ABox Abduction in the Description Logic $\mathcal{ALC}$

Szymon Klarman · Ulle Endriss · Stefan Schlobach

To appear in the *Journal of Automated Reasoning*
Accepted: February 03, 2010

Abstract Due to the growing popularity of Description Logics-based knowledge representation systems, predominantly in the context of Semantic Web applications, there is a rising demand for tools offering non-standard reasoning services. One particularly interesting form of reasoning, both from the user as well as the ontology engineering perspective, is abduction. In this paper we introduce two novel reasoning calculi for solving *ABox abduction problems* in the Description Logic $\mathcal{ALC}$, i.e. problems of finding minimal sets of ABox axioms, which when added to the knowledge base enforce entailment of a requested set of assertions. The algorithms are based on *regular connection tableaux* and *resolution with set-of-support* and are proven to be sound and complete. We elaborate on a number of technical issues involved and discuss some practical aspects of reasoning with the methods.

Keywords Description Logic · Abduction · Non-standard Reasoning Services · Semantic Tableaux · Resolution

S. Klarman
Department of Computer Science, Vrije Universiteit Amsterdam
de Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Tel.: +31-(0)-20-5987752
E-mail: sklarman@few.vu.nl

U. Endriss
Institute for Logic, Language and Computation, University of Amsterdam
Science Park 904, 1098 XH Amsterdam, The Netherlands
E-mail: ulle.endriss@uva.nl

S. Schlobach
Department of Computer Science, Vrije Universiteit Amsterdam
de Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
E-mail: schlobac@few.vu.nl

1 Introduction

In recent decades abduction has gained considerable attention in such fields as logic, artificial intelligence and philosophy of science. It has been widely recognized that the style of reasoning, usually illustrated as an inference from puzzling observations to explanatory hypotheses, is in fact inherent in a vast majority of problem solving and knowledge acquisition tasks. The scope of applications is immense and varies from scientific discovery, over medical and engineering diagnosis, design problems, planning, language and multimedia interpretation, to example generation in tutoring systems. In the face of such a widespread and diverse application interest much research has been devoted to gaining a better understanding of the theoretical foundations of abductive reasoning [1, 16, 40] and to developing computational frameworks for abduction, mainly in the context of logic programming [27, 14], but also with certain insightful proposals founded on the standard logical calculi, such as semantic tableaux [32, 33, 1] or resolution [11, 36], targeted at reasoning in propositional, first-order and several modal logics.

A new, challenging area for exploring the potential of abductive reasoning, which we aim to address in this paper, is Description Logic (DL) [3]. DL has become a leading paradigm of logic-based knowledge representation, a status that has been acknowledged in the course of standardization efforts for the Semantic Web, embracing DLs as the logical underpinning of the Web Ontology Language [26].[1] Due to the growing popularity of the formalism, there has been an ever rising demand for efficient tools providing different reasoning services for DL knowledge bases. Whereas highly optimized deductive reasoning algorithms for expressive DLs abound and are readily available [20, 24, 29, 35], the advances on non-standard types of inference —in particular abduction— are still very limited, though the need for them is obvious.

In their programmatic paper, Elsenbroich et al. [13] advocate initiating research on abduction in the context of DL ontologies, supporting their case with several application scenarios. For instance, the user of a medical ontology, covering descriptions of health disorders and their symptoms, should appreciate the possibility of querying the knowledge base for a short list of plausible diagnoses based on a patient's medical record. Ontology engineers, on the other hand, can benefit from having tool support for identifying minimal sets of axioms that should be inserted into a knowledge base for a certain entailment to hold [6]. Practically every research community interested in applying DL/Semantic Web technologies to their specific domains, such as *e*-Science, medical informatics, law and AI, computational linguistics or computer-supported engineering and design, can easily extend the list of feasible use cases for abduction over ontologies.

In this paper we study *ABox abduction* in DL, i.e. the type of abductive inference constitutive for problems of finding all minimal sets of ABox axioms, such that added to the knowledge base each of them triggers entailment of the initially specified set of ABox assertions. We propose a computational framework for solving this kind of problems in the DL $\mathcal{ALC}$, and argue for its adequacy and universal character that can facilitate extensions to more expressive DLs. Our work, being to the best of our knowledge a so far unique attempt of addressing such a form of reasoning, is thus the first step towards the creation of practical abductive reasoners for the family of DL languages.

[1] See `http://www.w3.org/TR/owl-features/` and `http://www.w3.org/TR/owl2-profiles/`.

$$
\begin{aligned}
\mathcal{T} = \{ \quad & \textsc{Optimist} \sqcup (\textsc{Nihilist} \sqcap \exists \mathsf{owns}.\textsc{Dog}) \sqsubseteq \textsc{Happy} \\
& \forall \mathsf{watches}.\textsc{Comedy} \sqsubseteq \textsc{Optimist} \quad \} \\
\mathcal{A} = \{ \quad & \textsc{Nihilist}(\mathsf{John}), \quad \textsc{Dog}(\mathsf{Snoopy}) \quad \} \\
Q = \quad & \textsc{Happy}(\mathsf{John}) \ ?
\end{aligned}
$$

Fig. 1 *Happy John* ABox abduction problem.

Example. For a motivating illustration of the problem, which will serve as the running example in the remainder of the paper, consider a simple knowledge base (Figure 1) defined by the terminology $\mathcal{T}$ (TBox) and the two assertions given in $\mathcal{A}$ (ABox). The terminology states that every individual who is an optimist or who is a nihilist owning a dog is happy, whereas every individual who watches only comedies is an optimist. Moreover, it is known that John is a nihilist and Snoopy a dog. Given this background one might want to find out what kind of facts, i.e. what ABox assertions, should be true in the described world for John to be naturally considered a happy individual, or more formally, for the assertion $\textsc{Happy}(\mathsf{John})$ to be derivable from the knowledge base. Clearly, there are several alternatives. For instance, one can conjecture that John is an optimist, which automatically renders him an instance of happy individuals. Another, more specific guess is that John watches only comedies, as then he is obviously an optimist, and thus a happy individual. Also, since $\textsc{Nihilist}(\mathsf{John})$ is already in the knowledge base, the requested statement is entailed by assuming that John owns a dog, or in particular, that John owns Snoopy, who is already known to be a dog. □

In the presented example, the background knowledge together with the query $\textsc{Happy}(\mathsf{John})$ form *an ABox abduction problem*, whereas the briefly listed alternatives constitute its *solutions*. Arguably, reasoning tasks of this sort comprise an essential share of all abduction problems in DL, in particular the majority of cases of potential attractiveness for end-user applications.

In this paper we introduce a framework for solving ABox abduction problems in the DL $\mathcal{ALC}$. As its central part we define two reasoning calculi, based on refinements of two well-known automated theorem proving techniques: *regular connection tableaux* and *resolution with set-of-support*. Both proof methods, enjoying the benefits of connection-driven decision procedures for satisfiability, in the sense originally formulated by Bibel [7], exhibit a goal-oriented behavior in solving abductive problems. Roughly, the approach allows for conducting the search only among those formulas that have good chances of contributing to the abductive solution, discarding possibly large parts of the knowledge base that are irrelevant for the problem. Next to the calculi, we define a special clausal transformation of DL axioms, which uses the techniques of flattening and Skolemization of formulas under standard translation, and finally we specify a procedure of reconstructing abductive solutions from the parts of the proofs generated by the calculi. The whole method is proven sound and complete.

The presentation of the work is organized as follows. In the next section we set out the formal framework for ABox abduction in $\mathcal{ALC}$, introducing basic notions and discussing the requirements for the procedure. We also review other work related to the problem. In Section 3 we provide a detailed account of our approach, describing the transformation of DL formulas, the two calculi and the method of reconstructing solutions. Section 4 contains the proofs of some logical properties of the framework, notably soundness and completeness, and addresses additional constraints typically applied for restricting the space of abductive solutions. Finally, we conclude the paper with a summary and a discussion of the results.

2 Problem definition

DLs are a family of logical languages intended particularly for representing knowledge about a domain of application. With their rich means of expressiveness and epistemically motivated clustering of knowledge bases into the terminological and factual layer, DLs provide a wide range of interesting contexts for investigating and applying forms of abductive reasoning. ABox abduction, for the first time formally identified in [13], comes here as conceptually the simplest, nevertheless computationally very demanding type of abduction, which requires reasoning over both layers of the knowledge base. In this section we introduce the preliminary notions and discuss the background necessary for explaining and justifying our approach to solving ABox abduction problems.

2.1 Preliminaries

A signature of a DL language $\mathcal{L}$ consists of a set of individual names N_I, a set of concept names N_C and a set of simple roles N_R [4]. The semantics is given by an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty domain of individuals and $\cdot^{\mathcal{I}}$ is an interpretation function defining the meaning of the vocabulary by mapping every individual name to an individual from $\Delta^{\mathcal{I}}$, every concept name to a subset of individuals, and every role name to a set of pairs of individuals from the domain. By default, we also treat $\top$ (top concept) and $\bot$ (bottom concept), where $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$, $\bot^{\mathcal{I}} = \emptyset$, as fixed symbols in the language. The remainder of the semantics is defined inductively on the construction rules for complex expressions available in the given language. In the following, we will consider only languages including concept constructors presented in Table 1.

Constructor	Syntax	Semantics
concept negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
concept intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
concept union	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \mid \exists_y(\langle x, y\rangle \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$
universal restriction	$\forall r.C$	$\{x \mid \forall_y(\langle x, y\rangle \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}})\}$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$

Table 1 The syntax and semantics of complex concept constructors.

A DL knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. The TBox is a formal representation of the terminological part of the knowledge base, establishing relationships between concepts and roles. We allow the liberal representation of general TBoxes, based on *general concept inclusions* (GCIs), that is, axioms of the type $C \sqsubseteq D$, where C and D are arbitrary concept descriptions. In this case C is said to be a subconcept of D. The equivalence of concepts, denoted as $C \equiv D$, is an abbreviation for two GCIs holding between C and D and vice versa. The ABox of a knowledge base consists of a set of assertions about individuals, of the form $C(a)$ or $r(a, b)$, where a, b are names of individuals, C is a concept description, and r is a role. The former states that a is an instance of C, whereas the latter expresses that individual a is related to b via role r.

The semantics of TBox and ABox axioms is defined in a standard way, presented in Table 2. An interpretation $\cdot^{\mathcal{I}}$ *satisfies* an axiom if and only if the semantics of the axiom is respected under $\cdot^{\mathcal{I}}$. An interpretation is a *model* of a knowledge base when it satisfies all its axioms. Finally, we say that a knowledge base is *satisfiable* if and only if it has at least one model. Else, the knowledge base is *unsatisfiable*.

Axiom	**Semantics**
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$C \equiv D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$
$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
$r(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in r^{\mathcal{I}}$

Table 2 Semantics of DL axioms.

2.2 ABox abduction problems and solutions

The following two definitions introduce the central notions of *ABox abduction problem* and *solution* to such a problem.

Definition 1 (ABox abduction problem) Let $\mathcal{L}_K$ and $\mathcal{L}_Q$ be DLs, $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ a **knowledge base** in $\mathcal{L}_K$ and Φ a set of ABox assertions in $\mathcal{L}_Q$, denoted as the **abductive query**. We call the tuple $\langle \mathcal{K}, \Phi \rangle$ an **ABox abduction problem** *iff* $\mathcal{K} \not\vDash \Phi$ and $\mathcal{K} \cup \Phi \not\vDash \bot$.

Definition 2 (ABox abduction solution) Let $\mathcal{L}_S$ be a DL and A a set of ABox assertions in $\mathcal{L}_S$. A is a (**plain**) **solution** to abductive problem $\langle \mathcal{K}, \Phi \rangle$ *iff* $\mathcal{K} \cup A \vDash \Phi$. Moreover, we call A:

1. **consistent** *iff* $\mathcal{K} \cup A \not\vDash \bot$.
2. **relevant** *iff* $A \not\vDash \Phi$.
3. **minimal** *iff* there is no solution B to $\langle \mathcal{K}, \Phi \rangle$ that is **minimal with respect** to A. We say that B is **minimal with respect to** A *iff* there exists a renaming $\rho : N_I^\star(B) \mapsto N_I^\star(A)$, where $N_I^\star(B)$ and $N_I^\star(A)$ are the sets of individual names from A and B that do not occur in $\mathcal{K}$, such that $A \vDash \rho B$, but for every renaming $\varrho : N_I^\star(A) \mapsto N_I^\star(B)$ it holds that $B \not\vDash \varrho A$.

ABox abduction, in the above sense, is the problem of finding a set of assertions A that, when added to the knowledge base $\mathcal{K}$, triggers entailment of a desired set of ABox axioms Φ, which otherwise does not follow from $\mathcal{K}$. The notion of entailment and its symbol $\models$ are understood here simply as the *classical consequence relation*, meaning that A entails B ($A \models B$) if every model of A is at the same time a model of B. We generalize the definition proposed in [13] by allowing multiple assertions as elements of an abductive query, interpreting them as implicitly connected by conjunction. We also adopt a purely logical perspective on abductive reasoning, and depart from the traditional, philosophically influenced nomenclature, which silently implies the existence of an explanatory or causal relationship between the premise and a conclusion of the inference. Instead of *explanandum* and *explanans*, common in the literature, e.g. Elsenbroich [12], we use the neutral notions of *the query* and *a solution* to the problem.[2] The only constitutive feature of an abductive problem in this setting is, therefore, indetermination of the truth value of the query given the background knowledge, whereas that of a solution is its potential of forcing this value to true when coupled with the knowledge base.

Since the space of abductive solutions can be in principle infinite, it is common to employ additional constraints to narrow it down, at least by excluding obviously unacceptable solutions, and even further, to a fragment of a higher pragmatic value from the application perspective. The choices proposed here, widely approved in the studies on the subject, e.g. Aliseda [1], Paul [36], should be to our opinion the least controversial, as they embrace arguably the most intuitive and universal criteria used in all applications of abductive reasoning.

- The *consistency* requirement discards solutions inconsistent with the knowledge base. For instance, if $\neg\textsc{Optimist}(\mathsf{John})$ followed from the knowledge base, in the introductory example, then $\textsc{Optimist}(\mathsf{John})$ would not be a consistent solution to the problem. Obviously, it is not rational to conjuncture something that is necessarily false.
- The *relevance* condition filters out those solutions that entail the query without any contribution of the background knowledge. Such outcomes trivialize the problem instead of really solving it, as it is the case, for example, with solution $\textsc{Happy}(\mathsf{John})$ to the *happy John* problem.
- The requirement for *minimality* or simplicity of abduced hypotheses is often a subject of debate in the literature, and in fact, several different, incompatible criteria have been proposed. Since it is not our intention to favor any particular view on abduction, we refrain from adjudicating between the proposals, and instead abide by the weakest and most fundamental notion of minimal solutions in the analytical sense of Quine's *prime implicants* [37]. The *minimality* criterion in this meaning ensures that solutions do not contain superfluous information, i.e. that one does not abduce more than is necessary. Clearly, there is no point in conjecturing that $(\textsc{Optimist} \sqcap \textsc{Nihilist})(\mathsf{John})$ if $\textsc{Optimist}(\mathsf{John})$ alone is already sufficient to solve the problem. Naturally, one can easily plug in a stronger notion of minimality on top of this one.

We will further address the problem of selection criteria, including the computational aspects of their verification, in Sections 4.2 and 4.3.

