Federated Knowledge Base Debugging in *DL-Lite*$_{\mathcal{A}}$

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Andreas Karl Nolle
aus Albstadt

Mannheim, 2021

Dekan: Dr. Bernd Lübcke, Universität Mannheim
Referent: Prof. Dr. Heiner Stuckenschmidt, Universität Mannheim
Korreferent: Prof. Dr. Stefan Schlobach, Vrije Universiteit Amsterdam, Netherlands

Tag der mündlichen Prüfung: 02.08.2021

To my family

Acknowledgement

At the early beginning Heiner emphasized that it will be challenging as an affiliated PhD student. What can I say – well, he was right. All the more I would like to thank Heiner, Christian and Mel for their support and especially their patience. A special thanks to Christian, a brilliant mentor and for me one of the best postdocs.

I am also very grateful to my wife Thuy who did and still do a great job and gave me the best possible support during all the time.

Many thanks to German for bringing me back to university and inducting me into the world of research. I would also like to extend my thanks to Stefan and Thomas for any kind of their assistance.

Last but not least, thanks to all of my colleagues and friends who supported me along the way and gave me the necessary distraction, occasionally also with some cold drinks.

Abstract

Due to the continuously growing amount of data the federation of different and distributed data sources gained increasing attention. In order to tackle the challenge of federating heterogeneous sources a variety of approaches has been proposed. Especially in the context of the Semantic Web the application of Description Logics is one of the preferred methods to model federated knowledge based on a well-defined syntax and semantics. However, the more data are available from heterogeneous sources, the higher the risk is of inconsistency – a serious obstacle for performing reasoning tasks and query answering over a federated knowledge base. Given a single knowledge base the process of knowledge base debugging comprising the identification and resolution of conflicting statements have been widely studied while the consideration of federated settings integrating a network of loosely coupled data sources (such as LOD sources) has mostly been neglected.

In this thesis we tackle the challenging problem of debugging federated knowledge bases and focus on a lightweight Description Logic language, called *DL-Lite*$_{\mathcal{A}}$, that is aimed at applications requiring efficient and scalable reasoning. After introducing formal foundations such as Description Logics and Semantic Web technologies we clarify the motivating context of this work and discuss the general problem of information integration based on Description Logics.

The main part of this thesis is subdivided into three subjects. First, we discuss the specific characteristics of federated knowledge bases and provide an appropriate approach for detecting and explaining contradictive statements in a federated *DL-Lite*$_{\mathcal{A}}$ knowledge base. Second, we study the representation of the identified conflicts and their relationships as a conflict graph and propose an approach for repair generation based on majority voting and statistical evidences. Third, in order to provide an alternative way for handling inconsistency in federated *DL-Lite*$_{\mathcal{A}}$ knowledge bases we propose an automated approach for assessing adequate trust values (i.e., probabilities) at different levels of granularity by leveraging probabilistic inference over a graphical model.

In the last part of this thesis, we evaluate the previously developed algorithms against a set of large distributed LOD sources. In the course of discussing the experimental results, it turns out that the proposed approaches are sufficient, efficient and scalable with respect to real-world scenarios. Moreover, due to the exploitation of the federated structure in our algorithms it further becomes apparent that the number of identified wrong statements, the quality of the generated repair as well as the fineness of the assessed trust values profit from an increasing number of integrated sources.

Zusammenfassung

Mit stetig zunehmender Menge an verfügbarer Daten wächst zunehmend auch der Bedarf unterschiedliche und verteilte Datenquellen zusammenzuschließen. Um die besondere Hausforderung der Föderation unterschiedlicher und heterogener Datenquellen anzugehen existiert eine Vielzahl verschiedenster Ansätze. Zur Modellierung föderierten Wissens basierend auf einer wohldefinierten Syntax und Semantik sind insbesondere im Kontext des Semantic Web Beschreibungslogiken eine präferierte Methode. Je mehr Daten allerdings aus heterogenen Quellen vorhanden sind, desto höher ist dann auch das Risiko einer Inkonsistenz – eine erhebliche Beeinträchtigung bei der Durchführung von Schlussfolgerungen (engl. Reasoning) sowie der Beantwortung von Abfragen in einer föderierten Wissensbasis. Der Prozess zum Debuggen einer einzelnen Wissensbasis, welcher die Identifizierung sowie die Aufhebung widersprüchlicher Aussagen umfasst, ist bereits umfassend erforscht, wohingegen föderative Gegebenheiten, sprich die Integration eines Netzwerks lose gekoppelter Datenquellen, bisher weitestgehend vernachlässigt wurden.

Im Rahmen der vorliegenden Arbeit nehmen wir exakt dieses anspruchsvolle Problem des Debuggens föderierter Wissensbasen in Angriff und konzentieren uns dabei auf *DL-Lite*$_{\mathcal{A}}$, eine weniger ausdrucksstarke Beschreibungslogik, welche insbesondere Anforderungen hinsichtlich eines effizienten und skalierbaren Reasonings gerecht wird. Nach einer Einführung in die formalen Grundlagen von Beschreibungslogiken und Semantic Web-Technologien beleuchten wir den als Motivation dienenden Kontext dieser Arbeit und diskutieren die allgemeine Problemstellung einer auf Beschreibungslogik basierten Informationsintegration.

Der daran anschließende Hauptteil gliedert sich in drei Bereiche. Zunächst diskutieren wir die spezifischen Merkmale föderierter Wissensbasen und bieten einen geeigneten Ansatz zum Erkennen und Begründen widersprüchlicher Aussagen in einer föderierten *DL-Lite*$_{\mathcal{A}}$-Wissensbasis. Im zweiten Themenfeld untersuchen wir die Repräsentation der identifizierten Konflikte und deren Beziehungsrelationen als Konfliktgraph und erörtern einen auf Mehrheitsentscheidungen und statistischen Evidenzen basierenden Ansatz zur Generierung einer geeigneten Reparatur. Als alternative Möglichkeit zum Umgang mit Inkonsistenzen in föderierten *DL-Lite*$_{\mathcal{A}}$-Wissensbasen erarbeiten wir im darauffolgenden Teil einen automatisierten Ansatz zur Bestimmumg adäquater Vertrauenswerte (sprich Wahrscheinlichkeiten) auf unterschiedlichen Granularitätsstufen mittels probabilistischer Inferenz in einem grafischen Modell.

Der letzte Teil umfasst eine experimentellen Evaluierung der zuvor entwickelten Algorithmen mittels eines Datensatzes bestehend aus einer Menge großer verteilter LOD-Datenquellen. Bei der Diskussion der Evaluationsergebnisse wird sich zeigen, dass die Ansätze aus dieser Arbeit hinsichtlich praxisbezogener Anwendungsfälle hinreichend, effizient und sklarierbar sind. Darüber hinaus wird zudem deutlich, dass durch Exploitieren der föderierten Struktur in unseren Algorithmen sich eine zunehmende Anzahl integrierter Datenquellen positiv auf die Identifizierung falscher Aussagen, die Qualität der generierten Reparatur, sowie auch auf die Güte der bemessenen Vertrauenswerte auswirkt.

Contents

Listings

List of Figures

List of Tables

List of Algorithms

List of Definitions

List of Examples

Part I

Prolegomenon

Chapter 1

Introduction

In this chapter we are discussing the motivation of this work in Section 1.1 before we define in Section 1.2 the covered research questions and outline the contents and contributions in Section 1.3.

1.1 Inconsistency in Federated Knowledge Bases

Information explosion leads to continuous growth of data distributed over different data sources. Especially the fact that data is often distributed in numerous independent sources explains the increasing interest on data source federation. Notably in the context of Semantic Web the amount of data published in the Linked Open Data cloud is growing continuously and thus opens new challenges in data and information integration [SH05; McC]. Additionally the use of different schemes makes the task of federating several data sources a difficult problem. In the last years Description Logics have been applied increasingly as a conceptual view facilitating the federation of numerous data sources using different access methods and data schemes. In this case, the conceptual view is defined by a central schema that comprises and extends the semantics of each integrated data source. Consequently, each data source is treated as a single knowledge base that is integrated in a federated knowledge base representing an interface to the distributed data. Approaches such as ontology-based data integration [Art+09; OŠ12; Cal+13] or ontology-based information integration [Wac+01] are aimed at this purpose. According to these approaches, queries formulated according to the central schema describing the knowledge domain as a whole are translated into queries referring to the related schema of each data source. However, in such integrative environments the increasing number of heterogeneous data sources increases the risk of inconsistency. Especially the federation of various data sources implies typically the amalgamation of ambiguous and possibly conflicting information and hence often leads to inconsistency (contradictory assertions) – a serious obstacle for leveraging the full potential of federative approaches. Generally, the process of identifying and resolving conflicts in Description Logic knowledge bases is called knowledge

base debugging (or ontology debugging) [SC03]. However, as previous approaches mostly have been developed for processing of single or locally available knowledge bases the objective of this thesis is to tackle the novel problem of debugging federated knowledge bases.

1.2 Research Questions

In order to address the topic of federated knowledge base debugging we propose successive and interrelated approaches for the identification and explanation of logical conflicts as well as the resolution (repair) and more generally the treatment of inconsistency in context of federated knowledge bases. In particular, we are covering the following research questions:

Q1: How can we formally describe the problem of inconsistency management in federated knowledge bases and what are its peculiarities?

Q2: How can the process of debugging federated knowledge bases be designed in a convenient, efficient, and eligible way?

Q3: Can the trustworthiness of individual assertions with respect to certain data sources be assessed based on the debugging results?

Q4: Is it possible to automatically transform an inconsistent federated knowledge base into a probabilistic one?

Q5: How do the proposed approaches perform in practice concerning runtime, scalability and quality?

Q6: What are the impacts of adding additional data sources into a federated knowledge base with respect to the quality of the debugging results?

1.3 Dissertation Outline and Contributions

After introducing formal foundations of Description Logics and Semantic Web technologies in Chapter 2, in the last Chapter 3 of Part I we are discussing the problem statement addressed by this work and introduce a formal definition of federated knowledge bases which can already be regarded as a first partial answer to research question Q1. Subsequently, we are tackling the debugging of federated knowledge bases in the following Part II: Theory and Methods. In particular, Part II is structured into three chapters where the addressed objectives of each chapter are as follows.

Federated Inconsistency Detection and Explanations

Compared to single knowledge bases, reasoning in a federated setting integrating a loosely coupled network of data sources becomes a challenging problem. In order to answer research question Q1, in Chapter 4 we analyze the peculiarities of federated knowledge bases. Moreover, due to the identified characteristics we justify our restriction to *DL-Lite*$_{\mathcal{A}}$ and propose an appropriate approach for detecting and explaining inconsistency.

Repair Plan Generation

In order to provide an appropriate repair for the identified conflicts constituting a complex network of correlated assertions, in Chapter 5 we develop an approach for the repair generation. By relying on majority voting and the use of statistical evidences we are able to exploit the characteristics of a federated knowledge base for improving the quality of the generated repair. The gained insights of this chapter coupled with the approach of Chapter 4 contribute to answer research question Q2.

Fine-grained Trust Assessment

As repairing an inconsistent knowledge base typically implies loss of information, in Chapter 6 we approach to provide an alternative strategy for handling inconsistency in federated knowledge bases. Based on the formalism of Markov networks we propose an algorithm for an automated assessment of adequate trust values (i.e., probabilities) at different levels of granularity and thus provide an answer to research question Q3 and Q4.

Subsequently, in order to empirically verify the practical application and to tackle research question Q5 and Q6 we evaluate in Part III each of the previously developed approaches against a set of large distributed LOD sources from the domain of library science.

The final Part IV completes this thesis by summarizing and critically discussing the main results of the proposed approaches.

As we have already published some parts of this thesis, we will explain in the introductory part of each chapter the origin of the respective contents and explicitly emphasize new contents and contributions. In order to present the entire work simply at one stretch by focusing the same unique keynote, we forgo to elucidate the origin of each single thought, definition, proof, example, graphics or similar parts.

Chapter 2

Description Logics and Semantic Web

In this chapter we introduce Description Logics in Section 2.1 by describing the syntax (Section 2.1.1) and semantics (Section 2.1.2) of the expressive Description Logic language $\mathcal{SROIQ}(\mathbf{D})$. Moreover, we discuss standard reasoning tasks (Section 2.1.3) within the context of Description Logics and its computational complexities with respect to $\mathcal{SROIQ}(\mathbf{D})$. Even if expressive Description Logics (such as $\mathcal{SROIQ}(\mathbf{D})$) are of theoretical interest, for many practical applications it is sufficient or at least desirable to rely on a Description Logic language with less expressive power. Because of that, we subsequently elaborate in Section 2.2 a family of lightweight Description Logic languages, called *DL-Lite*, and in particular its member *DL-Lite*$_{\mathcal{A}}$, the language we focus on within the scope of this thesis. By specifying the syntax and semantics of (unions of) conjunctive queries over a Description Logic knowledge base, we address in Section 2.3 the reasoning task of query answering. Finally, we discuss in Section 2.4 the Semantic Web and its underlying technologies, especially the de facto standard knowledge modeling language OWL 2 (Section 2.4.1) and (a subset) of the standard Semantic Web query language SPARQL 1.1 (Section 2.4.2). Note that most of the definitions in this chapter can be found in sightly modified versions in numerous publications related to Description Logics or Semantic Web, and some selected publications are referred at suitable positions.

2.1 Description Logics

Originating from formalisms of mathematical logics, *Description Logics* (DLs) [Baa+10] are a family of languages for representing knowledge in a structured and concise way. Based on formal semantics, a precise specification of the meaning of the described model, the inference of additional (implicit) information respectively knowledge via logical deduction out of explicitly stated facts is enabled. The task of inferring logical consequences is commonly referred to as *reasoning* and the

computational complexity of a sound and complete reasoning depends on the expressiveness of the DL language $\mathcal{L}$ used to describe the model. While *sound* means that each inferred consequence is indeed correct, *complete* claims to guarantee that all correct inferences are really computed. In fact, most DLs can actually be seen as decidable fragments of First-Order Logic (FOL), if not even as part of the two-variable fragment of FOL [Bor96], or, in some cases, as a slight extension of it, e.g., by adopt counting quantifiers. Each concrete DL $\mathcal{L}$ is defined by a set of constructs by which individual statements (closed FOL formulas) can be created. To present a comprehensive definition of DL syntax and semantics we consider in the following one of the most expressive DL languages, i.e., $\mathcal{SROIQ}(\mathbf{D})$ [HKS06] as one particular DL representative.

2.1.1 Syntax

Like in any natural language, in DLs the knowledge of a domain of interest is formulated according to a countably infinite *signature* (also referred to as alphabet or vocabulary) and *syntax* (also known as grammar). Formally, the signature can be defined as follows:

Definition 2.1 (Signature)**.** *The signature Σ is given by*

$$\Sigma = \langle \Sigma_I, \Sigma_V, \Sigma_C, \Sigma_D, \Sigma_R, \Sigma_A \rangle \tag{2.1}$$

and comprises six pairwise disjoint and countably infinite sets of symbols: individual names Σ_I, data values Σ_V, concept names Σ_C, value-domain names Σ_D, role names Σ_R and attribute names Σ_A.

Individual names represent singular entities (objects) in the considered domain of interest whereas *concept names* denote types or classes of such entities. Similarly, *value-domain names* (or data type names) denote sets of values and each single value is represented by a *data value*. Binary relations between individuals are represented by *role names* whereas binary relations between individuals and data values are denoted by *attribute names*.

Using the signature, complex expressions of concepts, value-domains, roles and attributes can be constructed by applying suitable constructs provided by the DL. Particularly, in $\mathcal{SROIQ}(\mathbf{D})$ expressions on *attributes* can be constructed according to the following definition:

Definition 2.2 (Attribute Expressions)**.** *Given a signature Σ, an attribute A is defined by the syntax*

$$A ::= \top_A \mid \sigma_A \,, \tag{2.2}$$

where $\top_A$ is the universal attribute and $\sigma_A \in \Sigma_A$ is an attribute name.

The *universal attribute* $\top_A$ is a special attribute that relates all individuals with all data values.

Similar to attributes, a *role* is either the *universal role* (a special role that relates all individual pairs), a role name or the inverse of a role name.

Definition 2.3 (Role Expressions)**.** *Given a signature Σ, a role R can be described by expressions of the form*

$$R ::= \top_R \mid \sigma_R \mid \sigma_R^- \,, \tag{2.3}$$

where $\top_R$ is the universal role, $\sigma_R \in \Sigma_R$ a role name and σ_R^- its inverse.

In contrast to the few expressions on attributes and roles, $\mathcal{SROIQ}(\mathbf{D})$ provides a variety of constructs for describing value-domains and concepts. However, as in [HS01] we assume that the set of *value-domains* (data types) Σ_D is already sufficiently structured and defined by a type system such that a creation of new value-domains do not have to be considered by the expression syntax. In addition, this assumption allows the consideration of any arbitrary set of value-domains while keeping the DL language concise. For the sake of simplicity, we further assume within the scope of this thesis that all data types in Σ_D are pairwise disjoint, which is without loss of generality as shown by Motik and Horrocks [MH08]. Due to these assumptions, the (simplified) syntax for value-domain expressions can be defined as follows:

Definition 2.4 (Value-Domain Expressions)**.** *Given a signature Σ, the description of a value-domain D can be expressed by*

$$D ::= \top_D \mid \bot_D \mid \sigma_D \mid \neg D \mid \{\sigma_V, \dots\} \,, \tag{2.4}$$

where $\top_D$ is the universal value-domain, $\bot_D$ the bottom value-domain, $\sigma_D \in \Sigma_D$ a value-domain name, $\neg D$ the negation of value-domain D and $\sigma_V, \dots \in \Sigma_V$ are data values.

Similar as for attributes and roles, the *universal value-domain* $\top_D$ comprises all data values, whereas $\bot_D$ denotes the *bottom value-domain* representing the empty value-domain (with no data values). The *negation* (or complement) of a value-domain D is denoted by $\neg D$ and represents the set of all data values that do not belong to the value-domain D. Finally, a value-domain can be defined by an enumeration of its data values and is denoted by $\{\sigma_V, \dots\}$, where $\sigma_V, \dots \in \Sigma_V$ are data values.

Besides the descriptions for value-domains, the syntax for expressions on *concepts* comprises additional constructs by which more complex concepts can be described. Concepts can thus not only formed by the *universal concept* $\top_C$, the *bottom concept*[1] $\bot_C$, a *concept name* $\sigma_C \in \Sigma_C$, a *negation* (or complement) of a concept denoted by $\neg C$, or by an enumeration of individual names $\{\sigma_I, \dots\}$ with $\sigma_I, \dots \in \Sigma_I$ to a so-called *nominal concept*, but also by a *conjunction* (intersection) or *disjunction* (union) of concepts represented by $C_1 \sqcap C_2$ respectively

[1] As the universal role $\top_R$ corresponds to the universal concept $\top_C$, by analogy there exists an *empty role* $\bot_R$ corresponding to the bottom concept $\bot_C$. Even though the empty role is not part of Definition 2.3, it can be simply defined by the axiom $\top_C \sqsubseteq \neg\exists\bot_R.\top_C$ or alternatively by the axiom $\top_C \sqsubseteq \forall\bot_R.\bot_C$ (see Definition 2.6 for more details on axioms). Similarly, the *empty attribute* $\bot_A$ can be described by $\top_C \sqsubseteq \neg\exists\bot_A.\top_D$ respectively $\top_C \sqsubseteq \forall\bot_A.\bot_D$.

$C_1 \sqcup C_2$. Moreover, complex concepts can be constructed by descriptions comprising both concepts and roles, i.e., *(qualified) existential restrictions* ($\exists R.C$), *universal restrictions* ($\forall R.C$), *(qualified) number restrictions* ($\geq n\ R.C$ and $\leq n\ R.C$) and *self restrictions* ($\exists R.\mathsf{Self}$). An existential restriction is used to describe a set of individuals that have at least one relation R to an individual which belongs to concept C. In contrast, a set of individuals that are related with role R to individuals which all belongs to concept C is described by an universal restriction. Note that this includes also individuals with no relation R. A constraint on the number of individuals that can be related via a certain role by other individuals is described by a number restriction (also called cardinality constraint). More precisely, *at-least restrictions* are of the form $\geq n\ R.C$ whereas *at-most restrictions* are of the form $\leq n\ R.C$, where n is a non-negative integer. In addition to that, *self restrictions* can be used to describe a set of individuals that are related by itself with a certain role (local reflexivity). Except of self restrictions, there exist comparable constructs for descriptions that comprise concepts and attributes. Formally, the expression syntax for concepts in $\mathcal{SROIQ}(\mathbf{D})$ is given by

Definition 2.5 (Concept Expressions)**.** *Given a signature Σ, a concept C can be defined according to the expression grammar*

$$\begin{aligned} C ::= {} & \top_C \mid \bot_C \mid \sigma_C \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \{\sigma_I, \dots\} \mid \\ & \exists R.C \mid \forall R.C \mid \geq n\ R.C \mid \leq n\ R.C \mid \exists R.\mathsf{Self} \mid \\ & \exists A.D \mid \forall A.D \mid \geq n\ A.D \mid \leq n\ A.D\,, \end{aligned} \tag{2.5}$$

where $\top_C$ is the universal concept, $\bot_C$ the bottom concept, $\sigma_C \in \Sigma_C$ a concept name, $\neg C$ the negation of concept C, C, C_1, C_2 are concepts, $\sigma_I, \dots \in \Sigma_I$ are individual names, R a role, A an attribute, D a value-domain and n is a non-negative integer.

Generally, in DLs a *knowledge base* (or ontology) $\mathcal{K}$ defined by $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ over a signature Σ consists of an intensional (or terminological) knowledge part $\mathcal{T}$ called *TBox* and an extensional (or assertional) knowledge part $\mathcal{A}$ called *ABox*. The TBox $\mathcal{T}$ of a knowledge base (KB) is a finite set of *axioms* by which properties of attributes, roles, value-domains and concepts can be specified by relating different expressions typically via an *inclusion* (or *subsumption*) relation $\sqsubseteq$.

Definition 2.6 (TBox Axioms)**.** *Given a signature Σ as well as attributes A, A_1, A_2, roles R, R_1, R_2, a value-domain D and concepts C, C_1, C_2 according to the corresponding expression definitions. The axioms in the TBox $\mathcal{T}$ of a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ are for value-domains of the form*

$$\exists A^{-}.C \sqsubseteq D\,, \tag{2.6}$$

for concepts given by the syntax

$$C_1 \sqsubseteq C_2 \qquad \textit{and} \qquad C_1 \equiv C_2\,, \tag{2.7}$$

for attributes of the form

$$A_1 \sqsubseteq A_2\,, \qquad A_1 \equiv A_2 \qquad \textit{and} \qquad A_1 \sqsubseteq \neg A_2\,, \tag{2.8}$$

and for roles according to the grammar

$$R_1 \sqsubseteq R_2\,, \qquad R_1 \equiv R_2 \qquad R_1 \sqsubseteq \neg R_2 \qquad \textit{and} \qquad R_1 \circ R_2 \sqsubseteq R\,. \tag{2.9}$$

Since we assume that the set of value-domains is already sufficiently defined and consequently no constructs for new value-domains are required, the syntax for value-domain axioms (2.6) is simply one axiom type that comprises on the left-hand side an *existential restriction* denoting an attribute's *range*, i.e., the value-domain to which the data values related by attribute A belong to. If the existential restriction is *unqualified*, i.e., $C ::= \top_C$ and often abbreviated as $\exists A^- \sqsubseteq D$, the range D holds for all data values linked by A, whereas a *qualified existential restriction* ($\exists A^-.C \sqsubseteq D$) denotes the qualified range of attribute A originating from individuals of concept C, i.e., the value-domain (set of data values) to which A relates individuals belonging to C.

