

Practical Methods for Concept Interpolation in Realistic Ontologies

Xiaoshuang Yang

Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands

Abstract

Ontologies formulated in description logics are widely used to formalise knowledge and terminologies in domains such as medicine and biology. One central reasoning task for ontologies is to decide whether one concept is subsumed by the other. While modern reasoners can efficiently determine the subsumptions of large-scale ontologies, they offer little insight into the reasoning process itself. Concept interpolation addresses this by computing an intermediate concept in a user-specified vocabulary that witnesses the subsumption. Concept interpolation can be used to solve a range of problems in KR, including explaining reasoning, extracting explicit definitions, and learning concepts from examples. Despite its relevance, no practical methods for concept interpolation in description logics have been developed. Furthermore, to be used for concept learning, an interpolation method needs to support nominals, which are known to make the interpolation problem harder. In this project, we will develop practical methods for deciding the existence and computing concept interpolants in different logics of the \mathcal{EL} and \mathcal{ALC} families. If they exist, we want to find the “optimal” interpolant based on specific criteria, and compute approximations otherwise.

Keywords

Description logics, Craig interpolation, automated reasoning, concept learning

1. Introduction

Ontologies formulated in *description logics* (DLs) are a central formalism for representing and reasoning about knowledge in many application domains, including biology, medicine, healthcare, knowledge graphs and AI [1, 2, 3]. The standard language for building ontologies is OWL [4], whose logical foundation is based on DLs [1], a family of decidable fragments of first-order logic. Prominent large-scale ontologies such as SNOMED CT [5], which contains around 370,000 axioms, and the Gene Ontology [6] are actively maintained and used by automated reasoning systems to derive implicit knowledge. A central reasoning task for ontologies is *subsumption testing*: determining whether $\mathcal{O} \models C \sqsubseteq D$, meaning that every instance of concept C is also an instance of D under the ontology \mathcal{O} . Modern reasoners can solve this task efficiently even for very large ontologies [7]. However, the result of reasoning is increasingly insufficient on its own. In high-stakes domains, users need to understand why a subsumption holds, not just that it holds. Two common approaches to this problem are justifications and proofs. Justifications [8] return a minimal set of axioms that together entail the subsumption, but offer no concept-level explanation of the reasoning. Proofs provide a derivation tree of the subsumption, but such trees can be lengthy and involve many intermediate reasoning steps, making it difficult for users to identify the essential parts of the reasoning. At the same time, the development and maintenance of ontology content remains a largely manual and labour-intensive process. Automatically generating ontology content is challenging, as extracted knowledge is often noisy and requires principled structuring [9]. As we will show, *concept interpolation* can address both challenges: it can provide concept-level explanations for subsumptions, and it can be used to learn new concepts, thereby supporting automated ontology construction.

The notion of *Craig interpolation* originates in classical logic: given two formulas ϕ and ψ such that $\phi \models \psi$, a *Craig interpolant* is a formula I satisfying $\phi \models I$ and $I \models \psi$, where I uses only the non-logical

Doctoral Consortium of the 23rd International Conference on Principles of Knowledge Representation and Reasoning (KR 2026 DC), July 20-23, 2026, Lisbon, Portugal

✉ x.s.yang@vu.nl (X. Yang)

ORCID 0009-0009-0735-3822 (X. Yang)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

symbols occurring in both ϕ and ψ [10, 11]. We apply this idea to DLs, thereby introducing the notion of *concept interpolation*. Here, a signature Σ is a set of concept names, role names, and individual names. For a DL \mathcal{L} , an $\mathcal{L}(\Sigma)$ concept is a concept expressible in \mathcal{L} using only non-logical symbols from Σ .

