# Explaining User Errors in Knowledge Base Completion

Barış Sertkaya*

TU Dresden, Germany
sertkaya@tcs.inf.tu-dresden.de

**Abstract.** Knowledge base completion is a method for extending both the terminological and assertional part of a Description Logic knowledge base by using information provided by a domain expert. It ensures that the extended knowledge base is complete w.r.t. a fixed interpretation in a certain, well-defined sense. Here we consider the problem of explaining user errors in knowledge base completion. We show that for this setting, the problem of deciding the existence of an explanation within a specified cardinality bound is NP-complete, and the problem of counting explanations that are minimal w.r.t. set inclusion is #P-complete. We also provide an algorithm that computes one minimal explanation by performing at most polynomially many subsumption tests.

## 1 Introduction

The most notable success of DLs so far is due to the fact that they provide the logical underpinning of OWL [HPSvH03], the standard ontology language for the semantic web [BLHL01]. As a consequence of this standardization, several ontology editors [KFNM04,OVSM04,KPS+06] now support OWL, and ontologies written in OWL are employed in more and more applications. As the size of these ontologies grows, tools that support improving their quality become more important. The tools available until now use DL reasoning to detect inconsistencies and to infer consequences. There are also promising approaches that allow to pinpoint the reasons for inconsistencies and for certain consequences, and that help the ontology engineer to resolve inconsistencies and to remove unwanted consequences [SC03,KPSG06,BPS07]. These approaches address the quality dimension of *soundness* of an ontology, both within itself (consistency) and w.r.t. the intended application domain (no unwanted consequences). In [BGSS06,BGSS07] we have considered a different quality dimension: *completeness*. Given an application domain and a DL knowledge base describing it, we have developed a method that supports the knowledge engineer in checking whether the knowledge base contains all the relevant information about the domain, namely:

– Are all the relevant constraints that hold between concepts in the domain captured by the TBox?

* Supported by DFG (BA 1122/12-1)

– Are all the relevant individuals existing in the domain represented in the ABox?

Such questions cannot be answered by an automated tool alone. Clearly, to check whether a given relationship between concepts—which does not follow from the TBox—holds in the domain, one needs to ask a domain expert, and the same is true for questions regarding the existence of individuals not described in the ABox.

In the aforementioned work we have developed a method for supporting the ontology engineer in checking whether an ontology contains all the relevant information about the application domain, and extending the ontology appropriately if this is not the case. The method achieves this by asking the ontology engineer questions of the form: "*is it true that instances of the concepts* $C_1, \ldots, C_n$ *are also instances of the concepts* $D_1, \ldots, D_m$*?*" When such a question is asked, the expert is expected to either confirm or reject it. If she confirms the question, a new axiom of the application domain that does not follow from the knowledge base has been found, and it is added to the TBox. Otherwise, the ontology engineer is asked to provide a counterexample to this question that will be added to the ABox. Once all such questions are answered this way, the knowledge base will be complete in a certain sense. The method is usable with any DL as long as it allows for conjunction and negation, and the TBox formalism allows for GCIs. The approach is based on *attribute exploration* [Gan84], which is a novel knowledge acquisition method developed in Formal Concept Analysis (FCA) [GW99] to acquire knowledge about an application domain by querying an expert. The use of attribute exploration ensures that, on the one hand, during knowledge base completion the interaction with the expert is kept to a minimum, and, on the other hand, the extended knowledge base is complete (w.r.t. a fixed interpretation) in a certain, well-defined sense.

Our first experiments with a prototype implementation of the method showed that during completion, the ontology engineer sometimes by mistake confirms a wrong question, in which case an axiom that actually does not hold in the application domain is added to the TBox. If she does not notice it immediately, as a result the completed knowledge base will have unwanted consequences. In the present work we consider the problem of pinpointing the axioms added during completion, which lead to certain unwanted consequences. As in [BPS07] we assume that the TBox consists of a static part that contains axioms whose correctness is undoubted, and a refutable part that contains axioms whose correctness is not yet for sure. In our setting, the static part is the initial knowledge base before completion, and the refutable part is the set of axioms added during completion. When we want to detect the axioms that have a certain unwanted consequence, we consider only the refutable part of the TBox. We call a subset of the refutable part that has a certain consequence, an *explanation* of this consequence. One important point in our setting that differs from [BPS07] is that, the axioms in the refutable part of the TBox are not arbitrary Horn axioms. They have a particular form that results in a canonical base called the Duquenne Guigues Base [GD86]. We show here that in the Horn case, despite the restricted form of the axioms in the refutable part, the problem of checking the existence of an explanation within a specified cardinality bound still remains NP-complete. We also show that the problem of determining the number of inclusionwise minimal explanations is #P-complete. In Section 2 we briefly recall the basic notions of knowledge base completion, and in Section 3 we show our main results. We conclude with Section 4 where we describe future work.