For axioms on concepts (2.7), the first axiom type represents a *general concept inclusion* (also called *subsumption*) describing an *is-a* relationship between the concepts C_1 and C_2 and the second kind describes a *concept equivalence*. According to the concept expressions (Definition 2.5), a general concept inclusions can be used to define the range of a role as well as the domain of a role or an attribute. Similar as for attributes, the unqualified existential restriction $\exists R^- \sqsubseteq C$ denotes concept C as the range of role R and the qualified existential restriction $\exists R^-.C_1 \sqsubseteq C_2$ defines concept C_2 as the qualified range of role R deriving form individuals of concept C_1. Correspondingly, the axioms $\exists A \sqsubseteq C$ and $\exists R \sqsubseteq C$ denote the *domain* of attribute A respectively of role R, i.e., the concept (set of individuals) that A links to some data value and R to some individuals. Again, the existential restrictions can be qualified like $\exists A.D \sqsubseteq C$ and $\exists R.C_2 \sqsubseteq C_1$ to define the domain of an attribute A or a role R with respect to (w.r.t.) a specific value-domain D or concept C_2. Conversely, if an existential restriction is used on the right-hand side of an axiom like $C \sqsubseteq \exists A$ respectively $C \sqsubseteq \exists R$ or $C \sqsubseteq \exists R^-$, all instances of concept C have a *mandatory participation* to attribute A respectively role R. Besides, a *mandatory non-participation* can be formalized by negating the corresponding existential restriction.

As can be expected, the first two forms in (2.8) represent a *(simple) attribute inclusion* axiom describing an *is-a* relationship between attributes such that every individual data value pair related by A_2 is linked by A_1 as well, and an *attribute equivalence* axiom, respectively. Since the definition of attribute expressions (Definition 2.2) does not contain negations, the third axiom type ($A_1 \sqsubseteq \neg A_2$) can be applied to denote an *attribute disjointness*. According to the negation of concepts, a *negated attribute* represents the set of all individual data value pairs that cannot be related by that attribute. Considering the attribute inclusion the attribute disjointness axioms specify that if an individual data value pair is linked by A_1 they

cannot be related by A_2 as well. In general, disjointness axioms are called *negative inclusions* whereas other axioms are *positive inclusions*.

Besides *(simple) role inclusion* axioms ($R_1 \sqsubseteq R_2$), *role equivalence* axioms ($R_1 \equiv R_2$) and *role disjointness* axioms ($R_1 \sqsubseteq \neg R_2$) there exists in (2.9) an additional type called *role composition* axioms ($R_1 \circ R_2 \sqsubseteq R$). Such axioms are a more complex kind of role inclusions and describe the formation of a new relation R from two given relations R_1 and R_2, such that if an individual σ_I relates via R_1 an individual σ_I' that again links an individual σ_I'' via role R_2, then σ_I directly relates σ_I'' via R.

According to the given syntax an attribute or role negation can only appear on the right-hand side and a role composition only on the left-hand side of an inclusion axiom. However, to ensure that sound and complete reasoning is decidable there exist for complex role inclusions some additional so called *structural restrictions* [KSH14]. These restrictions concern not the syntax of single axioms but the structure of the entire TBox $\mathcal{T}$.

Hence, to preserve decidability in $\mathcal{SROIQ}(\mathbf{D})$ for some axioms the application is restricted to *simple roles* [HKS06]. A role R is called simple if no role composition is subsumed by R, otherwise R is called *non-simple*. More precisely, a non-simple role is defined as follows:

- If $\mathcal{T}$ comprises an axiom of the form $R_1 \circ R_2 \sqsubseteq R$, then R is non-simple.
- If a role R is a non-simple role, then its inverse R^- is also a non-simple role.
- If a role R_1 is non-simple, then a role R_2 is also non-simple if $\mathcal{T}$ comprises an axiom of the form $R_1 \sqsubseteq R_2$ or $R_1 \equiv R_2$.

The restriction is now that qualified number restrictions ($\geq n\, R.C$ and $\leq n\, R.C$), self restrictions ($\exists R.\mathsf{Self}$) and role disjointness axioms ($R_1 \sqsubseteq \neg R_2$) must contain simple roles only.

Beside that, a second restriction is concerning the *regularity* of the role hierarchy in $\mathcal{T}$ and prevents the existence of arbitrary cyclic dependencies between roles. Formally, a regular order on the set of roles is a strict (irreflexive and transitive) partial order $\prec$ on roles that satisfies $R_1 \prec R_2$ if and only if (abbreviated as iff) $R_1^- \prec R_2$ and vice versa for all roles R_1 and R_2. A role inclusion axiom is called $\prec$-regular, if it is either

(i) $R \circ R \sqsubseteq R$,

(ii) $R^- \sqsubseteq R$,

(iii) $R_1 \circ \cdots \circ R_n \sqsubseteq R$,

(iv) $R \circ R_1 \circ \cdots \circ R_n \sqsubseteq R$, or

(v) $R_1 \circ \cdots \circ R_n \circ R \sqsubseteq R$,

where $R, R_1, \ldots, R_n$ are roles and $R_i \prec R$ for each $1 \leq i \leq n$. According to that, a $\mathcal{SROIQ}(\mathbf{D})$ TBox $\mathcal{T}$ respectively its role hierarchy is regular if there

exists a regular order $\prec$ on roles such that every role inclusion axiom is $\prec$-regular [HKS06].

While the TBox $\mathcal{T}$ captures general knowledge about concepts, value-domains, roles and attributes as well as their interdependencies in the considered domain of interest, the knowledge about single individuals is described in the ABox $\mathcal{A}$ by a finite set of *assertions*. In this way, features of single individuals are described by assigning concept memberships and relations to other individuals or data values.

Definition 2.7 (ABox Assertions)**.** *Given a signature Σ as well as a concept C, an attribute A, a role R, a data value $\sigma_V \in \Sigma_V$ and two individuals $\sigma_I, \sigma'_I \in \Sigma_I$. The syntax of individual assertions in the ABox $\mathcal{A}$ of a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is defined as*

$$C(\sigma_I)\,, \qquad R(\sigma_I, \sigma'_I)\,, \qquad \neg R(\sigma_I, \sigma'_I)\,, \qquad A(\sigma_I, \sigma_V)\,, \qquad \neg A(\sigma_I, \sigma_V)\,,$$
$$\sigma_I \approx \sigma'_I\,, \qquad \textit{and} \qquad \sigma_I \not\approx \sigma'_I\,. \tag{2.10}$$

Concept assertions ($C(\sigma_I)$), *role assertions* ($R(\sigma_I, \sigma'_I)$) and *attribute assertions* ($A(\sigma_I, \sigma_V)$) are the most common assertion types used to state that a certain individual belongs to a specific concept respectively to describe a specific relation between two single individual or between a certain individual and a concrete data value. According to the syntax defined above, concept assertions already capture negations while assertions on negated roles or attributes are defined by separated assertion forms ($\neg R(\sigma_I, \sigma'_I)$ respectively $\neg A(\sigma_I, \sigma_V)$). However, *negated role assertion* are additionally restricted to simple roles due to decidability aspects. We call assertions on non-negated concepts or roles *positive assertions* and *negative assertions* otherwise. Since a single entity in the considered domain of interest might be referred by different individual names, *individual equality* ($\sigma_I \approx \sigma'_I$) and *inequality statements* ($\sigma_I \not\approx \sigma'_I$) are used to state this information explicitly.

2.1.2 Semantics

In DLs the formal meaning of signature elements, axioms and assertions, and hence its logical consequences is given by its model-theoretic semantics defined in terms of *interpretations*.

Definition 2.8 (Interpretation)**.** *Given a signature Σ, an interpretation $\mathcal{I}$ is defined by the pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ represents a non-empty interpretation domain and $\cdot^{\mathcal{I}}$ an interpretation function. The interpretation domain is given by the union of the two non-empty disjoint sets $\Delta^{\mathcal{I}}_{\mathbf{O}}$, denoting the domain of objects (also called abstract domain), and $\Delta^{\mathcal{I}}_{\mathbf{D}}$, denoting the domain of values (also called concrete domain). The interpretation function maps each individual name $\sigma_I \in \Sigma_I$ to a domain element $\sigma_I^{\mathcal{I}} \in \Delta^{\mathcal{I}}_{\mathbf{O}}$, each data value $\sigma_V \in \Sigma_V$ to a value $\sigma_V^{\mathcal{I}} \in \Delta^{\mathcal{I}}_{\mathbf{D}}$, each concept name $\sigma_C \in \Sigma_C$ to a subset $\sigma_C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}_{\mathbf{O}}$, each value-domain name $\sigma_D \in \Sigma_D$ to a subset $\sigma_D^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}_{\mathbf{D}}$, each role name $\sigma_R \in \Sigma_R$ to a binary relation (set of ordered pairs) $\sigma_R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}_{\mathbf{O}} \times \Delta^{\mathcal{I}}_{\mathbf{O}}$, and each attribute name $\sigma_A \in \Sigma_A$ to a binary relation (set*

of ordered pairs) $\sigma_A^{\mathcal{I}} \subseteq \Delta_{\mathbf{O}}^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}$*. Moreover, based on the provided semantics of signature elements the interpretation function* $\cdot^{\mathcal{I}}$ *is extended for complex roles and concepts by*

$$
\begin{aligned}
\top_D^{\mathcal{I}} &= \Delta_{\mathbf{D}}^{\mathcal{I}}\,,\\
\top_C^{\mathcal{I}} &= \Delta_{\mathbf{O}}^{\mathcal{I}}\,,\\
\bot_D^{\mathcal{I}}, \bot_C^{\mathcal{I}} &= \emptyset\,,\\
\top_A^{\mathcal{I}} &= \Delta_{\mathbf{O}}^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}\,,\\
\top_R^{\mathcal{I}} &= \Delta_{\mathbf{O}}^{\mathcal{I}} \times \Delta_{\mathbf{O}}^{\mathcal{I}}\,,\\
(\neg D)^{\mathcal{I}} &= \Delta_{\mathbf{D}}^{\mathcal{I}} \setminus D^{\mathcal{I}}\,,\\
(\neg C)^{\mathcal{I}} &= \Delta_{\mathbf{O}}^{\mathcal{I}} \setminus C^{\mathcal{I}}\,,\\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}\,,\\
(C_1 \sqcup C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}\,,\\
(\neg R)^{\mathcal{I}} &= (\Delta_{\mathbf{O}}^{\mathcal{I}} \times \Delta_{\mathbf{O}}^{\mathcal{I}}) \setminus R^{\mathcal{I}}\,,\\
(R^-)^{\mathcal{I}} &= \{(\sigma_I, \sigma_I') \mid (\sigma_I', \sigma_I) \in R^{\mathcal{I}}\}\,,\\
(\exists R.C)^{\mathcal{I}} &= \{\sigma_I \in \Delta_{\mathbf{O}}^{\mathcal{I}} \mid \exists \sigma_I' \in \Delta_{\mathbf{O}}^{\mathcal{I}}.(\sigma_I, \sigma_I') \in R^{\mathcal{I}} \wedge \sigma_I' \in C^{\mathcal{I}}\}\,,\\
(\forall R.C)^{\mathcal{I}} &= \{\sigma_I \in \Delta_{\mathbf{O}}^{\mathcal{I}} \mid \forall \sigma_I' \in \Delta_{\mathbf{O}}^{\mathcal{I}}.(\sigma_I, \sigma_I') \in R^{\mathcal{I}} \rightarrow \sigma_I' \in C^{\mathcal{I}}\}\,,\\
(\geq n\, R.C)^{\mathcal{I}} &= \{\sigma_I \in \Delta_{\mathbf{O}}^{\mathcal{I}} \mid \#\{\sigma_I' \in \Delta_{\mathbf{O}}^{\mathcal{I}}.(\sigma_I, \sigma_I') \in R^{\mathcal{I}} \wedge \sigma_I' \in C^{\mathcal{I}}\} \geq n\}\,,\\
(\leq n\, R.C)^{\mathcal{I}} &= \{\sigma_I \in \Delta_{\mathbf{O}}^{\mathcal{I}} \mid \#\{\sigma_I' \in \Delta_{\mathbf{O}}^{\mathcal{I}}.(\sigma_I, \sigma_I') \in R^{\mathcal{I}} \wedge \sigma_I' \in C^{\mathcal{I}}\} \leq n\}\,,\\
(\exists R.\mathsf{Self})^{\mathcal{I}} &= \{\sigma_I \in \Delta_{\mathbf{O}}^{\mathcal{I}} \mid (\sigma_I, \sigma_I) \in R^{\mathcal{I}}\}\,,\\
(R_1 \circ R_2)^{\mathcal{I}} &= \{(\sigma_I, \sigma_I'') \in \Delta_{\mathbf{O}}^{\mathcal{I}} \times \Delta_{\mathbf{O}}^{\mathcal{I}} \mid \exists \sigma_I' \in \Delta_{\mathbf{O}}^{\mathcal{I}}.(\sigma_I, \sigma_I') \in R_1^{\mathcal{I}} \wedge (\sigma_I', \sigma_I'') \in R_2^{\mathcal{I}}\}\,,\\
(\neg A)^{\mathcal{I}} &= (\Delta_{\mathbf{O}}^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}) \setminus A^{\mathcal{I}}\,,\\
(A^-)^{\mathcal{I}} &= \{(\sigma_V, \sigma_I) \mid (\sigma_I, \sigma_V) \in A^{\mathcal{I}}\}\,,\\
(\exists A^-.C)^{\mathcal{I}} &= \{\sigma_V \in \Delta_{\mathbf{D}}^{\mathcal{I}} \mid \exists \sigma_I \in \Delta_{\mathbf{O}}^{\mathcal{I}}.(\sigma_I, \sigma_V) \in A^{\mathcal{I}} \wedge \sigma_I \in C^{\mathcal{I}}\}\,,\\
(\exists A.D)^{\mathcal{I}} &= \{\sigma_I \in \Delta_{\mathbf{O}}^{\mathcal{I}} \mid \exists \sigma_V \in \Delta_{\mathbf{D}}^{\mathcal{I}}.(\sigma_I, \sigma_V) \in A^{\mathcal{I}} \wedge \sigma_V \in D^{\mathcal{I}}\}\,,\\
(\forall A.D)^{\mathcal{I}} &= \{\sigma_I \in \Delta_{\mathbf{O}}^{\mathcal{I}} \mid \forall \sigma_V \in \Delta_{\mathbf{D}}^{\mathcal{I}}.(\sigma_I, \sigma_V) \in A^{\mathcal{I}} \rightarrow \sigma_V \in D^{\mathcal{I}}\}\,,\\
(\geq n\, A.D)^{\mathcal{I}} &= \{\sigma_I \in \Delta_{\mathbf{O}}^{\mathcal{I}} \mid \#\{\sigma_V \in \Delta_{\mathbf{D}}^{\mathcal{I}}.(\sigma_I, \sigma_V) \in A^{\mathcal{I}} \wedge \sigma_V \in D^{\mathcal{I}}\} \geq n\}\,,\\
(\leq n\, A.D)^{\mathcal{I}} &= \{\sigma_I \in \Delta_{\mathbf{O}}^{\mathcal{I}} \mid \#\{\sigma_V \in \Delta_{\mathbf{D}}^{\mathcal{I}}.(\sigma_I, \sigma_V) \in A^{\mathcal{I}} \wedge \sigma_V \in D^{\mathcal{I}}\} \leq n\}\,,\\
\{\sigma_V, \dots\}^{\mathcal{I}} &= \{\sigma_V^{\mathcal{I}}, \dots\}\,,\\
\{\sigma_I, \dots\}^{\mathcal{I}} &= \{\sigma_I^{\mathcal{I}}, \dots\}\,,
\end{aligned}
$$

where $\top_D$ *is the universal value-domain,* $\top_C$ *is the universal concept,* $\bot_D$ *the bottom value-domain,* $\bot_C$ *the bottom concept,* $\top_A$ *is the universal attribute,* $\top_R$ *is the universal role,* D *a value-domain,* C, C_1, C_2 *are concepts,* R, R_1, R_2 *are roles,* A *is an attribute,* $\sigma_V, \dots \in \Sigma_V$ *are data values,* $\sigma_I, \dots \in \Sigma_I$ *are individual names and* n *is a non-negative integer.*

Note that (unlike individuals and concepts) data values and value-domains have a fixed built-in interpretation because of our assumption that the set of value-domains (data types) Σ_D is already sufficiently defined by a type system.

Given a specific interpretation $\mathcal{I}$ fixing the meaning of each signature symbol (individual names, data values, concepts, value-domains, roles and attributes), we can determine if $\mathcal{I}$ *satisfies* an axiom or assertion $\alpha \in \mathcal{K}$. This is denoted by the *satisfaction relation* $\mathcal{I} \models \alpha$ and is defined by

Definition 2.9 (Satisfiability)**.** *Given a signature* Σ *and an interpretation* $\mathcal{I}$*,*

$$
\begin{aligned}
\mathcal{I} \models \exists A^{-}.C \sqsubseteq D \quad & \textit{iff} \quad (\exists A^{-}.C)^{\mathcal{I}} \subseteq D^{\mathcal{I}}\,, \\
\mathcal{I} \models C_1 \sqsubseteq C_2 \quad & \textit{iff} \quad C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}\,, \\
\mathcal{I} \models C_1 \equiv C_2 \quad & \textit{iff} \quad C_1^{\mathcal{I}} = C_2^{\mathcal{I}}\,, \\
\mathcal{I} \models A_1 \sqsubseteq A_2 \quad & \textit{iff} \quad A_1^{\mathcal{I}} \subseteq A_2^{\mathcal{I}}\,, \\
\mathcal{I} \models A_1 \equiv A_2 \quad & \textit{iff} \quad A_1^{\mathcal{I}} = A_2^{\mathcal{I}}\,, \\
\mathcal{I} \models A_1 \sqsubseteq \neg A_2 \quad & \textit{iff} \quad A_1^{\mathcal{I}} \subseteq (\Delta_{\mathbf{O}}^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}) \setminus A_2^{\mathcal{I}} \;\; (\textit{resp. } A_1^{\mathcal{I}} \cap A_2^{\mathcal{I}} = \emptyset)\,, \\
\mathcal{I} \models R_1 \sqsubseteq R_2 \quad & \textit{iff} \quad R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}\,, \\
\mathcal{I} \models R_1 \equiv R_2 \quad & \textit{iff} \quad R_1^{\mathcal{I}} = R_2^{\mathcal{I}}\,, \\
\mathcal{I} \models R_1 \sqsubseteq \neg R_2 \quad & \textit{iff} \quad R_1^{\mathcal{I}} \subseteq (\Delta_{\mathbf{O}}^{\mathcal{I}} \times \Delta_{\mathbf{O}}^{\mathcal{I}}) \setminus R_2^{\mathcal{I}} \;\; (\textit{resp. } R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} = \emptyset)\,, \\
\mathcal{I} \models R_1 \circ R_2 \sqsubseteq R \quad & \textit{iff} \quad (R_1 \circ R_2)^{\mathcal{I}} \subseteq R^{\mathcal{I}}
\end{aligned}
$$

for TBox axioms and

$$
\begin{aligned}
\mathcal{I} \models C(\sigma_I) \quad & \textit{iff} \quad \sigma_I^{\mathcal{I}} \in C^{\mathcal{I}}\,, \\
\mathcal{I} \models R(\sigma_I, \sigma'_I) \quad & \textit{iff} \quad (\sigma_I^{\mathcal{I}}, \sigma_I'^{\mathcal{I}}) \in R^{\mathcal{I}}\,, \\
\mathcal{I} \models \neg R(\sigma_I, \sigma'_I) \quad & \textit{iff} \quad \mathcal{I} \not\models R(\sigma_I, \sigma'_I) \;\; (\textit{resp. } (\sigma_I^{\mathcal{I}}, \sigma_I'^{\mathcal{I}}) \notin R^{\mathcal{I}})\,, \\
\mathcal{I} \models A(\sigma_I, \sigma_V) \quad & \textit{iff} \quad (\sigma_I^{\mathcal{I}}, \sigma_V^{\mathcal{I}}) \in A^{\mathcal{I}}\,, \\
\mathcal{I} \models \sigma_I \approx \sigma'_I \quad & \textit{iff} \quad \sigma_I^{\mathcal{I}} = \sigma_I'^{\mathcal{I}}\,, \\
\mathcal{I} \models \sigma_I \not\approx \sigma'_I \quad & \textit{iff} \quad \mathcal{I} \not\models \sigma_I \approx \sigma'_I \;\; (\textit{resp. } \sigma_I^{\mathcal{I}} \neq \sigma_I'^{\mathcal{I}})
\end{aligned}
$$

for ABox assertions, where $\sigma_D \in \Sigma_D$ *a value-domain name,* C, C_1, C_2 *are concepts,* R, R_1, R_2 *are roles,* A *is an attribute,* $\sigma_V, \ldots \in \Sigma_V$ *are data values,* $\sigma_I, \ldots \in \Sigma_I$ *are individual names and* n *is a non-negative integer.*

As the interpretation of the last two satisfaction relations for assertion types suggest, different individual names not necessarily refer to different individuals unless explicitly stated. Thus, $\mathcal{SROIQ}(\mathbf{D})$ do not follow the so called *unique name assumption* (UNA).

According to the satisfaction relation we can clearly determine if an axiom or assertion holds in an interpretation. If all axioms and assertions of a KB $\mathcal{K}$ hold in an interpretation $\mathcal{I}$ respectively if $\mathcal{I}$ satisfies all axioms and assertion of $\mathcal{K}$, $\mathcal{I}$ is called a *model* of $\mathcal{K}$.

Definition 2.10 (Model)**.** *An interpretation $\mathcal{I}$ is called a model of a knowledge base $\mathcal{K}$ and is denoted by $\mathcal{I} \models \mathcal{K}$, iff $\mathcal{I}$ satisfies all axioms in $\mathcal{T}$ and all assertions in $\mathcal{A}$ of $\mathcal{K}$, i.e., $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$.*

In conventional databases an instance of a database represents exactly one interpretation why the absence of information is understood as negated information. However, contrary to conventional databases in DL semantics the domain $\Delta^{\mathcal{I}}$ and the interpretation function $\cdot^{\mathcal{I}}$ are not fixed. Consequently, a KB may have several models (satisfying interpretations) and the absence of information is solely interpreted as incomplete information. DLs hence rely on the so called *open world assumption* (OWA) where databases adopt the *closed world assumption* (CWA) [HKR09; Baa+10]. The set of all models of a knowledge base $\mathcal{K}$ is denoted by $Mod(\mathcal{K})$.

2.1.3 Standard Reasoning

According to the formal semantics it is possible to infer logical consequences out of the information explicitly stated. Hence, the inference of additional (implicit) statements (i.e., axioms or assertions) via logical deduction is an important reasoning task [Rud11], called *entailment*, and is defined by

Definition 2.11 (Entailment)**.** *An axiom or assertion α' is entailed by (resp. a logical consequence of) a given knowledge base $\mathcal{K}$ denoted by $\mathcal{K} \models \alpha'$, iff each model $\mathcal{I}$ of $\mathcal{K}$, i.e., $\forall \mathcal{I} \in Mod(\mathcal{K})$ is a model of α', i.e., $\mathcal{I} \models \alpha'$ as well. Otherwise α' is not entailed by $\mathcal{K}$ and is denoted by $\mathcal{K} \not\models \alpha'$.*

If an assertion like $C(\sigma_I)$, $R(\sigma_I, \sigma'_I)$ or $A(\sigma_I, \sigma_V)$ is entailed by a knowledge base $\mathcal{K}$, i.e., the assertion holds in (is satisfied by) each model $\mathcal{I} \in Mod(\mathcal{K})$, then the individual σ_I respectively the individual pair (σ_I, σ'_I) or individual data value pair (σ_I, σ_V) is called an *instance* of concept C respectively role R or attribute A w.r.t. $\mathcal{K}$. Strongly related to entailment is the task of examining whether a certain individual, individual pair or individual data value pair is an instance of a given concept, role or attribute w.r.t. a specific KB and is called *instance checking*. Similarly, deciding whether a given concept C_1 (role R_1 or attribute A_1) is subsumed by another concept C_2 (role R_2 or attribute A_2) w.r.t. a knowledge base $\mathcal{K}$, i.e., $\mathcal{I} \models C_1 \sqsubseteq C_2$ ($\mathcal{I} \models R_1 \sqsubseteq R_2$ or $\mathcal{I} \models A_1 \sqsubseteq A_2$) for every $\mathcal{I} \in Mod(\mathcal{K})$, is called *subsumption checking*.