In this paper we restrict our attention to ABox abduction problems, the knowledge bases of which are expressed in $\mathcal{ALC}$, the basic attributive language with complex concept negation, which is the most prominent fragment of DL, covering an essential part of expressive means available in DLs. The syntax of the queries and abductive solutions is restricted to the conjunctive variant of $\mathcal{ALC}$, namely $\mathcal{ALE}$. The syntax of the two languages is recapped in Table 3. Noticeably, in $\mathcal{ALE}$ the union constructor is

[2] The variability of the abductive context has been recently highlighted also by Gabbay and Woods [19].

$$\mathcal{ALC}: \quad \top \mid \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall r.C \mid \exists r.C$$
$$\mathcal{ALE}: \quad \top \mid \bot \mid A \mid \neg A \mid C \sqcap D \mid \forall r.C \mid \exists r.C$$

Table 3 Concept constructors in the DLs $\mathcal{ALC}$ and $\mathcal{ALE}$.

prohibited and use of negation is reserved only for atomic concepts. No new expressive means with respect to $\mathcal{ALC}$ are introduced.

The syntactic restriction that we use is typically applied in the context of abduction, e.g. Aliseda [1], Elsenbroich [12], for obtaining fine-grained and interesting problems and solutions. For instance, the assertion $(\textsc{Optimist} \sqcup \forall \mathsf{watches}.\textsc{Comedy})(\mathsf{John})$ would not be a legal $\mathcal{ALE}$ solution to the *happy John* problem, although it does solve the problem in the $\mathcal{ALC}$ language. Still, among $\mathcal{ALE}$ solutions one can find separately all the disjuncts comprising this assertion, i.e. $\textsc{Optimist}(\mathsf{John})$ and $\forall \mathsf{watches}.\textsc{Comedy}(\mathsf{John})$, which in most cases is exactly what one seeks through abductive reasoning: a list of alternative ways the world should be for the query to hold. Given such knowledge it follows analytically that also the disjunctions of these alternatives solve the problem. There are, however, also other types of disjunctive solutions that cannot be reconstructed in a similar manner. For instance if $\forall \mathsf{watches}.(\textsc{Comedy} \sqcup \textsc{Musical})(\mathsf{John})$ was a valid $\mathcal{ALC}$ solution to the problem, then only $\forall \mathsf{watches}.\textsc{Comedy}(\mathsf{John})$ and $\forall \mathsf{watches}.\textsc{Musical}(\mathsf{John})$ would be retrieved in $\mathcal{ALE}$, where the union of the two $(\forall \mathsf{watches}.\textsc{Comedy} \sqcup \forall \mathsf{watches}.\textsc{Musical})(\mathsf{John})$ is obviously not equivalent to the original $\mathcal{ALC}$ assertion. Admittedly, in such cases the disjunctive solutions are simply lost and unrecoverable.

2.3 Related work

A discussion on the place for abduction in the context of DLs has been initiated by Elsenbroich et al. [13], who introduced a broad classification of the relevant types of abductive problems, coined some of the basic terminology, provided a number of use case scenarios, and finally outlined a far-reaching research programme on the subject. The call for tool support for abductive reasoning required in ontology engineering has been repeated also by Bada et al. [6].

Before that abduction in DL has been studied only by Colucci et al. [10], who proposed a tableaux-based algorithm for concept abduction, the problem of finding all subconcepts of a given concept, in order to support so-called matchmaking tasks. Also some attention has been given by Espinosa et al. [15], reported by Möller and Neumann [34], to the problem of facilitating interpretation of visual data using ontologies with DL rules. There, the authors discuss a simple inference mechanism for ABox abduction over the rule bodies, which suggests ways of enhancing the conceptualization of the data. Admittedly, the approach taken in both cases is quite narrowly scoped. The first one, although essentially based on similar principles as ours, is limited to unfoldable, acyclic terminologies in the DL $\mathcal{ALN}$. Potential extensions towards more expressive DLs, in particular ones offering at least all boolean operations in concept descriptions, do not seem trivial and would definitely require significant revisions in the employed calculus. Moreover, the approach is unsuitable for harder reasoning tasks, such as ABox abduction, considered in this paper. The second framework, presented by Espinosa et al., does not in fact involve genuine abductive reasoning for DL, but merely abduction over rules accompanying DL ontologies. Thus the style of inference used there falls much closer to the paradigm of abductive logic programming, whereas DL reasoning occurs only in the most standard form, as a support for deductive parts of the process. Both methods are therefore of a little help for the goal of this work, which is to propose a universal framework for ABox abduction in DL. Even though our focus is on the DL $\mathcal{ALC}$, we want the approach to be generic, so that lifting it to more expressive extensions of $\mathcal{ALC}$ is possible.

Other loosely related work includes results on the computational complexity of concept abduction in DL $\mathcal{EL}$, obtained by Bienvenu [8], and some proposals concerning other reasoning tasks, which reveal some affinity with the abductive style of inference. The latter include the work of Schlobach et al. [39] on debugging incoherent terminologies, where the problem is to find a minimally unsatisfiable subset of TBox axioms, and the proposals of Kalyanpur et al. [28], and later Horridge et al. [23], on finding justifications, i.e. minimal sets of axioms of an ontology that make a particular entailment of the ontology hold. Both tasks can be seen as borderline cases of abduction, where the query (the bottom concept in the former problem, and the specified entailment in the latter) already follows from the knowledge base, but still one has to check why it is the case, i.e. which subset of the ontology exactly guarantees the entailment. Regardless of this similarity, the formal properties of both problems permit a much simpler computational treatment, based on looking up into the inference graph generated by a standard DL reasoner, or even using the reasoner as a black-box. Hence, the solutions discussed by the cited authors bare no significant overlap with ours, which conversely, rest on deep adaptations of the standard automated reasoning techniques. On a different note, the works on justifications offer some interesting procedural explications of the notion of precision, which is closely related to our notion of minimality, but which in this paper is provided only with a semantic interpretation.

From the logical perspective, an important contribution has been delivered by Mayer and Pirri [32,33] and Aliseda-Llera [1], who laid down foundations for universal tableaux-based algorithms for abduction in propositional, modal logics (MLs) and first-order logic (FOL). Prior to these, quite different approaches to abduction in logic, built on linear resolution, have been investigated by Cox and Pietrzykowski [11] and employed in several applications. The main idea, promoted by these authors, of grounding abductive reasoning on the standard refutation proof systems has largely shaped the conceptual basis of the framework presented here, although a shift towards DLs presents a number of technical challenges not occurring in the case of logics addressed in the cited works. Moreover, we account for a goal-oriented style of reasoning, which was also not considered by the others. Such an approach to abduction in logic has only been studied by Elsenbroich [12], but rather than building on standard calculi, this work is based on the goal-directed approach to deduction of Gabbay and Olivetti [17, 18], and as such remains hardly comparable to ours. Likewise, goal-oriented abduction in the context of logic programming, as discussed e.g. by Kakas et al. [27], is incommensurable with the algorithms discussed here due to the fundamental discrepancy between the properties of the underlying formalisms.

2.4 Requirements

Based on a number of insights coming from the literature on abduction, DLs, and automated reasoning techniques, we have identified the following high-level requirements as the guiding principles for constructing the framework for ABox abduction.

- *Universality and flexibility*: The framework should be universal enough to be able to accommodate different expressive extensions of DL, thus enabling relatively uniform treatment of all DLs. Further, as signaled in Section 2.1, it should not incorporate any selection criteria for delimiting the scope of solutions, apart from ones that are supported by a firm epistemological justification, such as consistency, relevance and minimality in the sense defined above. Finally, it should not be confined to a particular search strategy, e.g. depth-first or breadth-first, leaving the issue open to customization. At the same time it has to be sufficiently flexible to allow tuning the solving strategy with respect to different dimensions. Preferably, one should be able to model within the framework a specific interpretation of the notion of *best hypothesis*, which is central to abduction, and adjust the balance of completeness/efficiency trade-off, inherent in abductive reasoning, according to the requirements of concrete application scenarios.
- *Utilization of standard reasoning methods*: Given the state-of-the-art advancements in the field of automated reasoning, especially concerning FOL, MLs and DLs, it is highly desirable to build the framework on reasoning calculi that offer good chances for integration with existing reasoners, thus facilitating reuse of well-developed and broadly applied methodologies and transfer of verified optimization techniques. In particular, since current DL reasoners are based almost exclusively on semantic tableaux [20, 24, 35] and resolution [29], the choice of refinements of these techniques as the foundation for abductive reasoning tools for DL knowledge bases seems most natural and promising.
- *Goal-orientedness*: Considering the issues of efficiency and basic principles of intelligent problem solving, ideally, the algorithm should exhibit a goal-oriented behavior. The reasoning should start from the abductive query as the goal to be explained and conduct a form of backward-chaining search for solutions through the formulas in the knowledge base. Consequently, the employed proof strategies should allow for a selective use of the background knowledge, such that those parts of the knowledge base that cannot contribute to solving the problem are not considered and thus do not introduce extra computational burden for reasoning. As will be pointed out in the beginning of the next section, one way of achieving goal-orientedness in this sense is employing *connection-driven* proof strategies [30].

The framework for ABox abduction, presented in the subsequent sections, uses a connection-based variant of semantic tableaux and a refinement of the resolution calculus. It involves standard transformation techniques for DL axioms, which can embed almost all expressive means of DLs, and finally, it does not depend on any selection criteria apart from those discussed in Definition 2. All known optimizations, search strategies, heuristics or other augmentations, which can be easily implemented in the calculi, are not intrinsic to the framework, and left merely as options. Our belief is that such a setting guarantees, to a sufficient extent, satisfaction of the requirements.

3 Computing solutions

In this section we present the computational framework for solving ABox abduction problems in the DL $\mathcal{ALC}$. We start with a general overview of the approach, indicating its formal and conceptual foundations, and provide an outline of its structure. In the subsequent parts we proceed with presenting the details. We close the section with an example of solving an ABox abduction problem in the framework.

3.1 Roadmap: A high-level overview

The reasoning mechanism constitutive to our approach to solving ABox abduction problems rests on two observations concerning the *model-theoretic* and *proof-theoretic* aspects of abductive inference, respectively.

On the *model-theoretic* side, solving an abduction problem can be seen as finding a formula that is unsatisfiable in all those models of the knowledge base in which the abductive query is not satisfied, therefore a formula that can eliminate all unintended models of the background knowledge. To solve a problem, one requires an overview of those models, or at least of their parts, in order to decide which formulas can succeed in eliminating them. This observation, endorsed by several authors [11, 32, 33, 1] and followed here, can be exploited within standard refutation proof systems such as resolution and semantic tableaux, which given the negated abductive query along with the knowledge base as input, provide a necessary insight into the structure of all the unintended models. In this context, recall that open branches in a tableau tree can be associated with possible models of the input, while resolvents in a resolution proof represent constraints that have to be satisfied in every such model.

From the *proof-theoretic* perspective, as shown elsewhere [30], the goal-oriented strategy of solving abduction problems, so typically employed by human reasoners, in which one sets the abductive query as the goal and moves backwards through the constraints of the knowledge base, setting intermediate goals, in order to identify consecutive, more indirect solutions, can be formally reinterpreted in terms of *connection proof methods*, in the sense originating from the works of Bibel [7] and Andrews [2]. These techniques, giving rise to a family of decision procedures for satisfiability in a number of logics, make central use of the notion of *connection*, i.e. an occurrence of complementary literals in two formulas in clausal form. Following the path of connections, one can easily identify all fragments of the knowledge base that are semantically interrelated, and so can be potentially relevant for solving a given abductive problem.

Our framework for ABox abduction incorporates features of both FOL and ML reasoning techniques. Thus we rely heavily on the well-known correspondence results between DL and the multi-modal logic K_n [38], and further, through the so-called *standard translation*, to FOL [9]. The proof system is based on the methods of *regular connection tableaux* [21] and *resolution with set-of-support* [31], both sound and complete reasoning calculi for FOL, and both hinging on connection-driven proof strategies.

Let us now roughly describe the procedure for solving an ABox abduction problem $\langle \mathcal{K}, \Phi \rangle$. Given the input transformed into FOL, we attempt to construct a refutation proof for $\mathcal{K} \vDash \Phi$. For this we use either of the two calculi, hence initiating the proof with $\neg\Phi$ and trying to show that $\mathcal{K} \cup \neg\Phi \vdash \bot$. Notice that the proof succeeds if it is possible to close all branches of the tableau tree or derive an empty resolvent, respectively. At each stage of the construction of the proof it is possible to force its completion by adding a suitable conjunctive formula A, which either closes all open branches or contradicts some resolvent. Such a formula will be a plain solution to the translated problem $\langle \mathcal{K}, \Phi \rangle$, as obviously it follows that $\mathcal{K} \cup A \vDash \Phi$. After transforming A from FOL back to DL, for which one has to account for an underlying relational structure of an essentially modal character, we obtain a set of ABox assertions that solve the original problem. The following sections address particular aspects of the framework.

1. In Section 3.2 we present the transformation of DL formulas into a clausal form, which uses the standard translation to FOL, but at the same time also encodes the modal structure of the formulas.
2. In Section 3.3 we elaborate on the two reasoning calculi, providing details of the proof construction rules.
3. Section 3.4 accounts for the aspects of bookkeeping of the relational structure underlying the proofs. To this end we introduce the notion of an *abductive graph*. Further, we define the conditions under which reverse transformation from FOL to the DL $\mathcal{ALE}$ is possible.
4. Section 3.5 describes the procedure of reverse transformation, i.e. of retrieving solutions from a proof. This defines our target notion of a $\vdash_{\text{ABox}}$-*solution* to an ABox abduction problem.

The procedure $\vdash_{\text{ABox}}$ described in this section computes *plain* solutions to an ABox abduction problem. How to extend the basic method to also account for consistency, relevance and minimality will be discussed in Sections 4.2 and 4.3.

3.2 Transformation

A complete transformation of DL formulas into the representation required by the abductive procedure comprises translation into *Negation Normal Form* of TBox axioms and concept descriptions in ABox assertions, followed by a reduction to *Conjunctive Normal Form* and *flattening* of the clauses, i.e. extraction of nested concept descriptions from quantification restrictions. Finally, all the clauses are *Skolemized*, while their modal structure is recorded in a way which enables a faithful reconstruction of the Kripke models underlying the abductive proofs. Overall, we obtain an *equisatisfiable* and *structure-preserving* transformation of the input DL formulas. The entire transformation procedure is summarized in Table 7, while in the following paragraphs we discuss its particular fragments.

We start by extending the signature of the language with a set $\mathbb{F}_{sko} = \{f_1, f_2, \ldots\}$ of Skolem functions and a set $\mathbb{P} = \{P_1, P_2, \ldots\}$ containing non-DL predicates, possibly of different arity. We also assume there is an infinite set $\text{Var} = \{x_1, x_2, \ldots\}$ of variables. We refer generically to any term, a variable or a Skolem term, using letters $t, t_1, t_2, \ldots$, and write $\bar{t} = t_1, \ldots, t_n$ to denote their sequence. We use $\cdot^\star$ to mark introduction of new symbols: $x^\star$ a new variable, $f^\star$ a new function and $P^\star$ a new non-DL predicate. Arbitrary predicates, DL or non-DL, are denoted by capital letters $L, L_1, L_2, \ldots$. We assume that in such contexts $\neg L_i$ stands for the complement of L_i. Below we present an outline of the three stages of transformation, marked by $\tau_\neg$, $\tau_\sqcap$ and $\tau_t^{\overline{x}}$, for NNF, CNF (involving flattening) and Skolemization, respectively. The layering of the process is rather schematic, as in practice it should be much more efficient to interleave the transformation steps belonging to different stages.