## 2 Knowledge base completion

We describe the knowledge base completion algorithm very briefly without going into any technical detail. For technical details, the reader is referred to the technical report [BGSS06]. Before we start describing the knowledge base completion algorithm, let us first introduce some basic notions.

**Definition 1.** *Let $M$ be a set of concept descriptions and $L, R \subseteq M$. We say that the* implication $L \rightarrow R$ *is* refuted *by $(\mathcal{T}, \mathcal{A})$ if there is an individual name $a$ occurring in $\mathcal{A}$ such that $\mathcal{T}, \mathcal{A} \models C(a)$ for all $C \in L$ and $\mathcal{T}, \mathcal{A} \models \neg D(a)$ for some $D \in R$. Similarly, $L \rightarrow R$ is refuted by the interpretation $\mathcal{I}$ if there is an element $d \in \Delta^{\mathcal{I}}$ such that $d \in C^{\mathcal{I}}$ for all $C \in L$ and $d \notin D^{\mathcal{I}}$ for some $D \in R$. If an implication is not refuted by $\mathcal{I}$, then we say that it* holds *in $\mathcal{I}$. In addition, we say that $L \rightarrow R$* follows *from $\mathcal{T}$ if $\sqcap L \sqsubseteq_{\mathcal{T}} \sqcap R$, where $\sqcap L$ and $\sqcap R$ respectively stand for the conjunctions $\bigsqcap_{C \in L} C$ and $\bigsqcap_{D \in R} D$.*

We are now ready to define what we mean by completing a DL knowledge base. Intuitively, a knowledge base is supposed to describe an intended model. For a fixed set $M$ of "interesting" concepts, the knowledge base is complete if it contains all the relevant knowledge about implications between these concepts. To be more precise, if an implication holds in the intended interpretation, then it should follow from the TBox, and if it does not hold in the intended interpretation, then the ABox should contain a counterexample. Based on the notions introduced above this can formally be defined as follows.

**Definition 2.** *Let $(\mathcal{T}, \mathcal{A})$ be a DL knowledge base, $M$ a finite set of concept descriptions, and $\mathcal{I}$ a model of $(\mathcal{T}, \mathcal{A})$. Then $(\mathcal{T}, \mathcal{A})$ is* $M$-complete *(or simply* complete *if $M$ is clear from the context)* w.r.t. $\mathcal{I}$ *if the following three statements are equivalent for all implications $L \rightarrow R$ over $M$:*

1. *$L \rightarrow R$ holds in $\mathcal{I}$;*
2. *$L \rightarrow R$ follows from $\mathcal{T}$;*
3. *$L \rightarrow R$ is not refuted by $(\mathcal{T}, \mathcal{A})$.*

In order to rephrase the definition of completeness, let us say that the element $d \in \Delta^{\mathcal{I}}$ of an interpretation $\mathcal{I}$ *satisfies* the subsumption statement $C \sqsubseteq D$ if $d \notin C^{\mathcal{I}}$ or $d \in D^{\mathcal{I}}$, and that $\mathcal{I}$ *satisfies* this statement if every element of $\Delta^{\mathcal{I}}$ satisfies it. In addition, let us call the individual name $a$ a *counterexample in* $(\mathcal{T}, \mathcal{A})$ to the subsumption statement $C \sqsubseteq D$ if $\mathcal{T}, \mathcal{A} \models C(a)$ and $\mathcal{T}, \mathcal{A} \models \neg D(a)$.