Taking all implicit information into account, another elementary reasoning task is to determine whether a KB implies contradicting statements (logical conflicts). The most primitive case is the *unsatisfiability* of an element in $\mathcal{T}$, which means that the empty set is assigned to a concept, role or attribute in every model of the knowledge base $\mathcal{K}$. Formally, the definition of an unsatisfiable concept, role or attribute is given by

Definition 2.12 (Unsatisfiability)**.** *Given a knowledge base $\mathcal{K}$ and its signature Σ. A concept, role or attribute σ in $\Sigma_C \cup \Sigma_R \cup \Sigma_A$ is unsatisfiable in $\mathcal{K}$ iff $\sigma^{\mathcal{I}} = \emptyset$ in every model $\mathcal{I} \in Mod(\mathcal{K})$, i.e., $\mathcal{K} \models \sigma \sqsubseteq \bot$.*

Since positive assertions on unsatisfiable elements in $\mathcal{T}$ cannot exist, an unsatisfiability of a concept, role or attribute may indicate some modeling errors [SC03]. Consequently, a knowledge base $\mathcal{K}$, i.e., its TBox $\mathcal{T}$ is called *incoherent* iff $\mathcal{T}$ comprises an unsatisfiable concept or role. Otherwise $\mathcal{K}$, i.e., $\mathcal{T}$ is called coherent.

Definition 2.13 (Incoherence)**.** *Given a knowledge base $\mathcal{K}$ and its signature Σ. $\mathcal{K}$ and $\mathcal{T}$ are incoherent iff $\exists \sigma \in (\Sigma_C \cup \Sigma_R \cup \Sigma_A) \mid \mathcal{K} \models \sigma \sqsubseteq \bot$. Otherwise $\mathcal{K}$ (and $\mathcal{T}$) are coherent.*

Even though unsatisfiability and hence incoherence are strongly related to classical contradictions, they concern merely the TBox of a KB as indicated by its definitions [Baa+10; Flo+06]. Hence, the definition of *inconsistency* is slightly different and also takes the ABox into consideration:

Definition 2.14 (Inconsistency)**.** *A given knowledge base $\mathcal{K}$ is inconsistent iff $Mod(\mathcal{K}) = \emptyset$. Otherwise $\mathcal{K}$ is consistent.*

Note that an unsatisfiable element in $\mathcal{T}$ does not necessarily imply that $\mathcal{K}$ is inconsistent since there may still exists a model for an incoherent KB. Because of that, a KB can be incoherent but still consistent or coherent but inconsistent [Flo+06]. However, the consistency of a DL KB is a crucial requirement since from an inconsistent KB any arbitrary statement is entailed (also called as *principle of explosion*). This is because the set of all models is empty why any statement is trivially satisfied in each model.

Since a KB holds (explicit and implicit) information about individuals, an intuitive purpose is the *instance retrieval* of a particular element in $\mathcal{T}$ like a concept, role or attribute. So a typical task could be for example to obtain all individuals σ_I that are instances of a concept C w.r.t. a given KB $\mathcal{K}$ such that $\mathcal{K} \models C(\sigma_I)$. In this case, the description of a concept, role or attribute is used as a query specifying the desired set of individuals. Thus, instance retrieval can be performed (not optimized) by simply checking for each individual $\sigma_I \in \Sigma_I$ if it is entailed by $\mathcal{K}$ as an instance of the query.

In expressive logics, like in $\mathcal{SROIQ}(\mathbf{D})$, all standard reasoning tasks can be expressed in terms of each other [Baa+10]. Since the computational complexity is measured w.r.t. the size of the input, we consider besides the *knowledge base complexity* also the *data complexity* [Var82; Don+94]. While the knowledge base complexity is measured in the size of the entire KB, in the data complexity only the size of the ABox is considered whereas the TBox is fixed. Especially when the size of the data, i.e., the ABox, dominates the size of the KB, the data complexity is of more interest on estimating the behavior for an increasing number of assertions. In $\mathcal{SROIQ}(\mathbf{D})$ KBs the knowledge base complexity for sound and complete standard reasoning tasks is N2EXPTIME and the decidability of standard reasoning tasks w.r.t. data complexity is known, but not its exact computational complexity, which is at least NP-hard [Kaz08; Mot+12].

Note that the complexity class N2EXPTIME denotes (decision) problems that can be solved by a nondeterministic Turing machine in time double exponential to

the input size while problems in NP are solvable in polynomial time on a nondeterministic Turing machine. Within the scope of this thesis we will further encounter the complexity classes depicted below according to its known inclusion relationships:

$$\mathrm{AC}^0 \subsetneq \textsc{LogSpace} \subseteq \textsc{NLogSpace} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \textsc{ExpTime} \subseteq \mathrm{N2}\textsc{ExpTime}$$

The complexity classes P and EXPTIME are defined analogously to NP respectively N2EXPTIME, with the exception that a deterministic Turing machine is applied. Problems belonging to P or below are referred to as *tractable*, whereas problems of complexity classes above are called *intractable*. If the space needed by a deterministic or nondeterministic Turing machine is of logarithmic size w.r.t. the input, the corresponding problem is in LOGSPACE and NLOGSPACE, respectively. Instead of relying on a Turing machine, the lowest complexity class AC^0 is defined by Boolean circuits [Koz06] that are of polynomial size and with a constant depth. Notably, a characteristic problem instance whose data complexity belongs to AC^0 is the query evaluation over relational databases. While the proper inclusion $\mathrm{AC}^0 \subsetneq$ LOGSPACE is known, for any other mentioned inclusions it is currently still open whether they are strict. For an extensive introduction to complexity theory and precise definitions of the complexity classes we refer to the textbooks of Papadimitriou [Pap94], Vollmer [Vol99], Kozen [Koz06], and Hopcroft et al. [HMU13].

2.2 *DL-Lite*$_\mathcal{A}$ and its Family

Due to the high computational complexity the application of expressive DLs in practical scenarios is limited. Especially when the data, i.e., the ABox part is huge the demand for efficient (tractable) and scalable reasoning increases. Driven by this motivation, lightweight DLs have been identified to gain a lower computational complexity by restricting the syntactical expressiveness [Krö12]. In the following, we briefly discuss the basic members of the *DL-Lite* family, one of the most well-known lightweight DLs. In particular, we are interested in *DL-Lite*$_\mathcal{A}$ [Pog+08] which is especially designed for efficiently dealing with and reasoning on huge ABoxes.

Aiming to find a trade-off between expressiveness and (sound and complete) reasoning complexity, Calvanese et al. [Cal+07b] proposed the *DL-Lite* family. The primary focus of *DL-Lite* is to handle challenges of data access and integration resulting from the continuously growing amount of available data from various sources. The approach of combining DLs with query optimization strategies of relational database management systems is known as *ontology-based data access* (OBDA) [Cal+07a; Cal+09]. In OBDA a TBox serves as conceptual and formal description of the considered domain of interest and mappings between this conceptual view (TBox) and the (relational) data schema, i.e., the data within the data source precisely specify the correspondences. The main intention of this approach

is to enable clients to access the data by using the conceptual view but without being aware of the source-specific data schema. In the more general case where several (independent) databases, i.e., data sources are considered within the mappings, the term *ontology-based data integration* (OBDI) is used [Cal+18].

DL-Lite$_{core}$ builds the base line of the *DL-Lite* family and allows the expression of cyclic concept inclusions, concept disjointnesses, role domains and ranges, as well as mandatory (non-)participations concerning roles. Based on *DL-Lite$_{core}$*, *DL-Lite$_\mathcal{F}$* (also called *DL-Lite$^\mathcal{F}_{core}$*) and *DL-Lite$_\mathcal{R}$* (also called *DL-Lite$^\mathcal{H}_{core}$*) are the simplest members extending *DL-Lite$_{core}$*. *DL-Lite$_\mathcal{F}$* extends the base line by *functionality constraints* for roles or its inverse, limiting the number of individuals related by an individual via a certain role or its inverse to one. In contrast, in *DL-Lite$_\mathcal{R}$* the core syntax is extended by role inclusion and disjointness axioms. The combination of both with some syntactical restrictions on functionality constraints is represented by *DL-Lite$_\mathcal{A}$*. Unlike *DL-Lite$_\mathcal{F}$* and *DL-Lite$_\mathcal{R}$*, *DL-Lite$_\mathcal{A}$* further distinguishes between concepts from value-domains and roles from attributes.

Like in any other DL, expressions in *DL-Lite* are built over a countably infinite *signature* Σ given in Definition 2.1.

Definition 2.15 (*DL-Lite* Syntax)**.** *Given a signature* $\Sigma = \langle \Sigma_I, \Sigma_V, \Sigma_C, \Sigma_D, \Sigma_R, \Sigma_A \rangle$*, the syntax for attributes, roles, value-domains, concepts, TBox axioms and ABox assertions in DL-Lite$_{core}$, DL-Lite$_\mathcal{F}$, DL-Lite$_\mathcal{R}$ and DL-Lite$_\mathcal{A}$ is defined by*

Table 2.1: The *DL-Lite* Family

	DL-Lite$_{core}$	*DL-Lite$_\mathcal{F}$*	*DL-Lite$_\mathcal{R}$*	*DL-Lite$_\mathcal{A}$*
$B ::=$	$\bot_C \mid \sigma_C \mid \exists Q$			$\bot_C \mid \sigma_C \mid \exists Q$ $\mid \delta(\sigma_A)$
$C ::=$	$\top_C \mid B \mid \neg B$		$\top_C \mid B \mid \neg B$ $\mid \exists Q.C$	$\top_C \mid B \mid \neg B$ $\mid \exists Q.C \mid \delta_D(\sigma_A)$
$Q ::=$	$\sigma_R \mid \sigma_R^-$			
$R ::=$	-		$Q \mid \neg Q$	
$E ::=$	-			$\rho(\sigma_A)$
$D ::=$	-			$\top_D \mid \sigma_D$
$A ::=$	-			$\sigma_A \mid \neg\sigma_A$
TBox	$B \sqsubseteq C$	$B \sqsubseteq C$ $(\mathsf{funct}\ Q)$	$B \sqsubseteq C$ $Q \sqsubseteq R$	$B \sqsubseteq C$ $Q \sqsubseteq R$ $E \sqsubseteq D$ $\sigma_A \sqsubseteq A$ $(\mathsf{funct}\ Q)^*$ $(\mathsf{funct}\ \sigma_A)^*$
ABox	$\sigma_C(\sigma_I)$ $\sigma_R(\sigma_I, \sigma_I')$			$\sigma_C(\sigma_I)$ $\sigma_R(\sigma_I, \sigma_I')$ $\sigma_A(\sigma_I, \sigma_V)$

* *iff Q and σ_A are primitive*

where $\top_C$ denotes the universal concept, $\bot_C$ the bottom concept, $\sigma_C \in \Sigma_C$ a concept name (atomic concept), B a basic concept, $\neg B$ the negation of concept B, C a general concept, $\sigma_R \in \Sigma_R$ a role name (atomic role) and σ_R^- its inverse, Q a basic role and $\neg Q$ its negation, $\exists Q$ an unqualified existential restriction and $\exists Q.C$ a qualified existential restriction, R a general role, $\sigma_A \in \Sigma_A$ an attribute name (atomic attribute), $\neg\sigma_A$ the negation of σ_A, A a general attribute, $\rho(\sigma_A)$ the range of attribute σ_A, E a basic value-domain, $\top_D$ the universal value-domain, $\sigma_D \in \Sigma_D$ a value-domain name, D a value-domain, $\delta(\sigma_A)$ the domain of attribute σ_A, $\delta_D(\sigma_A)$ the qualified domain of attribute σ_A w.r.t. the value-domain D, $B \sqsubseteq C$ a concept inclusion axiom, $Q \sqsubseteq R$ a role inclusion axiom, $E \sqsubseteq D$ a value-domain inclusion axiom, $\sigma_A \sqsubseteq A$ an attribute inclusion axiom, $(\mathsf{funct}\ Q)$ a role functionality assertion axiom, $(\mathsf{funct}\ \sigma_A)$ an attribute functionality assertion axiom, $\sigma_I, \sigma_I' \in \Sigma_I$ are individual names, $\sigma_V \in \Sigma_V$ a data value, $\sigma_C(\sigma_I)$ a concept assertion, $\sigma_R(\sigma_I, \sigma_I')$ a role assertion, and $\sigma_A(\sigma_I, \sigma_V)$ an attribute assertion.

Contrary to expressive DL languages like $\mathcal{SROIQ}(\mathbf{D})$ the syntax of the mentioned *DL-Lite* family members does not comprise conjunctions, disjunctions, universal restrictions or (arbitrary) number restrictions. Even if the syntax of the stated *DL-Lite* members does not allow conjunctions or disjunctions, the two axioms $C_1 \sqsubseteq C_2$ and $C_1 \sqsubseteq C_3$ may be expressed by $C_1 \sqsubseteq C_2 \sqcap C_3$. Similarly, the axiom $C_2 \sqcup C_3 \sqsubseteq C_1$ corresponds to the axioms $C_2 \sqsubseteq C_1$ and $C_3 \sqsubseteq C_1$, why conjunctions on the left-hand side of inclusion axioms and disjunctions on the right-hand side of inclusion axioms can be simply seen as syntactic sugar. Moreover, qualified existential restriction are only allowed for general concepts in *DL-Lite*$_\mathcal{R}$ and *DL-Lite*$_\mathcal{A}$, and according to the syntax for TBox axioms may only appear on the right-hand side of a concept inclusion. Note that $\delta(\sigma_A)$ and $\delta_D(\sigma_A)$ are just syntactical abbreviations for the existential restrictions $\exists\sigma_A.\top_D$ respectively $\exists\sigma_A.D$. Similarly, $\rho(\sigma_A)$ is an equivalent for $\exists\sigma_A^-.\top_C$. The value-domains Σ_D in *DL-Lite*$_\mathcal{A}$ correspond to the *XML Schema*[2] data types used by the *Resource Description Framework 1.1*[3] (RDF), where the sets of values represented by the individual data types are pairwise disjoint. Thus, like already for $\mathcal{SROIQ}(\mathbf{D})$ in Section 2.1.1, the data type system can be assumed to be already sufficiently structured and defined. *Functionality assertion axioms* in *DL-Lite*$_\mathcal{F}$ denoted by $(\mathsf{funct}\ Q)$ for a *basic role* Q, which comprises role names (*atomic roles*) or its inverse, are merely a shortcut for the axiom $\top_C \sqsubseteq \leq 1\, R.\top_C$ stating that an individual can relate via Q to not more than one individual. However, in *DL-Lite*$_\mathcal{A}$ functionality assertion axioms are further limited to basic roles and *atomic attributes* (i.e., attribute names) which are *primitive*, meaning that the role respectively attribute has no specialization, i.e., does not positively appear on the right-hand side of an inclusion axiom and is not used in a qualified existential restriction.

Corresponding to the classical approach in DLs, the semantics of the *DL-Lite* family members is given in terms of interpretations (Definition 2.8).

[2] `https://www.w3.org/XML/Schema`

[3] `https://www.w3.org/RDF/`, see also Section 2.4 for further details

Definition 2.16 (*DL-Lite* Semantics)**.** *Given a signature* $\Sigma = \langle \Sigma_I, \Sigma_V, \Sigma_C, \Sigma_D, \Sigma_R, \Sigma_A \rangle$ *and an interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$*, where* $\Delta^\mathcal{I} = \langle \Delta^\mathcal{I}_\mathbf{O}, \Delta^\mathcal{I}_\mathbf{D} \rangle$ *and* $\cdot^\mathcal{I}$ *maps each individual name* $\sigma_I \in \Sigma_I$ *to a domain element* $\sigma_I^\mathcal{I} \in \Delta^\mathcal{I}_\mathbf{O}$*, each data value* $\sigma_V \in \Sigma_V$ *to a value* $\sigma_V^\mathcal{I} \in \Delta^\mathcal{I}_\mathbf{D}$*, each concept name* $\sigma_C \in \Sigma_C$ *to a subset* $\sigma_C^\mathcal{I} \subseteq \Delta^\mathcal{I}_\mathbf{O}$*, each value-domain name* $\sigma_D \in \Sigma_D$ *to a subset* $\sigma_D^\mathcal{I} \subseteq \Delta^\mathcal{I}_\mathbf{D}$*, each role name* $\sigma_R \in \Sigma_R$ *to a binary relation (set of ordered pairs)* $\sigma_R^\mathcal{I} \subseteq \Delta^\mathcal{I}_\mathbf{O} \times \Delta^\mathcal{I}_\mathbf{O}$*, and each attribute name* $\sigma_A \in \Sigma_A$ *to a binary relation (set of ordered pairs)* $\sigma_A^\mathcal{I} \subseteq \Delta^\mathcal{I}_\mathbf{O} \times \Delta^\mathcal{I}_\mathbf{D}$*. Moreover,*

$$
\begin{aligned}
\top_D^\mathcal{I} &= \Delta^\mathcal{I}_\mathbf{D}\,, \\
\top_C^\mathcal{I} &= \Delta^\mathcal{I}_\mathbf{O}\,, \\
\bot_C^\mathcal{I} &= \emptyset\,, \\
(\neg B)^\mathcal{I} &= \Delta^\mathcal{I}_\mathbf{O} \setminus B^\mathcal{I}\,, \\
(\sigma_R^-)^\mathcal{I} &= \{(\sigma_I, \sigma'_I) \mid (\sigma'_I, \sigma_I) \in \sigma_R^\mathcal{I}\}\,, \\
(\neg Q)^\mathcal{I} &= (\Delta^\mathcal{I}_\mathbf{O} \times \Delta^\mathcal{I}_\mathbf{O}) \setminus Q^\mathcal{I}\,, \\
(\exists Q.C)^\mathcal{I} &= \{\sigma_I \in \Delta^\mathcal{I}_\mathbf{O} \mid \exists \sigma'_I \in \Delta^\mathcal{I}_\mathbf{O}.(\sigma_I, \sigma'_I) \in Q^\mathcal{I} \wedge \sigma'_I \in C^\mathcal{I}\}\,, \\
(\neg \sigma_A)^\mathcal{I} &= (\Delta^\mathcal{I}_\mathbf{O} \times \Delta^\mathcal{I}_\mathbf{D}) \setminus \sigma_A^\mathcal{I}\,, \\
(\rho(\sigma_A))^\mathcal{I} = (\exists A^-.\top_C)^\mathcal{I} &= \{\sigma_V \in \Delta^\mathcal{I}_\mathbf{D} \mid \exists \sigma_I \in \Delta^\mathcal{I}_\mathbf{O}.(\sigma_I, \sigma_V) \in \sigma_A^\mathcal{I}\}\,, \\
(\delta(\sigma_A))^\mathcal{I} = (\exists \sigma_A)^\mathcal{I} &= \{\sigma_I \in \Delta^\mathcal{I}_\mathbf{O} \mid \exists \sigma_V \in \Delta^\mathcal{I}_\mathbf{D}.(\sigma_I, \sigma_V) \in \sigma_A^\mathcal{I}\}\,, \\
(\delta_D(\sigma_A))^\mathcal{I} = (\exists \sigma_A.D)^\mathcal{I} &= \{\sigma_I \in \Delta^\mathcal{I}_\mathbf{O} \mid \exists \sigma_V \in \Delta^\mathcal{I}_\mathbf{D}.(\sigma_I, \sigma_V) \in \sigma_A^\mathcal{I} \wedge \sigma_V \in D^\mathcal{I}\},
\end{aligned}
$$

where $\top_D$ *is the universal value-domain,* $\top_C$ *the universal concept,* $\bot_C$ *the bottom concept,* B *a basic concept,* C *a general concept,* $\sigma_R \in \Sigma_R$ *a role name (atomic role),* Q *a basic role,* $\sigma_A \in \Sigma_A$ *an attribute name (atomic attribute),* $\delta(\sigma_A)$ *the domain of attribute* σ_A*,* $\delta_D(\sigma_A)$ *the qualified domain of attribute* σ_A *w.r.t. the value-domain* D*,* $\rho(\sigma_A)$ *the range of attribute* σ_A*,* $\sigma_I, \ldots \in \Sigma_I$ *are individual names and* $\sigma_V \in \Sigma_V$ *is a data value.*

For TBox axioms

$$
\begin{aligned}
\mathcal{I} \models B \sqsubseteq C \quad & \textit{iff} \quad B^\mathcal{I} \subseteq C^\mathcal{I}\,, \\
\mathcal{I} \models Q \sqsubseteq R \quad & \textit{iff} \quad Q^\mathcal{I} \subseteq R^\mathcal{I}\,, \\
\mathcal{I} \models E \sqsubseteq D \quad & \textit{iff} \quad E^\mathcal{I} \subseteq D^\mathcal{I}\,, \\
\mathcal{I} \models \sigma_A \sqsubseteq A \quad & \textit{iff} \quad \sigma_A^\mathcal{I} \subseteq A^\mathcal{I}\,, \\
\mathcal{I} \models (\mathsf{funct}\ Q) \quad & \textit{iff} \quad (\sigma_I, \sigma'_I) \in Q^\mathcal{I} \wedge (\sigma_I, \sigma''_I) \in Q^\mathcal{I} \rightarrow \sigma'_I = \sigma''_I\,, \\
\mathcal{I} \models (\mathsf{funct}\ \sigma_A) \quad & \textit{iff} \quad (\sigma_I, \sigma_V) \in \sigma_A^\mathcal{I} \wedge (\sigma_I, \sigma'_V) \in \sigma_A^\mathcal{I} \rightarrow \sigma_V = \sigma'_V\,,
\end{aligned}
$$

and for ABox assertions

$$
\begin{aligned}
\mathcal{I} \models \sigma_C(\sigma_I) \quad & \textit{iff} \quad \sigma_I^\mathcal{I} \in \sigma_C^\mathcal{I}\,, \\
\mathcal{I} \models \sigma_R(\sigma_I, \sigma'_I) \quad & \textit{iff} \quad (\sigma_I^\mathcal{I}, \sigma_I'^\mathcal{I}) \in \sigma_R^\mathcal{I}\,, \\
\mathcal{I} \models \sigma_A(\sigma_I, \sigma_V) \quad & \textit{iff} \quad (\sigma_I^\mathcal{I}, \sigma_V^\mathcal{I}) \in \sigma_A^\mathcal{I}\,,
\end{aligned}
$$

where B denotes a basic concept, C a general concept, Q a basic role, R a general role, E a basic value-domain, D a value-domain, $\sigma_A \in \Sigma_A$ an attribute name (atomic attribute), A a general attribute, $\sigma_I, \ldots \in \Sigma_I$ are individual names and $\sigma_V, \ldots \in \Sigma_V$ are data values.

Concerning the semantics of ABox assertions, the *DL-Lite* family imposes in contrast to $\mathcal{SROIQ}(\mathbf{D})$ the unique name assumption (UNA) [Cal+07b; Pog+08]. Therefore, for each individual name $\sigma_I \in \Sigma_I$ a distinct entity $\sigma_I \in \Delta^{\mathcal{I}}_{\mathbf{O}}$ is assigned such that $\sigma_I \neq \sigma'_I \rightarrow \sigma_I^{\mathcal{I}} \neq \sigma_I'^{\mathcal{I}}$ for each $\sigma_I, \sigma'_I \in \Sigma_I$ in every interpretation $\mathcal{I}$.

The specific tailoring of *DL-Lite* enables (standard) reasoning in NLOGSPACE in the size of the KB (knowledge base complexity) and in AC^0 in the size of the ABox (data complexity) for all mentioned *DL-Lite* family members [Art+09].[4]

2.3 Query Answering

Beside the mentioned standard reasoning problems more complex reasoning tasks are commonly required in practical applications. Especially the consideration of a KB as an information store implies among others the need for querying, i.e., the formulation and evaluation of queries. Within the scope of this thesis we consider (unions of) conjunctive queries which are a well-known fragment of FOL used as a query language. While instance retrieval can be seen as a simple form of querying, a conjunctive query is expressed in terms of a possibly open FOL formula and thus formulated by a more general and powerful query language providing a combination (conjunction and disjunction) of several statements, called *query atoms*. Since the specification of a query atom is based on the description of a concept, role, value-domain or attribute, its syntax is similar to ABox assertions with the extension of allowing variables. Especially the unrestricted sharing of variables by several query atoms facilitates flexible joins of pieces of information.