$\tau_\neg(\neg\neg C)$	$=$	$\tau_\neg(C)$	$\tau_\neg(\neg(C \sqcup D))$	$=$	$\tau_\neg(\neg C) \sqcap \tau_\neg(\neg D)$
$\tau_\neg(\neg A)$	$=$	$\neg A$	$\tau_\neg(\neg(C \sqcap D))$	$=$	$\tau_\neg(\neg C) \sqcup \tau_\neg(\neg D)$
$\tau_\neg(A)$	$=$	A	$\tau_\neg(\neg\forall r.C)$	$=$	$\exists r.\tau_\neg(\neg C)$
$\tau_\neg(\neg\bot)$	$=$	$\neg\bot$	$\tau_\neg(\neg\exists r.C)$	$=$	$\forall r.\tau_\neg(\neg C)$
$\tau_\neg(\bot)$	$=$	$\bot$	$\tau_\neg(C \sqcup D)$	$=$	$\tau_\neg(C) \sqcup \tau_\neg(D)$
$\tau_\neg(\top)$	$=$	$\neg\bot$	$\tau_\neg(C \sqcap D)$	$=$	$\tau_\neg(C) \sqcap \tau_\neg(D)$
$\tau_\neg(\neg\top)$	$=$	$\bot$	$\tau_\neg(\forall r.C)$	$=$	$\forall r.\tau_\neg(C)$
			$\tau_\neg(\exists r.C)$	$=$	$\exists r.\tau_\neg(C)$

Table 4 Negation Normal Form transformation.

$\tau_\sqcap(\bigsqcup_{1\leq i\leq n} C_i)$	$=$	$\bigsqcup_{1\leq i\leq n} C_i$, for $C_i \in \{L, \forall r.P, \exists r.P\}$
$\tau_\sqcap(B \sqcup (C \sqcup D) \sqcup E)$	$=$	$\tau_\sqcap(B \sqcup C \sqcup D \sqcup E)$
$\tau_\sqcap(B \sqcup (C \sqcap D) \sqcup E)$	$=$	$\tau_\sqcap(B \sqcup C \sqcup E)$, $\tau_\sqcap(B \sqcup D \sqcup E)$
$\tau_\sqcap(B \sqcup \forall r.C \sqcup D)$	$=$	$\tau_\sqcap(B \sqcup \forall r.P^\star \sqcup D), \tau_\sqcap(\neg P^\star \sqcup C)$
$\tau_\sqcap(B \sqcup \exists r.C \sqcup D)$	$=$	$\tau_\sqcap(B \sqcup \exists r.P^\star \sqcup D), \tau_\sqcap(\neg P^\star \sqcup C)$

Table 5 Conjunctive Normal Form transformation with flattening of the axioms.

The NNF and CNF transformations of $\mathcal{ALC}$ concept descriptions, presented in Tables 4 and 5, are standard and do not require any comments. The flattening technique, included in Table 5, is also a relatively common practice, used for instance in [41] and [23]. Under flattening the qualifying concept descriptions in the quantification restrictions are replaced with new predicate symbols, which are then related to the descriptions by means of separate GCIs. Such an approach is desirable for facilitating connection-driven construction of proofs, as given a flattened formula one obtains a direct access to all its literals at any depth. Figure 2 presents a small example of the transformation.

$\neg(\textsc{Optimist} \sqcup (\textsc{Nihilist} \sqcap \exists \mathsf{owns}.\textsc{Dog})) \sqcup \textsc{Happy}$	$\neg\textsc{Optimist} \sqcup \textsc{Happy}$
	$\neg\textsc{Nihilist} \sqcup \forall \mathsf{owns}.P_1 \sqcup \textsc{Happy}$
	$\neg P_1 \sqcup \neg\textsc{Dog}$
$\neg(\forall \mathsf{watches}.\textsc{Comedy}) \sqcup \textsc{Optimist}$	$\exists \mathsf{watches}.P_2 \sqcup \textsc{Optimist}$
	$\neg P_2 \sqcup \neg\textsc{Comedy}$
$\textsc{Nihilist}$	$\textsc{Nihilist}$
$\textsc{Dog}$	$\textsc{Dog}$
$\neg\textsc{Happy}$	$\neg\textsc{Happy}$

Fig. 2 *Happy John* problem: NNF and CNF transformations and flattening.

The transformation of a concept description through $\tau_\neg$ and $\tau_\sqcap$ results in a set of unions. With every such union $C = C_1 \sqcup \ldots \sqcup C_n$ we associate the clause comprising all its disjuncts $Cl = \{C_1, \ldots, C_n\}$. Within any set of such clauses Γ we distinguish the subset of their *roots* $\mathbf{R}(\Gamma)$, i.e. all those clauses that do not contain a literal $\neg P_i$ for any $P_i \in \mathbb{P}$, and the remaining subset of the *non-root* clauses $\mathbf{nR}(\Gamma)$.

Finally, all clauses are translated to FOL, as presented in Table 6, by means of the transformation $\tau_t^{\overline{x}}$, where the subscript contains a single term and the superscript a possibly empty sequence of variables. The translation function takes as the input a clause $Cl \in \Gamma$ along with the set of all non-root clauses $\mathbf{nR}(\Gamma)$. All quantification restrictions in the clause are replaced with the corresponding FOL expressions according to the standard translation, while the predicates are suitably Skolemized. During this process the modal structure of every clause, originally encoded in its quantification restrictions, is extracted and recorded as the so-called *modal core* μ of the clause, i.e. the graph describing role relationships between the terms occurring in the Skolemized clause. The modal cores of clauses are used later in the process of finding well-formed ABox solutions to abduction problems. The output of the transformation includes therefore a set of Skolemized clauses (so-called τ-clauses), each one accompanied by its modal core. Observe, that in order to Skolemize flattened formulas in a

$$
\begin{array}{rcll}
\tau_x^{\overline{x}}(\{A\} \cup Cl) & = & \{A(x)\} \cup \tau_x^{\overline{x}}(Cl) & \mu := \mu(Cl) \\
\tau_x^{\overline{x}}(\{\neg A\} \cup Cl) & = & \{\neg A(x)\} \cup \tau_x^{\overline{x}}(Cl) & \mu := \mu(Cl) \\
\\
\tau_x^{\overline{x}}(\{\forall r.P_i\} \cup Cl) & = & \{\neg r(x, x^\star), P_i(\overline{x}, x^\star)\} \cup \tau_x^{\overline{x}}(Cl), & \mu := \{r(x, x^\star)\} \cup \mu(Cl) \\
 & & \{\neg P_i(\overline{x}, x^\star)\} \cup \tau_{x^\star}^{\overline{x}, x^\star}(Cl') & \mu := \mu(Cl') \\
 & & \text{for every } \{\neg P_i\} \cup Cl' \in \mathbf{nR}(\mathit{\Gamma}) & \\
\\
\tau_x^{\overline{x}}(\{\exists r.P_i\} \cup Cl) & = & \{r(x, f^\star(\overline{x})\} \cup \tau_x^{\overline{x}}(Cl), & \mu := \{r(x, f^\star(\overline{x}))\} \cup \mu(Cl) \\
 & & \{P_i(\overline{x})\} \cup \tau_x^{\overline{x}}(Cl), & \mu := \{r(x, f^\star(\overline{x}))\} \cup \mu(Cl) \\
 & & \{\neg P_i(\overline{x})\} \cup \tau_{f^\star(\overline{x})}^{\overline{x}}(Cl') & \mu := \mu(Cl') \\
 & & \text{for every } \{\neg P_i\} \cup Cl' \in \mathbf{nR}(\mathit{\Gamma}) &
\end{array}
$$

Table 6 Skolemization and the modal core.

satisfiability preserving manner, the non-DL predicates have to be used for carrying over all universally bound variables into separated subformulas. The sequences of these variables are noted down as the superscripts of the transformation function. This way all Skolem functions that might possibly occur on the deeper levels of the nestings, obtain appropriate arguments. Since non-DL predicates uniquely identify the points of split, it is guaranteed that the connection can be established only in the original place and that all variables originally shared between a sub– and their superformulas will be fine-tuned by unification. An example of Skolemized clauses along with their modal cores is given in Figure 3.

Table 7 presents the complete procedure for computing the τ-transformation of the formulas that comprise the input of an ABox abduction problem, i.e. a knowledge base, consisting of TBox and ABox axioms, and the negated query. The latter case requires a slightly more elaborate approach, as the assertions in the query are implicitly connected by conjunction. The negation of the query is, therefore, equivalent to the disjunction of the negations of those assertions, which has to be properly reflected in defining the resulting root clauses and their modal cores.

As a final remark concerning transformation, we note that in practice it is not necessary to thoroughly pre-process the knowledge base via all the translation rules defined above. Since the proof procedures used in the framework are connection-driven it is possible to benefit from their specific character also on the level of transformation, rendering it equally goal-oriented. Notice that, having a formula translated into NNF, one can easily answer whether it is relevant for a given part of the proof. Given such knowledge, the remainder of the transformation of the formula can be deferred until a particular connection is actually requested.

$$
\begin{array}{rl}
\{\neg\textsc{Optimist}, \textsc{Happy}\} & \mid\ 1 : \{\neg\textsc{Optimist}(x_1), \textsc{Happy}(x_1)\} \\
\{\neg\textsc{Nihilist}, \forall\mathsf{owns}.P_1, \textsc{Happy}\} & \mid\ 2 : \{\neg\textsc{Nihilist}(x_1), \neg\mathsf{owns}(x_1, x_2), P_1(x_1, x_2), \textsc{Happy}(x_1)\} \\
\{\neg P_1, \neg\textsc{Dog}\} & \mid\ 3 : \{\neg P_1(x_1, x_2), \neg\textsc{Dog}(x_2)\} \\
\\
\{\exists\mathsf{watches}.P_2, \textsc{Optimist}\} & \mid\ 4 : \{\mathsf{watches}(x_1, f_1(x_1)), \textsc{Optimist}(x_1)\} \\
 & \mid\ 5 : \{P_2(x_1), \textsc{Optimist}(x_1)\} \\
\{\neg P_2, \neg\textsc{Comedy}\} & \mid\ 6 : \{\neg P_2(x_1), \neg\textsc{Comedy}(f_1(x_1))\} \\
\\
\{\textsc{Nihilist}\} & \mid\ 7 : \{\textsc{Nihilist}(\mathsf{John})\} \\
\{\textsc{Dog}\} & \mid\ 8 : \{\textsc{Dog}(\mathsf{Snoopy})\} \\
\{\neg\textsc{Happy}\} & \mid\ 9 : \{\neg\textsc{Happy}(\mathsf{John})\}
\end{array}
$$

$$
\begin{array}{rl}
1, 3, 6, 7, 8, 9 : & \mu = \emptyset \\
2 : & \mu = \{\mathsf{owns}(x_1, x_2)\} \\
4, 5 : & \mu = \{\mathsf{watches}(x_1, f_1(x_1))\}
\end{array}
$$

Fig. 3 *Happy John* problem: Skolemization and the modal core.

3.3 Tableaux and resolution-based abduction

Semantic tableaux [21] and resolution [5] are the two best known and most commonly used automated reasoning methods for FOL, with a plethora of refinements, optimization techniques and extensions to other logics available. In the following we explain the procedure of solving abductive problems based on application of two variants of the calculi: *regular connection tableaux* and *resolution with set-of-support.* Our approach diverges from similar proposals presented in the literature [11, 32, 33, 1], which differently, refer to the standard tableau and linear resolution in addressing abductive inference. We assume acquaintance with the basics of both calculi and only briefly characterize them below in order to introduce the respective refinements.

A *clause tableau* is a labeled tree, whose nodes are literals and whose root contains a set of clauses. The tree is developed by consecutive applications of the *beta expansion rule* to the clauses. Each clause can be expanded only once on a branch. Whenever a branch contains complementary literals or the symbol $\bot$, the *closure rule* can be applied. A tableau is *saturated* if no more expansion steps are possible. A tree T is a *tableau refutation proof* of Φ from $\mathcal{K}$, denoted as $\mathcal{K} \vdash \Phi$, if the root of T contains $\mathcal{K} \cup \neg\Phi$ and all branches of T are closed. A *regular connection tableau* is a clause tableau, whose construction is restricted by the following conditions:

- *Connectedness*: A clause can be expanded on a branch only if it contains a literal that is complementary to the literal in the current leaf.
- *Regularity*: A clause can be expanded on a branch only if it does not contain a literal that already occurs on the branch.

Table 8 summarizes the inference rules applicable in regular connection tableaux.

The *resolution* method is based on repetitive application of two inference rules — *binary resolution* and *factoring*— to a set of clauses. On every application of a rule, a new clause, a *resolvent* or a *factor*, is generated and added to the set of all clauses. A *resolution deduction* of Φ from a set of clauses $\mathcal{K}$, denoted as $\mathcal{K} \vdash \Phi$, is a derivation of an empty clause from $\mathcal{K} \cup \neg\Phi$ by means of the rules. The inference halts when none of the rules can be applied anymore. In such a case it is said that the resulting set of clauses is *saturated.* A resolution deduction of Φ from $\mathcal{K}$ is a *deduction with set-of-support* $S \subseteq \mathcal{K}$

axiom transformation	**COMPUTE** $\tau(\varphi)$: **IF** $\varphi = C \equiv D$ **THEN OUTPUT** $\tau(\varphi) = \tau(C \sqsubseteq D) \cup \tau(D \sqsubseteq C)$ **IF** $\varphi = C \sqsubseteq D$ **THEN** $\Gamma := \tau_{\sqcap} \circ \tau_{\neg}(\neg C \sqcup D)$ **OUTPUT** $\tau(\varphi) = \{\tau^{x^\star}_{x^\star}(Cl) \mid Cl \in \mathbf{R}(\Gamma)\}$ **IF** $\varphi = C(a)$ **THEN** $\Gamma := \tau_{\sqcap} \circ \tau_{\neg}(C)$ **OUTPUT** $\tau(\varphi) = \{\tau_a(Cl) \mid Cl \in \mathbf{R}(\Gamma)\}$ **IF** $\varphi = r(a,b)$ **THEN** $\mu(\{r(a,b)\}) = \{r(a,b)\}$ **OUTPUT** $\tau(\varphi) = \{\{r(a,b)\}\}$ **IF** $\varphi = \neg r(a,b)$ **THEN** $\mu(\{\neg r(a,b)\}) = \{r(a,b)\}$ **OUTPUT** $\tau(\varphi) = \{\{\neg r(a,b)\}\}$
knowledge base transformation	**COMPUTE** $\tau(\mathcal{K})$, for $\mathcal{K} = (\mathcal{T}, \mathcal{A})$: **OUTPUT** $\tau(\mathcal{K}) = \{\tau(\varphi) \mid \varphi \in \mathcal{T} \cup \mathcal{A}\}$
negated query transformation	**COMPUTE** $\tau(\neg\Phi)$, for $\Phi = \{\varphi_1, \ldots, \varphi_n\}$: $\Gamma := \mathbf{R}(\tau(\neg\varphi_1)) \times \ldots \times \mathbf{R}(\tau(\neg\varphi_n))$, where: $\neg\varphi = \neg r(a,b)$ *iff* $\varphi = r(a,b)$; $\neg\varphi = \neg C(a)$ *iff* $\varphi = C(a)$ $\mathbf{R}(\tau(\neg\Phi)) := \{\bigcup_{1 \leq i \leq n} Cl_i \mid \langle Cl_1, \ldots, Cl_n \rangle \in \Gamma\}$ $\mu(\bigcup_{1 \leq i \leq n} Cl_i) := \bigcup_{1 \leq i \leq n} \mu(Cl_i)$ $\mathbf{nR}(\tau(\neg\Phi)) := \bigcup_{\varphi \in \Phi} \mathbf{nR}(\tau(\neg\varphi))$ **OUTPUT** $\tau(\neg\Phi) = \mathbf{R}(\tau(\neg\Phi)) \cup \mathbf{nR}(\tau(\neg\Phi))$

Table 7 Complete τ-transformation procedure.

if every resolvent has at least one parent that is (a factor of) a resolvent or (a factor of) a member of S. Table 9 presents the two inference rules used in resolution with set-of-support.[3]

In case of both methods we shall assume that the selection procedure for inference steps is *fair*, i.e. that no clause that can be potentially used in the proof is persistently omitted. We also require that all variables in a clause included to the proof are consistently renamed, in order to avoid unintended interactions with already used variables. Under certain conditions concerning the choice of the first clause to be expanded on the tableau and the choice of the set-of-support for resolution, which we highlight in Section 4, the calculi are sound and complete for FOL [21,31]. As for now, let us take it for granted that in the context of abduction the choice of the negated abductive query as the initial clause or the set-of-support is sufficient to satisfy these conditions.

