept names: % w/ = percentage of runs introducing at least one. For the definitorial extension $\mathcal{X}$: clause-count statistics. The definer ratio is 0.00% at all three ratios (all definers are successfully eliminated). The minimum clause count is 1 for $\mathcal{H}$ and 3 for $\mathcal{X}$ at every ratio.

Ratio	Hypothesis $\mathcal{H}$					Unnamed Concept Names				Def. Extension $\mathcal{X}$			
	AvgSz	AvgCl	MedCl	P90Cl	MaxCl	% w/	Med	P90	Max	AvgCl	MedCl	P90Cl	MaxCl
10%	6.31	16.63	15	20	242	20.00	2.00	11.40	275	20.32	6.00	34.20	825
30%	6.36	65.16	14	19	18 624	29.44	4.50	19.50	140	30.15	13.50	58.50	420
50%	4.47	21.48	1	17	4 127	20.12	7.50	21.00	31	26.53	21.00	63.00	93

sampled from $\mathcal{O}$ and then removed. The remaining categories are algorithmic or resource-bounded: *Cyclic* means the acyclicity assumption of Theorem 2 is violated, while *TO* and *OOM* reflect time and memory limits rather than non-existence of a definitorial abduction solution. Even at the 50% ratio, cyclicity affects fewer than 9% of instances, suggesting that acyclicity is a reasonable assumption for the benchmark suite.

Success rate (SR). SR is measured over runs for which the solver returns $\langle \mathcal{H}, \mathcal{X} \rangle$ and the consistency and entailment checks complete within their timeout. A run counts as successful iff (i) $\mathcal{O} \cup \mathcal{X} \cup \mathcal{H} \not\models \bot$ and (ii) $\mathcal{O} \cup \mathcal{X} \cup \mathcal{H} \models \Phi$. Thus SR measures verified success under the particular $\mathcal{X}$ selected by Algorithm 1, consistent with the $\mathcal{X}$-complete guarantee in Theorem 3. The decline from 69% to 24% has two main causes. First, a larger $\mathcal{F}$ increases the chance of a *polarity mismatch*: if a concept name A occurs negatively in Φ but only positively in $\mathcal{O}$, elimination collapses the negative occurrence to $\bot$ via purification, producing a hypothesis that is logically consistent but explanatorily degenerate. Second, a larger forgetting signature simply leaves fewer concept names available to formulate the hypothesis.

5.3. Comparison with Lethe

To compare with existing methods, we utilized the Lethe-based abduction tool[2] [8] as a baseline. Lethe implements weak forgetting for $\mathcal{ALCI}$ and is the only available tool for signature-based TBox abduction. We restricted the comparison to ontologies for which both systems returned an explanation pair, yielding 34 ontologies. For both systems, 30% of the concept names were randomly selected as the forgetting signature. For each ontology, 10 axioms were randomly sampled as the observation and removed from the background ontology.

[2] https://lat.inf.tu-dresden.de/evaluation-kr2020-dl-abduction/

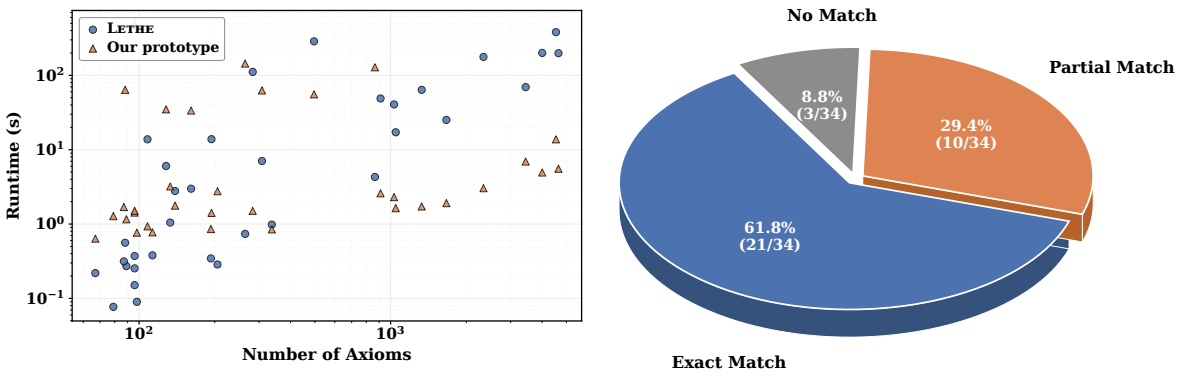

Figure 2: Comparison of our prototype and Lethe on 34 ontologies (30% forgetting ratio). Left: runtime against ontology size. Right: hypothesis match distribution.