Definition 2.17 (Query Syntax)**.** *Given a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ over a signature $\Sigma = \langle \Sigma_I, \Sigma_V, \Sigma_C, \Sigma_D, \Sigma_R, \Sigma_A \rangle$ and let $\mathcal{V}$ be a countably infinite set of variables. A query atom ϕ is given by the syntax*

$$\phi ::= C(t) \mid R(t_1, t_2) \mid D(t) \mid A(t_1, t_2) \mid t_1 = t_2\,, \tag{2.11}$$

where C is a concept name $\sigma_C \in \Sigma_C$, R is a role name $\sigma_R \in \Sigma_R$, D is a value-domain name $\sigma_D \in \Sigma_D$, A is an attribute name $\sigma_A \in \Sigma_A$ and the query terms t, t_1, t_2 are either individual names, data values or variables, i.e., $t, t_1, t_2 \in \Sigma_I \cup \Sigma_V \cup \mathcal{V}$.

A conjunctive query (CQ) q over the TBox $\mathcal{T}$ of a KB $\mathcal{K}$ is an expression of the form $q(\boldsymbol{x}) = \exists \boldsymbol{y}.conj(\boldsymbol{x}, \boldsymbol{y})$, where $q(\boldsymbol{x})$ denotes the head of query q comprising the free variables $\boldsymbol{x} \subseteq \mathcal{V}$ called distinguished variables (or answer variables).

[4] The data complexity AC^0 holds for all *DL-Lite* family members and results out of its FOL-rewritability (Definition 2.19) as explained in Section 2.3.

Moreover, the size of $\boldsymbol{x}$ is called the arity of query $q(\boldsymbol{x})$. The body $\exists \boldsymbol{y}.conj(\boldsymbol{x}, \boldsymbol{y})$ of q is a conjunction of query atoms and contains beside $\boldsymbol{x}$ the free variables $\boldsymbol{y} \subseteq \mathcal{V}$ called non-distinguished variables (and hence also named existentially quantified variables). While distinguished variables and non-distinguished variable occurring in at least two query atoms are called bound, non-distinguished non-shared variables are called unbound and are denoted by $\_$.

A disjunction of conjunctive queries is called a union of conjunctive queries (UCQ) and is denoted by $q(\boldsymbol{x}) = \bigvee_{n=1,\ldots,N} \exists \boldsymbol{y}_n.conj_n(\boldsymbol{x}, \boldsymbol{y}_n)$, where each $\exists \boldsymbol{y}_n.conj_n(\boldsymbol{x}, \boldsymbol{y}_n)$ is a CQ, i.e., a conjunction of query atoms as before.

The definition of (U)CQs can be extended by inequalities, denoted by (U)CQ$^{\neq}$, such that a query atom is either an expression according to Equation (2.11) or an expression of the form $t_1 \neq t_2$. Moreover, since in expressive DLs as, e.g., in $\mathcal{SROIQ}(\mathbf{D})$, the ABox of a KB may comprise also negative assertions on concepts, roles or attributes. Hence, another extension are (U)CQ with negations, denoted by (U)CQ$^{\neg}$, where the query body $\exists \boldsymbol{y}.conj(\boldsymbol{x}, \boldsymbol{y})$ is a conjunction of query atoms (ϕ) and negated query atoms ($\neg\phi$).

Definition 2.18 (Query Semantics)**.** *Given a consistent knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ over a signature $\Sigma = \langle \Sigma_I, \Sigma_V, \Sigma_C, \Sigma_D, \Sigma_R, \Sigma_A \rangle$, an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and a query q (either a CQ or an UCQ) over $\mathcal{K}$.*

Given a tuple $\boldsymbol{a}$ of individual names and data values in $\Sigma_I \cup \Sigma_V$ appearing in $\mathcal{A}$ of $\mathcal{K}$, $q(\boldsymbol{a})$ denotes the replacement of each distinguished variable $x_i \in \boldsymbol{x}$ by the respective $a_i \in \boldsymbol{a}$.

The tuple $\boldsymbol{a}$ is a certain answer to the query $q(\boldsymbol{x})$ over the knowledge base $\mathcal{K}$, if $\mathcal{K} \models q(\boldsymbol{a})$, i.e., $\mathcal{I} \models q(\boldsymbol{a})$ holds in every model $\mathcal{I} \in Mod(\mathcal{K})$. The finite set of all certain answers of $q(\boldsymbol{x})$ over $\mathcal{K}$ is denoted by $answ(q(\boldsymbol{x}), \mathcal{K})$.

If the query contains only non-distinguished variables, i.e., $\boldsymbol{x}$ is an empty tuple, then the query is called a Boolean query and the certain answer for $q()$ over $\mathcal{K}$ is $true$ if $\mathcal{I} \models q()$ for every model $\mathcal{I} \in Mod(\mathcal{K})$, i.e., $\mathcal{K} \models q()$, or $false$ otherwise.

The task of finding all certain answers for a (non-Boolean) query $q(\boldsymbol{x})$ over a (consistent) KB $\mathcal{K}$ is called *query answering*. In contrast, the problem of answering a Boolean query $q()$ is called *query entailment* and, although not efficient, query answering for any arbitrary (U)CQ $q(\boldsymbol{x})$ can be linearly reduced to query entailment by a replacement of the distinguished variables $\boldsymbol{x}$ in $q(\boldsymbol{x})$ by each possible assignment combination $\boldsymbol{a}$ and an entailment check of the corresponding Boolean query $q(\boldsymbol{a})$.

As for a DL KB there may exist several models with varying interpretation domains the task of query answering resp. query entailment is necessarily restricted to individual names and data values in $\Sigma_I \cup \Sigma_V$ explicitly named in $\mathcal{A}$ of $\mathcal{K}$ and to query answers that must hold in every model $\mathcal{I} \in Mod(\mathcal{K})$. Moreover, note that the definition of certain answers is restricted to consistent KBs, since if the set of all model is empty ($Mod(\mathcal{K}) = \emptyset$) any arbitrary statement would be trivially satisfied in each model why the certain answers of any (U)CQ would comprise all

possible tuples of individual names and data values for the distinguished variable of the query [Cal+09; Cal+13].

Since query languages are typically more expressive than DLs, query entailment and thus query answering can, in general, not be reduced to any standard reasoning task [OŠ12]. For evaluating the computational complexity of the reasoning task query entailment, and consequently of query answering, the size of the query may be considered as additional input parameter. While in the data complexity and the knowledge base complexity the query is regarded as fixed, the *combined complexity* considers not only the size of the whole KB but also the size of any arbitrary query [Var82; Don+94]. For $\mathcal{SROIQ}(\mathbf{D})$ KBs the data complexity, knowledge base complexity and combined complexity of answering (U)CQs are all unknown and even its decidability [OŠ12; Mot+12]. On the other hand, given a fixed (U)CQ, the knowledge base complexity and the data complexity for query answering *DL-Lite* members mentioned in Section 2.2 remain in NLOGSPACE respectively in AC^0 as already for standard reasoning tasks [Art+09]. Taking, however, an arbitrary (U)CQ into consideration, the combined complexity for query answering in a *DL-Lite* KB becomes NP-complete. Moreover, the usage of inequalities or negations in queries will lead quickly to intractability or even undecidability. In *DL-Lite*$_\mathcal{A}$ for example, a $CQ^{\neq}$ with even one inequality may causes that the data complexity is increased from AC^0 to undecidability as shown by Rosati [Ros07] and Gutiérrez-Basulto et al. [Gut+15].

Remarkable among them are especially the complexity bounds for answering (U)CQs over *DL-Lite* members, i.e., the knowledge base complexity and the data complexity, that both result from the notable property of *DL-Lite* called *FOL-rewritability* (or FOL-reducibility) [Cal+09].

Definition 2.19 (FOL-Rewritability)**.** *The task of query answering in a DL $\mathcal{L}$ is FOL-rewritable, if for any query $q(\boldsymbol{x})$ (either a CQ or an UCQ) over an $\mathcal{L}$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ there exist a FOL query $q^{\mathcal{T}}$ such that $answ(q(\boldsymbol{x}), \langle \mathcal{T}, \mathcal{A} \rangle) = answ(q^{\mathcal{T}}(\boldsymbol{x}), \langle \emptyset, \mathcal{A} \rangle)$ for every ABox $\mathcal{A}$.*

Correspondingly, by compiling the relevant TBox part of the KB into the given query, the task of query answering can be reduced to the plain evaluation of a FOL query, i.e., a SQL query over a relational database. Since this already holds in the absence of a TBox, i.e., $\mathcal{T} = \emptyset$, there is no smaller lower bound than AC^0 for the data complexity and than NP-complete for the combined complexity in any DL. Moreover, it has been shown that the *DL-Lite* family are one of the maximal logics allowing FOL-rewritability and therewith a processing of query answering through standard database technology (like in OBDA and OBDI) [Art+09; OŠ12; Cal+13].

2.4 Semantic Web

The content of the World Wide Web is mainly made up of hyperlinked text and media, intended to be consumed by humans and accessible via keyword-driven search

or link navigation. In contrast, the *Semantic Web* aims an automated locating, retrieval and integration of information on the web by employing an organization and formalization of knowledge [BHL01; HH01]. In order to achieve this objective, common and open standards have been established by the World Wide Web Consortium[5] (W3C) for representing information in a sufficiently formal and structured format. The basis of the Semantic Web is formed by the *Resource Description Framework 1.1*[6] (RDF) which facilitates a flexible description of resources on the web and its relationships by *triples*. A triple consists of a *subject*, a *predicate* and an *object*, where each referring a web resource by using an *Internationalized Resource Identifier* (IRI). Additionally, objects are not necessarily IRIs but can also be some data typed values, called *literals*, and are represented by the pair (l, d), where l denotes a string and d a data type IRI. Moreover, subjects and objects may also represent anonymous resources denoted by *blank nodes*. More precisely, an RDF triple is a tupel $(s, p, o) \in (I \cup B) \times I \times T$, where $T = I \cup B \cup L$ represents the set of *RDF terms* that comprises the three pairwise disjoint and countably infinite sets I of IRIs, B of blank nodes, and L of literals. Hence, the expressivity of RDF is comparable to the expressivity of an ABox of a DL KB. By representing the subjects and objects as nodes, and the predicates as edges, a set of triples form a directed, labeled graph. However, RDF itself determines solely a conceptual data model for expressing information, i.e., statements about resources, in terms of a graph but does not provide a definition of a (sufficient) specific language vocabulary that can be used to formalize semantic characteristics of the described resources. The *RDF Schema 1.1*[7] (RDFS) is a significant (semantic) extension of RDF providing basic constructs that are necessary to model knowledge. The main features of the RDFS vocabulary are especially the means to describe hierarchies of concepts, roles and attributes, as well as domains and ranges of roles respectively attributes.

2.4.1 OWL 2 Web Ontology Language

Even though RDFS already allows modeling (simple) knowledge, its capabilities are rather limited since it is, e.g., not possible to express disjointnesses or number restrictions. Based on and extending RDF(S), the W3C specification of the *OWL 2 Web Ontology Language*[8] (abbreviated as OWL 2) [HPH03] overcomes these limitations and is the de facto standard on the Semantic Web for modeling knowledge. Inspired by DLs, OWL 2 provides indeed very similar constructs and reasoning power, albeit with some diversity in their terms. Thus, in context of Semantic Web respectively OWL 2, knowledge bases are called *ontologies*, *classes* correspond to concepts, *object properties* to roles and *data properties* to attributes. Moreover, OWL 2 provides additionally some more axiom types like to define transitivity, (a)symmetry or (ir)reflexivity for roles, but which are just syntactic sugar since

[5] `https://www.w3.org/`
[6] `https://www.w3.org/RDF/`
[7] `https://www.w3.org/TR/rdf-schema/`
[8] `https://www.w3.org/TR/owl2-overview/`

they all can likewise be expressed by the DL axiom types already introduced in Section 2.1.1. For an extensive introduction of the OWL 2 syntax we refer the interested readers to Motik et al. [MPP12]. Due to the fact that OWL 2 is on the one hand a semantic extension of RDF(S) but is on the other hand also based on DL theory, there exists two different formal semantics. The so called *RDF-based Semantics*[9] extends the RDFS vocabulary by the expressive OWL 2 features and assigns meaning to the RDF graph that is obtained by translating OWL 2 statements, i.e., axioms, into RDF triples. In contrast, applying the semantics directly to OWL 2 statements by relating them to DL axioms is called *Direct Semantics*[10]. Since the RDF-based Semantics is defined for any arbitrary RDF graphs whereas Direct Semantics are restricted to some syntactical conditions in order to satisfy the close connection to DLs, OWL 2 ontologies (i.e., KBs) with RDF-based Semantics are the most expressive language which is hence also called *OWL 2 Full*. The syntactic fragment of OWL 2 (subset of OWL 2 Full) observing the Direct Semantics has the familiar model-theoretic semantics of DLs (see Section 2.1.2) and thus is called *OWL 2 DL*. Except a few negligible exceptions not considered in DL, the DL language $\mathcal{SROIQ}(\mathbf{D})$ represents the logical counterpart of OWL 2 DL [HKS06], why OWL 2 DL has the same complexity bounds, i.e., N2EXPTIME for standard reasoning tasks in the size of the KB. In Contrast, the higher expressivity of OWL 2 Full makes sound and complete reasoning undecidable.

Besides OWL 2 Full and OWL 2 DL, to overcome high computational complexity bounds there exists three lightweight, i.e., tractable OWL 2 profiles[11] (sublanguages of OWL 2 DL): OWL 2 EL, OWL 2 QL and OWL 2 RL, where each profile addresses certain application scenarios. *OWL 2 EL* is based on $\mathcal{EL}^{++}$ [BBL05; BBL08], a DL for which the knowledge base complexity for standard reasoning tasks is in P, why this profile is especially suitable for applications where the size of the TBox is huge, i.e., comprise vast amounts of concepts, roles or attributes. If, on the contrary, the amount of data is big and query answering is considered as the primary reasoning tasks, an application of the *OWL 2 QL* profile would be appropriated. Since this profile is based on the *DL-Lite* family (see Section 2.2), notably *DL-Lite*$_\mathcal{R}$, sound and complete answering of (U)CQs can be performed via relational database systems (cue OBDA/OBDI) and is thus in AC^0 w.r.t. the size of the ABox. However, contrary to the *DL-Lite* family, OWL 2 in general and thus OWL 2 QL does not adopt the UNA. While the computational properties of *DL-Lite*$_\mathcal{R}$ are not influenced by dropping the UNA, for other *DL-Lite* members like *DL-Lite*$_\mathcal{A}$ the complexity bounds strictly depend on the UNA [Art+09]. Because of that, to support such members on top of OWL 2 QL the UNA has to be explicitly axiomatized in OWL 2 QL and additional syntactic restrictions, like functional roles or attributes have to be primitive in *DL-Lite*$_\mathcal{A}$, may are required [MPP12]. Since both, OWL 2 EL and OWL 2 QL, suppose several syntactical restrictions

[9] https://www.w3.org/TR/owl2-rdf-based-semantics/
[10] https://www.w3.org/TR/owl2-direct-semantics/
[11] https://www.w3.org/TR/owl2-profiles/

quite limiting its expressivity, *OWL 2 RL* still aims scalable reasoning but without relinquishing significant expressive power. Encouraged by *Description Logic Programs* (DLPs) [Gro+03], a combination of logic programming, i.e., Horn clauses, and DLs, and by the pD* semantics [Hor05], an initially non-standard RDF(S)-compatible semantics involving a subset of the OWL 1 vocabulary, OWL 2 RL imposes some restrictions on OWL 2 constructs in order to support rule-based reasoning. As a result, by utilizing First-Order Horn Logic rules and confining the consideration only to individuals that are explicitly stated in a KB, i.e., in its ABox, all standard reasoning tasks are tractable, i.e., P-complete w.r.t. the size of the KB. For a more extensive introduction to the lightweight OWL 2 profiles we refer the interested reader to the work of Krötzsch [Krö12].

2.4.2 SPARQL

For querying the Semantic Web, i.e., RDF graphs, the W3C has specified the *SPARQL 1.1 Query Language*[12]. Basically, the evaluation of SPARQL queries is based on graph pattern matching, where a pattern is intuitively an RDF triple potentially comprising some variables. More precisely, given a set $T = I \cup B \cup L$ of RDF terms, where I denotes IRIs, B represents blank nodes and L are literals as before, and a countably infinite set $\mathcal{V}$ of variables that is disjoint from T, a *basic graph pattern* (BGP) is described by a finite set of triple patterns, where each *triple pattern* is a tupel of $(T \cup \mathcal{V}) \times (I \cup \mathcal{V}) \times (T \cup \mathcal{V})$ [PAG06]. Following the notation of Pérez et al. [PAG06] and concentrating on the core feature of SPARQL, a more complex *graph pattern* P can be expressed according to the syntax

$$P ::= B \mid \textsf{FILTER}(P, F) \mid \textsf{UNION}(P_1, P_2) \mid \textsf{JOIN}(P_1, P_2)\,, \tag{2.12}$$

where B is a basic graph pattern, P_1, P_2 are graph patterns and F is a filter condition expressed in terms of a formula constructed by using relational expressions of the form $(I \cup L \cup V) \times \{<, \leq, =, \neq, \geq, >\} \times (I \cup L \cup V)$, possibly connected via the logical operators $\wedge, \vee, \neg$. For graph patterns with $\textsf{FILTER}(P, F)$ we assume that every variable in F occurs in P, such that $var(F) \subseteq var(P)$, where $var(F)$ and $var(P)$ denotes the set of all variables in F respectively in P. Finally, a *SPARQL query* $Q ::= (P, V)$ consists of a graph pattern P and a set of variables $V \subseteq \mathcal{V}$ specifying the distinguished variables (also called answer or result variables). Besides the mentioned operators, SPARQL provides some further features like the declaration of optional results, bindings (assignments) of variables, aggregation functions, ordering functions, result modifications, and arithmetic operators or other functions especially for filter conditions. However, since we use the Semantic Web standard query language SPARQL just in our experimental evaluation for expressing and evaluating a particular type of CQs, we restrict within the scope of this thesis only to those SPARQL features that are necessary to implement this

[12] `https://www.w3.org/TR/sparql11-overview/`

task. For a detailed description of SPARQL and its syntax we refer to the official W3C recommendations [Bui+13; HS13] and other works like [PAG06; HKR09].

The results to a SPARQL query $Q ::= (P, V)$, i.e., the value bindings of the distinguished variables to RDF terms are given by partial maps $\mu\colon V \to T$, called *solution mappings*. The (possibly empty) domain of μ is the subset of V in which μ is defined and is denoted by $dom(\mu)$. Contrary to the specification of SPARQL that adopts the multiset (or bag) semantics, following Pérez et al. [PAG06] and Kontchakov et al. [Kon+14], we use the set-based semantics, similar to the answers as for (U)CQs. While a multiset may comprise an element more than once, under the set-based semantics the results to a query are unique and corresponds to the SPARQL sequence modifier DISTINCT[13]. Given an RDF graph G and a basic graph pattern B, the evaluation of B over G is the *answer set* $[\![B]\!]_G$ of solution mappings and is given by

$$[\![B]\!]_G = \{\mu\colon var(B) \to T \mid \mu(B) \subseteq G\}\,, \tag{2.13}$$

where $\mu(B)$ represents the substitution of each variable $v \in var(B)$ by $\mu(v)$. Based on that, the answer set $[\![P]\!]_G$ to a (complex) graph pattern P over G is inductively defined by

$$\textsf{FILTER}([\![P]\!]_G, F) = \{\mu \in [\![P]\!]_G \mid F^\mu = true\}\,, \tag{2.14}$$

$$\textsf{UNION}([\![P_1]\!]_G, [\![P_2]\!]_G) = \{\mu \mid \mu \in [\![P_1]\!]_G \cup [\![P_2]\!]_G\}\,, \tag{2.15}$$

$$\textsf{JOIN}([\![P_1]\!]_G, [\![P_2]\!]_G) = \{\mu_1 \oplus \mu_2 \mid \mu_1 \in [\![P_1]\!]_G \sim \mu_2 \in [\![P_2]\!]_G\}\,, \tag{2.16}$$

where F is a filter and $[\![P]\!]_G$, $[\![P_1]\!]_G$ and $[\![P_2]\!]_G$ are answer sets. Applying a filter F to a solution mapping μ is denoted by F^μ and its *truth-value* $F^s \in \{true, false\}$ is given by the classical truth-value of

$$\begin{aligned}((I \cup L \cup V) \times \{<, \leq, =, \neq, \geq, >\} \times (I \cup L \cup V))^\mu \\ = (I \cup L \cup \mu(V)) \times \{<, \leq, =, \neq, \geq, >\} \times (I \cup L \cup \mu(V))\,, \\ (\neg F)^\mu = \neg F^\mu\,, \\ (F_1 \times \{\wedge, \vee\} \times F_2)^\mu = F_1^\mu \times \{\wedge, \vee\} \times F_2^\mu\,.\end{aligned}$$

For the operation JOIN, the notation $\mu_1 \sim \mu_2$ states that the solution mappings μ_1 and μ_2 are *compatible*, i.e., $\mu_1(v) = \mu_2(v)$ for each $v \in dom(\mu_1) \cap dom(\mu_2)$ [Kon+14; Ahm+15]. In this case $\mu_1 \oplus \mu_2$ with domain $dom(\mu_1) \cup dom(\mu_2)$ is a solution mapping as well such that $(\mu_1 \oplus \mu_2)\colon v \to \mu_1(v)$ if $v \in dom(\mu_1)$ and $(\mu_1 \oplus \mu_2)\colon v \to \mu_2(v)$ otherwise. According to the set of variables V of a SPARQL query $Q ::= (P, V)$, the answer set $[\![Q]\!]_G$ to Q over G correspond to the solution mappings in $[\![P]\!]_G$ restricted to the variables in V.

The above mentioned semantics of SPARQL queries is known as *simple entailment*. However, to be able to consider on the evaluation of SPARQL queries also implicit statements that are entailed according to the semantic interpretation

[13] `https://www.w3.org/TR/sparql11-query/`

adopted by an RDF graph (such as RDF(S) Semantics, OWL 2 RDF-based Semantics or OWL 2 Direct Semantics), the specification of SPARQL 1.1 correspondingly defines several so called *entailment regimes*[14]. By redefining the evaluation of BGPs, an entailment regime extends the (basic) graph pattern matching by determining entailments that contribute to the answer set to a BGP. Formally, by redefining (2.13), the answer set $[\![B]\!]_G^{\mathsf{E}}$ to a BGP B over an RDF graph G under an entailment regime E is given by

$$[\![B]\!]_G^{\mathsf{E}} = \{\mu \colon var(B) \to T \mid G \models_{\mathsf{E}} \mu(B)\}\,, \tag{2.17}$$

where $\models_{\mathsf{E}}$ denotes the satisfaction relation given by E [Kon+14]. Based on that, the answer set $[\![Q]\!]_G^{\mathsf{E}}$ of solution mappings to a SPARQL query $Q ::= (P, V)$ over G under E is inductively defined as before, since the SPARQL operations do not depend on the underlying entailment regime.