Both reasoning methods share some important formal similarities. First, they employ a connection-driven proof strategy, which means that a clause can be included in a proof, constructed by any of the methods, only if it can be *connected* to it. More

[3] Note that derivation of Cl from $Cl \cup \{\bot\}$, included in Table 9, should be seen as a special case of an application of the binary resolution rule. Since by definition $\bot = L \wedge \neg L$ it follows that $Cl \cup \{\bot\}$ can be replaced in S with clauses $Cl \cup \{L\}$ and $Cl \cup \{\neg L\}$, which can be subsequently resolved against each other, resulting in clause Cl.

$$\begin{array}{c} \{L_1,\dots,L_n\}\in\mathcal{K} \\ \vdots \\ \neg L \\ \hline L_1 \quad | \quad \dots \quad | \quad L_n \end{array} \qquad\qquad \begin{array}{c} L_1 \\ \vdots \\ \neg L_2 \\ \hline \times \end{array} \qquad \begin{array}{c} \\ \vdots \\ \bot \\ \hline \times \end{array}$$

$\beta -$ rule

iff there exists an MGU σ of L and L_i for some $1 \leq i \leq n$, and σ is applied to the whole tableau; for no $1 \leq i \leq n$ there is L_i on the branch above.

Branch closure

iff there exists an MGU σ of L_1 and L_2 and σ is applied to the whole tableau.

Table 8 Regular connection tableau rules.

$$\frac{Cl_1 \cup \{L_1\} \in \mathcal{K} \qquad Cl_2 \cup \{\neg L_2\} \in S}{\sigma(Cl_1 \cup Cl_2) \in S} \qquad \frac{Cl \cup \{\bot\} \in S}{Cl \in S} \qquad \frac{Cl \cup \{L_1, L_2\} \in S}{\sigma(Cl \cup \{L_1\}) \in S}$$

Binary resolution **Factoring**

iff there exists an MGU σ of L_1 and L_2.

Table 9 Resolution with set-of-support rules.

specifically, a clause $Cl \cup \{L_1(t_1)\}$ can be connected to an abductive proof through a literal $L_2(t_2)$ occurring in that proof either as the leaf of an open branch or a literal in a resolvent only if there exists an MGU σ of $L_1(t_1)$ and $\neg L_2(t_2)$. As mentioned in the opening of this section, the proofs constructed according to such a strategy are typically structured in a more intuitive manner and involve less redundancy, in the sense of employing inference steps over clauses that are semantically irrelevant for the goal to be proved, and as such do not contribute to the proof. Second, both calculi are refutation proof systems. To prove that a certain conclusion is entailed by a set of premises one proves that the union of the premises and the negated conclusion is unsatisfiable. This characteristic makes both calculi especially attractive for abductive applications, allowing to identify all possible solutions of an abductive problem by attempting to construct a refutation proof for the negated abductive query, given the knowledge base.

Let us refer to a sample ABox abduction problem $\langle \mathcal{K}, \Phi \rangle$. Consider a borderline case when the query actually follows from the knowledge base. In such a situation there has to exist a refutation proof for $\mathcal{K} \models \Phi$, i.e. a closed tableau tree or a resolution deduction of an empty clause, initiated by $\neg \Phi$, with $\mathcal{K}$ as the set of premises. Naturally, both calculi operate only on FOL clauses, therefore the input has to be provided under the τ-transformation. The critical point here is to carefully select the initial clause for the tableau and the set-of-support in the resolution proof. In the latter case we will use $\mathbf{R}(\tau(\neg \Phi))$, i.e. the set of all root clauses obtained through transformation of the negated query, as defined in Table 7. In the tableau setting, we will be initiating alternative proofs with consecutive clauses from that set. If the proof $\tau(\mathcal{K}) \cup \tau(\neg \Phi) \vdash \bot$ succeeds we clearly do not deal with a genuine abductive problem, as no additional formula is needed to entail the query. Otherwise, we can identify such formulas by analyzing the structure of possible partial proofs constructed by either of the calculi. Observe, that at every stage of a proof it is possible to construct a formula A that forces its completion by simply closing all open branches of the tableau or enabling derivation of an empty clause via resolution. The simplest way of constructing A is to pick literals that are unifiable with the complements of the leaves of open branches of the tableau, or with the complements of the literals comprising any of the resolvents, and connect them with the conjunction symbol. Such a formula would obviously complete the proofs of the query, and thus it could be seen as a solution to the translated problem $\langle \mathcal{K}, \Phi \rangle$, as obviously $\tau(\mathcal{K}) \cup A \cup \tau(\neg\Phi) \vdash \bot$ and therefore $\mathcal{K} \cup A \models \Phi$. Since both calculi are sound and complete for FOL it is guaranteed that every such solution will be found at some point, provided it satisfies certain syntactic and semantic requirements. Eventually, we will be interested only in the formulas that can be translated back to the DL $\mathcal{ALE}$, so at this stage we call A only a *FOL base of a solution* to the original problem $\langle \mathcal{K}, \Phi \rangle$. The following definition gives the formal account of this notion.

Definition 3 (FOL-base of solution) Let $\langle \mathcal{K}, \Phi \rangle$ be an ABox abduction problem. A set of literals A_{FOL} is a **FOL-base of a solution** to $\langle \mathcal{K}, \Phi \rangle$ *iff* either of the following conditions holds:

1. **(tableau)**: There exists a regular connection tableau T such that:
   (a) the root of T contains all and only the clauses $\tau(\mathcal{K}) \cup \tau(\neg\Phi)$,
   (b) T was initiated by expansion of some clause $Cl_{init} \in \mathbf{R}(\tau(\neg\Phi))$,
   (c) $A_{FOL} = \{\neg L(\bar{t}) \mid L(\bar{t}) \in Cl\}$, where Cl is the set of the leaves of all the open branches of T.
2. **(resolution)**: There exists a sequence of resolution inference steps, with the resulting set of resolvents R, where:
   (a) the initial set of clauses comprised all and only $\tau(\mathcal{K}) \cup \tau(\neg\Phi)$,
   (b) $\mathbf{R}(\tau(\neg\Phi))$ was the set-of-support for that sequence,
   (c) $A_{FOL} = \{\neg L(\bar{t}) \mid L(\bar{t}) \in Cl\}$, where Cl is a resolvent in R.

As will be shown in Section 4 both conditions in fact coincide, hence both reasoning methods can be used interchangeably in the framework for ABox abduction.

3.4 Abductive proof constraints

In the context of FOL abduction one would typically apply *reverse Skolemization* [32] to the FOL-base retrieved from the proof in order to obtain an adequately quantified FOL formula solving the problem. Under this technique all free variables in a FOL-base get bound by existential quantifiers, whereas all Skolem terms are replaced by universally quantified variables. Thus the resulting formula can be immediately unified with the literals in the FOL-base and consequently force completion of the abductive proof.

Modal logics require a more sophisticated approach. The possibility of binding variables is limited to the use of modal operators, which allow to express statements concerning only the objects in the domain accessible from other objects through particular relations. For a sound reconstruction of an abductive solution from a modal proof one has to take into account the entire chain of relations and modalities that led to a particular term occurring in the proof [33]. This characteristic applies also to DLs and is handled in the framework for ABox abduction by means of *abductive graphs*, which encode the relational structure underlying the proof of each FOL-base.

An *abductive graph* is a tuple $\mathcal{G} = (V, E)$, whose vertices are terms (variables, individual names, Skolem terms) and edges are labeled with role names. With every abductive proof and its FOL-base we associate a single graph, which describes the relationships between all the terms occurring in the proof. The modal meaning of the terms is implied by their syntax: variables represent individuals bound by some universal restriction, Skolem terms, by existential restrictions, while individual names stand for individuals that were not originally bound in the used clauses. The graph is initiated at the start of the proof, by including all role assertions occurring in the ABox of the problem, and later it evolves along the construction of the proof. On each inference step it is extended with new edges and vertices present in the modal core (Section 3.2) of the connected clause, under the substitution applied to the proof at that step. We formalize the notion of abductive graph by the following inductive definition.

Definition 4 (Abductive graph) Let $\langle \mathcal{K}, \Phi \rangle$ be an ABox abduction problem and A_{FOL} a FOL-base obtained in an abductive proof for $\langle \mathcal{K}, \Phi \rangle$.

1. If A_{FOL} is derived from a clause Cl, such that Cl is the initial clause expanded on the tableau or one of the clauses in the initial set-of-support, then $\mathcal{G} = (V, E)$ is the **abductive graph** associated with the proof of A_{FOL} *iff* $V = \{a, b \mid r(a, b) \in E\}$ and $E = \{r(a, b) \mid r(a, b) \in \mathcal{A}\} \cup \mu(Cl)$, where $\mathcal{A}$ is the ABox in $\mathcal{K}$.
2. If A_{FOL} is obtained by connecting a clause Cl to an abductive proof involving application of an MGU σ, and $\mathcal{G}' = (V', E')$ is the abductive graph associated with that proof, then $\mathcal{G} = (V, E)$ is the **abductive graph** associated with the proof of A_{FOL} *iff* $V = \{a, b \mid r(a, b) \in E\}$ and $E = \sigma E' \cup \sigma\mu(Cl)$.

The abductive graphs, along with the associated FOL-bases, enable a sound reconstruction of ABox assertions from abductive proofs. Roughly, we will apply a *reverse relational Skolemization*, thus reversing the effects of Skolemization in a similar way as in the FOL case, but instead of quantified FOL formulas we derive $\mathcal{ALE}$ assertions involving nested quantification restrictions over the role chains encoded in the graph. The expressive power of $\mathcal{ALE}$ delimits the scope of graphs that can be submitted to such a procedure. The notion of *$\mathcal{ALE}$-admissible graph* reflects these limitations.

Definition 5 ($\mathcal{ALE}$-admissible graph) Graph $\mathcal{G} = (V, E)$ associated with the FOL-base A_{FOL} is **$\mathcal{ALE}$–admissible** *iff* the following requirements are satisfied:

1. for every Skolem term $t_1 \in V$ there is a unique $t_2 \in V$ and r, such that $r(t_2, t_1) \in E$,
2. for every Skolem term $t \in V$, t can be only succeeded by a tree-shaped subgraph in $\mathcal{G}$, which does not contain individual names from N_I.

The rationale behind the restrictions is rather obvious. DL formulas, like their modal counterparts, have the tree model property. Since after reverse relational Skolemization, Skolem terms can only stand at a position quantified by a universal restriction, all their successors have to be ordered in a way that can be captured by a complex concept assertion. Such an ordering has to give rise to a tree-shaped model. Moreover, every such model requires a single root, hence the single predecessor requirement. Finally there are no expressive means in $\mathcal{ALE}$ for ensuring that particular individuals belong to these models, hence no names from N_I can occur in such subgraphs. On the contrary to Skolem terms, relationships between named individuals and variables, which can be always replaced by new individual names in the process of reverse Skolemization, can take structures of arbitrary shapes, expressible via ABox assertions of the form $r(t_1, t_2)$. Figure 4 presents an example of an $\mathcal{ALE}$–admissible graph that could be associated with some abductive proof.

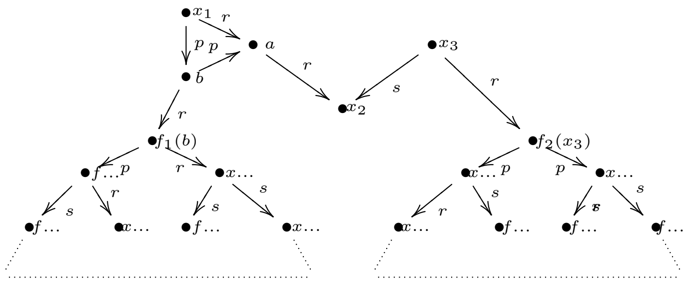

Fig. 4 An $\mathcal{ALE}$–admissible graph associated with an abductive proof: every Skolem vertex has exactly one predecessor and can be only succeeded by a tree-shaped subgraph containing no individual names.

Apart from discarding clauses associated with non-admissible graphs, we will also place restrictions on the FOL-bases, which are similarly inexpressible in $\mathcal{ALE}$.

Definition 6 **($\mathcal{ALE}$-admissible base)** Let A_{FOL} be a FOL-base obtained in an abductive proof for an ABox abduction problem. A_{FOL} is **$\mathcal{ALE}$-admissible** *iff* it contains none of the following literals:

1. $P_i(\overline{x})$ or $\neg P_i(\overline{x})$ for any $P_i \in \mathbb{P}$,
2. $r(t_1, f_i(t_2))$ for any $f_i \in \mathbb{F}$ and any r.

The first condition acknowledges impossibility of including non-DL predicates into DL assertions. The second one discards clauses that contain $r(t_1, f_i(t_2))$, a construct inexpressible in the DL $\mathcal{ALE}$ after reverse Skolemization. In fact the FOL-bases obtained through the procedure will only contain elements of the form:

1. $A(t)$ or $\neg A(t)$, where $A \in N_C$ and t is an individual name, Skolem term, or a variable;
2. $r(t_1, t_2)$, where $r \in N_R$ and t_2 is an individual name or a variable;
3. $\neg r(t_1, t_2)$, where $r \in N_R$ and t_2 is a Skolem term.

3.5 Solution retrieval

We can now describe the procedure of retrieving a set of $\mathcal{ALE}$ ABox assertions from a FOL-base and the associated abductive graph. The idea is to first address all tree-shaped subgraphs of the abductive graph that are rooted at Skolem terms. Proceeding bottom-up, we fold the relevant assertions from the FOL-base into nested concept descriptions. Next, we consider the remaining parts of the graph and the assertions applicable to them, and render them into DL axioms accordingly. To avoid syntactic ambiguity we will be referring only to the solutions in a certain normalized form. To this end we define a satisfiability preserving transformation π, presented in Table 10, which removes redundancy from $\mathcal{ALE}$ concept descriptions. For any concept description C in $\mathcal{ALE}$, we write $\pi(C)$ to refer to a concept equivalent to C, whose all subconcepts are closed under π. Further, if A is a set of ABox assertions in $\mathcal{ALE}$, we will call A *non-redundant* if and only if for all concept assertions $C(a) \in A$ it holds that $C = \pi(C)$, and there is not more than one concept assertion in A per individual name.