Definition 1.1 ($\mathcal{L}(\Sigma)$ -interpolant). [12, Chapter 15] *Let \mathcal{L} be a DL, C, D be \mathcal{L} concepts, \mathcal{O} an \mathcal{L} ontology and Σ a signature. Then, an $\mathcal{L}(\Sigma)$ -interpolant for C and D under \mathcal{O} is an $\mathcal{L}(\Sigma)$ concept I that satisfies $\mathcal{O} \models C \sqsubseteq I$ and $\mathcal{O} \models I \sqsubseteq D$. A Craig interpolant is an $\mathcal{L}(\Sigma)$ -interpolant for C and D under $\mathcal{O}_1 \cup \mathcal{O}_2$, where $\Sigma = (\text{sig}(\mathcal{O}_1) \cup \text{sig}(C)) \cap (\text{sig}(\mathcal{O}_2) \cup \text{sig}(D))$.*

Interpolants do not always exist. The *Craig Interpolation Property* (CIP) guarantees the existence of the special case of Craig interpolants, but not every DL has this property: for example, \mathcal{ALC} has the CIP, while \mathcal{ALCO} does not. If a DL has the CIP, one can reduce the existence problem of interpolants to a subsumption testing problem. Concept interpolation can serve multiple purposes: it can remove irrelevant symbols and isolate the essential information behind a subsumption, thereby providing *explanations*; it can restrict attention to a smaller vocabulary, a form of *forgetting*; and it can be used to compute *explicit definitions*, useful for ontology engineering [12, Chapter 15]. Further applications include *separating examples* [13] and *concept learning* [14, 15].

Example 1.1 (Pizza Ontology). *Consider the ontology \mathcal{O} containing the following axioms:*

$$\begin{aligned} \exists \text{hasTopping}.\top \sqsubseteq \text{Pizza} & \quad \text{Pepperoni} \sqsubseteq \text{Meat} & \quad \text{Meat} \sqsubseteq \neg \text{Vegetarian} \\ \text{VegetarianPizza} \equiv \text{Pizza} \sqcap \forall \text{hasTopping}.\text{Vegetarian} & \quad \text{NonVegetarianPizza} \equiv \text{Pizza} \sqcap \neg \text{VegetarianPizza} \\ \text{AmericanPizza} \sqsubseteq \text{Pizza} \sqcap \exists \text{hasTopping}.\text{Pepperoni} \sqcap \exists \text{hasTopping}.\text{Tomato} \sqcap \exists \text{hasTopping}.\text{Mozzarella} \end{aligned}$$

This ontology entails $\text{AmericanPizza} \sqsubseteq \text{NonVegetarianPizza}$. An $\mathcal{ALC}(\Sigma_1)$ -interpolant for this subsumption, where $\Sigma_1 = \{\text{hasTopping}, \text{Pepperoni}\}$, is $\exists \text{hasTopping}.\text{Pepperoni}$. The interpolant intuitively pinpoints the central reason why the pizza is not vegetarian. If the user is not familiar with pepperoni and instead provides $\Sigma_2 = \{\text{hasTopping}, \text{Meat}\}$, the interpolant becomes $\exists \text{hasTopping}.\text{Meat}$, which explains at a higher level of abstraction: it is non-vegetarian because it has a meat topping. To contrast, if we explain the subsumption with a justification, a widely used explanation method, we would use the set $\mathcal{O} \setminus \{\exists \text{hasTopping}.\top \sqsubseteq \text{Pizza}\}$: five axioms that are jointly responsible for the subsumption, which provides a much less concise explanation.

Beyond explaining subsumptions, concept interpolants can also be used to compute *explicit definitions*. Through the closely related notion of (*projective*) *Beth definability*, concept interpolation can determine whether a concept name A is definable in terms of a given signature Σ and, if so, produce a concept C over Σ such that $\mathcal{O} \models A \equiv C$ [16, 17]. To illustrate, if we add the following axioms to the above example:

$$\text{Pizza} \sqsubseteq \forall \text{hasTopping}.\text{(Fish} \sqcup \text{Meat} \sqcup \text{Vegetarian)} \quad \text{Fish} \sqsubseteq \neg \text{Vegetarian}$$

such an explicit definition for $\text{NonVegetarianPizza}$ in the signature $\text{sig}(\mathcal{O}) \setminus \{\text{NonVegetarianPizza}, \text{VegetarianPizza}, \text{Vegetarian}\}$ is

$$\text{NonVegetarianPizza} \equiv \text{Pizza} \sqcap \exists \text{hasTopping}.\text{(Fish} \sqcup \text{Meat)},$$

which may help in understanding the concept better.