Runtime comparison. Figure 2 (left) plots the runtime of both systems against ontology size. Our prototype is on average 36% faster (31.52 s vs. 49.32 s), with the advantage growing on larger ontologies. The visible scatter for small ontologies reflects the fact that runtime is not solely determined by ontology size: instances that trigger the $\exists$-surfacing rule incur additional overhead from constructing unnamed concept names and their definitorial axioms, and this cost can dominate on small inputs.

Hypothesis comparison. We compared the hypotheses produced by both systems by viewing each as a set of clauses and classifying each pair into three categories. *Exact Match* means the two clause sets are identical: the two systems derive the same explanation. *Partial Match* means the clause sets overlap but neither subsumes the other: they agree on part of the explanation but diverge on the rest. *No Match* means the clause sets are disjoint: the two explanations share no common clause. As Figure 2 (right) shows, the dominant outcome is Exact Match (61.8%), which occurs when the $\exists$-surfacing rule is not triggered (and $\mathcal{X} = \emptyset$); in these cases strong and weak forgetting coincide. Unnamed concept names appeared in 23.53% (8/34) of instances. Among the 13 non-exact-match cases, 8 are explained by unnamed concept names produced by our prototype (i.e., cases where weak forgetting degenerates but definitorial abduction succeeds), 3 by differing normal-form treatments of inverse roles, and in the remaining 2 Lethe returned $\bot$ while our prototype produced a non-degenerate hypothesis. These results confirm that definitorial abduction strictly subsumes standard signature-based abduction: it agrees with weak forgetting when sufficient, and extends it when necessary.

6. Conclusion and Future Work

We have introduced definitorial abduction, a generalization of signature-based TBox abduction that permits fresh concept names anchored by conservative definitorial extensions. Its computational backbone is a complete Ackermann-style strong forgetting calculus for acyclic $\mathcal{ALCI}$, whose key component, the $\exists$-surfacing rule, lifts pivots out of existential restrictions where no prior Ackermann-based method could. The calculus operates entirely at the TBox level, avoiding the nominal or ABox encodings required by existing approaches. Evaluation on 547 real-world ontologies confirms that roughly a quarter of abduction instances demand fresh concept names to avoid degenerate explanations.

Several directions remain open: extending the calculus to cyclic ontologies and richer DLs such as $\mathcal{SHIQ}$; tightening the complexity analysis, where we conjecture a triply-exponential worst case; and developing a ranking mechanism for the generated unnamed concept names to improve readability in practical ontology engineering.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT and Claude for grammar checking, sentence polishing, and rephrasing. After using these tools/services, the author(s) reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017.

[2] C. S. Peirce, Deduction, induction, and hypothesis, Popular Science Monthly 13 (1878) 470–482.

[3] M. Bienvenu, Complexity of Abduction in the $\mathcal{EL}$ Family of Lightweight Description Logics, in: Proc. KR'08, AAAI Press, 2008, pp. 220–230.

[4] C. Elsenbroich, O. Kutz, U. Sattler, A Case for Abductive Reasoning over Ontologies, in: Proc. OWLED'06, volume 216 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2006.

[5] S. Klarman, U. Endriss, S. Schlobach, ABox Abduction in the Description Logic $\mathcal{ALC}$, Journal of Automated Reasoning 46 (2011) 43–80.

[6] B. Glimm, Y. Kazakov, M. Welt, Concept Abduction for Description Logics, in: Proc. DL'12, volume 3263 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022.

[7] W. Del-Pinto, R. A. Schmidt, ABox Abduction via Forgetting in $\mathcal{ALC}$, in: Proc. AAAI'19, AAAI Press, 2019, pp. 2768–2775.

[8] P. Koopmann, W. Del-Pinto, S. Tourret, R. A. Schmidt, Signature-Based Abduction for Expressive Description Logics, in: Proc. KR'20, 2020, pp. 592–602.

[9] F. Lin, On strongest necessary and weakest sufficient conditions, Artificial Intelligence 128 (2001) 143–159.

[10] D. Poole, Probabilistic Horn abduction and Bayesian networks, Artificial Intelligence 64 (1993) 81–129.

[11] T. D. Noia, E. D. Sciascio, F. M. Donini, M. Mongiello, Abductive Matchmaking using Description Logics, in: Proc. IJCAI'03, Morgan Kaufmann, 2003, pp. 337–342.

[12] A. Kaya, S. Melzer, R. Möller, S. Espinosa, M. Wessel, Towards a Foundation for Knowledge Management: Multimedia Interpretation as Abduction, in: Proc. DL'07, volume 250 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2007.