**Lemma 1.** *The knowledge base $(\mathcal{T}, \mathcal{A})$ is complete w.r.t. its model $\mathcal{I}$ iff the following statements are equivalent for all subsets $L, R$ of $M$:*

1. $\sqcap L \sqsubseteq \sqcap R$ *is satisfied by* $\mathcal{I}$*;*
2. $\sqcap L \sqsubseteq_{\mathcal{T}} \sqcap R$ *holds;*
3. $(\mathcal{T}, \mathcal{A})$ *does not contain a counterexample to* $\sqcap L \sqsubseteq \sqcap R$*.*

As we have mentioned before, our knowledge base completion algorithm is an adaptation of the attribute exploration method of FCA, extended for completing a given knowledge base $(\mathcal{T}, \mathcal{A})$ w.r.t. a fixed model $\mathcal{I}$ of this knowledge base. The method successively produces implications that do not follow from $\mathcal{T}$, and that are at the same time not refuted by $\mathcal{A}$. It then asks the domain expert if this implication holds in $\mathcal{I}$. The questions are of the form: "*Is* $L \rightarrow R$ *refuted by* $\mathcal{I}$*?*". We assume that the domain expert has enough information about $\mathcal{I}$ to be able to answer such questions. If the answer is "no," then the GCI $\sqcap L \sqsubseteq \sqcap R$ is added to $\mathcal{T}$. Since $L \rightarrow R$ is not refuted by $\mathcal{I}$, the interpretation $\mathcal{I}$ is still a model of the new TBox obtained this way. If the answer is "yes," then the expert must extend $\mathcal{A}$ (by adding concept assertions) such that the extended ABox refutes $L \rightarrow R$ and $\mathcal{I}$ is still a model of this ABox. This procedure is repeated until all such questions are answered. Once all such questions are answered, the resulting knowledge base will be complete w.r.t. $\mathcal{I}$ in the sense that is introduced in Definition 2.

One important point here is that the algorithm actually does not naively enumerate all implications that do not follow from $\mathcal{T}$ and at the same time are not refuted by $\mathcal{A}$. This would mean too many and unnecessary questions to the expert. It produces implications in a specific lexicographic order such that the interaction with the expert is kept to minimum. More precisely, the algorithm asks the expert the minimum number of implication questions that result in a new GCI in the TBox. In FCA terminology, this set of implications is called the Duquenne Guigues Base [GD86]. It is well known that among all other sets of implications that have the same consequences, Duquenne Guigues Base contains the smallest number of implications. That is, no set of implications with smaller cardinality can have the same set of consequences as the Duquenne Guigues Base. The implications of Duquenne Guigues Base, thus the GCIs added to the TBox during completion, have the following property which is going to be used in Section 3.

**Lemma 2.** *Let $\mathcal{L}$ be a Duquenne Guigues Base. Every implication $L \rightarrow R$ in $\mathcal{L}$ has the following properties:*

- *$L$ is closed under the implications $\mathcal{L} \setminus \{L \rightarrow R\}$, i.e., $L = \mathcal{L}'(L)$ where $\mathcal{L}'$ is $\mathcal{L} \setminus \{L \rightarrow R\}$ and $\mathcal{L}(\cdot)$ denotes implicational closure,*
- *$L \cup R$ is closed under the implications $\mathcal{L} \setminus \{L \rightarrow R\}$, i.e., $L \cup R = \mathcal{L}'(L \cup R)$.*

Based on the results in [BGSS06,BGSS07], we have implemented a first experimental version of a knowledge base completion tool called InstExp [1], which

[1] available under `http://lat.inf.tu-dresden.de/~sertkaya/InstExp`

stands for INSTance EXPlorer. It is written in the Java programming language as an extension to version v2.3 beta 3 of the Swoop ontology editor [KPS+06] and communicates with the reasoner over the OWL API [BVL03].

Our first experiments with INSTEXP showed that during completion, unsurprisingly the expert sometimes makes errors when answering the questions. In the simplest case, the error makes the knowledge base inconsistent, which can easily be detected by DL reasoning and the expert can be notified about it. However, in this case an explanation for the reason of inconsistency is often needed to understand and fix the error. The situation gets more complicated if the error does not immediately lead to inconsistency, but the expert realizes in the later steps, or only after completion that she has accepted a wrong axiom in one of the previous steps. In this case, the completed knowledge base will have unwanted consequences. In the next section we are going to investigate axiom pinpointing in the knowledge base completion setting. We are going to look for methods for explaining user errors introduced to the TBox during knowledge base completion.

