Künstl Intell (2016) 30:177–182
DOI 10.1007/s13218-015-0399-3

TECHNICAL CONTRIBUTION

Abductive Conjunctive Query Answering w.r.t. Ontologies

Ralf Möller[1] · Özgür Özçep[1] · Volker Haarslev[2] · Anahita Nafissi[3] · Michael Wessel[4]

Received: 17 May 2015 / Accepted: 15 September 2015 / Published online: 13 October 2015

Abstract In this article we investigate *abductive* conjunctive query answering w.r.t. ontologies and show how use cases can benefit from this kind of query answering service. While practical reasoning systems such as Racer have supported abductive conjunctive query answering for 10 years now, and many projects have exploited this feature, few publications deal with A-box abduction from an implementation perspective. This article gives a generalized overview on features provided by practical systems and also explains optimization techniques needed to meet practical requirements.

Keywords Description logics · Query answering · Abduction

✉ Ralf Möller
moeller@ifis.uni-luebeck.de

Özgür Özçep
oezcep@ifis.uni-luebeck.de

Volker Haarslev
haarslev@cse.concordia.ca

Anahita Nafissi
a.nafissi@fz-juelich.de

Michael Wessel
michael_wessel@gmx.de

[1] University of Lübeck, Lübeck, Germany

[2] Concordia University, Montreal, Canada

[3] Forschungszentrum Jülich, Düren, Germany

[4] Hamburg University of Technology, Hamburg, Germany

1 Introduction

Ontologies allow for a formalization of conceptual and relational notions relevant for modeling application systems of a particular domain. Notions mentioned in the signature of an ontology can be used to formulate queries, which are used as part of algorithms to solve application problems. As a query language, for the projects we were involved in, so-called conjunctive queries were expressive enough. Conjunctive queries are specified by a set of query atoms (unary or binary predicates with constants or variables as arguments) with existentially quantified as well as free variables (queries without free variables are called boolean queries). In our opinion, existentially quantified variables are rarely important in practical applications, and therefore we deal with conjunctive queries without existential variables here. Queries are answered w.r.t. sets of descriptions for concrete data (an A-box) in such a way that for all (free) variables in a query, bindings to constants mentioned in the data descriptions need to be found, such that, for a given set of variable bindings and corresponding variable substitutions, all (ground) atoms are entailed w.r.t. a given set of terminological axioms (a T-box). Informally speaking, the result set for a query is the set of all variables bindings for which this holds. A variable binding which is an element of the result set is called a query result or a query answer. The axioms of an ontology have an impact on the query answers, and, due to our experiences, expressive ontologies allow for the derivation of more answers in practical applications. In addition, we argue that standard conjunctive query answering w.r.t. expressive ontologies is not enough.

Rather than having *all* (grounded) atoms of a conjunctive query entailed for generating answers, in a less strict

form of query answering, query atoms that are not fulfilled could be indicated as part of the query answer. We call this mode *abductive* conjunctive query answering w.r.t. ontologies, or *abduction-based data analysis* (ABDA), and in contrast to restrictions imposed by ontology-based data access (OBDA), we investigate this principle w.r.t. expressive ontologies in this article. Abductive query answering has also been explored in logic programming [1], and is sometimes referred to under the term query relaxation [2]. The article is written in a not too technical style and should be understandable with a background in logic, although basic knowledge in ontology description languages in general, and description logic systems in particular might be helpful.

2 Example

As an example for data analysis with abductive query answering we present a high-level image interpretation scenario in the spirit of the seminal article by Shanahan [3]. The example is basically taken from [4] and considers the interpretation of athletics images. We use Racer as an inference engine [5] for providing an actual implementation of ABDA principles. The purpose of the example is to demonstrate the central ideas behind abductive conjunctive query answering for data analysis used for high-level interpretation tasks. We do not deal with computer vision problems such as object detection in this article, and the application domain is certainly not completely covered by the ontologies and techniques we discuss as part of the example. The following small ontology is defined, with the signature specifying concept and role names relevant for the application context. In addition, we also provide sample axioms in the T-box relating the notions given in the signature.

```
(in-tbox athletics)
(signature
    :concept-names (pole bar human athlete
                    pole-vault high-jump
                    pv-in-start-phase
                    pv-in-turn-phase)
    :role-names (has-part
                 overlapping)

(disjoint pole bar human pole-vault high-jump)
(disjoint pv-in-start-phase pv-in-turn-phase)
(subconcept athlete human)
```