[13] F. Wei-Kleiner, Z. Dragisic, P. Lambrix, Abduction Framework for Repairing Incomplete $\mathcal{EL}$ Ontologies: Complexity Results and Algorithms, in: Proc. AAAI'14, AAAI Press, 2014, pp. 1120–1127.

[14] H. B. Enderton, A mathematical introduction to logic, Academic Press, 1972.

[15] F. Lin, R. Reiter, Forget It!, in: Proc. AAAI Fall Symposium on Relevance, AAAI Press, 1994, pp. 154–159.

[16] Y. Zhang, Y. Zhou, Forgetting Revisited, in: Proc. KR'10, AAAI Press, 2010.

[17] J. C. Jung, P. Koopmann, M. Knorr, Interpolation in Knowledge Representation, CoRR abs/2512.08833 (2025).

[18] S. Ghilardi, C. Lutz, F. Wolter, Did I Damage My Ontology? A Case for Conservative Extensions in Description Logics, in: Proc. KR'06, AAAI Press, 2006, pp. 187–197.

[19] C. S. Peirce, Collected Papers of Charles Sanders Peirce, volume 1–6, Belknap Press of Harvard University Press, Cambridge, MA, 1932.

[20] C. G. Hempel, The Theoretician's Dilemma: A Study in the Logic of Theory Construction, in: Concepts, Theories, and the Mind-Body Problem, volume II of *Minnesota Studies in the Philosophy of Science*, University of Minnesota Press, 1958, pp. 37–98.

[21] G. Schurz, Patterns of Abduction, Synthese 164 (2008) 201–234.

[22] S. Muggleton, W. Buntine, Machine Invention of First-Order Predicates by Inverting Resolution, in: Proc. ICML'88, Morgan Kaufmann, 1988, pp. 339–352.

[23] I. Stahl, The Appropriateness of Predicate Invention as Bias Shift Operation in ILP, Mach. Learn. 20 (1995) 95–117.
[24] P. Koopmann, Signature-Based Abduction with Fresh Individuals and Complex Concepts for Description Logics, in: Proc. IJCAI'21, ijcai.org, 2021, pp. 1929–1935.
[25] J. R. Shoenfield, Mathematical Logic, Addison-Wesley, 1967. Reprinted by AK Peters, 2001.
[26] C. Lutz, F. Wolter, Foundations for Uniform Interpolation and Forgetting in Expressive Description Logics, in: Proc. IJCAI'11, IJCAI/AAAI, 2011, pp. 989–995.
[27] B. Konev, C. Lutz, D. Walther, F. Wolter, Model-theoretic inseparability and modularity of description logic ontologies, Artificial Intelligence 203 (2013) 66–103.
[28] Y. Zhao, R. A. Schmidt, Forgetting Concept and Role Symbols in $\mathcal{ALCOIH}\mu^+(\nabla, \sqcap)$-Ontologies, in: Proc. IJCAI'16, IJCAI/AAAI Press, 2016, pp. 1345–1353.
[29] R. A. Schmidt, The ackermann approach for modal logic, correspondence theory and second-order reduction, J. Appl. Log. 10 (2012) 52–74.
[30] Y. Zhao, R. A. Schmidt, FAME: An Automated Tool for Semantic Forgetting in Expressive Description Logics, in: Proc. IJCAR'18, volume 10900 of *LNCS*, Springer, 2018, pp. 19–27.
[31] W. Ackermann, Untersuchungen über das Eliminationsproblem der mathematischen Logik, Mathematische Annalen 110 (1935) 390–413.
[32] F. Baader, S. Brandt, C. Lutz, Pushing the $\mathcal{EL}$ Envelope, in: Proc. IJCAI'05', Professional Book Center, 2005, pp. 364–369.
[33] B. ten Cate, E. Franconi, I. Seylan, Beth Definability in Expressive Description Logics, in: Proc. IJCAI'11, IJCAI/AAAI, 2011, pp. 1099–1106.
[34] P. Koopmann, LETHE: Forgetting and Uniform Interpolation for Expressive Description Logics, KI – Künstliche Intelligenz 34 (2020) 381–387.
[35] C. L. Duc, Category-theoretical Semantics of the Description Logic $\mathcal{ALC}$, in: M. Homola, V. Ryzhikov, R. A. Schmidt (Eds.), Proc. DL'21, volume 2954 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021.
[36] B. Parsia, N. Matentzoglu, R. S. Gonçalves, B. Glimm, A. Steigmiller, The OWL Reasoner Evaluation (ORE) 2015 Competition Report, Journal of Automated Reasoning 59 (2017) 455–482.
[37] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, Z. Wang, HermiT: An OWL 2 Reasoner, J. Autom. Reason. 53 (2014) 245–269.