$\pi(\forall r.\top)$	$=$	$\top$
$\pi(C \sqcap \top)$	$=$	C
$\pi(C \sqcap \neg C)$	$=$	$\bot$
$\pi(\exists r.\bot)$	$=$	$\bot$
$\pi(C \sqcap \bot)$	$=$	$\bot$
$\pi(C \sqcap C)$	$=$	C
$\pi(\forall r.C \sqcap \forall r.D)$	$=$	$\forall r.(C \sqcap D)$
$\pi(\exists r.C \sqcap \exists r.(C \sqcap D))$	$=$	$\exists r.(C \sqcap D)$
$\pi(\exists r.C \sqcap \forall r.\bot)$	$=$	$\bot$
$\pi(\exists r.\top \sqcap \exists r.C)$	$=$	$\exists r.C$

Table 10 Redundancy elimination from concept descriptions in $\mathcal{ALE}$.

The following definition introduces the notion of a $\vdash_{\text{ABox}}$-*solution* to an ABox abduction problem.

Definition 7 ($\vdash_{\mathbf{ABox}}$-solution) Let $\langle \mathcal{K}, \Phi \rangle$ be an ABox abduction problem, A_{FOL} an $\mathcal{ALE}$-admissible FOL-base obtained in an abductive proof for $\langle \mathcal{K}, \Phi \rangle$, and $\mathcal{G} = (V, E)$ an $\mathcal{ALE}$-admissible abductive graph associated with A_{FOL}. A non-redundant set of assertions A is a $\vdash_{\mathbf{ABox}}$**-solution** to $\langle K, \Phi \rangle$ *iff* it is semantically equivalent to a set of assertions A' (i.e. $A \vDash A'$ and $A' \vDash A$) generated according to the following procedure:

1. $A' := A_{FOL}$
2. For every term $t \in V$ with no successors in $\mathcal{G}$, if it is a Skolem term or has a Skolem predecessor, get $r(t', t)$ from E and begin:
   (a) If t is a variable then add $\exists r. \bigsqcap\{C \mid C(t) \in A'\}(t')$ to A' and remove every $C(t)$. In case there are no $C(t) \in A'$ add $\exists r.\top(t')$. Remove $r(t', t)$ from A'.
   (b) If t is a Skolem term then add $\forall r. \bigsqcap\{C \mid C(t) \in A'\}(t')$ to A' and remove every $C(t)$. In case there are no $C(t) \in A'$ but there is $\neg r(t', t) \in A'$ then add $\forall r.\bot(t')$. Remove $\neg r(t', t)$ from A'.
   (c) Remove t from V and $r(t', t)$ from E.
3. For every (remaining) term $t \in V$ with no successors in $\mathcal{G}$, begin:
   (a) If t is an individual name then for every $C(t) \in A'$ choose one option:
      – leave it unmodified

      OR

      – if there is $r(t', t) \in E$ add $(\forall r.C)(t')$ to A' and remove $C(t)$. If t' is a variable then instantiate it with a new individual name.
   (b) If t is a variable then choose one option:
      – instantiate t with a new individual name and consider it according to the previous rule

      OR

- if t has a unique immediate predecessor in $\mathcal{G}$ and $r(t', t) \in E$ then add $\exists r. \bigsqcap\{C \mid C(t) \in A'\}(t')$ to A' and remove every $C(t)$. In case there are no $C(t) \in A'$ add $(\exists r.\top)(t')$. Remove $r(t', t)$ from A'.

(c) Remove t from V and every $r(t', t)$ from E.

Provided the requirements for admissibility of the FOL-base and its graph are satisfied, the procedure returns a proper set of $\mathcal{ALE}$ ABox assertions A'. Let us shortly comment on the consecutive steps of the retrieval procedure. Initially (1) A' contains only concept and role literals. Note, that the former might be used only in simple concept assertions or as the qualifying concepts in quantification restrictions, whereas the latter, either in role assertions or implicitly in existential restrictions (positive role literals) or in the universal restrictions of the form $\forall r.\bot$ (negative role literals). First (2) we consider the tree-shaped subgraphs rooted at Skolem terms in $\mathcal{G}$. We start with their leaves and move node by node up the trees. Assertions over variables are converted into existential restrictions on their respective predecessors (2a), while assertions over Skolem terms are translated into universal restrictions (2b). Once all Skolem terms are removed from the graph we consider the remaining individual names and variables, similarly, starting from the leaves of the graph and proceeding upwards (3). For every individual name t and every assertion $C(t)$ one can choose between a solution, which uses $C(t)$ directly or another one, in which $C(t)$ is replaced with $(\forall r.C)(t')$, provided t' is an r-predecessor of t in $\mathcal{G}$ (3a). For every variable term t one has an option of treating it as a new "abduced" individual (3b'), or by considering it, like before (2a), as an individual entailed by an existential restriction placed on the predecessor of t, provided there exists a unique one (3b").

3.6 Example

To illustrate how ABox abduction problems can be solved in the framework discussed in this section we will now present a small example of using the approach in handling the *happy John* problem $\langle \mathcal{K}, \{\textsc{Happy}(\mathsf{John})\}\rangle$. Recall the content of the problem's knowledge base given in the introductory section and its transformation outlined in Section 3.2. For parsimony, we will compute parts of solutions using different calculi, though obviously both of them generate the same answers.

Figure 5 presents a regular connection tableau tree for the translated problem. Every subtree of the tableau with the root containing the clauses of the knowledge base and the negated query is a partial refutation proof for the query. The leaves of these subtrees surrounded by boxes form FOL-bases for proper ABox solutions, listed further in Table 11.

The resolution proofs for the query are included in Figure 6. Again, the resolvents printed in boxes give rise to FOL-bases of $\vdash_{\text{ABox}}$-solutions to the problem.

Finally, Table 11 gives a detailed account of all the $\vdash_{\text{ABox}}$-solutions to the problem found within the presented scope of computation. Every solution is derived from its respective FOL-base and the associated abductive graph. Observe the evolution of the graphs in the course of construction of the proofs. For instance, in the steps from 2. to 3. (tableau) and from 1. to 4. (resolution) the graphs are extended with two new vertices and an edge, after connecting a clause with non-empty modal core. In the step from 5. to 6. the substitution applied to the resolvent is used also over the associated graph.

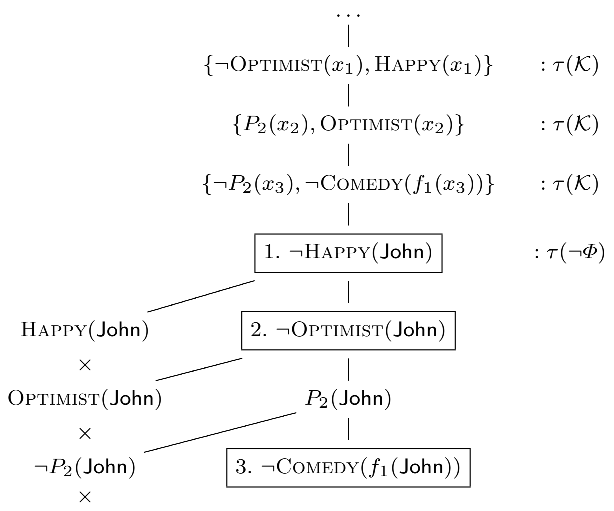

Fig. 5 *Happy John* problem: tableaux proofs.

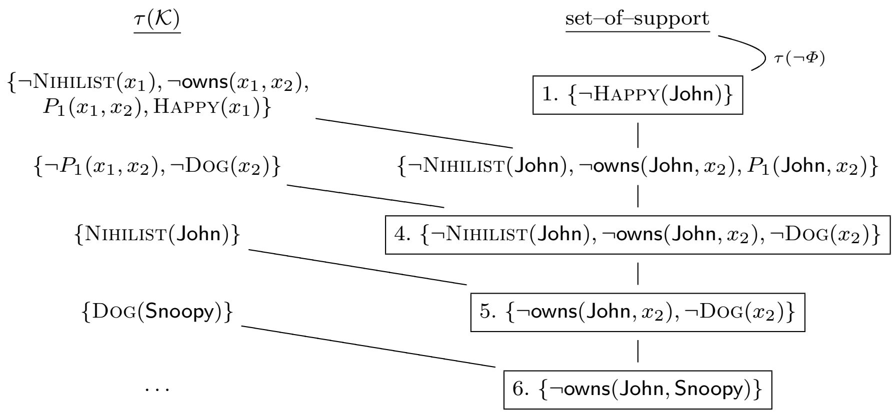

Fig. 6 *Happy John* problem: resolution proofs.

Clearly, all the generated sets of ABox assertions are plain solutions to the problem $\langle \mathcal{K}, \{\text{Happy}(\mathsf{John})\}\rangle$, although some of them might not be minimal, e.g. 4., relevant, e.g. 1., or consistent. In order to verify satisfaction of those criteria an additional post-processing is required. In Sections 4.2 and 4.3 we will devise a simple verification strategy, based on the use of a standard DL reasoner, as well as discuss some computational limitations to the problem of verifying minimality. Before that, in Section 4.1, we will show that the procedure of finding $\vdash_{\text{ABox}}$-solutions is correct.

	FOL-base	Abductive graph	Solution
1.	$\{\textsc{Happy}(\mathsf{John})\}$		$\{\textsc{Happy}(\mathsf{John})\}$
2.	$\{\textsc{Optimist}(\mathsf{John})\}$		$\{\textsc{Optimist}(\mathsf{John})\}$
3.	$\{\textsc{Comedy}(f_1(\mathsf{John}))\}$	$\underset{\mathsf{John}}{\bullet} \xrightarrow{\mathsf{watches}} \underset{f_1(\mathsf{John})}{\bullet}$	$\{\forall\mathsf{watches}.\textsc{Comedy}(\mathsf{John})\}$
4.	$\{\textsc{Nihilist}(\mathsf{John}),$ $\mathsf{owns}(\mathsf{John}, x_2), \textsc{Dog}(x_2)\}$	$\underset{\mathsf{John}}{\bullet} \xrightarrow{\mathsf{owns}} \underset{x_2}{\bullet}$	$\{\textsc{Nihilist}(\mathsf{John}),$ $\exists\mathsf{owns}.\textsc{Dog}(\mathsf{John})\}$
5.	$\{\mathsf{owns}(\mathsf{John}, x_2), \textsc{Dog}(x_2)\}$	$\underset{\mathsf{John}}{\bullet} \xrightarrow{\mathsf{owns}} \underset{x_2}{\bullet}$	$\{\exists\mathsf{owns}.\textsc{Dog}(\mathsf{John})\}$
6.	$\{\mathsf{owns}(\mathsf{John}, \mathsf{Snoopy})\}$	$\underset{\mathsf{John}}{\bullet} \xrightarrow{\mathsf{owns}} \underset{\mathsf{Snoopy}}{\bullet}$	$\{\mathsf{owns}(\mathsf{John}, \mathsf{Snoopy})\}$

Table 11 *Happy John* problem: solutions.

4 Correctness and selection criteria

In this section we elaborate on basic formal properties of the introduced procedure. First, we prove its soundness and completeness with respect to the semantics of plain solutions to ABox abduction problems, as specified in Definition 2. Following this, we address the task of applying additional selection criteria to the generated sets of plain solutions, and further, the problem of correctness and termination of reasoning under the criteria. We consider two cases: a general one (Section 4.2), involving no syntactic constraints on the knowledge bases, and the case of acyclic terminologies (Section 4.3), for which stronger results can be obtained.

4.1 Soundness and completeness

Below, we formally argue for adequacy of the procedure for solving ABox abduction problems. In Theorem 3 we claim that every solution that is found by the procedure is indeed a *plain solution* to the input problem. Conversely, Theorem 4 ensures that if A is a *consistent* and *minimal solution* to a given problem, then A will be found via $\vdash_{\text{ABox}}$ in a finite number of steps.

We start by recapping the results of soundness and completeness of the two calculi discussed in the paper.

Theorem 1 (Regular connection tableaux: completeness [21, Thm. 4.14], [22, Thm. 3]) *If a finite ground clause set S is unsatisfiable then there is a regular connection tableau proof for S, in which a relevant clause[4] $Cl \in S$ is the first to which a beta rule is applied.*

Proposition 1 (Regular connection tableaux: soundness) *If there exists a regular connection tableau proof for a set of ground clauses S then S is unsatisfiable.*

[4] A clause is *relevant* in S *iff* it belongs to a minimally unsatisfiable subset of S. See Definition 8 below.

Proof The proposition follows immediately from soundness of the standard semantic tableaux calculus [21, Thm 3.12] and by observing that every regular connection tableau proof is in fact a standard tableau proof. □

Theorem 2 (Resolution with set-of-support: completeness [31, Thm. 3.2.2.]) *If S is an unsatisfiable set of ground clauses and $T \subseteq S$ such that $S \setminus T$ is satisfiable then there exists a resolution refutation of S with set-of-support T.*

Proposition 2 (Resolution with set-of-support: soundness) *If there exists a resolution refutation of a set of ground clauses S with set-of-support $T \subseteq S$ then S is unsatisfiable.*

Proof The proposition follows immediately from soundness of the standard resolution method [31] and by observing that every resolution with set-of-support proof is in fact a resolution proof. □

To simplify the layout of the following arguments we will entirely adopt the FOL perspective on the procedure and the involved DL formulas. In order to do so, we must acknowledge that due to the specific character of the employed calculi (Section 3.3), the τ-transformation (Section 3.2), and the admissibility conditions (Section 3.4), reasoning in the framework can be seen as standard translation-based theorem proving for $\mathcal{ALC}$ and $\mathcal{ALE}$, where the input is reduced to Skolem Normal Form, i.e. Conjunctive Normal Form of Skolemized formulas. Notice, that for any DL axiom φ, the set of clauses $\tau(\varphi)$ closed under connection steps through the non-DL predicates, i.e. the set of all clauses that can be possibly constructed by pasting back each non-root clause to its original position marked by a non-DL predicate, is in fact equivalent to the set $SNF(st(\varphi))$, where $st(\varphi)$ denotes the standard translation of φ. In the remainder of this section, unless specified otherwise, we will hence assume that all considered DL formulas are simply sets of Skolemized FOL clauses constructed exactly in such a way. In particular, for any ABox abduction problem $\langle \mathcal{K}, \Phi \rangle$ and a solution A, we will assume the following abbreviations hold:

$$\begin{array}{rcl} \mathcal{K} & := & SNF(st(\mathcal{K})) \\ \neg\Phi & := & SNF(\neg \bigwedge st(\Phi)) \\ A & := & SNF(st(A)) \end{array}$$

Given the above conventions observe the following.

Proposition 3 *Let A be a set of ABox assertions in the DL $\mathcal{ALE}$ with the signature (N_I, N_C, N_R). Then:*

1. *Every clause $Cl \in A$ must be of the form $Cl = \bigcup_{0 \leq i \leq n}\{\neg r_i(t_i, t_{i'})\} \cup \{L(\bar{t})\}$, where $r_i \in N_R$, for all $1 \leq i \leq n$, and $L \in \{A, \neg A, r, \bot, \top\}$ for some $A \in N_C$ or $r \in N_R$.*
2. *For every clause $Cl^{\exists r} = \bigcup_{0 \leq i \leq n}\{\neg r_i(t_i, t_{i'})\} \cup \{r(t_{n'}, t_{n''})\} \in A$, where $t_{n''}$ is a Skolem term, there exists at least one clause $Cl^{\exists.C} \in A$, such that $Cl^{\exists r} \setminus \{r(t_{n'}, t_{n''})\} \subseteq Cl^{\exists.C}$ and $L(t_{n''}, \bar{t}) \in Cl^{\exists.C}$.*

Proof 1) The claim follows immediately from the analysis of the syntax of ABox assertions in $\mathcal{ALE}$. Notice that under the standard translation of such axioms the disjunction connective occurs only between negated role atom and the (translated) qualifying concept of every universal restriction. After distributing conjunction over disjunction and splitting the conjuncts we obtain a set of clauses of the form presented. 2) Whenever

$t_{n''}$ is a Skolem term then the literal $r(t_{n'}, t_{n''})$ in $Cl^{\exists r}$ must originate from an existential restriction. But then there has to also exist at least one more clause $Cl^{\exists.C}$ containing the term $t_{n''}$, which had to occur in the qualifying concept used in the same restriction. By the style of the standard translation and the SNF transformation, one can see that this clause has to contain $Cl^{\exists r} \setminus \{r(t_{n'}, t_{n''})\}$ as its proper subset. $\square$

The proof of soundness is relatively simple. We demonstrate that if a $\vdash_{\text{ABox}}$-solution A to $\langle \mathcal{K}, \Phi \rangle$ is added to the abductive proof from which it had been retrieved, the proof has a continuation that succeeds, thus guaranteeing that $\mathcal{K} \cup A \models \Phi$.