Although it seems at first glance that SPARQL queries are very similar to UCQs, there exists some considerable differences in both syntax and semantics. While in UCQs a query atom is either unary or binary, variables in BGPs may occur at any position of the triple and hence enabling in contrast a querying of the TBox as well. Moreover, the arbitrary use of variables in BGPs allows the formulation of much more expressive queries than UCQs. Another significant variation is that UCQs follow the OWA such that non-distinguished non-shared variables do not necessarily have to be assigned to known entities, while as stated in (2.13) and (2.17) for a SPARQL query it is required that all variables of a BGP are bound, i.e., are mapped to known RDF terms (as per CWA). In addition, according to Definition 2.18, on query answering an UCQ is considered holistically as an open FOL formula that have to be satisfied by the KB, i.e., by all its models. However, in SPARQL queries the UNION operations are downstream to the separate satisfiability checks of each BGP according to (2.17), why the answer set $[\![Q]\!]_G^{\mathsf{E}}$ is in contrast to the certain answers $answ(q(\boldsymbol{x}), \mathcal{K})$ not necessarily complete [Ahm+15]. In particular, this is for example the case if the TBox comprises an axiom of the form $C \sqsubseteq C_1 \sqcup C_2$, the ABox an assertion $C(\sigma_I)$ and a SPARQL query the graph pattern $P ::= \mathsf{UNION}(C_1, C_2)$. Since according to Equation (2.16) each BGP is checked independently of each other, i.e., since $G \not\models_{\mathsf{E}} C_1(\sigma_I)$ and $G \not\models_{\mathsf{E}} C_2(\sigma_I)$ the answer set $[\![P]\!]_G^{\mathsf{E}}$ will not contain σ_I whereas according to Definition 2.18 the certain answers to a query $q(x) = C_1(x) \vee C_2(x)$ indeed comprise σ_I [Pog16]. However, as this variation to UCQs comes only into effect if the TBox comprises disjunctions on the right-hand side of inclusion axioms, some (lightweight) DLs such as the *DL-Lite* members mentioned in Section 2.2 are not affected by that.

[14] https://www.w3.org/TR/sparql11-entailment/

Chapter 3

Problem Statement

After having introduced in the previous chapter the required formal foundations we will now clarify the motivating context of this work in Section 3.1. Subsequently, in Section 3.2, after discussing the general problem of information integration based on DLs, we formally define the term federated KB and clarify the research problem addressed by this work. In the last part of this chapter (Section 3.3) an example is given in order to illustrate on the one hand our motivation and on the other hand the problems and the proposed approaches in the later part of this thesis.

Some of the contents in this chapter originate from previously published papers [Nol+16; Nol+17]. This applies in particular to the definition of a federated KB in Section 3.2 and the example proposed in Section 3.3. However, for the sake of a unified notation within this thesis the definition of a federated KB is presented slightly different and the example is expanded by a few statements covering some additional cases.

3.1 Linked Open Data

Dealing with distributed and heterogeneous data sources has become an important research topic since the variety and amount of available data grows continuously in all aspects, such as in the business environment or in the public sector. Aside from that, following the idea of the Semantic Web, initiatives like DBpedia [Aue+07], an extraction of Wikipedia contents to RDF and OWL, have been launched. As a result, more and more sources are published on the Web via Semantic Web standards and led to a rapid increase of interlinked RDF datasets and OWL ontologies (i.e., KBs). The resulting network of linked sources is called the *Linked Open Data* (LOD) cloud[1] (also known as the *Web of data*) and currently comprises 1,234 (mostly distributed) datasets (as of January 2019) from previously 12 in 2007 [McC]. Note that an RDF dataset typically provided by a triple store represents in general a collection of RDF graphs, i.e., comprises a default graph and a possibly

[1] `http://linkeddata.org/`

empty set of named graphs that are identified by individual IRIs [CWL14]. Nevertheless, without loss of generality, we assume for simplicity that each triple store, i.e., data source, provides an RDF dataset comprising just a default graph.

The network of KBs, resulting from the provided links of and between LOD sources at the ABox but also at the TBox level, may suggest at first glance an exchange of information across different sources and domains. However, the LOD cloud lacks of semantic homogeneity and sufficient connectedness and thus of semantic interoperability. Since many LOD sources have been automatically generated from different semi-structured or even unstructured datasets but without any commitment for a shared signature or something similar, the sources lacking in a sufficient degree of linkings to other sources and varying in its details of semantic modeling and expressivity. As a result, even if two sources (KBs) are out of the same domain of interest, its data is not unusually described by different TBoxes with no or even slightly overlappings. Moreover, besides the intensional level (also called *schema level*) the heterogeneity also affects the extensional knowledge part (also denoted by *instance level*) such that same entities are denoted by various identifiers in different sources but without any (or sufficient) interlinkings representing its equalities.

3.2 Ontology-Based Information Integration

Hence, the LOD cloud can be critically seen as a semantic heterogeneous and loosely coupled network of KBs where the integration of information (and knowledge) from different sources is a major challenge. To address this shortcoming, *ontology matching* aims to find *alignments*, i.e., a set of correspondences (mappings) for semantically related TBox expressions of different KBs in terms of equivalence, inclusion or disjointness axioms. While ontology matching addresses primarily the alignment of TBoxes, *data interlinking* (also known as *instance matching*) pursuing similar objectives but focusing the extensional level by detecting and linking individuals in different sources that represent the same entity. Despite differing orientations both tasks mutually benefit from each other if they are applied collaboratively [SE11]. Because of the rising demand for combining information from different autonomous and heterogeneous sources, several automated approaches for ontology matching and data interlinking have been proposed. Corresponding surveys of approaches for ontology matching can be found for example in [ES13; SE13; ORG15; OK18] respectively for data interlinking in [KR10; Wöl+11; FNS11; Nen+17].

Due to the fact that some data sources are out of different domains of interest or rely on different interpretations of the same or similar domain, problems that pertain specifically to information integration like accessing and querying several heterogeneous data sources in an integrated manner usually require a global and unified description (TBox) of the federated domain of interest [SH05]. Thus, such a global TBox not just comprises alignments mapping to semantically related

expressions of the source-specific TBoxes but may provide a generalized description specifically tailored to the considered domain capturing complex interrelationships and constraints. According to the information integration paradigm [Wac+01; Len02], a *global TBox* serves as a shared conceptual view that comprises and possibly extends the semantics, i.e., the *local TBox*, of each integrated data source. Mappings between the global TBox and the different local TBoxes describing the diverse data sources are used to access and combine information from multiple independent, distributed and heterogeneous sources without being aware of each source-specific terminology (local TBox) or referring to the related signature of each data source. As a consequence, the global TBox represent an interface for accessing distributed information in a unified and integrated way, similar to OBDA respectively OBDI. However, while in OBDA respectively OBDI the data sources are typically considered as relational databases [Cal+07a; Cal+09; Cal+18], the approach of integrating data sources that are representing KBs addresses more the information level than the data level and is thus rather denoted as *ontology-based information integration* (OBII) [Wac+01].

Basically, for realizing an information integration there exist two different approaches, called materialized and virtual integration [CDL02; Len02]. In case of a *materialized integration* the information is extracted from each data source and loaded into a global information store why this may also be denoted as warehouse approach. On the contrary, in a *virtual integration* (also called mediator approach) the information stays in the original sources and the mappings are used to retrieve the required information on-the-fly. While in the former case the ABox represented by the global information store can be considered as a modifiable set of assertions, the latter offers more flexibility with respect to handling dynamic data (resp. information) and adding or removing additional sources. Following the vision of the Semantic Web, we adopt in context of this work the virtual integration approach. However, even if we focus on a virtual integration, the findings of this thesis can be easily adopted to a materialized integration where each integrated data source is represented at the global data source by a named RDF graph.

The virtual integration of distributed and heterogeneous KBs can be described in terms of a *federated knowledge base* which can be formally defined by

Definition 3.1 (Federated Knowledge Base)**.** *Given a set $\boldsymbol{\mathcal{K}} = \{\mathcal{K}_1, \ldots, \mathcal{K}_N\}$ of knowledge bases with $\mathcal{K}_n = \langle \mathcal{T}_n, \mathcal{A}_n \rangle$ over a signature Σ_n and $n = 1, \ldots, N$. Let $\mathcal{T}_I$ be an intermediary TBox over the signature $\Sigma_F = \Sigma_I \cup \bigcup_{n=1,\ldots,N} \Sigma_n$, where Σ_I denotes an additional signature used in $\mathcal{T}_I$ for the intermediary description of the federated domain of interest. Then, the federated knowledge base $\mathcal{K}_F$ is a knowledge base with $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$ over signature Σ_F, where $\mathcal{T}_F$ denotes the federated (global) TBox that is given with $\mathcal{T}_F = \mathcal{T}_I \cup \bigcup_{n=1,\ldots,N} \mathcal{T}_n$ and $\mathcal{A}_F$ is the federated ABox which is defined by $\mathcal{A}_F = \bigcup_{n=1,\ldots,N} \mathcal{A}_n$.*

Accordingly, the specific TBox of each source is integrated in the federated KB by alignments (mappings) within the intermediary TBox such that the federated (global) TBox $\mathcal{T}_F$ completely describes the federated domain of interest. Note

also that if the KBs, i.e., its ABoxes integrated in the federated KB are lacking in a sufficient degree of individual linking, the results of an approach for data interlinking can be published on an additional data source within the federated KB.

However, especially in the federation of heterogeneous KBs dealing with contradictive statements (incoherence or inconsistency) becomes a particular challenge. Generally, the process of identifying contradictive statements in DL KBs, generating explanations for them and proposing some repair plans to resolve found contradictions is called *knowledge base debugging* (or *ontology debugging*) [SC03]. Since the creation of a federated TBox relies not only on automated ontology matching techniques but usually requires manual effort for modeling the federated domain of interest, the TBox $\mathcal{T}_F$ of a federated KB can usually be assumed to be of high or at least sufficient quality and thus with less prone to errors. Moreover, there exist several approaches for detecting and resolving modeling errors, i.e., incoherence, like [SC03; HVT05; Haa+05; Kal06; Sch+07; Moo10]. Especially in the context of ontology matching, Meilicke [Mei11] proposed a framework for repairing incoherent ontology alignments based on diagnosis theory. Other works in this context are for example [Ji+09; QJH09; JC11; San+15]. However, even though an ontology matching respectively corresponding alignments will integrate autonomous KBs, i.e., its intensional part into the federated domain of interest, at the extensional part there are likely to be more contradictive assertions the larger the number of integrated KBs. Possible causes might be that the integrated data sources may are based on semi-structured or even unstructured data and may vary in the timeliness, accuracy and completeness of the provided information[2]. Thus, a KB or even the federated TBox may result in additional (explicit or implicit) information that may contradict the information already known from other sources. So even supposing that each integrated data source is self-consistent, the federated knowledge base can still be inconsistent.

In the scope of this thesis we study the research topic of automated debugging in relation to OBII targeting the identification and treatment of inconsistency in federated KBs. Since we primarily focus on inconsistency, we do not address integration problems related to incoherence and thus assume that the federated TBox $\mathcal{T}_F$ is *"semantically correct"*, i.e., is free from any modeling errors. While, to the best of our knowledge, previous approaches such as [Cal+07b; HQ07; Stu08; Stu13; SC03; Sch+07; Kal06][3] solely tackle the debugging of single DL KBs, *federated knowledge base debugging* in the light of OBII is a new problem and the contributions of this work are novel for DLs and the Semantic Web.

3.3 Running Example

Let us now introduce an example which is elucidating the motivation of this work. and that will be used in the further course of this thesis to illustrate the problems

[2]For works on data quality in the context of LOD see, e.g., [Zav+16]

[3]For details see Section 5.6 and Section 4.4

and the application of the proposed approaches.

Given a federated knowledge base $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$ that integrates a set of three distributed KBs $\{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$. Let $\mathcal{T}_F$ be the federated TBox of $\mathcal{K}_F$ that describes the domain of library science and comprises the following axioms:

$$\begin{aligned} \mathit{Book} \sqcup \mathit{Paper} &\sqsubseteq \mathit{Publication} & \mathit{Paper} &\sqsubseteq \neg \mathit{Book} \\ \mathit{Proceedings} &\sqsubseteq \mathit{Book} & \mathit{Publication} &\sqsubseteq \neg \mathit{SlideSet} \\ \exists \mathit{publishedIn} &\sqsubseteq \mathit{Paper} & \exists \mathit{publishedIn}^- &\sqsubseteq \mathit{Proceedings} \\ \exists \mathit{slideSetOf} &\sqsubseteq \mathit{SlideSet} & \exists \mathit{slideSetOf}^- &\sqsubseteq \mathit{Paper} \\ \delta(\mathit{edition}) &\sqsubseteq \mathit{Book} & \rho(\mathit{edition}) &\sqsubseteq \texttt{xsd:integer} \\ (\text{funct } \mathit{publishedIn}) & & (\text{funct } \mathit{edition}) & \end{aligned}$$

Note that according to Definition 3.1, the federated TBox $\mathcal{T}_F$ of $\mathcal{K}_F$ is given by the union of the source-specific TBoxes $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ and an intermediary one $\mathcal{T}_I$ aligning source-specific TBox expressions and possibly comprising generalized descriptions of the considered domain of interest. However, for the sake of simplicity we assume for our example that there is only one TBox that is used in all integrated KBs, i.e., $\mathcal{T}_F = \mathcal{T}_1 = \mathcal{T}_2 = \mathcal{T}_3$. Since the example can be easily extended to the more general case where each KB comprises a different terminology, this assumption is without loss of generality.

Moreover, let $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ denote the ABoxes of the three integrated KBs $\{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$ where the respective assertions of the individual ABoxes are given by the following table:

$\mathcal{A}_1$		$\mathcal{A}_2$	
$\mathit{Paper}(\mathbf{I1})$	(α_1)	$\mathit{Paper}(\mathbf{I1})$	(β_1)
$\mathit{publishedIn}(\mathbf{I1}, \mathbf{C1})$	(α_2)	$\mathit{Proceedings}(\mathbf{I1})$	(β_2)
$\mathit{Paper}(\mathbf{I2})$	(α_3)	$\mathit{publishedIn}(\mathbf{C1}, \mathbf{I1})$	(β_3)
$\mathit{publishedIn}(\mathbf{I4}, \mathbf{I5})$	(α_4)	$\mathit{publishedIn}(\mathbf{I4}, \mathbf{C2})$	(β_4)
$\mathit{publishedIn}(\mathbf{C2}, \mathbf{I5})$	(α_5)	$\mathit{Proceedings}(\mathbf{C2})$	(β_5)
$\mathit{publishedIn}(\mathbf{I6}, \mathbf{C3})$	(α_6)	$\mathit{publishedIn}(\mathbf{I6}, \mathbf{C3})$	(β_6)
$\mathit{Paper}(\mathbf{I6})$	(α_7)	$\mathit{Proceedings}(\mathbf{C3})$	(β_7)
$\mathit{Paper}(\mathbf{I7})$	(α_8)	$\mathit{Paper}(\mathbf{C4})$	(β_8)
$\mathit{edition}(\mathbf{B1}, \texttt{"2"\^\^xsd:integer})$	(α_9)	$\mathit{edition}(\mathbf{B1}, \texttt{"2nd"\^\^xsd:string})$	(β_9)

$\mathcal{A}_3$	
$\mathit{SlideSet}(\mathbf{I1})$	(γ_1)
$\mathit{SlideSet}(\mathbf{I2})$	(γ_2)
$\mathit{slideSetOf}(\mathbf{I2}, \mathbf{I3})$	(γ_3)
$\mathit{SlideSet}(\mathbf{I4})$	(γ_4)
$\mathit{slideSetOf}(\mathbf{I5}, \mathbf{C2})$	(γ_5)
$\mathit{Proceedings}(\mathbf{C3})$	(γ_6)
$\mathit{Proceedings}(\mathbf{C4})$	(γ_7)
$\mathit{slideSetOf}(\mathbf{C4}, \mathbf{I6})$	(γ_8)

By considering each data source separately, we can see that only KB_2 comprise contradicting assertions, e.g., β_1 contradicts β_2 and β_3, while KB_1 and KB_3 are both self-consistent. However, if, on the other hand, we look at the federated KB, the number of logical conflicts increases significantly. For example, the assertion α_1 in $\mathcal{A}_1$ stating that $\mathbf{I1}$ is a $Paper$ is obviously in contradiction to the assertions $SlideSet(\mathbf{I1})$ (γ_1) of $\mathcal{A}_2$ due to the axioms $Publication \sqsubseteq \neg SlideSet$ and $Paper \sqsubseteq Publication$ in $\mathcal{T}_F$. Moreover, as the assertion $Paper(\mathbf{I1})$ can also be found in $\mathcal{A}_2$ (assertion β_1) and additionally can be entailed from the assertion $publishedIn(\mathbf{I1}, \mathbf{C1})$ (α_2) of $\mathcal{A}_1$, both are contradicting (γ_1) as well.

But besides the fact that the amount of conflicting assertions is potentially increased, another effect is that the resolution of contradicting assertions could be influenced by the federation of several KBs as well. If we look for example only at the conflicting assertions β_1, β_2 and β_3 in $\mathcal{A}_2$, it seems that the assertion β_1 is presumably wrong since both, β_2 and β_3 are contradicting β_1. However, taking now $\mathcal{A}_1$ into consideration, α_1 and α_2 are also opposed to β_2 and β_3 so that one might get the impression that rather the assertions β_2 and β_3 should be declared as wrong.

Part II

Theory and Methods

Chapter 4

Federated Inconsistency Detection and Explanation

As shown by our example of Section 3.3, the integration of distributed and heterogeneous KBs in context of the Semantic Web raise the need for federated KB debugging. But before an appropriated diagnosis and repair of inconsistency can be performed, reasoning for inconsistency detection is the initial step in order to provide complete results. However, due to the loosely coupled network of data sources in context of OBII and the resulting large amounts of extensional knowledge that have to be handled, reasoning over a federated KB becomes a challenging problem. Hence, we initially discuss in Section 4.1 how interoperability over integrated sources can be implemented with the objective of performing federated reasoning and justify why we focus on *DL-Lite*$_{\mathcal{A}}$. Based on different types of logical conflicts that may occur in a *DL-Lite*$_{\mathcal{A}}$ KB we discuss in the subsequent parts of this chapter our approach of inconsistency detection in federated *DL-Lite*$_{\mathcal{A}}$ KB in terms of federated query answering (Section 4.2) and further describe the generation of corresponding explanations for the detected contradictions (Section 4.3). Finally, before concluding this chapter in Section 4.5, we compare approaches related to our work of this chapter in Section 4.4.

Most of the content of this chapter reflects our approach of efficiently detecting inconsistency in federated KBs which is based on the work of Lembo et al. [Lem+11] and Calvanese et al. [Cal+07b] and was first presented in [Nol+14] and subsequently in [Nol+16]. Especially the contents of Section 4.2.1, Section 4.2.2 and Section 4.2.3 originate from our previous published work [Nol+14] and are now presented in a unified manner within this thesis. However, while only pragmatically introduced in [Nol+14] and [Nol+16], the formal definition of federated querying in Section 4.2.4 as well as definitions like source-related ABox assertions, federated clash querying or back-translation in Section 4.3 are presented in this thesis for the first time. Likewise, the algorithm for inconsistency detection in Section 4.3 originating from [Nol+14] is correspondingly modified for the purpose of a complete and unified notation within this thesis.

4.1 Reasoning in Federated Knowledge Bases

One of the main intentions of knowledge representation based on DLs is obviously to perform reasoning tasks and in context of OBII a particular importance is attached to query answering. Traditionally, the implementation of reasoning can be based on a *bottom-up approach* (also called *forward chaining*), a *top-down approach* (also called *backward chaining*) or a combination of both (hybrid approach) [CGT89; CGT90; KD11]. Bottom-up approaches start from statements (assertions or axioms) explicitly stated in a KB and derive corresponding implications. Hence, all logical consequences can be cached (i.e., materialized) such that these statements can be used for efficiently processing a reasoning task. On the contrary, top-down approaches act in the opposite direction by starting from a reasoning task that is expressed in terms of a statement or a query and by verifying the given statement or finding all possible answers to the query with respect to the KB. As only those statements are entailed at runtime that are necessary for a particular reasoning task (also called 'pay-as-you-go' behavior) the KB is kept in its original state.

Since in context of an OBII, where several distributed and heterogeneous KBs are virtually integrated by a federated KB, each integrated KB commonly comprises only a subset of the federated TBox why a complete reasoning on the federated KB cannot be performed at the data source level but exclusively at the intermediary level (centralized). However, as we have to deal with a wealth of information originating from a possible huge amount of different data sources the requirements regarding scalability necessitate to keep the amount of data that has to be transferred from the integrated sources as low as possible, why the application of a top-down approach is preferred. As a consequence, there wont be a need to cache a significant huge amount of derived statements and the federated KB can essentially be kept virtual. To perform a federated reasoning over this loosely coupled network of distributed and autonomous data sources in terms of a selective retrieval of information from several sources we rely on the task of query answering. Thus, all other reasoning tasks, in particular those involving the ABox (such as inconsistency detection), have to be formulated in terms of queries. Similar to OBDA, before a query is evaluated, all relevant parts of the federated TBox comprising and extending the semantics of each integrated KB are compiled into the query such that the rewritten query only have to be evaluated against the assertions explicitly stated within the ABoxes of the integrated sources. Hence, for a complete answering of queries (i.e., finding all certain answers) no further reasoning have to be performed at the data source level and we may consider the integrated sources less as KBs but rather as simple repositories, i.e., KBs with an empty TBox.[1] Since the ABox is getting left out during query rewriting this method has proven to be crucial in applications where the ABox is getting huge and the intentional knowl-

[1] Note that this assumption is without loss of generalizability and the proposed work can be simply extended to the case where the integrated sources may support any entailment regime as long as these sources are self-consistent.

edge part is used to access external sources comprising the extensional knowledge part [Art+09; Lem+11; Cal+13]. However, to remain tractable in federated query answering w.r.t. data complexity and knowledge base complexity, the DL language of the federated KB has to be FOL-rewritable. Because of that, we limit the scope of this thesis to *DL-Lite*$_\mathcal{A}$ as one representative of lightweight DLs supporting tractable query answering, i.e., FOL-rewritability, and being specially tailored for dealing with and reasoning on large ABoxes. Moreover, since LOD sources, i.e., its TBoxes rather have a shallow structure describing a large amount of generic conceptualizations and hence are ordinarily of low expressivity [Dam+10; VN11], *DL-Lite*$_\mathcal{A}$ is sufficient in order to perform a complete reasoning (such as inconsistency detection) in context of the Semantic Web.

As already explained in Section 2.2, *DL-Lite*$_\mathcal{A}$ respectively the *DL-Lite* family imposes the UNA. But as a federated KB virtually integrates autonomous and hence heterogeneous KBs the UNA usually does not hold. Because of that it is not only essential to include alignments of the different TBoxes but also to consider individual equality statements ($\sigma_I \approx \sigma'_I$) that may already exist in the data sources or may be obtained by applying an approach for data interlinking. Especially in context of the Semantic Web, the explicit object property `owl:sameAs` of OWL is extensively used to denote that different individual names represent the same entity. However, by adding individual equality statements in *DL-Lite* (respectively in the corresponding OWL 2 QL profile) the FOL-rewritability will be lost in general [Art+09]. Although Calvanese et al. [Cal+15] could identify a set of restrictions under which it is even possible to take individual equality statements into account for query answering and the FOL-rewritability is retained, for the sake of simplicity but without loss of generality we will impose in the following the UNA.

4.2 Inconsistency Detection in Federated *DL-Lite*$_\mathcal{A}$ KBs

Having restricted our focus on *DL-Lite*$_\mathcal{A}$, we will now first discuss which types of logical conflicts may occur in a *DL-Lite*$_\mathcal{A}$ KB in Section 4.2.1. Based on that, we define in the subsequent Section 4.2.2 a corresponding translation function in order to generate queries for inconsistency detection out of the relevant axioms in $\mathcal{T}_F$. Before finally defining the federated evaluation of queries in Section 4.2.4, we first discuss in Section 4.2.3 the expansion of queries in terms of a top-down reasoning approach, i.e., for the purpose of a complete answering of queries.