We assume that data-driven image analysis provides detection and classification of objects in the form of A-boxes, i.e., sets of assertional axioms [4]. For the example image shown in Fig. 1 a temporary A-box `image-42` is declared relative to the above-mentioned T-box.

```
(in-abox image-42 athletics)
(instance p1 pole)
(instance h1 athlete)
```

Two objects, a pole `p1` and an athlete `h1`, are specified with A-box assertions as data descriptions. In addition the image interpretation module is assumed to use qualitative relations to spatially relate objects in a qualitative way. In the example we assume that the pole and the athlete overlap in the image, which gives us another data description mentioned in the A-box that is associated with the image: `(related p1 h1 overlapping)`.

Furthermore we assume that there are other A-boxes generated for representing alternative data-driven image processing results. Since data-driven processing is error-prone, we assume that one should be able to use background knowledge to "confirm" or "reject" data-driven processing results. The confirmation (or rejection) by background knowledge is done by data analysis, in our case to implement (high-level) image interpretation. The better the interpretation, the more confidence one can have in the correctness of a particular image processing A-box generated in a data-driven way. Note that well-engineered description logic systems allow for multiple A-boxes maintained efficiently at the same time, each of which effectively refers to the same T-box. In addition, new A-boxes can be created and deleted on the fly, without impacting the status of the T-box, exploiting previous computations on T-box axioms as effectively as possible.

Data analysis knowledge in our example is modeled using "symbolic patterns," which are represented as conjunctive queries. We assume that the set of patterns considered for computing image interpretations can be determined depending on the context and focus of attention. Due to space restrictions, we do not deal with this problem here. For the sake of the example we consider two patterns, namely a pole vault in the start phase and a pole vault in the turn phase:

```
(and (?z ?x has-part) (?z ?y has-part)
     (?x pole) (?y human) (?x ?y overlapping)
     (?z pole-vault) (?z pv-in-start-phase))

(and (?z ?x has-part) (?z ?y has-part)
     (?x bar) (?y human) (?x ?y overlapping)
     (?z pole-vault) (?z pv-in-turn-phase))
```

These patterns are used to query the image A-box. The procedure `retrieve-with-explanation` answers a conjunctive query w.r.t. an A-box and an associated T-box,

Fig. 1 Pole vault in starting phase

possibly explaining which query atoms are not entailed. The pattern for a pole vault in the start phase is used first.

```
(retrieve-with-explanation (?x ?y ?z)
   (and (?z ?x has-part) (?z ?y has-part)
        (?x pole) (?y human) (?x ?y overlapping)
        (?z pole-vault) (?z pv-in-start-phase))
   :abox image-42
   :final-consistency-checking-p t)
```

In the call to `retrieve-with-explanation` the first parameter denotes the variables for which bindings need to be generated. The result of the call is called an explanation, which is then used to represent an image interpretation. Explanation is meant here in the sense of "being part of a larger picture," namely certain patterns, possibly generated from previous experiences.

The call to `retrieve-with-explanation` returns the following data structure, explaining which query atoms must be hypothesized. In general, multiple explanations are possible, but here we have just one.

```
(t (((:tuple (?x p1) (?y h1) (?z IND-175))
      (:new-inds IND-175)
      (:hypothesized-assertions
         (instance IND-175 pole-vault)
         (instance IND-175 pv-in-start-phase)
         (related IND-175 p1 has-part)
         (related IND-175 h1 has-part)))))
```

The answer is `t` with additional information about new individuals, and hypothesized assertions. One new individual is generated in our example, namely `IND-175`, in order to represent the pole-vault event constructed such that the query result is `t`. The pole `p1` and the human `h1` are indicated to be parts of the pole-vault event. Note that `(h1 human)` is entailed by A-box `image-42` due to a T-box axiom and, thus, is not hypothesized.

The hypothesized assertions can be safely added to the A-box `image-42` since we have enforced consistency in the call (`:final-consistency-checking-p` is set to `t`). Neither `p1` nor `h1` can be used as a binding for `?z` due to disjointness axioms, so a new individual is used (and we get just one explanation). The new individual can be associated with the image object. The central idea is that this kind of high-level interpretation allows for retrieving media objects based on high-level symbolic queries (e.g., a query for pole-vault images, see [6] for details).

