

Interactive Dialogs with Logic-Based AI Systems

Tommaso Zendron¹

¹Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands

Abstract

Ontology-based systems can perform efficient query answering and automated reasoning given a set of constraints or rules. However, they work rather statically and offer only limited user adaptability. They may answer queries and provide explanations for their answers, but there is no real interaction with end-users and no general way of dealing with situations where additional input from the user is needed to solve the problem at hand. To overcome this limitation, we aim to develop a generic method for interacting with ontologies formulated in OWL in a dialog-like manner. The first objective of our project is to understand the theoretical foundations of these dialogs, where the system, in need of additional information to answer a query, determines questions to send to the user purely based on logical reasoning on available knowledge. As a basis for structuring this interaction, we focus on interactive abduction: a new interactive way to solve the abduction problem in a DL knowledge base. On this ground, we will develop a prototypical generic system that allows collaborative query answering with OWL ontologies. Finally, we want to investigate how to tailor interactions on the basis of user models, and develop personalised and interactive explanation methods for logical reasoning accordingly.

Keywords

Ontologies, Description Logics, Interaction, Abduction

1. Background

From its birth in the 1950s, AI research has focused on building machines able to solve tasks that would normally require human intelligence, such as theorem proving, game playing, and solving microworlds [1]. Since then, AI has been successfully applied to a wide range of domains, to the point that today AI systems are present in everyday life, supporting humans in handling data, making decisions, and optimising processes. For this reason, a flourishing research direction in AI is now committed to build systems that are not only capable of solving tasks, but that are also able to provide personalised outputs and adapt to specific users. Rather than achieving or substituting human intelligence, the new generation of AI systems should assist users and collaborate with them, thus augmenting human capabilities [2].

While much research has been conducted for chatbot personalisation, little work has been done in this direction for logic-based systems. These are systems in which knowledge has been formalised, able to perform automated reasoning given a set of constraint or rules. The rigour of logical formalisms ensures a high level of transparency and trustworthiness making symbolic systems highly desirable in critical domains where formal guarantees are needed. However, current logic-based systems offer only limited user adaptability. They may answer user queries [3] and provide explanations for their answers [4, 5, 6], but there is little space for personalised interaction with end-users and no general way to deal with situations where additional input from the user is needed to solve the problem at hand.

RuleML+RR'26: Companion Proceedings of the 10th International Joint Conference on Rules and Reasoning, August 24–26, 2026, Vilnius, Lithuania

✉ t.zendron@vu.nl (T. Zendron)

ORCID 0009-0009-3564-8783 (T. Zendron)



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

Table 1
System-Biologist interaction as Natural Language and DL Queries

	Dialog in Natural Language	DL formalisation
System:	Is the Great Barrier Reef bleached?	$q_1() = \text{Bleached}(\text{gbr})$
Biologist:	No	
System:	What animals live in the Great Barrier Reef?	$q_2(x) = \text{populatedBy}(\text{gbr}, x)$
Biologist:	Butterflyfish, damselfish, green sea turtle, dugong	
System:	Is the butterflyfish population declining?	$q_3() = \text{Declining}(\text{butterflyfish})$
Biologist:	Yes	
System:	The Great Barrier Reef is not healthy	$\neg \text{HealthyCoral}(\text{gbr})$

We consider logic-based systems that make use of description logics (DL) knowledge bases. DL are a family of knowledge representation formalisms which formalise domain knowledge by use of concept, role and individual names. Thanks to their balance between expressiveness and computational tractability, they have been proven very successful in many areas of AI, notably as the formalism underpinning the web ontology language OWL, as well as in ontology based data access, and knowledge representation and reasoning [7, 8, 9]. For an introduction to description logics we refer to [10].

A DL knowledge base consists of a TBox, expressing the structure of a domain, and an ABox, expressing the specific instances of a domain. Knowledge bases are used by logic-based systems to perform automated reasoning, with applications in domains such as healthcare, biology, planning, semantic technologies [11, 12, 13, 14]. In what follows, we will use $\mathcal{K}$, $\mathcal{T}$ and $\mathcal{A}$ to refer to a knowledge base, TBox and ABox respectively.