4.2.1 Inconsistency in *DL-Lite*$_\mathcal{A}$ Knowledge Bases

Given a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ over a signature $\Sigma = \langle \Sigma_I, \Sigma_V, \Sigma_C, \Sigma_D, \Sigma_R, \Sigma_A \rangle$ for which there exists at least one interpretation $\mathcal{I}$ that satisfies the TBox $\mathcal{T}$ of $\mathcal{K}$, i.e., $\mathcal{I} \models \mathcal{T}$. We denote the set of all interpretations satisfying $\mathcal{T}$ by $Mod(\mathcal{T})$. To determine if $\mathcal{K}$ is inconsistent, according to Definition 2.14 we have to check if there exists no interpretation $\mathcal{I} \in Mod(\mathcal{T})$ for which $\mathcal{I} \models \mathcal{A}$ holds and thus

implies that $Mod(\mathcal{K}) = \emptyset$. Hence, the detection of inconsistency can be performed by searching for assertions in $\mathcal{A}$ causing a logical conflict (also known as *clash*), i.e., violating $\mathcal{T}$ and thus resulting in $Mod(\mathcal{K}) = \emptyset$.

In context of a *DL-Lite*$_\mathcal{A}$ KB, assertions may contradict negative inclusion axioms, value-domain axioms (i.e., attribute ranges) and functionality assertion axioms for roles or attributes. More precisely, according to the work of Lembo et al. [Lem+11] there exist the following six different cases where ABox assertions may contradict the TBox and thus result in inconsistency.

(i) ABox assertions on unsatisfiable elements in $\mathcal{T}$, i.e., $\mathcal{T} \models C \sqsubseteq \neg C$ and $C(\sigma_I) \in \mathcal{A}$, where C is an atomic concept (concept name) $\sigma_C \in \Sigma_C$ and $\sigma_I \in \Sigma_I$ is an individual name, respectively $\mathcal{T} \models R \sqsubseteq \neg R$ and $R(\sigma_I, \sigma'_I) \in \mathcal{A}$, where R is an atomic role (role name) $\sigma_R \in \Sigma_R$ and $\sigma_I, \sigma'_I \in \Sigma_I$ are individual names, or $\mathcal{T} \models A \sqsubseteq \neg A$ and $A(\sigma_I, \sigma_V) \in \mathcal{A}$, where A is an atomic attribute (attribute name) $\sigma_A \in \Sigma_A$, $\sigma_I \in \Sigma_I$ is an individual name and $\sigma_V \in \Sigma_V$ a data value.

(ii) ABox assertions on roles that are restricted on interrelating individuals, i.e., $\mathcal{T} \models R \sqsubseteq \neg R^-$ or $\mathcal{T} \models \exists R \sqsubseteq \neg \exists R^-$ and $R(\sigma_I, \sigma_I) \in \mathcal{A}$ resp. $\{R(\sigma_I, \sigma'_I), R(\sigma'_I, \sigma_I)\} \subseteq \mathcal{A}$, where R is an atomic role $\sigma_R \in \Sigma_R$ and $\sigma_I, \sigma'_I \in \Sigma_I$ are individual names.

(iii) Attribute assertions comprising a data value of an incorrect data type, i.e., $\mathcal{T} \models \rho(A) \sqsubseteq D$, $A(\sigma_I, \sigma_V) \in \mathcal{A}$ and $\sigma_V^{\mathcal{I}} \notin D^{\mathcal{I}}$, where A is an atomic attribute $\sigma_A \in \Sigma_A$, D a value-domain $\sigma_D \in \Sigma_D$, $\sigma_I \in \Sigma_I$ an individual name and $\sigma_V \in \Sigma_V$ a data value.

(iv) ABox assertions contradicting a negative inclusion in $\mathcal{T}$ such that, e.g., $\mathcal{T} \models C \sqsubseteq \neg \exists R$ and $\{C(\sigma_I), R(\sigma_I, \sigma'_I)\} \subseteq \mathcal{A}$, where C denotes an atomic concept $\sigma_C \in \Sigma_C$, R is an atomic role $\sigma_R \in \Sigma_R$ and $\sigma_I, \sigma'_I \in \Sigma_I$ are individual names.

(v) ABox assertions violating the functionality constraint of a role R such that $(\mathsf{funct}\ R) \in \mathcal{T}$ and $\{R(\sigma_I, \sigma'_I), R(\sigma_I, \sigma''_I)\} \subseteq \mathcal{A}$, respectively the functionality constraint of an inverse role such that $(\mathsf{funct}\ R^-) \in \mathcal{T}$ and $\{R(\sigma'_I, \sigma_I), R(\sigma''_I, \sigma_I)\} \subseteq \mathcal{A}$, where R is an atomic role $\sigma_R \in \Sigma_R$, $\sigma_I, \sigma'_I, \sigma''_I \in \Sigma_I$ are individual names and $\sigma'_I \neq \sigma''_I$.

(vi) ABox assertions violating the functionality constraint of an attribute A such that $(\mathsf{funct}\ A) \in \mathcal{T}$ and $\{A(\sigma_I, \sigma_V), A(\sigma_I, \sigma'_V)\} \subseteq \mathcal{A}$, where A is an atomic attribute $\sigma_A \in \Sigma_A$, $\sigma_I \in \Sigma_I$ is an individual name, $\sigma_V, \sigma'_V \in \Sigma_V$ are data values and $\sigma_V \neq \sigma'_V$.

4.2.2 Clash Query Generation

According to the clash types given above, we can now define appropriate queries in order to perform inconsistency detection in a federated KB relying on query

answering. More precisely, based on the work of Calvanese et al. [Cal+07b] we are able to define a translation function τ that generates *clash queries*, i.e., open FOL formulas in terms of CQ bodies, out of negative inclusion axioms, functionality assertion axioms and value-domain inclusion axioms in $\mathcal{T}$.

Definition 4.1 (Translation Function τ)**.** *Given a DL-Lite$_\mathcal{A}$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ over a signature $\Sigma = \langle \Sigma_I, \Sigma_V, \Sigma_C, \Sigma_D, \Sigma_R, \Sigma_A \rangle$, where the set of positive inclusion axioms in $\mathcal{T}$ is denoted by $\mathcal{T}_{PI}$, the set of disjointness axioms in $\mathcal{T}$ is denoted by $\mathcal{T}_{disj}$, the set of functionality assertion axioms in $\mathcal{T}$ is denoted by $\mathcal{T}_{\mathsf{funct}}$ and the set of value-domain inclusion axioms in $\mathcal{T}$ is denoted by $\mathcal{T}_{VD}$ with the result that $\mathcal{T} = \mathcal{T}_{PI} \cup \mathcal{T}_{NI}$, where $\mathcal{T}_{NI}$ denotes the set of negative inclusions constituted by $\mathcal{T}_{disj} \cup \mathcal{T}_{\mathsf{funct}} \cup \mathcal{T}_{VD}$.*

Let φ be a function that maps concept and role expressions of $\mathcal{T}$ into query atoms as follows:

$$\begin{array}{lll} C \mapsto C(x) & \exists R^- \mapsto R(\_, x) & R \mapsto R(x, x') \\ \exists R \mapsto R(x, \_) & \delta(A) \mapsto A(x, \_) & R^- \mapsto R(x', x) \end{array}$$

where C is an atomic concept (concept name) $\sigma_C \in \Sigma_C$, R is an atomic role (role name) $\sigma_R \in \Sigma_R$, A is an atomic attribute (attribute name) $\sigma_A \in \Sigma_A$, $\delta(A)$ denotes the domain of attribute A, x, x' are (bound) variables, and $\_$ denotes an unbound variable.

Based on that, the translation function τ maps the clash types (i)–(vi) respectively the corresponding axioms in $\mathcal{T}_{NI}$ into conjunctive query bodies, i.e., open FOL formulas, as follows:

(i) $\tau(C_1 \sqsubseteq \neg C_1) = \varphi(C_1)$,
$\tau(R_1 \sqsubseteq \neg R_1) = \varphi(R_1)$,
$\tau(A \sqsubseteq \neg A) = A(x, x')$,

(ii) $\tau(R \sqsubseteq \neg R^-) = \varphi(R) \wedge \varphi(R^-)$,
$\tau(\exists R \sqsubseteq \neg \exists R^-) = \varphi(\exists R) \wedge \varphi(\exists R^-)$,

(iii) $\tau(\rho(A) \sqsubseteq D) = A(\_, x) \wedge \mathit{datatype}(x) \neq D$,

(iv) $\tau(C_1 \sqsubseteq \neg C_2) = \varphi(C_1) \wedge \varphi(C_2)$,
$\tau(R_1 \sqsubseteq \neg R_2) = \varphi(R_1) \wedge \varphi(R_2)$,
$\tau(A_1 \sqsubseteq \neg A_2) = A_1(x, x') \wedge A_2(x, x')$,

(v) $\tau(\mathsf{funct}\ R) = R(x, x') \wedge R(x, x'') \wedge x' \neq x''$,
$\tau(\mathsf{funct}\ R^-) = R(x', x) \wedge R(x'', x) \wedge x' \neq x''$,

(vi) $\tau(\mathsf{funct}\ A) = A(x, x') \wedge A(x, x'') \wedge x' \neq x''$,

where $C, R, A, \delta(A), x, x'$ and $\_$ are as before, C_1, C_2 are basic concepts, R_1, R_2 are basic roles, $\rho(A)$ denotes the range of the atomic attribute A, D is a value-domain $\sigma_D \in \Sigma_D$, A_1, A_2 are atomic attributes, and x'' is a (bound) variable as x, x'. Moreover, $\mathit{datatype}(x)$ is an external function that returns the value-domain σ_D of a data value σ_V, i.e., for which $\sigma_V^\mathcal{I} \in \sigma_D^\mathcal{I}$ holds.

Since we assume that the set of value-domains Σ_D is already sufficiently structured and defined by a type system, where all value-domains $\sigma_D \in \Sigma_D$ are pairwise disjoint, it follows that $\sigma_V^{\mathcal{I}} \notin \Sigma_D^{\mathcal{I}} \setminus \sigma_D^{\mathcal{I}}$ holds for each data value $\sigma_V^{\mathcal{I}} \in \sigma_D^{\mathcal{I}}$. Hence, from an axiom of the form $\rho(A) \sqsubseteq D$ we can directly conclude $\rho(A) \sqsubseteq \neg D_i$ for each $D_i \in \Sigma_D \setminus D$. However, instead of verifying $A(\_, x) \wedge D_i(x)$ for each $D_i \in \Sigma_D \setminus D$ it is sufficient to validate if $A(\_, x) \wedge \mathit{datatype}(x) \neq D$ employing the external function $\mathit{datatype}(x)$ as defined above.

Definition 4.2 (Boolean Clash Queries)**.** *Given a DL-Lite$_\mathcal{A}$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ over a signature $\Sigma = \langle \Sigma_I, \Sigma_V, \Sigma_C, \Sigma_D, \Sigma_R, \Sigma_A \rangle$, where the TBox $\mathcal{T}$ is again subdivided into $\mathcal{T}_{PI}$ denoting the set of positive inclusion axioms and $\mathcal{T}_{NI}$ representing the set of negative inclusions. The complete set of Boolean clash queries is then given with*

$$\mathcal{Q}_{clash}() = \bigcup_{\alpha \in \mathcal{T}_{NI}} \{\tau(\alpha)\}\,, \tag{4.1}$$

where τ is a translation function according to Definition 4.1.

Obviously, given an axiom $\alpha \in \mathcal{T}_{NI}$, the CQ body $\tau(\alpha)$ corresponds to the negation of α in terms of a (open) FOL formula. Thus, by evaluating the set $\mathcal{Q}_{clash}() = \bigcup_{\alpha \in \mathcal{T}_{NI}} \{\tau(\alpha)\}$ of *Boolean clash queries* obtained by applying the translation function $\tau(\alpha)$ for each $\alpha \in \mathcal{T}_{NI}$, we can conclude that $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is inconsistent, i.e., $\mathit{Mod}(\mathcal{K}) = \emptyset$, iff $\langle \mathcal{T}_{PI}, \mathcal{A} \rangle \models q()$ holds for at least one Boolean clash query $q_i() = \exists \boldsymbol{y}_i.\mathit{conj}_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{Q}_{clash}()$, where $\boldsymbol{x}_i = \emptyset$ is an empty tuple. Note that unlike $\langle \mathcal{T}, \mathcal{A} \rangle$, $\langle \mathcal{T}_{PI}, \mathcal{A} \rangle$ is always consistent why the computation of certain answers to a query over $\langle \mathcal{T}_{PI}, \mathcal{A} \rangle$ becomes even possible. It is easy to see that the generation of clash queries also holds for federated *DL-Lite$_\mathcal{A}$* KBs.

Example 4.1 (Generation of Boolean Clash Queries)**.** *Given, for instance, the axiom $\mathit{Paper} \sqsubseteq \neg \mathit{Book}$, the translation function τ of Definition 4.1 produces the clash query body (FOL formula) $\tau(\mathit{Paper} \sqsubseteq \neg \mathit{Book}) = \mathit{Paper}(x) \wedge \mathit{Book}(x)$.*

Applying τ to the set $\mathcal{T}_{NI} \subseteq \mathcal{T}_F$ of our running defined in Section 3.3, the following complete set of Boolean clash queries can be derived:

$$\begin{aligned}
q() &= \mathit{Paper}(x) \wedge \mathit{Book}(x)\,,\\
q() &= \mathit{Publication}(x) \wedge \mathit{SlideSet}(x)\,,\\
q() &= \mathsf{edition}(_, x) \wedge \mathit{datatype}(x) \neq \texttt{xsd:integer}\,,\\
q() &= \mathit{publishedIn}(x, x') \wedge \mathit{publishedIn}(x, x'') \wedge x' \neq x''\,,\\
q() &= \mathsf{edition}(x, x') \wedge \mathsf{edition}(x, x'') \wedge x' \neq x''\,.
\end{aligned}$$

4.2.3 Clash Query Expansion

Besides explicit statements, the evaluation of clash queries or more generally of (U)CQs requires also the consideration of implicit statements that are entailed by

the KB. However, since we focus on the context of OBII, each integrated data source will commonly comprises only a subset of the federated TBox. Hence, according to the FOL-rewritability of *DL-Lite*$_\mathcal{A}$, for a complete entailment we incorporate the federated TBox into the query such that the integrated KBs can be kept in its original state and the answering of a query (CQ or UCQ) over a federated KB can be reduced to answering queries over the federated ABox only. More precisely, given a (U)CQ q over a *DL-Lite*$_\mathcal{A}$ KB $\mathcal{K}_F = \langle \mathcal{T}, \mathcal{A}_F \rangle$, where again $\mathcal{T}_{PI}$ denotes the set of positive inclusion axioms and $\mathcal{T}_{NI}$ the set of negative inclusions in $\mathcal{T}$, the query q is *expanded*[2] into a query $q^{\mathcal{T}_{PI}}$ according to the axioms in $\mathcal{T}_{PI}$, such that $q^{\mathcal{T}_{PI}}$ will return, when evaluated over $\mathcal{A}_F$, all certain answers to q over $\langle \mathcal{T}_{PI}, \mathcal{A}_F \rangle$, i.e., $answ(q, \langle \mathcal{T}_{PI}, \mathcal{A}_F \rangle) = answ(q^{\mathcal{T}_{PI}}, \langle \emptyset, \mathcal{A}_F \rangle)$.

Definition 4.3 (Query Expansion)**.** *Given a query $q(\boldsymbol{x})$ (either a CQ or an UCQ) over a DL-Lite$_\mathcal{A}$ TBox $\mathcal{T}$, where again the set of positive inclusion axioms in $\mathcal{T}$ is denoted by $\mathcal{T}_{PI}$ and the set of negative inclusion axioms is denoted by $\mathcal{T}_{NI}$ which is constituted by disjointness axioms, functionality assertion axioms and value-domain inclusion axioms in $\mathcal{T}$, i.e., $\mathcal{T}_{NI} = \mathcal{T}_{disj} \cup \mathcal{T}_{\mathsf{funct}} \cup \mathcal{T}_{VD}$. An expansion of $q(\boldsymbol{x})$ is a UCQ denoted by $q^{\mathcal{T}_{PI}}(\boldsymbol{x}) = \mathsf{Expand}\big(q(\boldsymbol{x}), \mathcal{T} \setminus \mathcal{T}_{NI}\big)$, where $\mathsf{Expand}\big(q(\boldsymbol{x}), \mathcal{T} \setminus \mathcal{T}_{NI}\big)$ is an algorithm that returns a rewriting of $q(\boldsymbol{x})$ w.r.t. $\mathcal{T}_{PI}$, such that $\langle \mathcal{T}_{PI}, \mathcal{A} \rangle \models q(\boldsymbol{a})$ iff $\mathcal{A} \models q^{\mathcal{T}_{PI}}(\boldsymbol{a})$, for an arbitrary ABox $\mathcal{A}$ and any tuple $\boldsymbol{a}$ of individual names and data values in $\mathcal{A}$, i.e., $answ(q(\boldsymbol{x}), \langle \mathcal{T}_{PI}, \mathcal{A} \rangle) = answ(q^{\mathcal{T}_{PI}}(\boldsymbol{x}), \langle \emptyset, \mathcal{A} \rangle)$.*

Exploiting the FOL-rewritability, Calvanese et al. [Cal+07b] proposed the first algorithm called PerfectRef for implementing query answering by query expansion in *DL-Lite*. Informally, the algorithm applies axioms in $\mathcal{T}_{PI}$ from right to left to each query atom and thus obtain a union of CQs covering all possibilities that imply the query atoms of the original query. However, since the size of the resulting UCQ from the PerfectRef algorithm is exponential in the input query, several optimizations [Rod10; RC11] and alternative algorithms for query expansion in *DL-Lite* and other lightweight DLs have been proposed (see [BO15] for a survey). One of these is the TreeWitness algorithm constituted by Kikot et al. [KKZ12] for *DL-Lite* (resp. OWL 2 QL) that produces simpler and shorter query expansions than most of the other approaches [RKZ13]. While the implementations of both, PerfectRef and TreeWitness, were originally part of the mature open-source OBDA framework –ontop–[3], PerfectRef is meanwhile replaced by the more efficient TreeWitness algorithm [Kha+17].

In the following we illustrate the approach of query expansion with an example and refer the interested reader to works like [Cal+07b; KKZ12; RKZ13; BO15] for a detailed introduction.

Example 4.2 (Query Expansion)**.** *Given a query $q(x) = Book(x)$ that simply selects all books, the expansion of that query with respect to the TBox $\mathcal{T}_F$ of our*

[2] The approach of query expansion is also known as query rewriting or query reformulation.

[3] `http://ontop.inf.unibz.it`

running example results in an UCQ which can be represented by the following set of CQ:

$$
\begin{aligned}
q(x) &= \mathit{Book}(x)\,,\\
q(x) &= \mathit{Proceedings}(x)\,,\\
q(x) &= \mathit{publishedIn}(\_, x)\,,\\
q(x) &= \mathit{edition}(x, \_)\,.
\end{aligned}
$$

As we can observe according to Definition 4.1, the FOL formulas generated by the translation function τ are of fixed length, i.e., one query atom in case (i) and (iii) or two query atoms in the remaining ones, and may comprise limited forms of inequalities (case (iii), (v) and (vi)). Moreover, subsumption axioms in *DL-Lite*$_\mathcal{A}$ (see Definition 2.15 for the syntax of *DL-Lite*$_\mathcal{A}$) generally comprise only one element on the left and one element on the right hand side of the subsumption relation, or can be normalized to that form, i.e., without any syntactic sugar such as $C_1 \sqsubseteq C_2 \sqcap C_3$ or $C_2 \sqcup C_3 \sqsubseteq C_1$. Consequently, an expansion of a clash query is a UCQ where each conjunct has again one (case (i) and (iii) including an inequality) or at most two query atoms (case (ii) and (iv)). As functionality assertions are restricted to basic roles and atomic attributes that are primitive, i.e., that do not positively appear on the right-hand side of an inclusion axiom and are not used in a qualified existential restriction, except of equivalent roles or attributes there exist no further expansions for clash queries w.r.t. functional roles or attributes (case (v) and (vi)).

Example 4.3 (Clash Query Expansion)**.** *With reference to our running example of Section 3.3, the expansion for the first clash query* $q() = \mathit{Paper}(x) \wedge \mathit{Book}(x)$ *mentioned in Example 4.1 results in the following set of CQs:*

$$
\begin{aligned}
q() &= \mathit{Paper}(x) \wedge \mathit{Book}(x)\,,\\
q() &= \mathit{Paper}(x) \wedge \mathit{Proceedings}(x)\,,\\
q() &= \mathit{Paper}(x) \wedge \mathit{publishedIn}(\_, x)\,,\\
q() &= \mathit{Paper}(x) \wedge \mathit{edition}(x, \_)\,,\\
q() &= \mathit{publishedIn}(x, \_) \wedge \mathit{Book}(x)\,,\\
q() &= \mathit{publishedIn}(x, \_) \wedge \mathit{Proceedings}(x)\,,\\
q() &= \mathit{publishedIn}(x, \_) \wedge \mathit{publishedIn}(\_, x)\,,\\
q() &= \mathit{publishedIn}(x, \_) \wedge \mathit{edition}(x, \_)\,,\\
q() &= \mathit{slideSetOf}(\_, x) \wedge \mathit{Book}(x)\,,\\
q() &= \mathit{slideSetOf}(\_, x) \wedge \mathit{Proceedings}(x)\,,\\
q() &= \mathit{slideSetOf}(\_, x) \wedge \mathit{publishedIn}(\_, x)\,,\\
q() &= \mathit{slideSetOf}(\_, x) \wedge \mathit{edition}(x, \_)\,.
\end{aligned}
$$

While the expansion for the second clash query is carried out in the same way, the respective expansion of the three remaining clash queries only comprises the query itself.

Since the expansion of a query only depends on the axioms in $\mathcal{T}_{PI}$ (and obviously the query), this implies a data complexity of AC^0 for the evaluation of a (U)CQ, as already mentioned in Section 2.3. However, as also mentioned in Section 2.3 answering a CQ$^{\neq}$ with even one inequality over a *DL-Lite*$_\mathcal{A}$ KB may lead to undecidability even in data complexity [Ros07; Gut+15]. But indeed, inequalities only occur in clash queries w.r.t. data values of attributes (case (iii) in Definition 4.1) and in clash queries for functional roles or attributes (case (v) and (vi)). Since by definition there exist no expansion for data types, functional roles or attributes (except of roles or attributes that are declared to be equivalent), the limited forms of inequalities in clash queries represent an exceptional case that does not affect the computational complexity of answering (U)CQs over a *DL-Lite*$_\mathcal{A}$ KB [Art+09; Gut+15]. Hence, the data complexity for answering expanded clash queries is still kept in AC^0. Moreover, because of that, and due to the fixed length of the clash queries the combined complexity for checking if a *DL-Lite*$_\mathcal{A}$ KB is consistent by query answering becomes equivalent to the knowledge base complexity and is thus in NLOGSPACE as already for standard reasoning tasks [Art+09].