We also investigate the result of the second interpretation pattern:

```
(retrieve-with-explanation (?x ?y ?z)
   (and (?z ?x has-part)
        (?z ?y has-part)
        (?x bar)
        (?y human)
        (?x ?y overlapping)
        (?z pole-vault)
        (?z pv-in-turn-phase))
   :abox image-42
   :final-consistency-checking-p t)
```

As can be seen in the result

```
(t (((:tuple (?x IND-185) (?y h1) (?z IND-176))
     (:new-inds IND-176 IND-185)
     (:hypothesized-assertions
        (instance IND-176 pole-vault)
        (instance IND-176 pv-in-turn-phase)
        (related IND-176 IND-185 has-part)
        (instance IND-185 bar)
        (related IND-176 h1 has-part)))))
```

the second pattern requires more hypothesized assertions (and more new individuals). Therefore, the first one is defined to be better. In addition to a pole-vault individual, here also a new bar individual must be generated (similarly as above, neither `p1` nor `h1` can be used as a binding for `?x` or `?z` due to T-box axioms). Another pattern to interpret the image refers to a high jump event can be specified as follows.

```
(and (?z ?x has-part) (?z ?y has-part)
     (?x bar) (?y human)
     (?x ?y overlapping) (?z high-jump))
```

Note that this kind of abductive processing scheme definitely avoids the brittleness of standard rule-based symbolic processing approaches (see, e.g., [7]). A pattern is not subject to a boolean match, but induces a "fitness score", which will be described in more detail below. For now we can safely assume that very many patterns will be present in a suitable library for practical purposes. A naive

algorithm uses `retrieve-with-explanation` with each pattern (albeit possibly in parallel), and derives a score vector based on all patterns that are active in a certain context. Obviously, we are only interested in the topmost (top-k) patterns w.r.t. this vector, and we should be able to find some means to easily sort out patterns that are less suited for explaining things described in a particular input A-box. However, even for a single pattern a combinatorial space has to be investigated to cope with all possible variable substitutions. In the next section we give more details about computing and ranking abductive answers for conjunctive queries, and discuss how the search space for answers can be pruned based on a score for evaluating answers. Later on we also discuss how multiple interpretation patterns, viz. conjunctive queries, can be handled effectively.

3 Answering Conjunctive Queries with Abduction

Abduction has a long tradition in logical programming. See, e.g., [8] for a seminal paper imposing the view that the abduction problem is the problem to compute a set of formulae Δ such that $\Sigma \cup \Delta \models \Gamma$ with Σ representing the knowledge base and Γ being the set of formulae which should be entailed. As side conditions, $\Sigma \cup \Delta$ must be consistent and Δ should be minimal in some formal sense. Depending on the application contexts, different logical languages have been investigated for the formulae involved in an abduction problem as well as for specifying how formulae in Δ are to be built.

A specific variant of abduction, namely A-box abduction, is relevant for applications of ontology languages (see [9, 10] for an early publication for A-box abduction inspired by abduction in logic programming). For RDF, the idea later on has been considered as query relaxation as well [11]. In addition, see also [12–14] for a survey on using A-box abduction for media data interpretation. While [9, 10] and successors [15, 16] have used the atoms in a query for specifying the space of abducibles, later approaches have used just concept and role names for specifying the abducibles [17–19], resulting in a much more complex abduction problem. For the latter approaches no stable implementations are available.

Common to all approaches is however that in solutions to abduction problems also new individuals (constants) can be mentioned. Furthermore, since an abduction problem can have multiple solutions, techniques for ranking solutions to abduction problems have been investigated right from the beginning, mostly from an application-specific perspective (based on weights [20] or probabilities [21, 22] to name just a few).

Here we focus on abduction problems for which the space of abducibles is given by the atoms mentioned in conjunctive queries, and, based on a lightweight scoring technique, we analyze how results to a single abduction problem, viz. one call to `retrieve-with-explanation` can be computed effectively such that practical application scenarios can actually be handled (cf. [12–14]). As we have seen in the examples above, for each pattern a call to `retrieve-with-explanation` returns the hypothesized assertions (the Δ from above). The idea of defining a score (and minimizing it) is that the more needs to be hypothesized the less perfect a pattern "matches" the input A-box, and the more atoms are entailed the higher a pattern should be ranked because the context expressed in the pattern is better captured in the input A-box. Therefore, the score S for a result of a call to `retrieve-with-explanation` w.r.t. an A-box $\mathcal{A}$ is defined to be the difference of the number of entailed assertions $|\mathcal{E}|$ from pattern $\mathcal{P}$ and the number of hypothesized assertions in Δ, i.e., $S = |\mathcal{E}| - |\Delta|$ with $\mathcal{P} = \mathcal{E} \cup \Delta$.