Since dialogs are a popular, efficient, and user-friendly method for interacting with users, our aim is to develop a generic method for interacting with logic-based systems in a dialog-like manner. This type of interaction would be beneficial for both knowledge engineers and end users.

As an example, consider a logic-based system for monitoring marine ecosystems. Sensors provide information about water temperature and water quality, while the system also stores information about different fish species. A marine biologist consults the system to determine whether the Great Barrier Reef is healthy. Since the system lacks information about the status both of the coral in the Great Barrier Reef, and of the fishes that populate it, it starts a dialog with the marine biologist. To answer the query, the system interacts with the biologist and combines the acquired information with its background knowledge. One possible dialog is presented in Table 1.

Dialogs like this are not new and are easy to hard-code if we know exactly what kind of interaction we would want to have. However, our aim is to address system-user interactions in a generalised way to determine such questions purely based on logical reasoning on the available knowledge. The column "DL formalisation" in Table 1 shows that our dialog can be seen as a series of logical queries and answers between the user and the system.

Formally, we can see the problem in the example as an abduction problem. In abduction, we are given a knowledge base and a fact whose truth cannot be established purely based on the available knowledge [15]. Abductive reasoning then produces a set of hypotheses for the fact, which could be verified by the biologist afterward. Since our aim is to produce only the correct hypothesis by interactively constructing it through questions to the user, we call our problem *interactive abduction*. Interactive abduction has applications that go beyond the dialog scenario above: while it can be used

by AI systems to answer questions in interaction with a human user or some external system, it can also be used for interactive configuration tasks or for knowledge engineering to interactively construct statements to be added to a knowledge base.

2. Related Work

Current software to interact with logic-based systems include Protégé and Evonne. Protégé is an interactive software for editing and creating ontologies [16], whereas Evonne is a Protégé plug-in that allows users to see the derivation steps of an entailment by unfolding a proof in a tree-like structure [17].

In the realm of knowledge representation, there are works that consider dialog-like interactions between a system and a user. Research about planning has shown how explanations can be formulated as model reconciliation problem, where dialog is used to update user knowledge with the information necessary to understand why a specific plan was formulated by the system [18, 19]. Within this line of research, Vasileiou et al. [20] has studied how to apply argumentation to user interaction, to provide more fine-grained and tailored explanations. The idea of interacting through queries has also been considered in the scope of concept learning, where the purpose is to learn a new concept to add to the knowledge of the system through a sequence of queries sent to an expert [21]. With respect to ontology debugging, our previous work has considered how such task can be undertaken interactively, by merging user answers to queries with automated computation [22]. Furthermore, in controlled query evaluation, a system that answers user queries keeps track of previous interactions with the user, to avoid disclosing protected information by adapting its answers according to what has already been said [23].

As anticipated in the introduction, in our project on interactive dialogs with logic-based systems we focus on interactive abduction as a basis for user interactions. In DL, abduction has been proposed as a method to explain missing entailments. Given a KB $\mathcal{K}$, and an observation Φ s.t. $\mathcal{K} \not\models \Phi$, abduction provides hypotheses $\mathcal{H}$ s.t. $\mathcal{K} \cup \mathcal{H} \models \Phi$. We speak of TBox or ABox abduction problem depending on whether the generated hypotheses consist of TBox or ABox axioms. The complexity of computing which assertions are necessary or relevant for a given ABox abduction problem has been studied for *DL-Lite* [24] and Existential Rules [25]. Also, different constraints have been proposed to compute hypotheses: consistency $\mathcal{H} \cup \mathcal{K} \not\models \perp$, relevance $\mathcal{H} \not\models \Phi$, minimality, signature $\text{sig}(\mathcal{H}) \subseteq \Sigma$ [15, 26, 27, 28, 29]. However, these are just parameters to adjust before (or after) computing hypotheses, rather than actual options to interactively build a subset of desired and factually correct hypotheses. This gap is also visible in current practical methods for solving the abduction problem, as they do not involve the user [26, 30, 31]. The only systems that offer some form of interaction are CATS, that allows the user to choose the best suited algorithm to compute the hypotheses according to the case [32], and Eevee, whose workflow facilitates incremental adaptation of the signature of the hypotheses [33].