## 3 Explaining user errors in knowledge base completion

In [BPS07] Baader et. al. have considered axiom pinpointing in a setting where the TBox consists of two kinds of axioms, namely the trusted ones whose correctness is no longer doubted, and the refutable ones whose correctness is not yet for sure. Trusted axioms form so-called the *static* part of the TBox, and others form the *refutable* part. The static part of the TBox is assumed to be always present, and axioms explaining a certain consequence are searched only in the refutable part of the TBox. In our knowledge base completion scenario we have a similar situation. We assume that the axioms in the initial TBox, which we have at the beginning of completion, are trusted i.e., they have no unwanted consequences. However, as mentioned above, during completion the user sometimes wrongly accepts an axiom into the TBox. As a result the axioms added during completion can lead to unwanted consequences. Therefore we consider them as refutable axioms. When we need to find axioms responsible for a certain consequence, we only look at the axioms added during completion.

One important point here that differs from [BPS07] is that, the axioms added during completion are not arbitrary axioms. They form a Duquenne Guigues Base, thus they have a specific form satisfying the property in Lemma 2. At this point, one might think that in our setting we have background knowledge in the initial TBox, thus as shown in [Stu96], the set of axioms resulting from the completion process will not form a Duquenne Guigues Base. However, this is not true. We do not use this background knowledge in the way mentioned in [Stu96], i.e., we do not make use of the axioms already existing in the initial TBox for generating the implication questions. We make use of them only when we want to answer the implication questions, i.e., whenever a new implication question is asked, we first check if it already follows from the TBox. Thus, the resulting set of axioms will indeed be a Duquenne Guigues Base.

In [BPS07] it was shown that in the Horn case, i.e., both the left and right handsides of the axioms only consist of conjunctions of concept names, a given axiom can have exponentially many minimal explanations, i.e., minimal subsets of the refutable part of the given TBox that have the given axiom as consequence. The following example shows that this is also the case if the refutable part of the TBox is restricted to have the form of a Duquenne Guigues Base, i.e., if it satisfies the property in Lemma 2.

*Example 1.* Consider the TBox

$$\mathcal{T} := \{X \sqcap B_{i-1} \sqsubseteq P_i \sqcap Q_i, Y \sqcap P_i \sqsubseteq B_i, Y \sqcap Q_i \sqsubseteq B_i \mid 1 \leq i \leq n\}.$$

Assume that all axioms in $\mathcal{T}$ are refutable. It is not difficult to see that none of the left handsides is contained in another left handside or in the union of left and right handsides of another axiom, i.e., it obeys the property mentioned in Lemma 2. Moreover its size is linear in $n$, and it has $2^n$ minimal subsets that explain the axiom $B_0 \sqcap X \sqcap Y \sqsubseteq B_n$ since for each $i$, $1 \leq i \leq n$, $B_i$ can be generated by the axiom $Y \sqcap P_i \sqsubseteq B_i$ or by $Y \sqcap Q_i \sqsubseteq B_i$.

Apart from the example showing that there can be exponentially many minimal explanations, in [BPS07] Baader et. al. have shown that even in the Horn case, the problem of checking the existence of a minimal explanation within a specified cardinality bound is NP-complete. Here we show that the problem still remains NP-complete despite the restricted form of the axioms in the refutable part of the TBox. Let us first formally define our problem. In the following, for a set of concept names $L$, $\bigsqcap L$ denotes the conjunction $\bigsqcap_{C \in L} C$.

**Problem:** MINIMUM CARDINALITY EXPLANATION
*Input:* A Horn TBox $\mathcal{T}$ satisfying the properties in Lemma 2, sets $L$ and $R$ of concept names occurring in $\mathcal{T}$ such that $\bigsqcap L \sqsubseteq_{\mathcal{T}} \bigsqcap R$, a natural number $n$.
*Question:* Is there an explanation of $\bigsqcap L \sqsubseteq \bigsqcap R$ in $\mathcal{T}$ with cardinality less than or equal to $n$, i.e., is there a set of axioms $\mathcal{T}' \subseteq \mathcal{T}$ such that $\bigsqcap L \sqsubseteq_{\mathcal{T}'} \bigsqcap R$ and $|\mathcal{T}'| \leq n$?