For the discussion of scores we use a simplified example that demonstrates the main insights about optimizing abductive query answering. In particular, we use an empty T-box (called `empty-tbox`).

```
(in-abox abox-17 empty-tbox)
(instance i c)
(related i j r)
```

The abduction problem is specified as follows, with the solution being indicated below.

```
(retrieve-with-explanation (?x ?y)
   (and (?x c) (?x ?y r) (?y d))
   :abox abox-17
   :final-consistency-checking-p t
   :show-score-p t)

(t
 (((:tuple (?x i) (?y j))
   (:new-inds)
   (:hypothesized-assertions (instance j d))
   (:score 1 ...) ...))
```

There is one hypothesized assertion, namely the assertion `(instance j d)`. The search space is illustrated in Fig. 2. We assume depth-first search as indicated by the curled arrow in Fig. 2. In the A-box there are two (old) individuals mentioned, `i` and `j`, which can be used as bindings for variables x and y, respectively. In addition, for each variable new individuals can be used (new_1 and new_2). Each path in the tree corresponds to a particular binding. If resulting query atoms are not entailed, they are hypothesized. In the example, there are 9 possible sets Δ

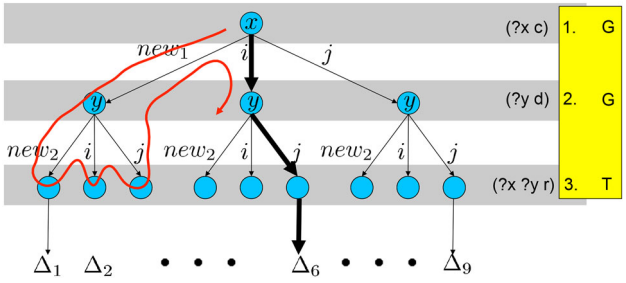

Fig. 2 Search space

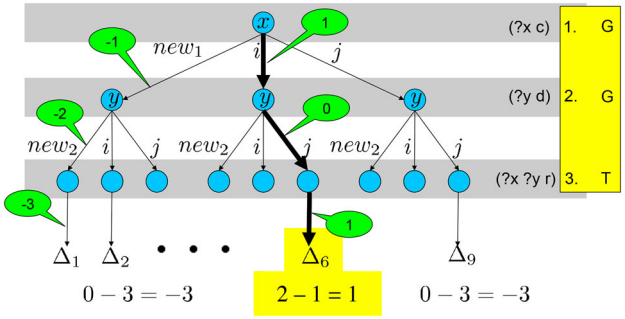

Fig. 3 Search graph with path scores

for hypothesized assertions. The solution presented above corresponds to the path with thicker lines leading to Δ_6. The first two assertions are generators (G) for variables x and y, respectively, the last one is then a tester (T).

Since introducing a new individual will always lead to a hypothesized assertion, the first path receives the worst score (see Fig. 3). Path 6, the solution, binds x to `i` and uses `j` as a binding for y (see the thick arrow in Fig. 3). Given the A-box, `(instance i c)` is entailed (the assertion is even a member of the A-box here), hence the score is increased by 1. However, `(instance j d)` needs to be hypothesized (reduction of the score by 1 to 0). Last, the assertion `(related i j r)` is entailed (score incremented by 1). Δ_6 is the best solution so far. As shown in Fig. 4 all remaining paths yield lower scores, and thus, the remaining paths can be pruned. It is important to understand that for pruning correctly, the score must be monotone. The score definition we gave above, i.e., $S = |\mathcal{E}| - |\Delta|$ does not have this property, and we need to slightly reformulate it. Since $\mathcal{P} = \mathcal{E} \cup \Delta$, we get $|\mathcal{P}| = |\mathcal{E}| + |\Delta|$, or $|\mathcal{P}| - |\mathcal{E}| = |\Delta|$. Thus, $S = 2|\mathcal{E}| - |\mathcal{P}|$. The size of $\mathcal{P}$ is constant, and the size of $\mathcal{E}$ can only grow. With this reformulation, the score S becomes monotone, and hence pruning w.r.t. maximization of S works correctly because one skip branches of the search space if it can be shown that for the best-so-far solution $|\mathcal{E}|$ is maximal (see Fig. 4).