Concept interpolation can also be used for *concept learning*. Given sets of positive and negative examples (denoted by **Pos** and **Neg**, respectively), the goal is to find a concept that describes all positive and excludes all negative examples [18]. This supports tasks such as describing query results, finding concept definitions from data, and extracting interpretable representations of subsymbolic models [19]. For a separating concept to exist, the positive and negative examples must be disjoint, and can be expressed as a subsumption in DLs with nominals.

$$\mathcal{O} \models \bigsqcup \{\{a\} \mid a \in \mathbf{Pos}\} \sqsubseteq \bigsqcap \{\neg\{a\} \mid a \in \mathbf{Neg}\}$$

A concept interpolant for this subsumption is directly a separating concept. In fact, strict concept learning and concept interpolation are interchangeable in DLs with nominals [20].

2. Problem Statements and Related Work

Despite the range of applications outlined above, concept interpolation in DLs remains largely theoretical. We identify four research challenges that motivate this project.

Challenge 1: Only few DLs enjoy the CIP. The CIP has been established for \mathcal{EL} and \mathcal{EL} with role hierarchies [21, 22]. However, it has been shown that none of the remaining classical extensions of \mathcal{EL} enjoy the CIP [23]. For the more expressive \mathcal{ALC} , any extension with transitive roles, inverse roles and functionality restrictions has the CIP [17]. However, \mathcal{ALCO} (\mathcal{ALC} with nominals) and \mathcal{ALCH} (\mathcal{ALC} with role hierarchies) and their extensions with transitive roles, inverse roles, and functionality restrictions do not enjoy the CIP [17]. This means that in these logics, one cannot reduce the existence of interpolants to a subsumption test. Moreover, deciding the existence is usually one exponential higher in DLs without the CIP. For more expressive DLs, Lyon and Karge [24] recently introduced sequent calculi for \mathcal{RSQ} and established the concept-based Beth definability property, a weaker variant that only restricts concept names (not role names) in the interpolant. However, this does not entail that \mathcal{RSQ} enjoys the CIP.

Challenge 2: Existing algorithms are not practical. For the DLs that do enjoy the CIP, algorithms for computing interpolants have been proposed, but none have been implemented for realistic ontologies. Fortin et al. [23] give algorithms and optimal complexity bounds for the \mathcal{EL} family, and ten Cate et al. [17] propose a tableaux-based algorithm for \mathcal{ALC} that constructively computes interpolants. For settings where the CIP fails, Jung et al. [25] recently gave the first elementary algorithms for constructing \mathcal{ALC} interpolants under \mathcal{ALCH} and \mathcal{ALCQ} (\mathcal{ALC} with qualified number restrictions) ontologies, but with double- to triple-exponential complexity. However, a naive implementation of these algorithms will likely not work well in practice, since real-world ontologies can contain hundreds of thousands of axioms [26]. Craig interpolation is closely related to the notion of *uniform interpolation* (forgetting), for which tools have been developed [27, 28, 29, 30]. While uniform interpolation can also be used to compute concept interpolants, its complexity is even higher and practical methods do not scale well. Furthermore, many DLs that enjoy the CIP do not enjoy the uniform interpolation property. Uniform interpolation also has a formal connection to *abduction*, which computes hypotheses whose addition to a knowledge base would entail a given observation [31, 32, 33]: the negation of a Σ -uniform interpolant can serve as a logically weakest abductive hypothesis over Σ [34, 35, 36]. However, both uniform interpolation and abduction operate on the entire knowledge base, whereas Craig interpolation targets a specific subsumption, making it more suitable for explaining individual subsumptions and providing concept-level explanations.

Challenge 3: Concept learning requires nominals, which are often incompatible with the CIP. Concept learning, the most promising application of concept interpolation, requires nominals to separate positive and negative examples. However, nominals are often incompatible with the CIP: as noted in Challenge 1, adding nominals to \mathcal{EL} or \mathcal{ALC} destroys the CIP [23, 20], and there is no decidable extension of \mathcal{ALCO} that enjoys the CIP [37]. This creates a fundamental tension: the application that most needs interpolation is precisely the setting where the CIP fails. In DLs with nominals, one must first decide whether an interpolant exists at all, which is 2EXPTIME-complete for \mathcal{ALCO} [20], and then compute it if it does.