**Theorem 1.** MINIMUM CARDINALITY EXPLANATION *is* NP-*complete.*

*Proof.* The problem is clearly in NP. We can nondeterministically guess a subset $\mathcal{T}'$ of $\mathcal{T}$ with cardinality $n$, and in polynomial time check whether $\bigsqcap L \sqsubseteq_{\mathcal{T}'} \bigsqcap R$.

In order to show NP-hardness, we are going to give a reduction from the NP-complete problem HITTING SET [GJ90], which is defined as follows:

**Problem:** HITTING SET
*Input:* A collection $S_1, \ldots, S_k$ of subsets of a finite set $S$, a natural number $n$.
*Question:* Is there a subset $S' \subseteq S$ with $|S'| \leq n$ such that $S'$ contains at least one element from each $S_i$, i.e., $S' \cap S_i \neq \emptyset$ for $1 \leq i \leq k$.

Consider an instance of the HITTING SET problem. We denote elements of $S_i$ with concept names $P_{i1}, \ldots, P_{i\ell_i}$, and introduce additional fresh concept names

$A, B, X_1, \ldots, X_k, Q_1, \ldots, Q_k$, and $Y_p$ for $p \in S_1 \cup \ldots \cup S_k$. We construct the set of axioms

$$\begin{aligned}\mathcal{T} := &\{X_i \sqcap P_{ij} \sqsubseteq Q_i \mid 1 \leq i \leq k, 1 \leq j \leq \ell_i\} \cup \\ &\{A \sqcap Y_P \sqsubseteq P \mid P \in S_1 \cup \ldots \cup S_k\} \cup \\ &\{Q_1 \sqcap \ldots \sqcap Q_k \sqsubseteq B\}\end{aligned}$$

and the axiom $A \sqcap X_1 \sqcap \ldots \sqcap X_k \sqcap \bigsqcap_{P \in S_1 \cup \ldots S_k} Y_P \sqsubseteq B$ that follows from $\mathcal{T}$. Note that none of the axioms in $\mathcal{T}$ contains the left handside of another axiom in its left handside or in the union of its left and right handsides. That is, the axioms in $\mathcal{T}$ satisfy the property mentioned in Lemma 2. Obviously, this construction can be done in polynomial time.

We claim that $S_1, \ldots, S_k$ has a hitting set of cardinality less than or equal to $n$ iff the axiom $A \sqcap X_1 \sqcap \ldots \sqcap X_k \sqcap \bigsqcap_{P \in S_1 \cup \ldots S_k} Y_P \sqsubseteq B$ has a minimal explanation in $\mathcal{T}$ with cardinality less than or equal to $n + k + 1$. Assume $S_1, \ldots, S_k$ has a hitting set $S'$ such that $|S'| \leq n$. Then it is not difficult to see that the following subset of $\mathcal{T}$ constructed by using $S'$ is a minimal explanation:

$$\mathcal{T}' := \{A \sqcap Y_P \sqsubseteq P \mid P \in S'\} \cup \{X_i \sqcap P_{ij} \sqsubseteq Q_i \mid P_{ij} \in S'\} \cup \{Q_1 \sqcap \ldots \sqcap Q_k \sqsubseteq B\}$$

Indeed, the first set of axioms derive the concept names $P \in S'$. Since $S'$ is a hitting set, at least one such $P$ is derived for each $1 \leq i \leq k$. Thus the second set of axioms derive the concept names $Q_i$ for each $1 \leq i \leq k$. Finally, using the only axiom in the last set, such $Q_i$ altogether derive the concept name $B$. Note that $\mathcal{T}'$ is a minimal explanation and contains exactly $n + k + 1$ axioms. The other direction of the claim is shown easily in the similar way. □

In applications where one is interested in all explanations that are minimal w.r.t. set inclusion, it might be useful to know in advance how many of them exist. Next we consider this counting problem. It turns out that it is hard for the counting complexity class #P [Val79a], i.e., it is intractable. Let us first formally define the problem.