3. Problem

To formally capture the type of dialog presented in the introduction, we propose the interactive abduction problem (IAP). We start by focusing on IAP that have atomic queries (AQ) as query language, that is, queries of the form $C(x)$ or $r(x, y)$, $r(a, y)$, $r(x, a)$ where C is a concept name, r is a role name, a is an individual and $x, y \in N_v$ (the countable, infinite set of variables). In what follows, we will use Ass to refer to the set of all assertions.

Definition 1. An Interactive Abduction Problem (IAP) is a tuple $(\mathcal{K}, q, U)$, which consists of:

1. a background knowledge base $\mathcal{K}$;
2. a query q in the form of a Boolean query;
3. a User U modeled as an Oracle $U : \{\text{Ass}, \text{AQ}\} \rightarrow \{\text{yes}, \text{no}\} \cup \bigcup_{k=1}^2 \mathcal{P}(N_I^k)$ s.t. $U(\alpha) \in \{\text{yes}, \text{no}, \emptyset\}$ if $\alpha \in \text{Ass}$ and $U(\alpha) \in \mathcal{P}(N_I^k)$ otherwise, where k is the number of free variables in α , and N_I is the countable, infinite set of individual names.

Definition 2. A solution to IAP is a set $S = (\mathcal{H}, \text{ans})$, where $\mathcal{H} \subseteq \text{Ass}$, $\text{ans} \in \{\text{yes}, \text{no}\}$ and

1. $\mathcal{K} \cup \mathcal{H} \models q$ if $\text{ans} = \text{yes}$ or $\mathcal{K} \cup \mathcal{H} \models \neg q$ if $\text{ans} = \text{no}$;
2. for all $\alpha \in \mathcal{H}$, $U(\alpha) = \text{yes}$ or $U(\neg\alpha) = \text{no}$

Definition 3. A dialog is a sequence $\Pi = (\alpha_1, \text{ans}_1), \dots, (\alpha_n, \text{ans}_n)$ where $\alpha_i \in \text{Ass} \cup \text{AQ}$ and $\text{ans}_i = U(\alpha_i)$ which is successful if for $\mathcal{H} = \{\alpha_i \mid (\alpha_i, \text{yes}) \in \Pi\} \cup \{\neg\alpha_i \mid (\alpha_i, \text{no}) \in \Pi\} \cup \{\alpha_i(\vec{c}) \mid (\alpha_i(\vec{x}), S) \in \Pi, \vec{c} \in S\}$, $(\mathcal{H}, \text{yes})$ or $(\mathcal{H}, \text{no})$ is a solution.

Intuitively, IAP can be thought of as a game where, given an input question q , the system asks questions to a user, and for each question in a dialog Π , the user returns "yes", "no", "I do not know" (the empty set) or individuals. The system adds the assertions that are confirmed by the user to the set $\mathcal{H}$, until $\mathcal{K} \cup \mathcal{H} \models q$ or $\mathcal{K} \cup \mathcal{H} \models \neg q$.

At this stage of our project, we have already analysed how the number of queries required to find a solution to an IAP changes given different DL, query languages and constraints on the type of individuals allowed (fresh or from the ABox). Interestingly, for the case of an IAP that uses $\mathcal{EL}$, AQ as the query language and allows for fresh individuals, a solution can always be obtained with a polynomial number of queries, if it exists.

With the formal definitions of the elements of an IAP at hand, we can now formalise the example presented in the introduction.