Theorem 3 ($\vdash_{\text{ABox}}$: soundness) *If A is a $\vdash_{\text{ABox}}$-solution to the ABox abduction problem $\langle \mathcal{K}, \Phi \rangle$ then A is a plain solution to $\langle \mathcal{K}, \Phi \rangle$.*

Proof Let $\langle \mathcal{K}, \Phi \rangle$ be an Abox abduction problem and A a $\vdash_{\text{ABox}}$-solution to it. To demonstrate that A is a plain solution to $\langle \mathcal{K}, \Phi \rangle$ we have to show that $\mathcal{K} \cup A \models \Phi$ (Definition 2). We rest on the soundness of regular connection tableaux and resolution with set-of-support (Proposition 1 and 2) and show there is a refutation proof for $\mathcal{K} \cup A \cup \neg\Phi$ constructed with either of the calculi. Recall the form of clauses in A from Proposition 3. Also, note that FOL-base A_{FOL} of A must have contained all concept and positive role literals occurring in those clauses (except for $\top$), plus the negative role literals preceding $\perp$, modulo reverse Skolemization of the terms involved (Definition 3 and 7). Focus on the abductive proof from which A was retrieved, add A to the set of premises and continue the proof. Consider one of the concept literals $C(s) \in A_{FOL}$ (analogical argument will hold also for positive and negative role literals). Clearly there must be a clause in A, which contains the corresponding literal. Let $Cl = \{\neg r_1(t_1, t_{1'}), \ldots, \neg r_n(t_n, t_{n'}), C(t)\}$ be that clause (where we allow the set of negative role literals to be empty). Connect Cl to the corresponding leaf on the tableau or to the respective resolvent via $C(t)$. Unification is naturally possible due to the style of reverse Skolemization involved in the reconstruction of A from A_{FOL}. In particular, one of the following must be the case:

1. s is a variable: Then t must be a Skolem term or an individual name.
2. s is an individual name: Then t must be a variable or an individual name such that $s = t$.
3. s is a Skolem term: Then t must be a variable.

After including Cl in the proof, the connecting literal on the tableau or in the resolvent is replaced by the sequence of literals $\{\neg r_1(t_1, t_{1'}), \neg r_2(t_2, t_{2'}), \ldots, \neg r_n(t_n, t_{n'})\}$. Notice, that every occurrence of such a literal was originally motivated by the presence of a corresponding edge in the abductive graph associated with the FOL-base of A (Definition 4). Note also that for every $1 \leq i \leq n$, term $t_{i'}$ is a variable. Consider the last literal $\neg r_n(t_n, t_{n'})$. Clearly there must be an edge $r_n(s_n, s_{n'})$ in the graph such that $s_{n'}$ is an individual name or a Skolem term, or otherwise $t_{n'}$ would not have been reversely Skolemized into a variable. Consider the following cases:

1. s_n and $s_{n'}$ are individual names: Then one of the following has to hold:
   - There is a role assertion $r_n(s_n, s_{n'})$ in the ABox, included in the graph by default. Then $r_n(s_n, s_{n'})$ can be used as a connection to $\neg r_n(s_n, s_{n'})$.
   - There is a role assertion $r_n(s_n, s_{n'})$ in A_{FOL}, and consequently in A. Then $r_n(s_n, s_{n'})$ can be used as a connection to $\neg r_n(s_n, s_{n'})$.

2. $s_{n'}$ is a Skolem term: Then there must be a clause Cl' used in the same proof, whose modal core contains $r_n(s_n, s_{n'})$. Consider this clause. Notice that by the definition of the τ-transformation (Table 6) either Cl' contains $r(s_n, s_{n'})$ (since $s_{n'}$ is a Skolem term it has to originate from some existential restriction), or there is another clause that contains $r_n(s_n, s_{n'})$, whose remaining literals belong to Cl'. In either case it is possible to use such clause in the proof, connecting it to $\neg r_n(t_n, t_{n'})$. Again unification is possible due to soundness of reverse Skolemization. From that point on construction of the proof should mimic the inference steps that were used in the context of Cl' in the same proof. Eventually, all literals from the connected clause must obtain the same connections as the ones from Cl' and so this fragment of the proof, started with $\neg r_n(t_n, t_{n'})$, succeeds.

Repeat the argument for all remaining negative role literals and for all clauses in A. Clearly the refutation proof has to succeed, resulting in a closed tableau tree or an empty resolvent, which shows that indeed $\mathcal{K} \cup A \vDash \Phi$. □

The completeness result, presented in Theorem 4, holds under an additional restriction, which reveals a certain limitation of the procedure. Namely, any consistent and minimal solution A to a problem $\langle \mathcal{K}, \Phi \rangle$ is guaranteed to be found, only if every subconcept C occurring in the assertions from A, such that $C \neq \bot$, is satisfiable with respect to $\mathcal{K}$, i.e. for which there exists a model of the knowledge base $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that $C^{\mathcal{I}} \neq \emptyset$. Consider for instance the problem $\langle \mathcal{K}, \Phi \rangle$, where $\mathcal{K} = \{C \sqsubseteq B \sqcap \neg B, \forall r.(D \sqcap \neg D) \sqsubseteq A\}$ and $\Phi = \{A(a)\}$. The procedure will return solution $\forall r.\bot(a)$, but will fail to output $\forall r.C(a)$, even though it clearly solves the problem, as $\mathcal{K} \models C \sqsubseteq \bot$. Nevertheless, the connectedness requirement makes it impossible to use the clauses from $C \sqsubseteq B \sqcap \neg B$ for solving the problem, as they could not be connected to the proof starting from $\neg A(a)$. As the use of unsatisfiable concepts other than $\bot$ is practically always unintended in applications of Description Logics, we believe that this limitation does not diminish the pragmatic value of the procedure.

The proof of the result is more involved than that of soundness and requires additional formal machinery, which we now introduce. First, we assume that solutions to ABox abduction problems, which are under consideration in this section, are always non-redundant in the sense defined in Section 3.5. For any non-redundant set of assertions A under SNF transformation, we define an operation $A^{\setminus\top} = \{Cl \in A \mid \top(t), \neg\bot(t) \notin Cl\}$, i.e. an operation of removing from A all clauses containing the literal $\top(t)$ or $\neg\bot(t)$ for any term t. Now we note a simple observation.

Proposition 4 *Let A be a consistent solution to an ABox abduction problem $\langle \mathcal{K}, \Phi \rangle$. Any solution B to $\langle \mathcal{K}, \Phi \rangle$ such that $B^{\setminus\top} \subset A^{\setminus\top}$ is minimal with respect to A.*

Proof Observe that once all clauses containing symbol $\top$ are removed from A, then any proper subset $A' \subset A^{\setminus\top}$ is deductively weaker than $A^{\setminus\top}$, i.e. $A' \not\models A^{\setminus\top}$. Any solution B, such that $B^{\setminus\top} \subset A^{\setminus\top}$, must be therefore deductively weaker from A and therefore, by Definition 2, minimal with respect to A. □

Let us recall the notion of *minimal unsatisfiability* and present three lemmas, building upon it, which will play a pivotal role in the following part.

Definition 8 (Minimal unsatisfiability) A set of clauses is **minimally unsatisfiable (MU)** if it is unsatisfiable and each of its proper subsets is satisfiable. A clause is **relevant** in a set of clauses S if it belongs to a MU subset of S.

Lemma 1 *Let S be a MU set of ground clauses, $A \subseteq S$ a set of unit clauses, and Cl any clause in $S \setminus A$. The following claims hold:*

1. *There exists a regular connection tableau tree initiated with beta expansion of Cl, whose root is $S \setminus A$, such that $\{\neg L \mid \{L\} \in A\}$ is the set of the leaves of its open branches.*
2. *There exists a sequence of resolution inference steps from $S \setminus A$ with set-of-support $\{Cl\}$ resulting in the clause $\{\neg L \mid \{L\} \in A\}$.*

Proof 1. Since Cl is relevant in S, it follows from Theorem 1 that there exists a closed tableau initiated with Cl, whose root contains S. Notice that if you remove any unit clause from A the proof cannot succeed, or else S would not be MU. Hence every clause from A contains a literal complementary to the leaf on (at least) one of the closed branches. Now if A is removed from the root we get a tableau, for which $\{\neg L \mid \{L\} \in A\}$ is the set of the leaves of its open branches.

2. Since $S \setminus \{Cl\}$ is satisfiable, it follows from Theorem 2 that there has to be a resolution refutation of S with set-of-support $\{Cl\}$. Assume now that A is not present in S. We prove the claim of the lemma by induction on the cardinality of A.

Consider $|A| = 0$. Then $\{\neg L \mid \{L\} \in A\}$ is an empty clause. Clearly, derivation of such a clause in the specified setting is guaranteed by Theorem 2.

Assume that for some k and any A, such that $|A| = k$, the clause $Cl_{\bigcup \neg A} = \{\neg L \mid \{L\} \in A\}$ is derivable. Consider $|A| = k+1$ and let $A = B \cup \{\{L'\}\}$, such that $|B| = k$. Assume $\{L'\}$ is added to the set of premises. By inductive assumption it follows that the clause $Cl_{\bigcup \neg B} = \{\neg L \mid \{L\} \in B\}$ is now derivable. We argue that there exists a derivation of $Cl_{\bigcup \neg B}$ in which the last inference step is resolution of $\{L'\}$ against $\{\neg L'\} \cup Cl_{\bigcup \neg B}$. First, note that $\{L'\}$ is necessary for deriving $Cl_{\bigcup \neg B}$ or otherwise S would not have been MU at the first place. Therefore there has to be a clause $Cl' \cup \{\neg L'\}$ against which $\{L'\}$ has to be resolved, such that either $Cl' = Cl_{\bigcup \neg B}$, which would prove the point, or Cl' is further resolved against other clauses. But in the latter case it is possible to defer resolution of $\{L'\}$ until Cl' is first resolved. Switch the sequence of resolution steps in such a derivation and repeat the argument. Since derivation of $Cl_{\bigcup \neg B}$ has to be finite, it follows that at some point one has to arrive at a sequence in which $\{\neg L'\} \cup Cl_{\bigcup \neg B}$ appears as the last clause to be resolved. But then there also exists a derivation of $Cl_{\bigcup \neg A}$, which is equivalent to $\{\neg L'\} \cup Cl_{\bigcup \neg B}$. □

Lemma 2 ([31, Lemma 2.3.2]) *Let S be a MU set of clauses, let Cl' be a subset of clause $Cl \in S$, and let $S' = (S \setminus \{Cl\}) \cup \{Cl'\}$; i.e. replace Cl in S with Cl'. Then every MU subset of S' contains Cl'.*

Lemma 3 (MU set under a solution) *Let A be a consistent and minimal solution to an ABox abduction problem $\langle \mathcal{K}, \Phi \rangle$, such that except for $\perp$ all subconcepts occurring in the assertions from A are satisfiable with respect to $\mathcal{K}$. There exists a finite MU set $S = K \cup A^{\setminus \top} \cup Q$, where $K \subseteq \mathcal{K}$ and $Q \neq \emptyset \subseteq \neg \Phi$, a ground substitution σ and a MU set of ground clauses σS, call it a* ***MU set under*** *A, which contains at least one instance of every clause from S.*

Proof By Definition 2 it holds that $\mathcal{K} \cup A \vDash \Phi$, hence the set $\Theta = \mathcal{K} \cup A \cup \neg \Phi$ is unsatisfiable. By Herbrand's theorem Θ is unsatisfiable *iff* there exists a finite set of ground instances of Θ which is unsatisfiable. Thus there must exist a ground substitution σ whose application to the instances of the clauses of Θ results in such a set. Naturally this set must have a MU subset σS. First we show that $\sigma A^{\setminus\top} \subseteq \sigma S$. Recall from Proposition 3 two types of clauses that can occur in $\sigma A^{\setminus\top}$: $Cl_1 = \bigcup_{0\leq i\leq n}\{\neg r_i(t_i, t_{i'})\}\cup\{C(t)\}$ and $Cl_2 = \bigcup_{0\leq i\leq n}\{\neg r_i(t_i, t_{i'})\}\cup\{r(t_{n'}, t_{n''})\}$. Suppose for some clause $Cl_1 \in \sigma A^{\setminus\top}$ it holds that $Cl_1 \notin S$. This would mean that one can construct a solution B which differs from A only in that $C(t)$ is replaced with $\top(t)$ in Cl_1. But then, by Proposition 4, it would follow that B is minimal with respect to A, which contradicts the assumed minimality of A. Suppose the same holds for some clause of type Cl_2. One of the following has to be the case: a) if $t_{n''}$ is not a Skolem term or there is no clause $Cl_2^{\exists.C} \in \sigma S$, where $Cl_2^{\exists.C}$ is defined as in Proposition 3, then a solution minimal with respect to A can be constructed, which contradicts the assumption; b) (else) if $t_{n''}$ is a Skolem term and there is a clause $Cl_2^{\exists.C} \in S$, then A contains unsatisfiable subconcepts different from $\perp$, which is also not true by assumption. Observe that the literal $r(t_{n'}, t_{n''})$ does not occur in σS, and hence, neither does $\neg r(t_{n'}, t_{n''})$. Moreover, there cannot be any other term t' nor role $r' \in N_R$ such that $r'(t', t_{n''})$ is in σS, as every Skolem term is introduced by a unique existential restriction. Since $t_{n''}$ has no predecessors in σS, it follows that neither $t_{n''}$ nor any of its successors in σS can occur in σQ. Replace $Cl_2^{\exists.C}$ in σS with $Cl_2^{C} = Cl_2^{\exists.C} \setminus Cl_2$ (clauses corresponding to the qualifying concept C), and focus on the resulting MU set (Lemma 2). Repeat for every $Cl_2^{\exists.C} \in \sigma S$. Eventually no clauses from σQ can be present in the resulting MU set. Since all clauses Cl_2^{C} that remain in the set are rooted at $t_{n''}$, it is possible to render them back into $\mathcal{ALE}$, as a concept description C, which is clearly unsatisfiable with respect to $\mathcal{K}$. Hence $\sigma S = \sigma K \cup \sigma A^{\setminus\top} \cup \sigma Q$, where $\sigma K \subseteq \sigma\mathcal{K}$ and $\sigma Q \subseteq \sigma\neg\Phi$. Moreover, $\sigma\mathcal{K} \cup \sigma A$ is satisfiable (by consistency of A), hence $\sigma Q \neq \emptyset$. Therefore $S = K \cup A^{\setminus\top} \cup Q$. □

Finally, we demonstrate the proof of the main result. We first argue that for any solution one can find the corresponding FOL-base and its abductive graph and then we show that the solution retrieval method presented in Definition 7 effectively reconstructs the solution given such input.