4.2.4 Clash Query Federation

Since the query expansion incorporate all relevant parts of the federated TBox, all source-specific TBoxes of the integrated KBs are addressed by the resulting UCQ. However, as we also have to deal with distributed ABoxes each query atom potentially addressing several sources have to be evaluated at each integrated KB and the corresponding answers have to be merged according to the logical operators within the UCQ. Formally, the answering of a *federated query* (federated UCQ) can be defined as follows:

Definition 4.4 (Federated Querying)**.** *Given an expansion $q^{\mathcal{T}_{PI}}(\boldsymbol{x})$ of a UCQ over a federated DL-Lite$_\mathcal{A}$ knowledge base $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$ integrating a set $\boldsymbol{\mathcal{K}} = \{\mathcal{K}_1, \ldots, \mathcal{K}_N\}$ of knowledge bases with $\mathcal{K}_n = \langle \mathcal{T}_n, \mathcal{A}_n \rangle$ over its signature Σ_n, where $\mathcal{T}_I$ is an intermediary TBox over the signature Σ_F, $\mathcal{T}_F = \mathcal{T}_I \cup \bigcup_{n=1,\ldots,N} \mathcal{T}_n$ denotes the federated TBox, $\mathcal{A}_F = \bigcup_{n=1,\ldots,N} \mathcal{A}_n$ the federated ABox and $\mathcal{T}_{PI}$ denotes the set of positive inclusion axioms in $\mathcal{T}_F$.*

The certain answers to $q^{\mathcal{T}_{PI}}(\boldsymbol{x})$ over $\mathcal{K}_F$ are given by

$$answ\big(q^{\mathcal{T}_{PI}}(\boldsymbol{x}), \langle \emptyset, \mathcal{A}_F \rangle\big) = \bigcup_{conj_i \in q^{\mathcal{T}_{PI}}(\boldsymbol{x})} answ\big(q_i(\boldsymbol{x}) = \exists \boldsymbol{y}_i.conj_i(\boldsymbol{x}, \boldsymbol{y}_i), \langle \emptyset, \mathcal{A}_F \rangle\big)\,, \quad (4.2)$$

where in turn the certain answers to a CQ $q_i(\boldsymbol{x})$ given by $q_i(\boldsymbol{x}) = \exists \boldsymbol{y}_i.conj_i(\boldsymbol{x}, \boldsymbol{y}_i) \in q^{\mathcal{T}_{PI}}(\boldsymbol{x})$ are defined with

$$answ\big(q_i(\boldsymbol{x}), \langle \emptyset, \mathcal{A}_F \rangle\big) = \left\{ \boldsymbol{a} \;\middle|\; \boldsymbol{a} \in \bigoplus_{\phi_j \in q_i(\boldsymbol{x}) \setminus q_i^c(\boldsymbol{x})} answ\big(q_j(\boldsymbol{x} \cup \boldsymbol{y}_j) = \phi_j, \langle \emptyset, \mathcal{A}_F \rangle\big) \wedge \bigwedge_{\phi_k^c \in q_i^c(\boldsymbol{x})} \phi_k^c(\boldsymbol{a}) \right\}, \tag{4.3}$$

where $q_i^c(\boldsymbol{x})$ denotes the set of equality and inequality query atoms (i.e., query atoms of the form $t_1 = t_2$ and $t_1 \neq t_2$) in $q_i(\boldsymbol{x})$, $\boldsymbol{y}_j \subseteq \boldsymbol{y}_i$ denotes all non-distinguished variables in ϕ_j and $\phi_c(\boldsymbol{a})$ denotes the replacement of each variable $x_k \in \boldsymbol{x}$ in ϕ_c by the respective $a_k \in \boldsymbol{a}$. Moreover, the conjunction operation $\oplus$ is defined as

$$\begin{aligned} &\bigoplus_{\phi_j \in q_i(\boldsymbol{x})} answ\big(q_j(\boldsymbol{x}) = \phi_j, \langle \emptyset, \mathcal{A}_F \rangle\big) \\ &\qquad = \Big(\Big(\Big(answ\big(q_1(\boldsymbol{x}), \langle \emptyset, \mathcal{A}_F \rangle\big) \oplus answ\big(q_2(\boldsymbol{x}), \langle \emptyset, \mathcal{A}_F \rangle\big) \Big) \oplus \dots \Big) \\ &\qquad\qquad \oplus answ\big(q_n(\boldsymbol{x}), \langle \emptyset, \mathcal{A}_F \rangle\big) \Big), \end{aligned} \tag{4.4}$$

where as per Equation (2.16) $answ\big(q'(\boldsymbol{x}), \langle \emptyset, \mathcal{A}_F \rangle\big) \oplus answ\big(q''(\boldsymbol{x}), \langle \emptyset, \mathcal{A}_F \rangle\big) = \{\boldsymbol{a}' \oplus \boldsymbol{a}'' \mid \boldsymbol{a}' \in answ(q'(\boldsymbol{x}), \langle \emptyset, \mathcal{A}_F \rangle) \sim \boldsymbol{a}'' \in answ(q''(\boldsymbol{x}), \langle \emptyset, \mathcal{A}_F \rangle)\}$, $\boldsymbol{a}' \sim \boldsymbol{a}''$ means that $\boldsymbol{a}'$ and $\boldsymbol{a}''$ are compatible, i.e., $\boldsymbol{a}'(x) = null \vee \boldsymbol{a}''(x) = null \vee \boldsymbol{a}'(x) = \boldsymbol{a}''(x)$ holds for each $x \in \boldsymbol{x}$. The resulting tuple of $\boldsymbol{a}' \oplus \boldsymbol{a}''$ is given by $(\boldsymbol{a}' \oplus \boldsymbol{a}'')\colon x \to \boldsymbol{a}'(x)$ if $\boldsymbol{a}'(x) \neq null$ and $(\boldsymbol{a}' \oplus \boldsymbol{a}'')\colon x \to \boldsymbol{a}''(x)$ otherwise. On the other hand the certain answers to $q_j(\boldsymbol{x} \cup \boldsymbol{y}_j) = \phi_j$, i.e., a query atom ϕ_j in a CQ $q_i(\boldsymbol{x})$ over a federated ABox $\mathcal{A}_F$ are defined by

$$answ\big(q_j(\boldsymbol{x} \cup \boldsymbol{y}_j), \langle \emptyset, \mathcal{A}_F \rangle\big) = \bigcup_{\mathcal{A}_n \in \mathcal{A}_F} answ\big(q_j(\boldsymbol{x} \cup \boldsymbol{y}_j), \langle \emptyset, \mathcal{A}_n \rangle\big). \tag{4.5}$$

Note that according to Definition 4.4 a query expansion $q^{\mathcal{T}_{PI}}(\boldsymbol{x})$, i.e., a UCQ, is decomposed into parallelizable subqueries since each query atom ϕ_j of a CQ $q_i(\boldsymbol{x}) = \exists \boldsymbol{y}_i.conj_i(\boldsymbol{x}, \boldsymbol{y}_i) \in q^{\mathcal{T}_{PI}}(\boldsymbol{x})$ is independently evaluated at each integrated source. Accordingly to the above definition, in other words, we can say that $\boldsymbol{a}'$ and $\boldsymbol{a}''$ are compatible, i.e., $\boldsymbol{a}' \sim \boldsymbol{a}''$ if $\boldsymbol{a}'(x) = \boldsymbol{a}''(x)$ for each $x \in dom(\boldsymbol{a}') \cap dom(\boldsymbol{a}'')$, where $dom(\boldsymbol{a})$ denotes the domain of $\boldsymbol{a}$ that is a subset of $\boldsymbol{x}$ where $\boldsymbol{a}$ is defined. Thus, we can also say that the resulting tuple of $\boldsymbol{a}' \oplus \boldsymbol{a}''$ is given by $(\boldsymbol{a}' \oplus \boldsymbol{a}'')\colon x \to \boldsymbol{a}'(x)$ if $x \in dom(\boldsymbol{a}')$ and $(\boldsymbol{a}' \oplus \boldsymbol{a}'')\colon x \to \boldsymbol{a}''(x)$ otherwise. Consequently, the definition of the conjunction operation $\oplus$ in Equation (4.4) is in conformity with the definition of the JOIN operation in Equation (2.16) for SPARQL queries with simple entailment (see Section 2.4.2).

In order to subsequently combine the answers to each query atom $\phi_j \in q_i(\boldsymbol{x})$ according to CQ $q_i(\boldsymbol{x})$, all bound variables (distinguished and non-distinguished variable) of $q_i(\boldsymbol{x})$, i.e., $\boldsymbol{x} \cup \boldsymbol{y}_j$, have to be part of the query atom answers why the head of a query $q_j(\boldsymbol{x} \cup \boldsymbol{y}_j) = \phi_j$ is different from the head of the CQ $q_i(\boldsymbol{x})$.

Example 4.4 (Query Federation)**.** *Following up our running example and taking, for instance, the third query* $q() = Paper(x) \wedge publishedIn(\_, x)$ *from Example 4.3, the following Figure 4.1 depicts a schematic representation of the corresponding federated evaluation:*

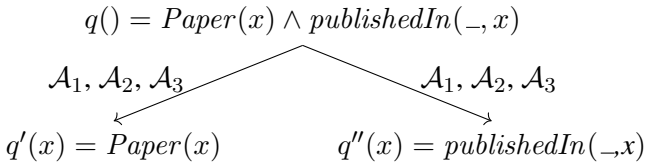

Figure 4.1: Query Federation

According to Definition 4.4, both query atoms of $q()$ *are evaluated independently at each data source, i.e., its ABox. However, even the query* $q()$ *is a Boolean query, the head of both,* q' *and* q''*, is extended by the bound variable* x *such that the query answers can be combined subsequently. The query answers of each query evaluation are as follows:*

$$
\begin{aligned}
answ(q'(x), \langle \emptyset, \mathcal{A}_1 \rangle) &= \{\, (\mathbf{I1}), (\mathbf{I2}), (\mathbf{I6}), (\mathbf{I7}) \,\}\,, \\
answ(q'(x), \langle \emptyset, \mathcal{A}_2 \rangle) &= \{\, (\mathbf{I1}), (\mathbf{C4}) \,\}\,, \\
answ(q'(x), \langle \emptyset, \mathcal{A}_3 \rangle) &= \emptyset\,, \\
answ(q''(x), \langle \emptyset, \mathcal{A}_1 \rangle) &= \{\, (\mathbf{C1}), (\mathbf{I5}), (\mathbf{C3}) \,\}\,, \\
answ(q''(x), \langle \emptyset, \mathcal{A}_2 \rangle) &= \{\, (\mathbf{I1}), (\mathbf{C2}), (\mathbf{C3}) \,\}\,, \\
answ(q''(x), \langle \emptyset, \mathcal{A}_3 \rangle) &= \emptyset\,.
\end{aligned}
$$

Since both, $answ(q'(\boldsymbol{x}), \langle \emptyset, \mathcal{A}_1 \rangle)$ *and* $answ(q'(\boldsymbol{x}), \langle \emptyset, \mathcal{A}_2 \rangle)$ *are compatible with* $answ(q''(\boldsymbol{x}), \langle \emptyset, \mathcal{A}_2 \rangle)$ *for* $x = \mathbf{I1}$*, the subsequent join of the query answers results in* $answ(q(), \langle \emptyset, \mathcal{A}_F \rangle) = true$*.*

It is easy to see that not all data sources are likely to return some results to each atom of a query expansion, since the integrated sources of a federated KB ordinarily comprise different TBox signatures with small or even no intersections. Hence, the query evaluation can be optimized such that an atom is evaluated only over those ABoxes that (probably) will return some results. Besides, a more advanced optimization could be a result estimation of the conditions (i.e., conjunctions or inequalities) formulated within a CQ in order that query atoms are only evaluated over those sources, the results of which will (probably) be relevant for answering the CQ. However, as we will not focus on optimizing execution plans for federated queries within the context of this work, we just apply a plain query federation and refer the interested reader to works like [QL08; NN13; Li13; Rak+13b; Rak+13a; LNZ14; Sal+16; NS16] for approaches related to optimized evaluations of federated queries.

4.3 Explanations for Inconsistency in Federated *DL-Lite*$_\mathcal{A}$ Knowledge Bases

According to our definition of Boolean clash queries and its expansions we can determine that a federated KB is inconsistent iff at least one Boolean query $q() = \exists \boldsymbol{y}_i.conj_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{Q}^{\mathcal{T}_{PI}}_{clash}()$, i.e., the set of all clash queries and its expansions, is evaluated to $true$. However, to be able to resolve the inconsistency we have to pinpoint those assertions and axioms causing the logical conflicts.

In context of entailment, an *explanation* (or justification) for an inferred axiom or assertion (resp. closed FOL formula) α' is a minimal subset of a KB comprising exactly all those statements that are responsible for $\mathcal{K} \models \alpha'$. According to Kalyanpur et al. [Kal+07] an explanation can be formally defined as follows:

Definition 4.5 (Explanation)**.** *Given a knowledge base $\mathcal{K}$ and let $\mathcal{K} \models \alpha'$. An explanation for $\mathcal{K} \models \alpha'$ is a subset $\mathcal{K}'$ of $\mathcal{K}$ such that $\mathcal{K}' \models \alpha'$ while $\mathcal{K}'' \not\models \alpha'$ holds for all $\mathcal{K}'' \subset \mathcal{K}'$.*

Intuitively, an explanation can be interpreted as a minimal reason explaining why α' is entailed by $\mathcal{K}$. It is easy to see that there may exist several, possibly overlapping, explanations for a specific entailment.

Analogously, given an inconsistent KB, we are especially interested in explanations for the inconsistency, called *minimal inconsistent subsets* (MISs). Referring to the definition of minimal incoherence preserving sub-TBox (MIPS) by Schlobach and Cornet [SC03], we can define a MIS accordingly to Definition 4.5:

Definition 4.6 (MIS)**.** *Given an inconsistent knowledge base $\mathcal{K}$, an explanation for the inconsistency of $\mathcal{K}$ is a minimal inconsistent subset $\mathcal{K}'$ of $\mathcal{K}$ such that $\mathcal{K}'$ is inconsistent, i.e., $Mod(\mathcal{K}') = \emptyset$ and there exists no proper subset $\mathcal{K}'' \subset \mathcal{K}'$ for which $Mod(\mathcal{K}'') = \emptyset$ holds.*

Consequently, a MIS comprises exactly those assertions and axioms causing a logical conflict in $\mathcal{K}$.

However, since we assume that the federated TBox is already *"semantically correct"* (free from any modeling errors) such that $Mod(\mathcal{T}_F) \neq \emptyset$, we are only interested in the ABox part of a MIS. We refer to such a subset of a MIS comprising only ABox assertions as a *minimal inconsistent sub-ABox* (MISA).

Definition 4.7 (MISA)**.** *Given an inconsistent knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, a minimal inconsistent sub-ABox $\mathcal{A}'$ is a subset of $\mathcal{A}$ such that $Mod(\langle \mathcal{T}, \mathcal{A}' \rangle) = \emptyset$ while $Mod(\langle \mathcal{T}, \mathcal{A}'' \rangle) \neq \emptyset$ for each $\mathcal{A}'' \subset \mathcal{A}'$.*

According to the definition of the translation function τ (Definition 4.1) we can notice that a MISA of an inconsistent *DL-Lite*$_\mathcal{A}$ KB is either unary (case (i) and (iii)) or binary (case (ii), (iv), (v) and (vi)). Moreover, while an unary MISA always explains by definition a *local conflict*, i.e., a conflict with respect to one single data source, a binary MISA may explains a local or a *federated conflict*, where the

conflicting ABox assertions are originating from the same or two different sources, respectively.

So far, we have only considered the Boolean form of clash queries which is sufficient in order to determine if a KB is inconsistent. However, to be able to generate a complete set of explanations, i.e., MISAs, we have to modify the set $\mathcal{Q}_{clash}^{\mathcal{T}_{PI}}()$ of all expanded Boolean clash queries such that the corresponding ABox assertions can be reproduced out of the certain answers $answ(q_i(\boldsymbol{x}_i), \langle \emptyset, \mathcal{A} \rangle)$ for each $q_i(\boldsymbol{x}_i) = \exists \boldsymbol{y}_i.conj_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{Q}_{clash}^{\mathcal{T}_{PI}}()$.

As Boolean clash queries or its expansions may comprise unbound variables (resulting from existential restrictions), we have to replace all those variables by new variables in order to be able to make precise distinctions between different instantiations of them and to completely reconstruct the respective ABox assertions. Formally, we can define *clash queries* as follows:

Definition 4.8 (Clash Queries)**.** *Given a set $\mathcal{Q}_{clash}^{\mathcal{T}_{PI}}()$ of expanded Boolean clash queries for a DL-Lite$_\mathcal{A}$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, each unbound variable $\_$ in an atom ϕ_j of a Boolean clash query $q_i() = \exists \boldsymbol{y}_i.conj_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{Q}_{clash}^{\mathcal{T}_{PI}}()$ is replaced by a new variable y_{new} yet not occurring in $q()$, i.e., $y_{new} \notin \boldsymbol{x}_i \cup \boldsymbol{y}_i$.[4] After the elimination of unbound variables, the clash queries are extended with distinguished variables (answer variables) by adding all the non-distinguished variables of each query atom ϕ_j in $\exists \boldsymbol{y}_i.conj_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{Q}_{clash}^{\mathcal{T}_{PI}}()$ into the head of the resulting clash query, i.e., $q_i(\boldsymbol{x}_i') = \exists \boldsymbol{y}_i.conj_i(\boldsymbol{x}_i, \boldsymbol{y}_i)$ where $\boldsymbol{x}_i' = \boldsymbol{x}_i \cup \boldsymbol{y}_i$. We denote the set of all (non-Boolean) clash queries including its expansions by $\mathcal{Q}_{clash}^{\mathcal{T}_{PI}}(\boldsymbol{x})$.*

Example 4.5 (Non-Boolean Clash Queries)**.** *Given, for example, the seventh Boolean query $q() = publishedIn(x, \_) \wedge publishedIn(\_, x)$ mentioned in Example 4.3, each unbound variables $\_$ is replaced by a new variable and all variables of each query atom are added to the head of query. The corresponding non-Boolean clash query is then given with $q(x, y', y'') = publishedIn(x, y') \wedge publishedIn(y'', x)$. Given again the Boolean clash query $q() = Paper(x) \wedge publishedIn(\_, x)$ discussed in Example 4.4, according to Definition 4.8 the unbound variable $\_$ of the second query atom is replaced by a new variable and all variables of each query atom are added to the head of query. As a consequence, the corresponding non-Boolean clash query is then given with $q(x, y') = Paper(x) \wedge publishedIn(y', x)$. While the query answers to the first atom are equivalent to those mentioned in Example 4.4, due to the variable extensions, the query answers to the second atom are now given with*

$$
\begin{aligned}
&answ(q''(x, y'), \langle \emptyset, \mathcal{A}_1 \rangle) = \{\, (\mathbf{I1}, \mathbf{C1}), (\mathbf{I4}, \mathbf{I5}), (\mathbf{C2}, \mathbf{I5}), (\mathbf{I6}, \mathbf{C3}) \,\}\,, \\
&answ(q''(x, y'), \langle \emptyset, \mathcal{A}_2 \rangle) = \{\, (\mathbf{C1}, \mathbf{I1}), (\mathbf{I4}, \mathbf{C2}), (\mathbf{I6}, \mathbf{C3}) \,\}\,, \\
&answ(q''(x, y'), \langle \emptyset, \mathcal{A}_3 \rangle) = \emptyset\,,
\end{aligned}
$$

[4] Note that since such a replacement of unbound variables would affect the query expansion, the replacement have to be performed not on the set $\mathcal{Q}_{clash}()$ of Boolean clash queries but on the set $\mathcal{Q}_{clash}^{\mathcal{T}_{PI}}()$ of expanded Boolean clash queries.

and the subsequent join of the query answers results in $answ(q(x, y'), \langle \emptyset, \mathcal{A}_F \rangle) = \{ (\mathbf{I1}, \mathbf{C1}) \}$.

Similar to Boolean clash queries we can conclude that the federated KB is consistent, iff the evaluation of $\mathcal{Q}^{\mathcal{T}_{PI}}_{clash}(\boldsymbol{x})$ over $\mathcal{A}_F$ ends up with an empty answer. Otherwise, the individuals that are delivered by a clash query $q_i(\boldsymbol{x}'_i) = \exists \boldsymbol{y}_i . conj_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{Q}^{\mathcal{T}_{PI}}_{clash}(\boldsymbol{x})$ in conjunction with the respective query atoms in $q_i(\boldsymbol{x}'_i)$ can be used to reconstruct the corresponding ABox assertions causing a logical conflict.

Before we will define a back-translation reproducing the corresponding ABox assertions, i.e., MISAs, out of the query answers, we first discuss the peculiarity of ABox assertions in the specific context of federated KBs. If we would use the common syntax of ABox assertions within the back-translation the information about the source stating the assertion will be lost. Because of that, we employ an augmented form of an assertion syntax preserving the information about the originating source, called *source-related ABox assertion*.

Definition 4.9 (Source-Related ABox Assertions)**.** *Given a federated DL-Lite*$_\mathcal{A}$ *knowledge base* $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$ *integrating a set* $\boldsymbol{\mathcal{K}} = \{\mathcal{K}_1, \ldots, \mathcal{K}_N\}$ *of knowledge bases with* $\mathcal{K}_n = \langle \mathcal{T}_n, \mathcal{A}_n \rangle$ *where* $\mathcal{T}_I$ *is an intermediary TBox,* $\mathcal{T}_F = \mathcal{T}_I \cup \bigcup_{n=1,\ldots,N} \mathcal{T}_n$ *denotes the federated TBox and* $\mathcal{A}_F = \bigcup_{n=1,\ldots,N} \mathcal{A}_n$ *the federated ABox. A source-related ABox assertion is a tuple* $\langle \alpha, n \rangle$ *where* α *is a conventional DL-Lite*$_\mathcal{A}$ *ABox assertion according to Definition 2.15 and* n *denotes an integrated knowledge base* $\mathcal{K}_n$ *for which* $\alpha \in \mathcal{A}_n$ *holds.*

Hence, source-related ABox assertions facilitate the distinction of equivalent assertions that are present in several sources. Moreover, as we can easily identify the source where a specific assertion stems from, we may benefit from this information on dealing with found contradictions. Note that since all approaches of the following chapters are based on the notion of source-related ABox assertions for the sake of brevity we will preferably use the designation assertions.

In order to assign the corresponding source to a certain answer tuple we have to extend the federated querying (Definition 4.4) by additional distinguished variables binding the identifier of the origin data source.

Definition 4.10 (Federated Clash Querying)**.** *Given a clash query* $q_i(\boldsymbol{x}'_i)$ *with* $q_i(\boldsymbol{x}'_i) = \exists \boldsymbol{y}_i . conj_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{Q}^{\mathcal{T}_{PI}}_{clash}(\boldsymbol{x})$ *over a federated DL-Lite*$_\mathcal{A}$ *knowledge base* $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$*. The certain answers* $answ(q_i(\boldsymbol{x}'_i), \langle \emptyset, \mathcal{A}_F \rangle)$ *are given according to Definition 4.4 but with the modification that the distinguished variables* $\boldsymbol{x}'_i$ *of* $q_i(\boldsymbol{x}'_i)$ *are extended by the additional variables* $\boldsymbol{s}$ *and the certain answers (Equation (4.5)) to* $q_j(\boldsymbol{x}'_i \cup \boldsymbol{s}) = \phi_j$*, i.e., to a query atom* ϕ_j *in the CQ* $q_i(\boldsymbol{x}'_i)$ *over the federated ABox* $\mathcal{A}_F$ *are given by*

$$answ\big(q_j(\boldsymbol{x}'_i \cup \boldsymbol{s}), \langle \emptyset, \mathcal{A}_F \rangle\big) = \bigcup_{\mathcal{A}_n \in \mathcal{A}_F} answ\big(q_j(\boldsymbol{x}'_i) = \phi_j \wedge (s_j = n), \langle \emptyset, \mathcal{A}_n \rangle\big) . \tag{4.6}$$

Having defined the federated clash querying and the syntax of source-related ABox assertions, we can now define the back-translation of clash query answers into the corresponding MISAs as follows:

Definition 4.11 (Back-Translation)**.** *Given a certain answer tuple $\boldsymbol{a} \in answ(q_i(\boldsymbol{x}_i' \cup \boldsymbol{s}), \langle \emptyset, \mathcal{A}_F \rangle)$, where $\mathcal{A}_F$ is the ABox of an inconsistent federated DL-Lite$_\mathcal{A}$ knowledge base $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$, $q_i(\boldsymbol{x}_i')$ is a clash query (CQ) given by $q_i(\boldsymbol{x}_i') = \exists \boldsymbol{y}_i . conj_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{Q}_{clash}^{\mathcal{T}_{PI}}(\boldsymbol{x})$ and $\boldsymbol{s}$ are additional variables as in Definition 4.10. The corresponding MISA $\boldsymbol{m}$ is an unary or binary set of source-related ABox assertions explaining a logical conflict detected by evaluating $q_i(\boldsymbol{x}_i' \cup \boldsymbol{s})$ over $\mathcal{K}_F$, i.e., $\mathcal{A}_F$, and is given by*

$$\boldsymbol{m} = \bigcup_{\phi_j \in q_i(\boldsymbol{a})} \left\{ \langle \phi_j, \boldsymbol{a}(s_j) \rangle \right\}, \tag{4.7}$$

where $q_i(\boldsymbol{a})$ denotes the replacement of each variable $x_k \in \boldsymbol{x}_i'$ in $q_i(\boldsymbol{x}_i')$ by the respective $a_k \in \boldsymbol{a}$ and $\boldsymbol{a}(s_j)$ the replacement of variable $s_j \in \boldsymbol{s}$ by the corresponding value in $\boldsymbol{a}$.