**Problem:** #MINIMAL EXPLANATION
*Input:* A Horn TBox $\mathcal{T}$ satisfying the properties in Lemma 2, sets $L$ and $R$ of concept names occurring in $\mathcal{T}$ such that $\bigsqcap L \sqsubseteq_{\mathcal{T}} \bigsqcap R$.
*Output:* Number of all minimal explanations of $\bigsqcap L \sqsubseteq \bigsqcap R$ in $\mathcal{T}$, i.e.,
$|\{\mathcal{T}' \subseteq \mathcal{T} \mid \bigsqcap L \sqsubseteq_{\mathcal{T}'} \bigsqcap R \text{ and } \forall \mathcal{T}'' \subsetneq \mathcal{T}'. \bigsqcap L \not\sqsubseteq_{\mathcal{T}''} \bigsqcap R\}|$ .

**Theorem 2.** #MINIMAL EXPLANATION *is* #P*-complete.*

*Proof.* The problem is clearly in #P. It can be in polynomial time verified that a given $\mathcal{T}' \subseteq \mathcal{T}$ is an explanation, and it is minimal w.r.t. set inclusion.

For showing #P-hardness, we are going to use the same construction used in the proof of Theorem 1. It is common folklore that the problem of counting hitting sets is #P-complete, which can be easily shown by a parsimonious reduction from #MONOTONE 2-SAT. #MONOTONE 2-SAT is the problem of counting the models of a monotone Boolean formula in CNF with exactly 2 variables in each clause. It was shown to be #P-complete in [Val79b].

**Algorithm 1** Computing one minimal explanation

**Input:** The set of axioms $\mathcal{T}$ obtained from completion, and sets of concept names $L$ and $R$ s.t. $\bigsqcap L \sqsubseteq_{\mathcal{T}} \bigsqcap R$.
$\mathcal{T}' := \mathcal{T}$
**for all** $t \in \mathcal{T}'$ **do**
  **if** $\bigsqcap L \sqsubseteq_{\mathcal{T}' \setminus \{t\}} \bigsqcap R$ **then** {if $\mathcal{T}' \setminus \{t\}$ is still an explanation}
    $\mathcal{T}' := \mathcal{T}' \setminus \{t\}$
  **end if**
**end for**
**return** $\mathcal{T}'$

Our construction in the proof of Theorem 1 maps each hitting set to exactly one minimal explanation, and each minimal explanation to exactly one hitting set. That is, it establishes a bijection between hitting sets and minimal explanations. Thus it preserves the number of solutions, i.e., it is parsimonious. Since the problem of counting hitting sets is #P-complete, #MINIMAL EXPLANATION is also #P-complete. □

Despite these negative results, it is not difficult to find one minimal explanation with at most polynomially many subsumption tests. We can just start with the whole set of axioms obtained from the completion process, iterate over these axioms and eliminate an axiom if the remaining ones still have the consequence in question. It is formally described in Algorithm 1.

## 4 Concluding remarks

In [BPS07] it was shown that given a set of minimal explanations, the problem of checking whether there exists a minimal explanation that is not contained in the given set is NP-complete. This means that, the set of all minimal explanations cannot be computed in output polynomial time [JPY88]. We do not know whether this is also the case in our setting for axioms with restricted form. As future work we are going to consider this problem of computing all minimal explanations in the knowledge base completion setting.

Alternatively, our results can be obtained from the results on functional dependencies in relational databases [Mai83]. In relational databases, it is known that, obtaining a minimum (w.r.t. cardinality) cover from a given set of functional dependencies $F$ can be done in time polynomial in the size of $F$ [Mai80]. A corresponding result in the FCA setting has been mentioned in [Rud07]. Using these results and the results of [BPS07], one can also obtain our results here.

## References

[BGSS06] F. Baader, B. Ganter, U. Sattler, and B. Sertkaya. Completing Description Logic Knowledge Bases using Formal Concept Analysis. LTCS-Report

LTCS-06-02, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2006. See http://lat.inf.tu-dresden.de/research/reports.html.