Theorem 4 **($\vdash_{\text{ABox}}$: completeness)** *If A is a consistent and minimal solution to the ABox abduction problem $\langle\mathcal{K}, \Phi\rangle$, such that except for $\perp$ all subconcepts occurring in the assertions from A are satisfiable with respect to $\mathcal{K}$, then A is a $\vdash_{\text{ABox}}$-solution to $\langle\mathcal{K}, \Phi\rangle$.*

Proof Let A be a consistent and minimal solution to the ABox abduction problem $\langle\mathcal{K}, \Phi\rangle$. Consider a ground substitution σ and a MU set σS under A (Lemma 3), where $\sigma A^{\setminus\top} \subseteq \sigma S$ is a set of clauses containing at least one ground instance of every clause from $A^{\setminus\top}$ and $\sigma Q \subseteq \sigma S$ is a set of ground instances of $\neg\Phi$. Obviously, there has to exist a refutation proof for σS, constructed with either of the calculi, initiated by some clause from σQ. Given that, we show that there exists an abduction proof for $\langle\mathcal{K}, \Phi\rangle$, which is associated with the FOL-base A_{FOL} and the graph $\mathcal{G}$, such that $\sigma\mathcal{G}$ is the graph associated with the refutation proof of σS, and $[\sigma A^{\setminus\top}] = \sigma A_{FOL}$, where $[\sigma A^{\setminus\top}]$ is the set of unit clauses obtained by pruning clauses from $\sigma A^{\setminus\top}$.

1. Finding A_{FOL}: Take a non-unit clause $Cl = \bigcup_{0\leq i\leq n}\{\neg r_i(t_i, t_{i'})\}\cup\{L(\bar{t})\} \in \sigma A^{\setminus\top}$ and consider two cases: 1) $L(\bar{t}) = \perp(t_{n'})$: Since we assume A is non-redundant there has to exist $\neg r(t_n, t_{n'}) \in Cl$ (Table 10). Replace Cl in σS with the unit clause $Cl' = \{\neg r_i(t_n, t_{n'})\}$; 2) $L(\bar{t}) \neq \perp(t_{n'})$: Replace Cl in σS with the unit clause $Cl' = \{L(\bar{t})\}$. From Lemma 2 it follows that Cl' has to belong to every MU subset of the resulting set of clauses. Repeat the procedure for every non-unit clause in $\sigma A^{\setminus\top}$. We now show that:

(*) after the operation there exists a MU subset $[\sigma S]$, which contains all unit clauses $[\sigma A^{\setminus\top}]$ (both originally unit and those obtained by pruning), and at least one clause $Cl_{init} \in \sigma Q$.

The argument rests on induction over pruning steps. First, note that only the literals of the form $\bot(t)$ and $\neg r(t_1, t_2)$ are being left out. The former case is straightforward: if $Cl \cup \{\bot(t)\}$ is replaced with Cl in a MU set, then the set still remains MU. For the latter case consider a clause Cl, where $\neg r(t_1, t_2) \in Cl$ is the leftmost occurrence of a negative role literal that is to be pruned:

(a) if t_2 is an individual name then the only clause that can possibly fall out from the original MU set after the pruning is a role assertion $r(t_1, t_2)$ from the knowledge base. Note, that $r(t_1, t_2)$ cannot be an assertion in A, as then it would be possible to construct a solution minimal with respect to A, which contradicts the assumption. Also there are no assertions of the form $r(t_1, t_2)$ in σQ. Repeat the step for all clauses in $\sigma A^{\setminus\top}$ and all leftmost occurrences of such literals. Observe that at some point all the remaining negative role literals will not contain any more individual names.

(b) if t_2 is a Skolem term then the only clause that can possibly fall out from the original MU set after the pruning is some clause $Cl_1^{\exists r}$ such that $r(t_1, t_2) \in Cl_1^{\exists r}$. Consider the possible origin of that clause:
   - $Cl_1^{\exists r} \in \sigma K$: then the claim holds.
   - $Cl_1^{\exists r} \in \sigma Q$: then one of the two must be true:
     - there has to be a clause $Cl_1^{\exists .C} \in \sigma Q$, which remains in the MU set.
     - there is no $Cl_1^{\exists .C} \in \sigma Q$, because $Cl_1^{\exists .C} = (Cl_1^{\exists r} \setminus \{r(t_1, t_2)\}) \cup \{\top(t_2)\}$. But in such case the clause $Cl_1^{\exists r}$ will not be removed in the first place, or else it would mean that A contains unsatisfiable subconcepts different from $\bot$. Since t_2 would not have a predecessor in the resulting MU set and it would not occur in $\sigma\neg\Phi$, it follows that the clause $Cl \setminus \{\neg r(t_1, t_2)\}$ would have to correspond to a concept unsatisfiable with respect to $\mathcal{K}$. If that concept was $\bot$, then in fact Cl would not be pruned from $\{\neg r(t_1, t_2)\}$, else the concept would have to be different from $\bot$ which contradicts the assumption.
   - $Cl_1^{\exists r} \in \sigma A^{\setminus\top}$: this is not possible, or else a solution minimal with respect to A could be constructed. Let Cl' and ${Cl'}_1^{\exists r}$ denote the clauses Cl and $Cl_1^{\exists r}$, respectively, before the pruning. Replace Cl' with $({Cl'}_1^{\exists r} \setminus \{r(t_1, t_2)\}) \cup (Cl \setminus \{\neg r(t_1, t_2)\})$ in A and observe that the resulting set of assertions also solves the problem, but it is deductively weaker from the original one.

By the inductive hypothesis we arrive at the initial claim (*). By Lemma 1 it follows that given the set $[\sigma S]$ it is possible to find $\{\overline{L} \mid L \in [\sigma A^{\setminus\top}]\}$ using any of the two calculi, provided the proof is initiated by Cl_{init} (this condition is clearly satisfied in our setting, as we expect that gradually every clause from $\neg\Phi$ will be used as an initial clause for abductive proofs). Note, that in the actual abductive proof not all terms in the literals from $[\sigma A^{\setminus\top}]$ will be ground. Hence one obtains a FOL-base A_{FOL} unifiable with $[\sigma A^{\setminus\top}]$.

2. Finding $\mathcal{G}$: Among the proofs for A_{FOL} there has to exist at least one which is associated with a graph $\mathcal{G}$, which is unifiable with $\sigma\mathcal{G}$ by a substitution that subsumes the one unifying A_{FOL} with $[\sigma A^{\setminus\top}]$. Consider a proof for A_{FOL}, such that it would succeed provided that A was added to the set of formulas available in the proof. Given the assumed properties of A, it follows that every literal occurring

in each clause from $A^{\setminus\top}$ has to obtain a connection in the proof, where the connecting literal has to originate from a clause not in $A^{\setminus\top}$. Thus, in particular, for every positive role literal occurring in A there must exist a complementary one, unifiable with it, already in the proof; for every negative one, there has to exist a complementary one, unifiable with it, which is either already in the proof, or it occurs in some clause $Cl^{\exists r}$, such that $Cl^{\exists .C}$ is already in the proof. In either case the modal cores of the clauses included already in the proof form the graph $\mathcal{G}$, which after completion of the proof becomes grounded with the same substitution that unifies A_{FOL} with $[\sigma A^{\setminus\top}]$.

Finally, we can argue that given this input, the solution retrieval method outputs A', such that it is semantically equivalent to the non-redundant $\vdash_{\text{ABox}}$-solution A. Take a unit clause $\{L(\bar{t})\} \in A$ and consider two cases:

1. $L(\bar{t}) = C(t)$: Note that t is an individual name. There must exist a corresponding literal $C(t') \in A_{FOL}$, such that $t' = t$ or t' is a variable. We retrieve $C(t)$.
2. $L(\bar{t}) = r(t_1, t_2)$: Note that t_1 and t_2 are individual names. There must exist a corresponding literal $r(t'_1, t'_2) \in A_{FOL}$, such that $t'_1 = t_1$ or t'_1 is a variable and $t'_2 = t_2$ or t'_2 is a variable. We retrieve $r(t_1, t_2)$.

Take a non-unit clause $Cl = \bigcup_{0 \leq i \leq n}\{\neg r_i(t_i, t_{i'})\} \cup \{L(\bar{t})\} \in A$ and consider the following cases:

1. $L(\bar{t}) = \bot(t_{n'})$: Note that $t_{n'}$ must be a variable. There must exist a corresponding literal $\neg r_n(t_1, t_2) \in A_{FOL}$ and $r_n(t_1, t_2) \in \mathcal{G}$, such that t_2 is a Skolem term, and there is no $C(t_2) \in A_{FOL}$. We retrieve $\forall r.\bot(t_1)$, which translates to $\{\neg r_n(t_1, t_{n'}), \bot(t_{n'})\}$, and add it to A_{FOL}.
2. $L(\bar{t}) = C(t_{n'})$ and $t_{n'}$ is a variable: There must exist a corresponding literal $C(t_2) \in A_{FOL}$ and $r_n(t_1, t_2) \in \mathcal{G}$, such that t_2 is a Skolem term or an individual name. We retrieve $\forall r.\bigsqcap\{C \mid C(t_2) \in A_{FOL}\}(t_1)$, which for each C translates to a clause $\{\neg r_n(t_1, t_{n'}), C(t_{n'})\}$, and add it to A_{FOL}.
3. $L(\bar{t}) = C(t)$, $C \neq \top$ and t is a Skolem term: There must exist a corresponding literal $C(t_2) \in A_{FOL}$, $r_n(t_1, t_2) \in A_{FOL}$ and $r_n(t_1, t_2) \in \mathcal{G}$, such that t_2 is a variable. We retrieve $\exists r.\bigsqcap\{C \mid C(t_2) \in A_{FOL}\}(t_1)$, which for each C translates to clauses $\{r_n(t_1, t)\}, \{C(t)\}$, and add it to A_{FOL}.
4. $L(\bar{t}) = \top(t)$: Note that t is a Skolem term: There must exist $r_n(t_1, t_2) \in A_{FOL}$ and $r_n(t_1, t_2) \in \mathcal{G}$, such that t_2 is a variable and no $C(t_2) \in A_{FOL}$. We retrieve $\exists r.\top(t_1)$, which translates to clauses $\{r_n(t_1, t)\}, \{\top(t)\}$, and add it to A_{FOL}.
5. $L(\bar{t}) = r(t_1, t_2)$: Note that t_2 is a Skolem term, hence there must be another clause $Cl \cup \{C(t_2)\}$ in A and one of the two cases above must hold.

On each retrieval step consecutive literals from the clauses obtain appropriate interpretation. By applying the inductive hypothesis we conclude that A', equivalent to the solution A, will be appropriately reconstructed by the retrieval procedure. □

4.2 Selection Criteria. Correctness in the general case

Every computed $\vdash_{\text{ABox}}$-solution can be verified against the additional selection criteria imposed on the solution space of an abduction problem, among others relevance, consistency and minimality, all presented in Definition 2. For this purpose it is possible to use the services of standard reasoning tools for DLs. In the following we propose one such approach based on checking consistency of the ABox with respect to the TBox. For a proper representation of the queries it is necessary to shift to a more expressive DL $\mathcal{ALCO}$, which additionally to $\mathcal{ALC}$ allows *nominals*, i.e. concept constructors of the form $\{a\}$, where $a \in N_I$ is a specified individual name. The expressiveness of $\mathcal{ALCO}$ enables rendering $\mathcal{ALE}$ ABox assertions into equisatisfiable TBox axioms, where every concept assertion $C(a)$ is translated into $\{a\} \sqsubseteq C$, while every role assertion $r(a,b)$ into $\{a\} \sqsubseteq \exists r.\{b\}$. By employing this technique one can easily map the complement of a set of ABox axioms Φ into the corresponding TBox expression:

$$\neg\Phi := \top \sqsubseteq \bigsqcup_{C(a)\in\Phi} (\neg\{a\} \sqcup \neg C) \sqcup \bigsqcup_{r(a,b)\in\Phi} (\neg\{a\} \sqcup \forall r.\neg\{b\}) \tag{1}$$

Table 12 outlines the decision procedures for satisfaction of the particular criteria by a solution A to an ABox abduction problem $\langle \mathcal{K}, \Phi \rangle$, where $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.

relevance	ABox := A
	TBox := $\neg\Phi$
	IF ABox consistent with respect to TBox
	THEN OUTPUT A is relevant
	ELSE OUTPUT A is not relevant
consistency	ABox := $\mathcal{A} \cup A$
	TBox := $\mathcal{T}$
	IF ABox consistent with respect to TBox
	THEN OUTPUT A is consistent
	ELSE OUTPUT A is not consistent
minimality	**FOR** every solution B:
	FOR every renaming $\rho : N_I^\star(B) \mapsto N_I^\star(A)$:
	ABox := A
	TBox := $\neg\rho B$
	IF ABox inconsistent with respect to TBox
	THEN FOR every renaming $\varrho : N_I^\star(A) \mapsto N_I^\star(B)$:
	ABox := B
	TBox := $\neg\varrho A$
	IF ABox inconsistent with respect to TBox
	THEN GO TO $(*)$
	REPEAT
	OUTPUT A is not minimal
	(B is minimal with respect to A)
	TERMINATE
	REPEAT
$(*)$	**REPEAT**
	OUTPUT A is minimal

Table 12 Decision procedures for selection criteria via standard reasoning services in $\mathcal{ALCO}$.

Observe that whereas relevance and consistency can be unconditionally decided for any solution, minimality checking remains in the worst case a semi-decidable procedure, guaranteed to output the answer in finite time only if the answer is negative. The consequence follows from the fact that in order to verify minimality, the solution has to be compared with every other solution to the same problem, of which there can be infinitely many. Consider for instance an extension of the *happy John* problem, in which we include an additional TBox axiom $\forall \mathsf{hasFriend}.\textsc{Happy} \sqsubseteq \textsc{Happy}$, stating that an individual is happy if all his friends are happy. In such a scenario all the following assertions are minimal and consistent solutions to the problem:

$$(\forall \mathsf{hasFriend}.\textsc{Happy})(\mathsf{John})$$
$$(\forall \mathsf{hasFriend}.(\forall \mathsf{hasFriend}.\textsc{Happy}))(\mathsf{John})$$
$$(\forall \mathsf{hasFriend}.(\forall \mathsf{hasFriend}.(\forall \mathsf{hasFriend}.\textsc{Happy})))(\mathsf{John})$$
$$\ldots$$

Clearly, the sequence of solutions above, generated due to repeated use of the additional TBox axiom in the abductive procedure, is infinite. In such cases if a solution A is not minimal, it can be at most guaranteed that a solution minimal with respect to it will be found at some point (Theorem 4), and thus will allow for removing A from the set of minimal solutions to the problem. Since the procedure does not terminate, however, the positive answer regarding minimality of A cannot be in principle obtained.

One way of dealing with this inconvenience is to suitably relax the requirement for minimality. A relatively weaker notion, whose verification is decidable even for infinite solution spaces, is that of *local minimality*. This criterion shares with the original one exactly the same conceptual foundation of preference for *prime implicants* of a formula, and presents a high pragmatic value from the user perspective in typical application scenarios. It is weaker in the sense of only approximating the proper minimality in a gradual, controllable manner over the progress of computation. Roughly, we delimit the allowed distance of the entailed and abduced individuals from the known ones in abductive proofs, by some arbitrarily fixed bound.