Challenge 4: Optimising interpolants for different criteria. Even within the same signature, multiple interpolants can exist, and not all are equally useful. Under $\Sigma_3 = \{\text{Pizza, hasTopping, Meat, Pepperoni}\}$, the following are all $\mathcal{ALC}(\Sigma_3)$ -interpolants:

1. $\exists \text{hasTopping.Pepperoni}$ (concept size 2)
2. $\exists \text{hasTopping.Meat}$ (concept size 2)

3. $\text{Pizza} \sqcap \exists \text{hasTopping.Pepperoni} \sqcap \exists \text{hasTopping.Meat}$ (concept size 7)

Arguably, the last one is less preferable as it is longer. Indeed, it adds information (*Pizza*, *Meat*) that is already implied. One may also argue that the first one is more useful as an explanation than the second, as it is more specific. This illustrates that rather than just computing some interpolant, we would often want to find one that optimises some criterion, which may depend on the application at hand. Even for the simplest \mathcal{EL} , in the worst case, interpolants can be exponential in size with respect to the input [23], making it essential to select smaller ones for a given criterion. Works on optimising proofs for DL subsumptions [38, 39] suggest that similar techniques could be applied to compute interpolants, but no work for computing optimal interpolants for a given signature currently exists.

3. Research Plan

This project addresses all four challenges through three research problems:

Problem 1: Concept Interpolation for \mathcal{EL} Family. While \mathcal{EL} itself enjoys the CIP, most of its extensions do not. Our main theoretical reference for this research question is [23], which provides an algorithm for computing Craig interpolants in extensions of \mathcal{EL} and $\mathcal{EL}\mathcal{I}$ when they exist, together with optimal complexity bounds, though the complexity can be quite high in the worst case. There are several possible implementation strategies, and we will initially focus on a proof-theoretic approach, using Maehara’s method [40], in analogy with the treatment of interpolants for classical propositional logic and modal logic K [12, Chapter 4,5]. We aim to extract concept interpolants from derivations by finding or designing a sequent calculus with the rule interpolation property. We will build on existing sequent systems for DLs [41, 24], and use proof libraries for Java such as EVEE [42] to support proof search. Besides the proof-theoretic approach, we may also look at other methods. For example, we may use canonical models as introduced in [43] and construct derivation trees following [23]. We may also take inspiration from existing implementations of uniform interpolation in DLs [27, 28, 29, 30]. For cases where the CIP fails in $\mathcal{EL}\mathcal{O}$, we will explore approaches such as *repair* to compute approximate interpolants, though the main treatment of nominals is deferred to Problem 3, where we adapt our methods to DLs with nominals for concept learning.

Problem 2: Concept Interpolation for Expressive \mathcal{ALC} Family. Since \mathcal{ALC} is more expressive than \mathcal{EL} , the implementation is likely to be more difficult. For this problem, we will use [17] as our main theoretical reference. Still, we can build on the results of Problem 1 and first try a proof-theoretic approach. We will look at existing proof systems and, if needed, develop our own system with the rule interpolation property, so that interpolants can again be computed from proof trees. As an alternative direction, we may study reduction to modal logic. Subsumptions in \mathcal{ALC} can be reduced to subsumptions in modal logic K, and then we can reuse existing techniques for computing interpolants in K [12, Chapter 4].

Problem 3: Interpolation-Based Concept Learning with Nominals. For this problem, we want to use our interpolation methods to implement a practical system for learning concept expressions from examples. The first challenge is scalability: applying interpolation methods to large sets of examples will require specialised data structures and indexing techniques. The second, more fundamental challenge is that concept learning typically requires the use of nominals, and they are often incompatible with the CIP. To address this lack of CIP, one possible strategy is to *repair* interpolation by allowing interpolants in a richer logic [12, Chapter 15]. However, there is no decidable extension of $\mathcal{ALC}\mathcal{O}$ that enjoys CIP [37]. Other possible approaches include changing the signature or working with suitable approximation formulas.