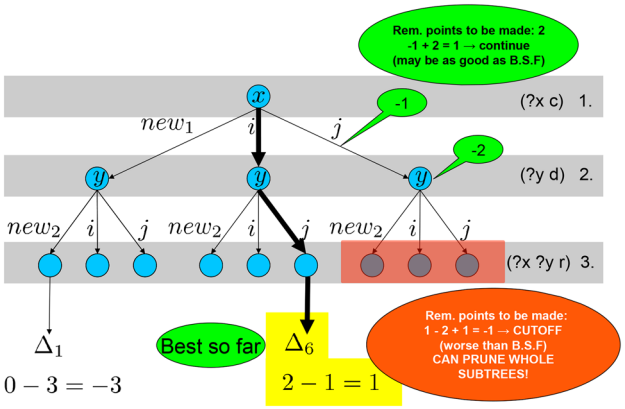

Fig. 4 Pruning

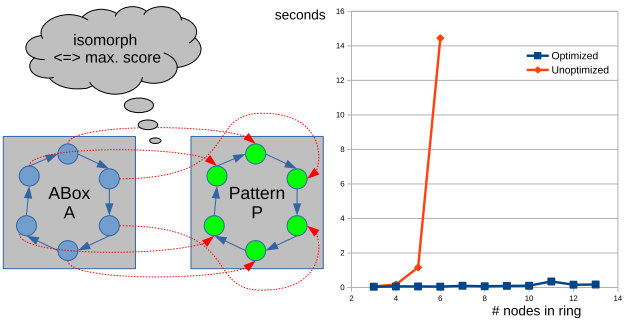

Fig. 5 Effectiveness

Figure 5 demonstrates the effectiveness of pruning for more complex abduction problems (graph isomorphism between two graphs A and B with different node names and edges all labelled R is formulated as an abduction problem). Graph A is mapped into an A-box directly, whereas graph B is mapped into a query pattern P (node names are replaced with corresponding variable names). In our experiments we have used growing circles to investigate the effectiveness of pruning for abductive query answering on A-boxes. A "circular" query pattern P is to be answered w.r.t. a "circular" A-box $\mathcal{A}$, and if nodes are mapped as indicated in Fig. 5 the lowest number of assertions need to be hypothesized. As can be seen in the diagram showing runtimes for growing numbers of nodes in a circle (see the righthand side of Fig. 5), the optimized version (with pruning) yields promising results whereas the version without pruning (unoptimized) immediately runs into combinatorial explosion.

Given the set of entailed assertions mentioned in the output that `retrieve-with-explanation` generates for an input pattern (see above), one can determine the minimal sets of assertions (MinAs) from A-box $\mathcal{A}$ responsible for the entailment (see, e.g., pinpointing

algorithms in [23] and justification techniques explained in [24]). The score of a query answer for a pattern can be used as a score for all MinAs for the answer. Adding the MinA scores for all query patterns in focus yields a score for a complete A-box. Given A-boxes represent different interpretation results, like in our image processing example from above, comparing the A-box scores allows us to rank different interpretation possibilities in the media interpretation context we have sketched.[1]

4 Conclusion and Future Work

Using high-level image interpretation as an example, we have seen that abduction-based data analysis can be optimized such that reasoning systems can be used for data analysis in practical application contexts. Brittleness of rule systems is avoided by computing query answering solutions with scored hypothesized assertions. With the features demonstrated here, it becomes clear that description logic reasoning systems provide more interesting features than just consistency checking, T-box classification, or strict OBDA. For many more details on ABDA with `retrieve-with-explanation`, see the Racer system documentation. In the future we will extend ABDA to stream-based data analysis (see, e.g., [25]).