Example 4. Consider the IAP $(\mathcal{K}, q, U)$ with $\mathcal{K} = (\mathcal{T} \cup \mathcal{A})$

$$\begin{aligned} \mathcal{T} := & \{ \neg \text{Bleached} \sqcap \text{StableTemp} \sqcap \forall \text{populatedBy}.(\text{Fish} \sqcap \text{Thriving}) \sqsubseteq \text{HealthyCoral}, \\ & \text{Polluted} \sqcap \exists \text{populatedBy}.(\text{Fish} \sqcap \text{Declining}) \sqsubseteq \neg \text{HealthyCoral}, \\ & \text{Bleached} \sqcap \text{Polluted} \sqsubseteq \neg \text{HealthyCoral}, \quad \text{Thriving} \sqcap \text{Declining} \sqsubseteq \perp \} \end{aligned}$$

$$\mathcal{A} := \{ \text{Fish}(\text{butterflyfish}), \quad \text{Fish}(\text{damselfish}), \quad \text{Fish}(\text{salmon}), \quad \text{StableTemp}(\text{gbr}), \quad \text{Polluted}(\text{gbr}) \}$$

The query of this IAP is $q = \text{HealthyCoral}(\text{gbr})$. The same problem could be stated as an ABox abduction problem. In this case, we would produce a set of hypotheses (depending on the constraints used) for q , many of which are incorrect not just as explanations for the specific case, but also factually incorrect w.r.t. the real world. As an example, current abduction algorithms would produce, a set of hypotheses $\mathcal{E}$ with $\{\text{Bleached}(\text{gbr})\} \in \mathcal{E}$ which is incorrect, and also $\{\text{populatedBy}(\text{gbr}, \text{salmon})\} \in \mathcal{E}$ which is incorrect since salmon do not live in tropical seas. Furthermore, to obtain the hypotheses for $\neg q = \neg \text{HealthyCoral}(\text{gbr})$, we would have to instantiate a new abduction problem, with $\neg \text{HealthyCoral}(\text{gbr})$ as a query.

To interact with the user U , we need a strategy that selects the questions to ask.

Definition 5. A strategy is a function that takes a KB $\mathcal{K}$ and an assertion q from an IAP $(\mathcal{K}, q, U)$, as well as a dialog Π , and returns an assertion or an AQ.

Given different answers from the user, different dialogs are produced by the same strategy. Hence, we introduce the notion of *dialog tree* for a strategy, that are all the possible dialogs the strategy would create for a given IAP. In the tree, branching happens through the user answers. Each node in the tree is a state in a dialog game, and the depth of a dialog tree is the length of its longest branch.

To make things easier, we focus on the case where the system only asks assertions, to which the user can answer "yes", "no", or the empty set ("I do not know"). This means that from each node of a dialog tree there are three possible answers, and the number of possible dialogs is at most 3^k , where $k = |\Pi|$.

Given these notions, how can we compare strategies and choose one to interact with the user? First of all, to make strategies comparable, we assume that if the strategy is underspecified the next question is chosen based on a global, fixed order (this way every strategy is deterministic). Hence, we assume a total ordering $<$ on all possible assertions.

That said, one could think that a good way to compare strategies is to consider a strategy better than another if, given any user, it produces a shorter (or equal) dialog. However, it is possible to show that this way of comparing does not allow to find an overall better strategy, as it is always possible to find a random strategy that asks exactly the questions needed to obtain a solution.

We need a definition that considers the entire dialog tree produced by a strategy. Currently, we consider the following:

Definition 6. *Given an IAP P , two strategies S_A and S_B , we say that S_A is better than S_B , written $S_A \preceq S_B$, if there is a $n \in \mathbb{N}$ s.t. for all $k < n$, the dialog trees $S_A(P)$ and $S_B(P)$ produced by S_A and S_B , respectively, have the same number of completed dialogs of length k , and S_A has more or equal completed dialogs of length n than S_B .*

Here, when comparing strategies we focus on positive solutions to IAP. This means that whenever it becomes clear that it is not possible to find a hypothesis H s.t. $\mathcal{K} \cup \mathcal{H} \models q$, we terminate the dialog.

Given $\mathcal{K}$, q , a partial solution $\mathcal{H}$, the strategies that we have considered are:

1. S_1 : if there is a relevant assertion β for $(\mathcal{K} \cup \mathcal{H}, q)$ s.t. $\mathcal{K} \cup \mathcal{H} \not\models \beta$, ask β ;
2. S_2 : if there is a necessary assertion β for $(\mathcal{K} \cup \mathcal{H}, q)$ ask β , otherwise use S_1 ;
3. S_3 : ask necessary assertion first, then necessary entailed, otherwise use S_1 .