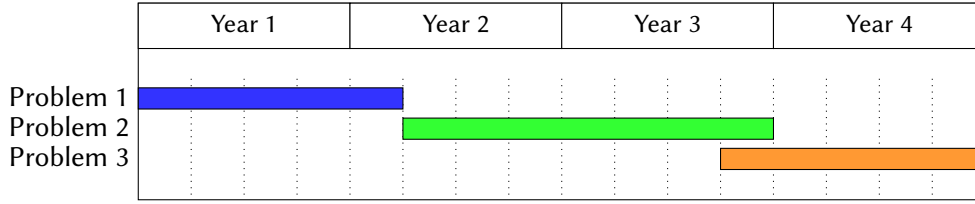


Figure 1: Research timeline.

Problem 1 and Problem 2 address Challenges 1, 2 and 4 for the \mathcal{EL} and \mathcal{ALC} families respectively, including both practical computation and optimality criteria. Problem 3 addresses Challenge 3 by extending and adapting interpolation methods for DLs with nominals to enable concept learning from examples. Figure 1 shows the planned timeline for this project.

4. Current Progress

We are currently addressing the problem for the \mathcal{EL} family. Our approach to computing concept interpolants is proof-theoretic. Sequent calculus is a simple and versatile formalism, widely used to define proof systems for modal and non-classical logics [44]. Maehara’s method [40] is a classical technique that extracts an interpolant from a sequent calculus proof by defining an interpolation property for each inference rule, propagating partial interpolants from premises to conclusion. This method has been used to establish the CIP of classical propositional logic and the minimal modal logic K [12, Chapter 4,5]. Inspired by this approach, our goal is to extract concept interpolants directly from proofs in \mathcal{EL} by finding or designing a calculus that admits the *rule interpolation property*. We have investigated several calculi for \mathcal{EL} towards this goal.

Hofmann’s Calculus We first considered Hofmann’s calculus [41], which closely resembles a sequent calculus. For this calculus, we defined a deterministic procedure that computes exactly one Craig interpolant of the conclusion from the Craig interpolant(s) of the premise(s). We proved that this method is *sound and complete*, in the sense that whenever a subsumption holds, we can compute a Craig interpolant for it. Since \mathcal{EL} enjoys the CIP, the existence of an $\mathcal{EL}(\Sigma)$ -interpolant can be reduced to a subsumption problem in \mathcal{EL} . Specifically, the following two statements are equivalent for any DL \mathcal{L} :

- there is an $\mathcal{L}(\Sigma)$ -interpolant for C and D under \mathcal{O} ;
- there is a Craig interpolant for C and D_Σ under $\mathcal{O} \cup \mathcal{O}_\Sigma$, where \mathcal{O}_Σ and D_Σ are obtained respectively, by renaming all symbols not in Σ uniformly to fresh symbols [12, Chapter 15].

Moreover, whenever $\mathcal{O} \cup \mathcal{O}_\Sigma \vDash C \sqsubseteq D_\Sigma$ holds, the CIP guarantees that a Craig interpolant exists, and such an interpolant is an $\mathcal{L}(\Sigma)$ -interpolant for C and D under \mathcal{O} .

Following this reduction, our method can be adapted to compute $\mathcal{EL}(\Sigma)$ -interpolants. However, Hofmann’s calculus has not been implemented in practice, and there is currently no efficient way to extract $\mathcal{EL}(\Sigma)$ -interpolants using this approach. Nevertheless, the results we have for this calculus confirm that the proof-theoretic approach is viable in principle, and the interpolation procedure we defined serves as a blueprint for the practically implemented calculi discussed next.