References

1. Kakas A, Kowalski R, Toni F (1993) Abductive logic programming. J Logic Comput 2(6):239–770
2. Gaasterland T, Godfrey P, Minker J (1992) Relaxation as a platform for cooperative answering. J Intell Inf Syst 1:293–321
3. Shanahan M (2005) Perception as abduction: turning sensor data into meaningful representation. Cognit Sci 29(1):103–134
4. Petasis G, Möller R, Karkaletsis V (2013) Boemie: reasoning-based information extraction. In: Proceedings of the 1st workshop on natural language processing and automated reasoning co-located with 12th international conference on logic programming and nonmonotonic reasoning (LPNMR 2013), pp 60–75
5. Haarslev V, Hidde K, Möller R, Wessel M (2012) The RacerPro knowledge representation and reasoning system. Seman Web J 3(3):267–277
6. Espinosa S, Kaya A, Möller R (2009) The BOEMIE semantic browser: a semantic application exploiting rich semantic metadata. In: Proceedings of the applications of semantic technologies workshop (AST-2009), Lübeck, Germany
7. Lenat DB, Feigenbaum EA (1991) On the thresholds of knowledge. Artif Intell 47(1–3):185–250
8. Denecker M, Kakas AC (2002) Abduction in logic programming. In: Kakas AC, Sadri F (eds) Computational logic: logic programming and beyond, essays in honour of Robert A. Kowalski, part I. Lecture notes in computer science, vol 2407. Springer, Berlin, pp 402–436
9. Möller R, Neumann B (2008) Ontology-based reasoning techniques for multimedia interpretation and retrieval. In: Semantic multimedia and ontologies : theory and applications. Springer, Berlin, pp 55–98
10. Espinosa Peraldi S, Kaya A, Melzer S, Möller R (2008) On ontology based abduction for text interpretation. In: Gelbukh A (ed) Proceedings of 9th international conference on intelligent text processing and computational linguistics (CICLing-2008)
11. Dolog P, Stuckenschmidt H, Wache H, Diederich J (2009) Relaxing rdf queries based on user and domain preferences. J Intell Inf Syst 33:239–260
12. Espinosa S (2011) Content management and knowledge management: two faces of ontology-based text interpretation. PhD thesis, Hamburg University of Technology
13. Espinosa S, Atila K, Möller R (2011) Knowledge-driven multimedia information extraction and ontology evolution. In: LNCS, chapter logical formalization of multimedia interpretation, vol 6050. Springer, Berlin, pp 110–133
14. Kaya A (2010) A logic-based approach to multimedia interpretation. PhD thesis, Hamburg University of Technology
15. Castano S, Peraldi ISE, Ferrara A, Karkaletsis V, Kaya A, Möller R, Montanelli S, Petasis G, Wessel M (2009) Multimedia interpretation for dynamic ontology evolution. J. Log. Comput. 19(5):859–897
16. Espinosa S, Kaya A, Möller R (2009) Formalizing multimedia interpretation based on abduction over description logic aboxes. In: Proceedings of the 2009 international workshop on description logics DL- 2009, 27 to 30 July 2009. CEUR workshop proceedings, vol 477, Oxford, UK
17. Du J, Guilin Q, Yi-Dong S, Jeff PZ (2012) Towards practical abox abduction in large description logic ontologies. Int. J. Seman. Web Inf. Syst. 8(2):1–33
18. Klarman S, Endriss U, Schlobach S (2011) Abox abduction in the description logic ALC. J Autom Reas 46(1):43–80
19. Ma Y, Gu T, Xu B, Chang L (2012) An abox abduction algorithm for the description logic alci. In: Intelligent information processing VI. IFIP advances in information and communication technology, vol 385. Springer, Berlin, pp 125–130
20. Hobbs JR, Stickel ME, Appelt DE, Martin PA (1993) Interpretation as abduction. Artif Intell 63(1–2):69–142
21. Gries O, Möller R, Nafissi A, Rosenfeld M, Sokolski K, Wessel M (2010) A probabilistic abduction engine for media interpretation based on ontologies. In: Hitzler P, Lukasiewicz T (eds) Web reasoning and rule systems—fourth international conference, RR 2010, Bressanone/Brixen, Italy, September 22–24, 2010. Proceedings. Lecture notes in computer science, vol 6333. Springer, Berlin, pp 182–194
22. Nafissi A (2013) Applying markov logics for controlling abox abduction. PhD thesis, Hamburg University of Technology
23. Baader F, Peñaloza R (2010) Axiom pinpointing in general tableaux. J Logic Comput 20(1):5–34 **(Special Issue: Tableaux and Analytic Proof Methods)**
24. Kalyanpur A, Parsia B, Horridge M, Sirin E (2007) Finding all justifications of owl dl entailments. In: The semantic web. Springer, Berlin, pp 267–280
25. Özçep OL, Möller R, Neuenstadt C (2014) A stream-temporal query language for ontology based data access. In: KI 2014, vol 8736. LNCS, pp 183–194

[1] A slightly different approach for ranking different interpretation possibilities based on probabilistic logic is presented in [22].