Definition 9 (Local minimality) A solution A to abductive problem $\langle \mathcal{K}, \Phi \rangle$ is **locally minimal** if there exists a natural number n for which A is an n-locally minimal solution. A is n-**locally minimal** *iff*

1. there exists a MU set σS under A (see Lemma 3) such that for every term t_k occurring in σS there exists a sequence $r_1(t_0, t_1), \ldots, r(t_{k-2}, t_{k-1}), r_k(t_{k-1}, t_k)$ of positive role occurrences in σS, such that $k \leq n$ and $t_0 \in N_I$.
2. there is no other solution B to $\langle \mathcal{K}, \Phi \rangle$ for which the above condition is satisfied and B is minimal with respect to A.

More indirectly, the definition implies that in order to find an n-locally minimal solution it is enough to consider possible models of $\mathcal{K} \cup \neg\Phi$ only to depth bounded by n. What further follows, is that the set of locally minimal solutions coincides with the set of minimal solutions in the limit of computation (for $n \to \infty$). Given above formulation we can easily show that the following claim holds.

Proposition 5 (n-local minimality completeness) *For any $n \in \mathbb{N}$, if A is a plain, n-locally minimal and consistent solution to ABox abduction problem $\langle \mathcal{K}, \Phi \rangle$ then A is an $\vdash_{\text{ABox}}$-solution to $\langle \mathcal{K}, \Phi \rangle$. Furthermore, n-local minimality of A can be decided in a finite number of steps.*

Proof First note, that since we require that there exists a MU set under A then by Theorem 4 A can be found as one of the $\vdash_{\text{ABox}}$-solutions to $\langle \mathcal{K}, \Phi \rangle$. In order to decide

n-local minimality of A we constrain the search for solutions by blocking connection steps that involve clauses introducing new terms, whose distance from the individual names in the abductive graph of the proof is greater than n. Notice, that the abductive graph of a proof for A corresponds to the set of positive role occurrences in the MU set under A, modulo substitution of ground terms for the variables in the graph. Given a finite number of clauses and finite number of individual names, clearly there can be only a finite number of connection-driven proofs for a given abduction problem, whose abductive graph satisfies the constraint. Hence, if there exist an n-locally minimal solution to the problem it has to be found among one of these proofs and all non-minimal solutions can be eliminated via pairwise entailment checks. $\square$

4.3 Correctness for acyclic terminologies

Alternatively to introducing a weaker notion of minimality, one can try to ensure termination by identifying a restricted form of TBoxes, which cannot result in the construction of infinite abductive proofs. The standard distinction, used in similar contexts, between *cyclic* and *acyclic* terminologies [4] allows for eliminating TBoxes involving certain forms of definitional loops. Since these categories, however, do not apply to general TBoxes, which are considered in this paper, we propose a new notion of *acyclicity* scoped particularly for reasoning in connection-driven proof systems, such as discussed here.

A clause Cl is *directly connectible* to Cl' if there exists a connection between Cl and Cl'. We define the relation *connectible* as the transitive closure of *directly connectible* and say that a terminology $\mathcal{T}$ under τ-transformation is *connection acyclic* if for every non-DL predicate $P \in \mathbb{P}$ occurring in $\mathcal{T}$ and every two clauses $Cl, Cl' \in \mathcal{T}$ such that $P(\overline{x}) \in Cl$ and $\neg P(\overline{x}) \in Cl'$, the clauses are connectible only through P. Otherwise we call $\mathcal{T}$ *connection cyclic*.

If the requirement of connection acyclicity of the TBox is satisfied it is possible to guarantee termination of reasoning for any ABox abduction problem.

Proposition 6 (Termination for acyclic TBoxes) *Let $\langle \mathcal{K}, \Phi \rangle$ be an ABox abduction problem, with $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a connection acyclic terminology. The procedure of solving $\langle \mathcal{K}, \Phi \rangle$ via $\vdash_{ABox}$ terminates in a finite number of steps.*

Proof Recall that occurrences of non-DL predicates in the clauses under τ-transformation mark the original positions of the quantification restrictions in the DL axioms. The connection acyclicity condition prohibits generation of infinite chains of terms (Skolem terms and variables) by cyclic reuse of clauses. Given a finite number of clauses and named individuals it is impossible to infinitely expand tableau branches or create infinite sequences of resolution inference steps. Thus the reasoning terminates in finite time. $\square$

As a consequence, for ABox abduction problems based on connection acyclic terminologies it is possible to decide the strong notion of minimality for every computed solution.

Also, as pointed out in [30], given the acyclicity condition is satisfied it is possible to engage more sophisticated goal-oriented reasoning methods for checking consistency of abductive solutions. Such techniques, based on the same connection-driven calculi as the abductive reasoning itself, do not require a computationally expensive consistency checking of the whole knowledge base for deciding consistency of a single solution with respect to that base. Instead they are meant precisely for verifying whether a newly generated solution does not *trigger* inconsistency in the otherwise consistent knowledge base.

5 Conclusions

ABox abduction is a particularly interesting form of abductive reasoning over DL ontologies. It is constitutive for problems of identifying minimal sets of ABox axioms that, if added to the knowledge base, trigger entailment of a requested set of assertions. Possible application scenarios for this inference service are numerous and the need for a practical tool support in dealing with abductive tasks, especially in the context of OWL applications, has increasingly been reported [13, 6]. Nevertheless, the amount of work addressing the problem that has been so far undertaken is very limited.

In this paper we have introduced a formal computational framework for ABox abduction in the DL $\mathcal{ALC}$. The employed reasoning mechanism rests on regular connection tableaux and resolution with set-of-support, refinements of two well-known and commonly applied automated theorem proving techniques. Essentially, an ABox abduction problem is reduced to the task of constructing a refutation proof, with either of the two methods, for the complement of the abductive query, given the background knowledge base. Any set of assertions that can force completion of such a proof is a solution to the original problem. Along with the algorithms, we have developed a special satisfiability- and structure-preserving clausal transformation for DL axioms, and a method for retrieving well-formed $\mathcal{ALE}$ ABox assertions from abductive proofs. Finally, we have discussed the possibility of using standard DL reasoning services for applying selection criteria on the generated sets of solutions, and considered the special case of infinite solution spaces. The whole procedure has been proven sound and complete for solving ABox abduction problems in $\mathcal{ALC}$.

The framework has a universal and flexible character, encouraging customization towards specific use cases. The transformation procedure allows general and cyclic TBoxes and easily covers all expressive means available in $\mathcal{ALC}$. Consequently, abductive reasoning is not dependent on a particular syntactic structure of the input. Also, we have not committed ourselves to any specific solving heuristics or arbitrary preference criteria over potential solutions, except for the most fundamental constraints such as consistency, relevance and minimality. Due to the connection-driven proof strategy, inherent to both of the calculi employed, the framework exhibits a goal-oriented behavior on the search level, enabling a more efficient and focused form of computation. Moreover, it is guaranteed to provide interesting results even for problems with an infinite number of solutions. Another strong feature is the structural modularity of the approach, which allows for considering different phases of solving a problem independently from the others, and handling them be means of the most suitable tools. For instance, it can be reasonable to use separate algorithms for finding solutions and for their post-processing, depending on the average performance of particular computation techniques on the respective tasks.

In order to foster the progress towards designing practical abductive DL reasoners, the work on the framework should be advanced at least on three levels. First, it is necessary to systematically extend the approach to other DLs, especially to ones being of a special application interest, such as highly expressive DLs underlying OWL languages, i.e. $\mathcal{SHOIN}$, $\mathcal{SROIQ}$ [25] and their specific fragments underpinning different profiles of OWL. A shift towards more expressive languages should require extra transformation rules, covering additional constructors in DL axioms, and will have to involve revisions at least in the definitions of an admissible abductive graph, admissible FOL-base, and the procedure of extracting ABox assertions from an abductive proof. In all these cases increased expressivity permits more structural possibilities that should be accounted for in the procedure. Second, it is desirable to tighten the links between the proposed procedures and existing DL reasoning tools, in order to enable a convenient integration of ABox abduction in larger reasoning and knowledge representation infrastructures, as well as to save on the effort of reinventing well-developed optimization techniques for reasoning with DLs. A promising way of aligning the proposed approach with mainstream reasoning methods in DL, which we want to investigate in the future, is by incorporating the connectedness requirement into standard tableau-based reasoners. A successful integration is obviously not a straightforward prospect, as the presented procedure requires an extensive support for FOL features, predominantly Skolemization, that are not present in the standard DL reasoners. However, it is likely that even without them connectedness could be to some extent approximated in such calculi, for instance by expanding in the tableau only those axioms in NNF that contain concept names complementary to the ones present anywhere on the considered branch. Such a strategy would lead to a noticeable loss in the goal-directedness of the procedure, but in return it would not introduce a need for a fundamental reconstruction of the reasoning paradigm. Finally, the framework should be accompanied by a menu of optional extensions and plugins, such as additional selection strategies, the possibility of marking abducibles, or the integration of efficient search heuristics, for which appropriate formal foundations will have to be developed.

Acknowledgements. We would like to thank two anonymous reviewers of the *Journal of Automated Reasoning* for their helpful comments.

References

1. A. Aliseda-Llera. *Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence.* PhD thesis, ILLC Disertation Series, University of Amsterdam, 1997.
2. P. B. Andrews. Theorem proving via general matings. *Journal of the ACM*, 28(2):193–214, 1981.
3. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.
4. F. Baader and W. Nutt. Basic description logics. In F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 47–100. Cambridge University Press, 2003.
5. L. Bachmair and H. Ganzinger. Resolution theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 1–82. Elsevier Science Publishers B.V., 2001.
6. M. Bada, C. Mungall, and L. Hunter. A call for an abductive reasoning feature in OWL-reasoning tools toward ontology quality control. In *Proceedings of the 5th International Workshop OWL: Experiences and Directions 2008 (OWLED08 Karlsruhe)*, 2008.
7. W. Bibel. On matrices with connections. *Journal of the ACM*, 28(4):633–645, 1981.
8. M. Bienvenu. Complexity of abduction in the EL family of lightweight description logics. In *Proceedings of KR2008, 11th International Conference on Principles of Knowledge Representation and Reasoning*, 2008.

9. P. Blackburn and J. van Benthem. Modal logic: a semantic perspective. In P. Blackburn, J. van Benthem, and F. Wolter, editors, *Handbook of Modal Logic*, pages 1–82. Elsevier, 2006.
10. S. Colucci, T. D. Noia, E. D. Sciascio, F. M. Donini, and M. Mongiello. A uniform tableaux-based approach to concept abduction and contraction in ALN. In *Proceedings of the 2004 International Workshop on Description Logics (DL2004)*, 2004.
11. P. T. Cox and T. Pietrzykowski. Causes for events: Their computation and applications. In *Proceedings of the 8th International Conference on Automated Deduction*, pages 608–621, 1986.
12. C. Elsenbroich. *Instinct for Detection.* PhD thesis, Department of Computer Science, King's College London, 2005.
13. C. Elsenbroich, O. Kutz, and U. Sattler. A case for abductive reasoning over ontologies. In B. C. Grau, P. Hitzler, C. Shankey, and E. Wallace, editors, *Proceedings of the OWLED'06 workshop on OWL: Experiences and Directions 2006*, volume 216, 2006.
14. U. Endriss, P. Mancarella, F. Sadri, G. Terreni, and F. Toni. The CIFF proof procedure for abductive logic programming with constraints. In J. J. Alferes and J. Leite, editors, *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA-2004)*, volume 3229 of *LNAI*, pages 31–43. Springer-Verlag, September 2004.
15. S. Espinosa Peraldi, A. Kaya, S. Melzer, R. Möller, and M. Wessel. Multimedia interpretation as abduction. In *International Workshop on Description Logics (DL-2007)*, 2007.
16. P. Flach and A. Kakas, editors. *Abduction and Induction: Essays on their relation and integration.* Kluwer Academic Publishers, 2000.
17. D. M. Gabbay. *Elementary Logics: a Procedural Perspective.* Prentice Hall Europe, UK, 1998.
18. D. M. Gabbay and N. Olivetti. *Goal-Directed Proof Theory*, volume 21 of *Applied Logic Series.* Kluwer Academic Publishers, 2000.
19. D. M. Gabbay and J. Woods. Advice on abductive logic. *Logic Journal of the IGPL*, 14(2):182–219, 2006.
20. V. Haarslev and R. Möller. RACER system description. In R. Goré, A. Leitsch, and T. Nipkow, editors, *International Joint Conference on Automated Reasoning, IJCAR'2001*, pages 701–705, 2001.
21. R. Hähnle. Tableaux and related methods. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 101–178. Elsevier Science Publishers B.V., 2001.
22. R. Hähnle, N. Murray, and E. Rosenthal. Linearity and regularity with negation normal form. *Theoretical Computer Science*, 328(4):325–354, 2004.
23. M. Horridge, B. Parsia, and U. Sattler. Laconic and precise justifications in OWL. In *The International Semantic Web Conference 2008 (ISWC 08)*, 2008.
24. I. Horrocks. The FaCT system. In H. de Swart, editor, *Proceedings of the 2nd Int. Conf. on Analytic Tableaux and Related Methods (TABLEAUX'98)*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 307–312. Springer, 1998.
25. I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible SROIQ. In *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 57–67, 2006.
26. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.
27. A. C. Kakas, R. A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1992.
28. A. Kalyanpur, B. Parsia, and B. C. Grau. Beyond asserted axioms: Fine-grain justifications for OWL-DL entailments. In *Proceedings of the International Workshop on Description Logics DL'06*, 2006.
29. Y. Kazakov and B. Motik. A resolution-based decision procedure for SHOIQ. *Journal of Automated Reasoning*, 40(2–3):89–116, 2008.
30. S. Klarman. ABox abduction in Description Logic. Master's thesis, ILLC, University of Amsterdam, http://www.illc.uva.nl/Publications/ResearchReports/MoL-2008-03.text.pdf, 2008.
31. D. W. Loveland. *Automated Theorem Proving: A Logical Basis*, volume 6 of *Fundamental Studies in Computer Science.* North-Holland Publishing, 1978.
32. M. C. Mayer and F. Pirri. First order abduction via tableau and sequent calculi. *Bulletin of the IGPL*, 1 (1):99–117, 1993.

33. M. C. Mayer and F. Pirri. Propositional abduction in modal logic. *Journal of the Interest Group in Pure and Applied Logics*, 3 (6):907–919, 1995.
34. R. Möller and B. Neumann. Ontology-based reasoning techniques for multimedia interpretation and retrieval. In Y. Kompatsiaris and P. Hobson, editors, *Semantic Multimedia and Ontologies: Theory and Applications*. Springer, 2008.
35. B. Motik, R. Shearer, and I. Horrocks. Optimized Reasoning in Description Logics using Hypertableaux. In F. Pfenning, editor, *Proceedings of the 21st Conference on Automated Deduction (CADE-21)*, volume 4603, pages 67–83, 2007.
36. G. Paul. Approaches to abductive reasoning: an overview. *Artificial Intelligence Review*, 7(2):109–152, 1993.
37. W. V. O. Quine. On cores and prime implicants of truth functions. *The American Mathematical Monthly*, 66(9):755–760, 1959.
38. K. Schild. A correspondence theory for terminological logics: preliminary report. In *Proceedings of IJCAI-91, 12th International Joint Conference on Artificial Intelligence*, pages 466–471, Sidney, AU, 1991.
39. S. Schlobach, Z. Huang, R. Cornet, and F. van Harmelen. Debugging incoherent terminologies. *Journal of Automated Reasoning*, 33(3):317–349, 2007.
40. G. Schurz. Models of abductive reasoning. TPD Preprint 1, University of Düsseldorf, 2002.
41. T. Tammet. *Resolution Methods for Decision Problems and Finite-Model Building*. PhD thesis, Göteborg University, 1992.