Theorem 7. *Given an IAP A , $S_3(A) \preceq S_2(A) \preceq S_1(A)$.*

Necessary and relevant assertions can be computed in polynomial time [24]. The complexity of determining necessarily entailed assertions is not included in [24], but is a consequence of the following theorem.

Theorem 8. *For every DL that allows for concept and role disjointness axiom, the problem of deciding whether an assertion is necessary entailed can be polynomially reduced to deciding existence of a solution.*

Proof. Let $(\mathcal{K}, \mathcal{H}, \Phi)$ be an abduction problem, and $\phi(\vec{b})$ be an assertion. To check if $\phi(\vec{b})$ is a necessary entailed assertion, we construct the knowledge base $\mathcal{K}' = \mathcal{K} \cup \{\psi \sqsubseteq \neg\phi, \psi(\vec{b})\}$. If the abduction problem $(\mathcal{K}', \mathcal{H}, \Phi)$ does not admit a solution, $\phi(\vec{b})$ is necessary entailed. $\square$

4. Plan

The core of our PhD project is to establish the formal foundation of interactive abduction for different description logics. As mentioned above, we are currently investigating strategies to interact with the

user and the associated decision problems, as well as their computational complexity, focusing on AQ as a query language. Extending the query language to complex concepts or union of conjunctive queries is beyond the scope of our research, but we plan to also investigate the case where the user is free to return individuals, and not just "yes", "no", "I do not know" as answer.

Once efficient strategies for asking questions are found, we will develop a first practical system that implements interactive abduction. For a first evaluation of this prototype, we will model also the user as a logic-based system, which will allow us to do large scale experiments based on available ontology benchmarks without human users. In the next step, we will use our system with a knowledge base developed specifically for users affected by type 2 diabetes [34]. In this application, diabetic users can consult the system to receive daily advice on their lifestyle and diet. The system, that stores medical information together with data about the patient (age, blood sugar level...) and their habits, initialises a dialog when additional information is necessary to answer questions such as: "Is this meal suitable for me before going to bed?" This application will allow us to evaluate our system in a user study and understand how this type of dialog is perceived by human users.

In a subsequent stage, we plan to personalise the interaction on the basis of user-models. In this setting, the system has background information about the user before the interaction begins and uses it to tailor the interaction according to what the user knows. A starting point will be to focus on *signatures* as a way to restrict the space of possible answers and questions, as our user model will make sure that we can infer the appropriate signature of the current situation. We will build on interpolation [35, 36, 37], which allows to project (portions of) knowledge into different vocabularies. We will use it to restrict and project questions and answers to the vocabulary that suits the user type and the current situation.

As a last step, we plan to develop interactive explanation methods for logical reasoning based on user models. Current research on explanations in DL knowledge bases, consisting of both theoretical techniques for providing explanations [4, 5, 6, 38] and practical systems for visualising them [17, 33] lacks user adaptability. For this reason, we will develop flexible variants of existing logic-based explanation methods that allow for this kind of user adaptation. For instance, we want our system to dynamically adapt the length of an explanation based on user preferences. Once these aspects have been considered, extended and efficiently merged in a single logic-based system, our aim is to conduct a final user study.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

- [1] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach (4th Edition), Pearson, 2020.
- [2] Z. Akata, D. Balliet, M. de Rijke, F. Dignum, V. Dignum, G. Eiben, A. Fokkens, D. Grossi, K. Hindriks, H. Hoos, H. Hung, C. Jonker, C. Monz, M. Neerincx, F. Oliehoek, H. Prakken, S. Schlobach, L. van der Gaag, F. van Harmelen, H. van Hoof, B. van Riemsdijk, A. van Wynsberghe, R. Verbrugge, B. Verheij, P. Vossen, M. Welling, A research agenda for Hybrid Intelligence: Augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence, *Computer* 53 (2020) 18–28. doi:10.1109/MC.2020.2996587.
- [3] O. Corcho, F. Priyatna, D. Chaves-Fraga, Towards a new generation of ontology based data access, *Semantic Web* 11 (2020) 153–160. doi:10.3233/SW-190384.