Practically Implemented Calculi To bridge this practicality gap, we turned to calculi that already have working implementations, such as the calculus from the DL textbook [1] and the calculus underlying the ELK reasoner [45]. The key advantage of these calculi is the availability of the proof library EVEC [42], which can generate proofs of subsumptions and allows us to search through the space of proofs. This is beneficial for our purposes, since different proofs may give different interpolants, and we aim to find the “best” one. EVEC implements several calculi, which have been compared in [46].

$$\begin{array}{c}
\text{CR1} \frac{}{A \sqsubseteq A} \quad \text{CR2} \frac{}{A \sqsubseteq \top} \\
\text{CR3} \frac{C_1 \sqsubseteq C_2 \quad C_2 \sqsubseteq C_3}{C_1 \sqsubseteq C_3} \quad \text{CR4} \frac{C \sqsubseteq C_1 \quad C \sqsubseteq C_2 \quad C_1 \sqcap C_2 \sqsubseteq D}{C \sqsubseteq D} \\
\text{CR5} \frac{C \sqsubseteq \exists r.C_1 \quad C_1 \sqsubseteq C_2 \quad \exists r.C_2 \sqsubseteq D}{C \sqsubseteq D}
\end{array}$$

Figure 2: Rule interpolation for calculus from [1]

For these calculi, we also revised our interpolation strategy in an important way. The classical Maehara method computes a single interpolant per inference rule. Moreover, it has been shown that applying Maehara’s method to cut-free sequent calculi for classical propositional logic is *incomplete up to logical equivalence*: it cannot produce all interpolants of a given subsumption [47]. Instead, our approach computes *all possible interpolants* for each inference rule, and at each rule step, selects the best one according to a given signature and a specified optimality criterion, such as minimising concept size.

Figure 2 shows the rule interpolation we defined for the textbook calculus. The interpolant formula(s) above each subsumption in the premises represent an arbitrary interpolant for that subsumption. In the conclusion, except for CR1, we list multiple red formulas: these are all the candidate interpolants one may choose from. Note that a finite conjunction of interpolants, or a conjunction of an interpolant with \top , is also an interpolant of the subsumption. These trivial combinations are omitted here.

5. Conclusion

In this paper, we outlined a research project on practical methods for concept interpolation in realistic ontologies, identifying four challenges: the limited scope of the CIP, the lack of practical algorithms, the incompatibility of nominals with the CIP, and the need for optimising interpolants. We proposed three research problems across the \mathcal{EL} and \mathcal{ALC} families, leading to an interpolation-based concept learning system.

We presented a proof-theoretic approach for the \mathcal{EL} family, defining rule interpolation procedures for Hofmann’s calculus and the textbook calculus, the latter supported by the proof library *EVEE*. Unlike the classical Maehara method, our approach computes all possible interpolants at each inference step, enabling optimisation for a given criterion. Current work focuses on completing the implementation, proving the completeness of the methods, and evaluating them on ontology benchmarks. Key challenges ahead include extending to \mathcal{ALC} , where negation and disjunction introduce additional complexity, and supporting nominals for concept learning, where the CIP fails and new strategies are needed.

Acknowledgements

I thank my supervisor Patrick Koopmann for his valuable suggestions. This publication is part of the project “*PICON – Practical methods for Concept Interpolation in Realistic Ontologies*” with file number OCENW.M.23.448 of the research programme Open Competition Domain Science, which is financed by the Dutch Research Council (NWO) under the grant <https://doi.org/10.61686/OTFBL33387>.