[BGSS07] F. Baader, B. Ganter, B. Sertkaya, and U. Sattler. Completing Description Logic Knowledge Bases using Formal Concept Analysis. In M. M. Veloso, ed., *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)*, pp. 230–235. AAAI Press, 2007.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[BPS07] F. Baader, R. Peñaloza, and B. Suntisrivaraporn. Pinpointing in the Description Logic $\mathcal{EL}^+$. In J. Hertzberg, M. Beetz, and R. Englert, eds., *Proceedings of the 30th German Conference on Artificial Intelligence (KI2007)*, volume 4667 of *Lecture Notes in Artificial Intelligence*, pp. 52–67. Springer-Verlag, 2007.

[BVL03] S. Bechhofer, R. Volz, and P. W. Lord. Cooking the Semantic Web with the OWL API. In D. Fensel, K. P. Sycara, and J. Mylopoulos, eds., *Proceedings of the Second International Semantic Web Conference, (ISWC 2003)*, volume 2870 of *Lecture Notes in Computer Science*, pp. 659–675. Springer-Verlag, 2003.

[Gan84] B. Ganter. Two Basic Algorithms in Concept Analysis. Technical Report Preprint-Nr. 831, Technische Hochschule Darmstadt, Darmstadt, Germany, 1984.

[GD86] J.-L. Guigues and V. Duquenne. Familles minimales d'implications informatives resultant d'un tableau de données binaries. *Mathématiques, Informatique et Sciences Humaines*, 95:5–18, 1986.

[GJ90] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Company, New York, NY, USA, 1990. ISBN 0716710455.

[GW99] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, Berlin, Germany, 1999.

[HPSvH03] 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.

[JPY88] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. On Generating All Maximal Independent Sets. *Information Processing Letters*, 27(3):119–123, 1988.

[KFNM04] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In S. A. McIlraith, D. Plexousakis, and F. van Harmelen, eds., *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pp. 229–243. Springer-Verlag, 2004.

[KPS+06] A. Kalyanpur, B. Parsia, E. Sirin, B. C. Grau, and J. A. Hendler. Swoop: A Web Ontology Editing Browser. *Journal of Web Semantics*, 4(2):144–153, 2006.

[KPSG06] A. Kalyanpur, B. Parsia, E. Sirin, and B. C. Grau. Repairing Unsatisfiable Concepts in OWL Ontologies. In Y. Sure and J. Domingue, eds., *The Semantic Web: Research and Applications. Proceedings of the 3rd European Semantic Web Conference (ESWC 2006)*, volume 4011 of *Lecture Notes in Computer Science*, pp. 170–184. Springer-Verlag, 2006.

[Mai80] D. Maier. Minimum Covers in Relational Database Model. *Journal of the ACM*, 27(4):664–674, 1980.

[Mai83] D. Maier. *The Theory of Relational Databases*. Computer Science Press, Maryland, 1983.

[OVSM04] D. Oberle, R. Volz, S. Staab, and B. Motik. An Extensible Ontology Software Environment. In S. Staab and R. Studer, eds., *Handbook on Ontologies*, International Handbooks on Information Systems, pp. 299–320. Springer-Verlag, 2004.

[Rud07] S. Rudolph. Some Notes on Pseudo-closed Sets. In S. O. Kuznetsov and S. Schmidt, eds., *Proceedings of the 5th International Conference on Formal Concept Analysis, (ICFCA 2007)*, volume 4390 of *Lecture Notes in Computer Science*, pp. 151–165. Springer-Verlag, 2007.

[SC03] S. Schlobach and R. Cornet. Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. In G. Gottlob and T. Walsh, eds., *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*, pp. 355–362. Morgan Kaufmann, 2003.

[Stu96] G. Stumme. Attribute Exploration with Background Implications and Exceptions. In H.-H. Bock and W. Polasek, eds., *Data Analysis and Information Systems. Statistical and Conceptual approaches. Proceedings of GfKl'95. Studies in Classification, Data Analysis, and Knowledge Organization 7*, pp. 457–469. Springer-Verlag, 1996.

[Val79a] L. G. Valiant. The Complexity of Computing the Permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[Val79b] L. G. Valiant. The Complexity of Enumeration and Reliability Problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.