- [4] C. Alrabbaa, S. Borgwardt, P. Koopmann, A. Kovtunova, Explaining ontology-mediated query answers using proofs over universal models, in: Proceedings of the 6th International Joint Conference on Rules and Reasoning (RuleML+RR), volume 13752 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 167–182. URL: https://doi.org/10.1007/978-3-031-21541-4_11.
- [5] M. Horridge, Justification based explanation in ontologies, Ph.D. thesis, University of Manchester, UK, 2011. URL: <http://www.manchester.ac.uk/escholar/uk-ac-man-scw:131699>.
- [6] S. Schlobach, Explaining subsumption by optimal interpolation, in: European Workshop on Logics in Artificial Intelligence, Springer, 2004, pp. 413–425. doi:10.1007/978-3-540-30227-8_35.
- [7] I. Horrocks, P. F. Patel-Schneider, F. Van Harmelen, From SHIQ and RDF to OWL: The making of a web ontology language, *Journal of web semantics* 1 (2003) 7–26.
- [8] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family, *Journal of Automated reasoning* 39 (2007) 385–429. doi:10.1007/S10817-007-9078-X.
- [9] F. Van Harmelen, V. Lifschitz, B. Porter, Handbook of knowledge representation, volume 1, Elsevier, 2008.
- [10] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An introduction to description logic, Cambridge University Press, 2017.
- [11] 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.
- [12] 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.
- [13] The Gene Ontology resource: enriching a GOLD mine, *Nucleic Acids Research* 49 (2021) D325–D334. doi:10.1093/nar/gkaa1113.
- [14] Y. Gil, Description logics and planning, *AI magazine* 26 (2005) 73–84.
- [15] C. Elsenbroich, O. Kutz, U. Sattler, A case for abductive reasoning over ontologies, in: Proceedings of the OWLED’06 Workshop on OWL: Experiences and Directions, CEUR-WS.org, 2006. URL: https://ceur-ws.org/Vol-216/submission_25.pdf.
- [16] M. A. Musen, The protégé project: a look back and a look forward, *AI Matters* 1 (2015) 4–12. doi:10.1145/2757001.2757003.
- [17] C. Alrabbaa, F. Baader, S. Borgwardt, R. Dachsel, P. Koopmann, J. Méndez, Evonne: Interactive proof visualization for description logics (system description), in: Automated Reasoning - 11th International Joint Conference, IJCAR, volume 13385 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 271–280. doi:10.1007/978-3-031-10769-6_16.
- [18] T. Chakraborti, S. Sreedharan, Y. Zhang, S. Kambhampati, Plan explanations as model reconciliation: Moving beyond explanation as soliloquy, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017, pp. 156–163. doi:10.24963/ijcai.2017/23.
- [19] S. L. Vasileiou, W. Yeoh, T. C. Son, A. Kumar, M. Cashmore, D. Magazzeni, A logic-based explanation generation framework for classical and hybrid planning problems, *J. Artif. Intell. Res.* 73 (2022) 1473–1534. doi:10.1613/JAIR.1.13431.
- [20] S. L. Vasileiou, A. Kumar, W. Yeoh, T. C. Son, F. Toni, Dialectical reconciliation via structured argumentative dialogues, in: Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR, 2024. doi:10.24963/KR.2024/73.
- [21] D. Angluin, Queries and concept learning, *Mach. Learn.* 2 (1987) 319–342. doi:10.1007/BF00116828.
- [22] T. Zendron, R. Peñaloza, Guiding interactive ontology repair through prolific and relevant