Declaration on Generative AI

During the preparation of this work, the author used Claude Opus 4.6 to: grammar and spelling check, paraphrase and reword. After using this service, the author reviewed and edited the content as needed and takes 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] R. Hoehndorf, P. N. Schofield, G. V. Gkoutos, The role of ontologies in biological and biomedical research: a functional perspective, *Briefings Bioinform.* 16 (2015) 1069–1080.
- [3] M. de Sousa Ribeiro, J. Leite, Aligning artificial neural networks and ontologies towards explainable AI, in: *Proceedings of AAI*, AAAI Press, 2021, pp. 4932–4940.
- [4] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, U. Sattler, OWL 2: The next step for OWL, *Journal of Web Semantics* 6 (2008) 309–322.
- [5] E. Chang, J. Mostafa, The use of SNOMED CT, 2013-2020: a literature review, *Journal of the American Medical Informatics Association* 28 (2021) 2017–2026.
- [6] The Gene Ontology resource: enriching a Gold mine, *Nucleic Acids Research* 49 (2021) D325–D334.
- [7] B. Parsia, N. Matentzoglou, R. S. Gonçalves, B. Glimm, A. Steigmiller, The OWL reasoner evaluation (ORE) 2015 competition report, *Journal of Automated Reasoning* 59 (2017) 455–482.
- [8] M. Horridge, B. Parsia, U. Sattler, Laconic and precise justifications in OWL, in: *International semantic web conference*, Springer, 2008, pp. 323–338.
- [9] I. Tiddi, S. Schlobach, Knowledge graphs as tools for explainable machine learning: A survey, *Artificial Intelligence* 302 (2022) 103627.
- [10] W. Craig, Linear reasoning. A new form of the Herbrand-Gentzen theorem, *The Journal of Symbolic Logic* 22 (1957) 250–268.
- [11] W. Craig, Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory, *The Journal of Symbolic Logic* 22 (1957) 269–285.
- [12] B. ten Cate, J. C. Jung, P. Koopmann, C. Wernhard, F. Wolter (Eds.), *Theory and Applications of Craig Interpolation*, Ubiquity Press, 2026. URL: <https://cibd.bitbucket.io/taci/>, To appear. Open access.
- [13] J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Logical separability of labeled data examples under ontologies, *Artificial Intelligence* 313 (2022) 103785.
- [14] J. Lehmann, P. Hitzler, Concept learning in description logics using refinement operators, *Machine Learning* 78 (2010) 203–250.
- [15] M. Funk, J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Learning description logic concepts: When can positive and negative examples be separated?, in: *Proceedings of IJCAI*, ijcai.org, 2019, pp. 1682–1688.
- [16] B. ten Cate, W. Conradie, M. Marx, Y. Venema, et al., Definitorially complete description logics., *Proceedings of KR* 6 (2006) 79–89.
- [17] B. ten Cate, E. Franconi, I. Seylan, Beth definability in expressive description logics, *Journal of Artificial Intelligence Research* 48 (2013) 347–414.
- [18] J. Lehmann, DL-learner: Learning concepts in description logics, *Journal of Machine Learning Research* 10 (2009) 2639–2642.
- [19] P. Hitzler, M. Sarker, Human-centered concept explanations for neural networks, *Neuro-symbolic artificial intelligence: The state of the art* 342 (2022) 2.
- [20] A. Artale, J. C. Jung, A. Mazzullo, A. Ozaki, F. Wolter, Living without Beth and Craig: Definitions and interpolants in description and modal logics with nominals and role inclusions, *ACM Transactions on Computational Logic* 24 (2023) 34:1–34:51.
- [21] B. Konev, C. Lutz, D. K. Ponomaryov, F. Wolter, Decomposing description logic ontologies, in: *Proceedings of KR*, AAAI Press, 2010.
- [22] C. Lutz, I. Seylan, F. Wolter, The data complexity of ontology-mediated queries with closed predicates, *Logical Methods in Computer Science* 15 (2019).
- [23] M. Fortin, B. Konev, F. Wolter, Interpolants and explicit definitions in extensions of the description logic \mathcal{EL} , in: *Proceedings of KR*, ijcar.org, 2022.
- [24] T. S. Lyon, J. Karge, Constructive interpolation and concept-based Beth definability for description logics via sequents, in: *Proceedings of IJCAI*, ijcai.org, 2024, pp. 3484–3492.

- [25] J. C. Jung, J. Kolodziejcki, F. Wolter, Computation of interpolants for description logic concepts in hard cases, in: Proceedings of DL, CEUR Workshop Proceedings, CEUR-WS.org, 2025.
- [26] D. T. Cucala, B. C. Grau, I. Horrocks, 15 years of consequence-based reasoning, in: Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday, volume 11560 of LNCS, Springer, 2019, pp. 573–587.
- [27] B. Konev, D. Walther, F. Wolter, Forgetting and uniform interpolation in large-scale description logic terminologies., in: Proceedings of IJCAI, Morgan Kaufmann Publishers Inc., 2009, pp. 830–835.
- [28] M. Ludwig, B. Konev, Practical uniform interpolation and forgetting for \mathcal{ALC} TBoxes with applications to logical difference, in: Proceedings of KR, AAAI Press, 2014.
- [29] Y. Zhao, R. A. Schmidt, FAME(Q): an automated tool for forgetting in description logics with qualified number restrictions, in: Proceedings of CADE, Springer, 2019, pp. 568–579.
- [30] P. Koopmann, LETHE: forgetting and uniform interpolation for expressive description logics, *Künstliche Intelligenz* 34 (2020) 381–387.
- [31] S. Klarman, U. Endriss, S. Schlobach, Abox abduction in the description logic \mathcal{ALC} , *Journal of Automated Reasoning* 46 (2011) 43–80.
- [32] M. Obeid, Z. Obeid, A. Moubaidin, N. Obeid, Using description logic and Abox abduction to capture medical diagnosis, in: Proceedings of IEA/AIE, LNCS, Springer, 2019, pp. 376–388.
- [33] J. Du, K. Wang, Y. Shen, A tractable approach to Abox abduction over description logic ontologies, in: Proceedings of AAAI, AAAI Press, 2014, pp. 1034–1040.
- [34] D. M. Gabbay, R. A. Schmidt, A. Szalas, Second-Order Quantifier Elimination - Foundations, Computational Aspects and Applications, volume 12 of *Studies in logic : Mathematical logic and foundations*, College Publications, 2008.
- [35] W. Del-Pinto, R. A. Schmidt, Abox abduction via forgetting in \mathcal{ALC} , in: Proceedings of AAAI, AAAI Press, 2019, pp. 2768–2775.
- [36] P. Koopmann, W. Del-Pinto, S. Tournet, R. A. Schmidt, Signature-based abduction for expressive description logics, in: Proceedings of KR, 2020, pp. 592–602.
- [37] B. ten Cate, Interpolation for extended modal languages, *The Journal of Symbolic Logic* 70 (2005) 223–234.
- [38] C. Alrabbaa, F. Baader, S. Borgwardt, P. Koopmann, A. Kovtunova, Finding small proofs for description logic entailments: Theory and practice, in: Proceedings of LPAR, EPiC Series in Computing, EasyChair, 2020, pp. 32–67.
- [39] C. Alrabbaa, F. Baader, S. Borgwardt, P. Koopmann, A. Kovtunova, Finding good proofs for description logic entailments using recursive quality measures, in: Proceedings of CADE, LNCS, Springer, 2021, pp. 291–308.
- [40] S. Maehara, On the interpolation theorem of Craig (in japanese), *Sūgaku* 12 (1960) 235.
- [41] M. Hofmann, Proof-theoretic approach to description-logic, in: Proceedings of LICS, IEEE, 2005, pp. 229–237.
- [42] C. Alrabbaa, S. Borgwardt, T. Friese, A. Hirsch, N. Knieriemen, P. Koopmann, A. Kovtunova, A. Krüger, A. Popovic, I. S. R. Siahaan, Explaining reasoning results for OWL ontologies with E_{VEE}, in: Proceedings of KR, 2024.
- [43] F. Baader, S. Brandt, C. Lutz, Pushing the \mathcal{EL} envelope, in: Proceedings of IJCAI, Professional Book Center, 2005, pp. 364–369.
- [44] A. S. Troelstra, H. Schwichtenberg, Basic Proof Theory, Cambridge University Press, 2000.
- [45] Y. Kazakov, M. Krötzsch, F. Simančík, The incredible ELK: From polynomial procedures to efficient reasoning with \mathcal{EL} ontologies, *Journal of Automated Reasoning* 53 (2014) 1–61.
- [46] C. Alrabbaa, S. Borgwardt, P. Herrmann, M. Krötzsch, The shape of \mathcal{EL} proofs: A tale of three calculi, in: Proceedings of DL, CEUR-WS.org, 2025.
- [47] S. Hetzl, R. Jalali, On the completeness of interpolation algorithms, in: Proceedings of LICS, ACM, 2024, pp. 45:1–45:13.