- axioms, in: Proceedings of the Workshop on Foundations and Future of Change in Artificial Intelligence (FCAI 2025), CEUR Workshop Proceedings, CEUR-WS, 2025, pp. 8–26.
- [23] G. Cima, D. Lembo, L. Marconi, R. Rosati, D. F. Savo, Controlled query evaluation in ontology-based data access, in: International Semantic Web Conference, Springer, 2020, pp. 128–146.
 - [24] D. Calvanese, M. Ortiz, M. Simkus, G. Stefanoni, Reasoning about explanations for negative query answers in *DL-Lite*, J. Artif. Intell. Res. 48 (2013) 635–669. doi:10.1613/JAIR.3870.
 - [25] Ī. Ī. Ceylan, T. Lukasiewicz, E. Malizia, C. Molinaro, A. Vaicnavicius, Explanations for negative query answers under existential rules, in: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, 2020, pp. 223–232. doi:10.24963/KR.2020/23.
 - [26] P. Koopmann, W. Del-Pinto, S. Tournet, R. A. Schmidt, Signature-based abduction for expressive description logics, in: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, volume 17, 2020, pp. 592–602. doi:10.24963/kr.2020/59.
 - [27] D. Mrózek, J. Pukancová, M. Homola, ABox abduction solver exploiting multiple DL reasoners, in: Proceedings of the 31st International Workshop on Description Logics co-located with 16th International Conference on Principles of Knowledge Representation and Reasoning, CEUR-WS.org, 2018. URL: <https://ceur-ws.org/Vol-2211/paper-24.pdf>.
 - [28] S. Klarman, U. Endriss, S. Schlobach, ABox abduction in the description logic $\mathcal{ALC}$, J. Autom. Reason. 46 (2011) 43–80. doi:10.1007/S10817-010-9168-Z.
 - [29] F. Haifani, P. Koopmann, S. Tournet, C. Weidenbach, Connection-minimal abduction in $\mathcal{EL}$ via translation to FOL, in: International Joint Conference on Automated Reasoning, Springer, 2022, pp. 188–207. doi:10.1007/978-3-031-10769-6_12.
 - [30] J. Du, H. Wan, H. Ma, Practical TBox abduction based on justification patterns, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press, 2017, pp. 1100–1106.
 - [31] I. Gocev, G. Meditskos, N. Bassiliades, An abduction-based method for explaining non-entailments of semantic matching, Semantic Web 16 (2025). doi:10.1177/22104968251386880.
 - [32] J. Kloc, J. Boborová, M. Homola, J. Pukancová, CATS solver: The rise of hybrid abduction algorithms, in: Proceedings of the 38th International Workshop on Description Logics - DL, volume 4091 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025.
 - [33] C. Alrabbaa, S. Borgwardt, T. Friesse, A. Hirsch, N. Knieriemen, P. Koopmann, A. Kovtunova, A. Krüger, A. Popovič, I. Siahaan, Explaining reasoning results for OWL ontologies with Eeve, in: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, volume 21, 2024, pp. 709–719. doi:10.24963/kr.2024/67.
 - [34] P. Y. Chen, S. B. Santamaria, M. H. De Boer, F. Den Hengst, B. A. Kamphorst, Q. Smit, S. Wang, J. Wolff, Intelligent support systems for lifestyle change: Integrating dialogue, information extraction, and reasoning, in: 3rd International Conference on Hybrid Human-Artificial Intelligence, HHAI, IOS Press BV, 2024, pp. 457–459. doi:10.3233/FAIA240223.
 - [35] P. Koopmann, R. A. Schmidt, Uniform interpolation and forgetting for $\mathcal{ALC}$ ontologies with ABoxes, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI Press, 2015, pp. 175–181. URL: <https://doi.org/10.1609/aaai.v29i1.9206>.
 - [36] J. C. Jung, P. Koopmann, M. Knorr, Interpolation in knowledge representation, in: [37], 2025. Open access, <https://cibd.bitbucket.io/taci/>.
 - [37] B. ten Cate, J. C. Jung, P. Koopmann, C. Wernhard, F. Wolter (Eds.), Theory and Applications of Craig Interpolation, Ubiquity Press, to appear in 2026. Open access, <https://cibd.bitbucket.io/taci/>.
 - [38] P. Koopmann, Y. Mahmood, A. N. Ngomo, B. Tiwari, Can you tell the difference? Contrastive explanations for ABox entailments, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, AAAI Press, 2026, pp. 19189–19197. doi:10.1609/AAAI.V40I23.38993.