The finite set of all MISAs for a clash query $q_i(\boldsymbol{x}_i') = \exists \boldsymbol{y}_i . conj_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{Q}_{clash}^{\mathcal{T}_{PI}}(\boldsymbol{x})$ is denoted by $expl\big(q_i(\boldsymbol{x}_i'), answ(q_i(\boldsymbol{x}_i' \cup \boldsymbol{s}), \langle \emptyset, \mathcal{A}_F \rangle)\big)$ in order that the complete set of MISAs for $\mathcal{K}_F$ is given by

$$\mathcal{M} = \bigcup_{conj_i \in \mathcal{Q}_{clash}^{\mathcal{T}_{PI}}(\boldsymbol{x})} expl\big(q_i(\boldsymbol{x}_i') = \exists \boldsymbol{y}_i . conj_i(\boldsymbol{x}_i, \boldsymbol{y}_i), answ(q_i(\boldsymbol{x}_i' \cup \boldsymbol{s}), \langle \emptyset, \mathcal{A}_F \rangle)\big). \tag{4.8}$$

Example 4.6 (Generation of MISAs)**.** *If we would take the query answers $answ(q(x, y'), \langle \emptyset, \mathcal{A}_F \rangle) = \{\langle \mathbf{I1}, \mathbf{C1} \rangle\}$ of Example 4.5, the corresponding set of MISAs for $q(x, y')$ would just be*

$$\left\{ \{Paper(\mathbf{I1}), publishedIn(\mathbf{C1}, \mathbf{I1})\} \right\}.$$

However, due to our definition of source-related ABox assertions as well as our corresponding Definition 4.10 of federated clash querying the information that the assertion $Paper(\mathbf{I1})$ is stated by both, $\mathcal{K}_1$ and $\mathcal{K}_2$, is preserved. As a consequence, the resulting set of MISAs for $q(x, y')$ is

$$\left\{ \begin{array}{l} \{\langle Paper(\mathbf{I1}), 1 \rangle, \langle publishedIn(\mathbf{C1}, \mathbf{I1}), 2 \rangle\}, \\ \{\langle Paper(\mathbf{I1}), 2 \rangle, \langle publishedIn(\mathbf{C1}, \mathbf{I1}), 2 \rangle\} \end{array} \right\}$$

and hence facilitates a further debugging based on a more detailed set of explanations.

Algorithm 4.1 summarizes our approach of inconsistency detection in federated *DL-Lite*$_\mathcal{A}$ KBs including the generation of the corresponding set of explanations.

Algorithm 4.1: DetectInconsistency($\mathcal{K}_F$)

Input: *DL-Lite*$_\mathcal{A}$ knowledge base $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$
Output: complete set $\mathcal{M}$ of MISAs

begin
  $\mathcal{M}, \mathcal{T}_{NI} \leftarrow \emptyset$
  foreach $\alpha \in \mathcal{T}_F$ **do**
    if α *is not a positive inclusion axiom* **then**
      $\mathcal{T}_{NI} \leftarrow \mathcal{T}_{NI} \cup \{\alpha\}$
  foreach $\alpha \in \mathcal{T}_{NI}$ **do**
    $q(\boldsymbol{x}) \leftarrow \tau(\alpha)$
    $q^{\mathcal{T}_{PI}}(\boldsymbol{x}) \leftarrow \mathsf{Expand}\big(q(\boldsymbol{x}), \mathcal{T}_F \setminus \mathcal{T}_{NI}\big)$
    foreach $q_i(\boldsymbol{x}) \in q^{\mathcal{T}_{PI}}(\boldsymbol{x})$ **do**
      $q_i(\boldsymbol{x}) \leftarrow \mathsf{EliminateUnboundVariables}\big(q_i(\boldsymbol{x})\big)$
      $q_i'(\boldsymbol{x} \cup \boldsymbol{y}_j) \leftarrow \mathsf{FirstAtom}\big(q_i(\boldsymbol{x})\big)$
      $q_i''(\boldsymbol{x} \cup \boldsymbol{y}_j) \leftarrow \mathsf{SecondAtom}\big(q_i(\boldsymbol{x})\big)$
      $c_i \leftarrow \mathsf{ConstraintAtom}\big(q_i(\boldsymbol{x})\big)$
      foreach $\mathcal{A}_n \in \mathcal{A}_F$ **do**
        $R' \leftarrow answ\big(q_i'(\boldsymbol{x} \cup \boldsymbol{y}_j), \langle \emptyset, \mathcal{A}_n \rangle\big)$
        if $q_i''(\boldsymbol{x} \cup \boldsymbol{y}_j) \neq \emptyset$ **then**
          foreach $\mathcal{A}_m \in \mathcal{A}_F$ **do**
            $R'' \leftarrow answ\big(q_i''(\boldsymbol{x} \cup \boldsymbol{y}_j), \langle \emptyset, \mathcal{A}_m \rangle\big)$
            foreach $\boldsymbol{a}' \in R'$ **do**
              foreach $\boldsymbol{a}'' \in R''$ **do**
                if $\boldsymbol{a}' \sim \boldsymbol{a}'' \wedge \big(c_i \neq \emptyset \vee c_i(\boldsymbol{a}' \oplus \boldsymbol{a}'')\big)$ **then**
                  $\boldsymbol{m} \leftarrow \big\{\langle q_i'(\boldsymbol{a}'), n\rangle, \langle q_i''(\boldsymbol{a}''), m\rangle\big\}$
                  $\mathcal{M} \leftarrow \mathcal{M} \cup \{\boldsymbol{m}\}$
        else
          foreach $\boldsymbol{a}' \in R'$ **do**
            if $c_i \neq \emptyset \vee c_i(\boldsymbol{a}')$ **then**
              $\boldsymbol{m} \leftarrow \big\{\langle q_i'(\boldsymbol{a}'), n\rangle\big\}$
              $\mathcal{M} \leftarrow \mathcal{M} \cup \{\boldsymbol{m}\}$
  return $\mathcal{M}$
end

Given a federated *DL-Lite*$_\mathcal{A}$ KB $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$, we first iterate over all axioms in $\mathcal{T}_F$ for creating the set $\mathcal{T}_{NI} \subseteq \mathcal{T}_F$ of negative inclusion axioms, functionality assertion axioms and value-domain inclusion axioms, which is given by $\mathcal{T}_{NI} = \mathcal{T}_F \setminus \mathcal{T}_{PI}$, where $\mathcal{T}_{PI}$ denotes the set of positive inclusion axioms. Iterating over each axiom α in $\mathcal{T}_{NI}$ we apply the translation function τ, given in Definition 4.1, which translates α into the corresponding (non-Boolean) clash query

$q(\boldsymbol{x})$. Subsequently, we apply on $q(\boldsymbol{x})$ a query expansion algorithm denoted with Expand incorporating $\mathcal{T}_{PI}$ given by $\mathcal{T}_{PI} = \mathcal{T}_F \setminus \mathcal{T}_{NI}$ into the resulting UCQ $q^{\mathcal{T}_{PI}}(\boldsymbol{x})$ and iterate over each CQ $q_i(\boldsymbol{x}) \in q^{\mathcal{T}_{PI}}(\boldsymbol{x})$. According to Definition 4.1 and Definition 4.3, a clash query and its expansions are constituted either by one or by two query atoms, and may comprise an additional inequality constraint. However, before we are constructing a query for each atom in $q_i(\boldsymbol{x})$, denoted by $q_i'(\boldsymbol{x} \cup \boldsymbol{y}_j) \leftarrow \mathsf{FirstAtom}\big(q_i(\boldsymbol{x})\big)$ and $q_i''(\boldsymbol{x} \cup \boldsymbol{y}_j) \leftarrow \mathsf{SecondAtom}\big(q_i(\boldsymbol{x})\big)$, and extracting the inequality atom by $c_i \leftarrow \mathsf{ConstraintAtom}\big(q_i(\boldsymbol{x})\big)$, all unbound variables in $q_i(\boldsymbol{x})$ are eliminated by $\mathsf{EliminateUnboundVariables}\big(q_i(\boldsymbol{x})\big)$ replacing each $\_$ with a new variable $y_{new} \notin \boldsymbol{x} \cup \boldsymbol{y}_i$. In case that $q_i(\boldsymbol{x})$ comprises two query atoms, both atom queries q_i' and q_i'' are evaluated at each data source $\mathcal{A}_n \in \mathcal{A}_F$. For each answer tuple pair $\boldsymbol{a}' \leftarrow answ\big(q_i'(\boldsymbol{x} \cup \boldsymbol{y}_j), \langle \emptyset, \mathcal{A}_n \rangle\big)$ and $\boldsymbol{a}'' \leftarrow answ\big(q_i''(\boldsymbol{x} \cup \boldsymbol{y}_j), \langle \emptyset, \mathcal{A}_m \rangle\big)$ that is compatible, i.e., $\boldsymbol{a}' \sim \boldsymbol{a}''$, and if $c_i \neq \emptyset$ for which the joint answer tuple $\boldsymbol{a}' \oplus \boldsymbol{a}''$ holds the inequality query atom c_i in $q_i(\boldsymbol{x})$, a corresponding MISA $\boldsymbol{m}$ with two source-related ABox assertions (according to Definition 4.9) is constructed by $\boldsymbol{m} \leftarrow \big\{\langle q_i'(\boldsymbol{a}'), n\rangle, \langle q_i''(\boldsymbol{a}''), m\rangle\big\}$, where $q_i'(\boldsymbol{a}')$ and $q_i''(\boldsymbol{a}'')$ denote the replacement of each variable in $q_i'(\boldsymbol{x} \cup \boldsymbol{y}_j)$ resp. $q_i''(\boldsymbol{x} \cup \boldsymbol{y}_j)$ by the respective value in $\boldsymbol{a}'$ resp. $\boldsymbol{a}''$, and n, m denote the source, i.e., ABox, where the answer tuple $\boldsymbol{a}'$ resp. $\boldsymbol{a}''$ originate from. In the other case, where $q_i(\boldsymbol{x})$ consists of only one query atom, a MISA $\boldsymbol{m}$ with one source-related ABox assertion is generated by $\boldsymbol{m} \leftarrow \big\{\langle q_i'(\boldsymbol{a}'), n\rangle\big\}$ if $q_i(\boldsymbol{x})$ contains no inequality, i.e., $c_i \neq \emptyset$, or $c_i(\boldsymbol{a}') = true$. Finally, the set $\mathcal{M}$ of all MISAs is returned.

Proposition 4.1. *Let $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$ be a federated DL-Lite$_\mathcal{A}$ knowledge base. Then* $\mathsf{DetectInconsistency}(\mathcal{K}_F)$ *returns $\mathcal{M} = \emptyset$ iff $\mathcal{K}_F$ is consistent, and otherwise the complete set $\mathcal{M}$ of MISAs for $\mathcal{K}_F$.*

Proof. Since the clash types given in Section 4.2.1 are based on the work of Lembo et al. [Lem+11], we can assume that these patterns are correct and complete for *DL-Lite$_\mathcal{A}$* KBs. Moreover, due to works like that of Calvanese et al. [Cal+07b] and Kikot et al. [KKZ12] we can also assume that the algorithm $\mathsf{Expand}\big(q(\boldsymbol{x}), \mathcal{T}_F \setminus \mathcal{T}_{NI}\big)$ returns an expansion of a (U)CQ that is complete with respect to $\mathcal{T}_{PI}$ so that $answ(q(\boldsymbol{x}), \langle \mathcal{T}_{PI}, \mathcal{A} \rangle) = answ(q^{\mathcal{T}_{PI}}(\boldsymbol{x}), \langle \emptyset, \mathcal{A} \rangle)$. As a consequence, the set $\mathcal{Q}_{clash}^{\mathcal{T}_{PI}}(\boldsymbol{x})$ of expanded clash queries is complete such that every clash query $q_i(\boldsymbol{x}_i') = \exists \boldsymbol{y}_i.conj_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{Q}_{clash}^{\mathcal{T}_{PI}}(\boldsymbol{x})$ expresses the negation of a possible violation of $\mathcal{T}_F$ in terms of a (open) FOL formula. By evaluating each $q_i(\boldsymbol{x}_i') = \exists \boldsymbol{y}_i.conj_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{Q}_{clash}^{\mathcal{T}_{PI}}(\boldsymbol{x})$ over $\langle \emptyset, \mathcal{A}_F \rangle$ all assertions actually violating $\mathcal{T}_F$ are returned. Hence, $Mod(\langle \mathcal{T}_F, \mathcal{A}_F \setminus (\bigcup_{\boldsymbol{m} \in \mathcal{M}} \boldsymbol{m}) \rangle) \neq \emptyset$ holds and if no assertion is returned for any clash query in $\mathcal{Q}_{clash}^{\mathcal{T}_{PI}}(\boldsymbol{x})$, we can conclude that $\mathcal{K}_F$ is consistent. □

Note that for performance reasons, the evaluation of atom queries at different sources can be parallelized. Moreover, to avoid that an atom query possibly occurring in several clash queries is repeatedly evaluated, the corresponding query answers could also be temporarily cached until all axioms in $\mathcal{T}_{NI}$ are proceeded.

Since $\mathcal{T}_{NI}$, the set of all negative inclusion axioms, functionality assertion axioms and value-domain inclusion axioms in $\mathcal{T}_F$ is finite and the termination of $\mathsf{Expand}\big(q(\boldsymbol{x}), \mathcal{T}_F \setminus \mathcal{T}_{NI}\big)$ is assumed to be already established (see, e.g., Calvanese et al. [Cal+07b] and Kikot et al. [KKZ12]), the termination of this algorithm is given.

Moreover, none of the operations in DetectInconsistency has an impact on the computational complexity, why the complexity bounds remain in AC^0 in the size of the ABox and in NLOGSPACE in the size of the whole KB, given a fixed set of data sources.

4.4 Related Work

There exists several state-of-the-art DL (or OWL) reasoners that can be used for inconsistency detection and generation of explanations. Basically, they are varying in the supported language expressiveness and the underlying reasoning method such as widely used tableau algorithms as in FaCT++ [TH06], Pellet [Sir+07], or RacerPro [HM01], the hypertableau technique of HermiT [MSH09; HMW12], consequence-driven approaches like those described by Kazakov [Kaz09] or Simančík et al. [SKH11], or resolution-based methods described by Motik and Sattler [MS06] or Kazakov and Motik [KM08]. However, all of such reasoners essentially process local KBs and hence are not designed for distributed environments.

To the best of our knowledge there currently exists no approach that is applicable for detecting and explaining inconsistency in the context of OBII i.e., in a loosely coupled network of KBs such as the LOD cloud. Nevertheless, there are some works pursuing a similar direction.

Besides the initial definition of the *DL-Lite* family the work of Calvanese et al. [Cal+07b] includes, inter alia, a definition of a translation function δ transforming negative inclusions and functionality assertions into CQs (open FOL formulas). By applying this translation function to each negative inclusions and functionality assertion that can be entailed from the TBox of a given KB, the authors present an algorithm, called Consistent, that evaluates the resulting Boolean UCQ comprising the union of all CQs generated by δ over the ABox of the given KB. An implementation of this algorithm for determining if a KB is consistent is included in the –ontop– framework, already mentioned in Section 4.2.3. Compared to our approach, the work of Calvanese et al. [Cal+07b] are only meant for *DL-Lite*$_{\mathcal{F}}$ and *DL-Lite*$_{\mathcal{R}}$ KBs. Moreover, [Cal+13] proposed an expansion of their approach to a new member of the *DL-Lite* family, called *DLR-Lite*$_{\mathcal{A},\sqcap}$, that is designed for permitting the use of conjunctions and n-ary relations on the left-hand side of inclusion assertions while preserving the FOL-rewritability. Even though the Consistent algorithms of [Cal+07b] and [Cal+13] are similar to our approach, both algorithms are only designed to identify if a given KB is (in)consistent but not to specify an inconsistency in greater detail or to generate corresponding explanations. Further-

more, both works are focusing the context of OBDA and hence do not support the federation of distributed KBs.

Another approach that facilitates meaningful query results over an inconsistent *DL-Lite* KB under different inconsistency-tolerant semantics is proposed by Lembo et al. in [Lem+11; Lem+12]. In order to implement inconsistency tolerance on query answering an additional rewriting under the defined semantics is applied to query expansions generated by the algorithm PerfectRef proposed in [Cal+07b]. Roughly speaking, expanded (U)CQs are modified such that assertions causing inconsistency are not considered on query answering. Similar to our approach the algorithm for generating an inconsistency-tolerant (U)CQ uses any TBox axiom that may be contradicted by ABox assertions. A subsequent work to that is proposed by Savo [Sav13] and addresses additionally the issue of updating inconsistent KBs. However, while all these works primary targeting the exclusion of all assertions causing inconsistency from query evaluation, our claim is exactly the opposite as we precisely select those assertions. Hence, even applicable to the context of OBII, i.e., to access inconsistent and distributed KBs, such approaches are not designed for KB debugging.

Nevertheless, there exist approaches enabling distributed reasoning capabilities over interrelated data sources (repositories) such as the tableau-based DL reasoning algorithm offered by Serafini and Tamilin [ST04]. However, this approach follows principles of peer-to-peer networks where each integrated data source has to implement a peer ontology manager and must provide local and global reasoning services. Because of these imposed requirements, Especially these imposed requirements restrict the integration of arbitrary data sources such as in a loosely coupled network of KBs and are thus contradictory to the principles followed in OBII.

More concerned with the context of OBII, Ji et al. [Ji+09] proposed an approach tackling the debugging of an inconsistent KB network. However, even applicable to loosely coupled LOD sources, the presented strategies are based on the assumption that the individual KBs are locally consistent and the debugging is solely performed on the mappings between the KBs.

4.5 Summary

In this chapter we have mainly answered research question Q1 that asks for a formal description of inconsistency management in federated knowledge bases and its peculiarities. Starting with a discussion about the problem of federated reasoning over a network of loosely coupled KBs, we could identify FOL-rewritability as an appropriate property of the underlying DL language and justified why *DL-Lite*$_{\mathcal{A}}$ is sufficient for our purpose. Based on the identified peculiarities and requirements we have subsequently proposed our approach of efficient inconsistency detection in federated *DL-Lite*$_{\mathcal{A}}$ KBs based on clash queries. After defining the generation of clash queries and federated query answering we have introduced our notion of

source-related ABox assertions, based on which gave a definition for generating the corresponding explanations, called MISAs. Since our proposed definition of federated querying is compatible with the syntax of SPARQL queries, our approach does not place any specific requirements to the integrated data sources, but relies only on the usage of existing standard SPARQL interfaces.

By introducing our notion of source-related ABox assertions we gave also a first partial answer to question Q2**:** How can the process of debugging federated knowledge bases be designed in a convenient, efficient, and eligible way? As we preserve the information of the originating source we are able to differentiate equivalent assertions of various sources and may benefit from this information on resolving the identified found contradictions.

Chapter 5

Repair Plan Generation

After having found an efficient approach for detecting and explaining inconsistency in a federated *DL-Lite*$_\mathcal{A}$ KB, we can now tackle the challenge of providing an appropriate repair for the inconsistent KB. In Section 5.1 we start with the notion of repair and introduce in Section 5.2 the representation of the conflicting assertions and their relationships as a conflict graph. Subsequently, we propose in Section 5.3 our approach of repair generation based on majority voting. As only a subset of all MISAs may are resolved, we use in Section 5.4 the majority voting-based repair to determine a data source specific measure of validity for certain types of assertions, called signature accuracy. Based on these statistical evidences of the previous repair we proposed in Section 5.5 a complementary approach to resolve all remaining MISAs. In the last parts of this chapter (Section 5.6 and Section 5.7) we discuss the relation of our approach to other works and summarize our results.

Our basic approach of repair generation was originally published in [Nol+16]. However, since the proposed algorithms for majority voting and learned repairs were incomplete in the sense that on the one hand entailment relations between assertions were not considered and on the other hand the generated repair was not ensured to be minimal. Hence, the corresponding extensions in Section 5.3 and Section 5.5 have not yet been published. Moreover, Section 5.4 of this chapter comprising our definition of signature accuracy is directly adapted from the work we have already presented in [Nol+17].

5.1 Notion of Repair

One of the most common approaches to resolve an inconsistent KB $\mathcal{K}$ is to find an appropriate subset $\mathcal{K}'$ of $\mathcal{K}$ that is consistent. While repairing an inconsistent KB $\mathcal{K}$, i.e., by removing some statements (axioms and assertions), an obvious intension is to keep the changes in $\mathcal{K}$ to a minimum. The set $\mathcal{R}$ of statements that are removed from $\mathcal{K}$ to gain a consistent KB is called a *repair* (sometimes also called repair plan or repair solution). Formally, we can define a repair as follows:

Definition 5.1 (Repair)**.** *Given an inconsistent knowledge base $\mathcal{K}$, a repair $\mathcal{R}$ of $\mathcal{K}$ is defined as a subset $\mathcal{R} \subseteq \mathcal{K}$ for which $Mod(\mathcal{K} \setminus \mathcal{R}) \neq \emptyset$ holds and there exists no proper subset $\mathcal{R}' \subset \mathcal{R}$ for which $Mod(\mathcal{K} \setminus \mathcal{R}') \neq \emptyset$ holds.*

Hence, a repair $\mathcal{R}$ is always minimal, i.e., there exist no repair that is a proper subset of $\mathcal{R}$. The resulting KB $\mathcal{K}'$ given by $\mathcal{K}' = \mathcal{K} \setminus \mathcal{R}$ is called a *knowledge base solution*. Since we assume that the federated TBox $\mathcal{T}_F$ of $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$ is already correct, we focus within the scope of this thesis only on *ABox repairs*, i.e., in order that $\mathcal{R} \subseteq \mathcal{A}_F$.

Note that there exists a dual notion of repairs used in the context of databases where, in contrast to the definition above, a repair of an inconsistent database is a new database that is consistent, i.e., satisfies the given integrity constrains, and differs from the initial database to a minimum [ABC99]. However, we will rely on the notion of repairs as defined above since this notion has already been established in context of KBs resp. ontologies by works like [Kal06; Kal+06; HPS09].

The computation of a repair, and more generally the task of KB debugging directly corresponds to the field of *model-based diagnosis* [Rei87; KW87]. Given a model describing the behavior of a system, the model-based diagnosis targets the identification and correction of faulty system behavior. Based on *minimal conflict sets* representing minimal faulty subsets of the system, an appropriate *diagnosis* is computed, which represents a minimal set of system components that if removed from the system will repair the faulty behavior. Hence, an explanation is equivalent to a minimal conflict set and a repair corresponds to a diagnosis. Hence, a minimal conflict set is equivalent to an explanation and a diagnosis corresponds to a repair. A prominent algorithm that is used in model-based diagnosis is the hitting set tree algorithm proposed by Reiter [Rei87]. Given a collection $\mathcal{S}$ of sets, a *hitting set* $\mathcal{H}$ is a set that comprise at least one element of each set in the collection, i.e., $\mathcal{H} \subseteq \bigcup_{S \in \mathcal{S}} S$ such that $\mathcal{H} \cap S \neq \emptyset$ for every set $S \in \mathcal{S}$. Similar to Definition 5.1 of repairs, a hitting set $\mathcal{H}$ is called *minimal* iff there exists no proper subset $\mathcal{H}' \subset \mathcal{H}$ that is also a hitting set. Moreover, a hitting set $\mathcal{H}$ is called *smallest minimal* iff there is no other hitting set $\mathcal{H}'$ with a smaller number of elements, i.e., $\#\mathcal{H} \leq \#\mathcal{H}'$ holds for every hitting set $\mathcal{H}'$ of $\mathcal{S}$. To compute possible minimal hitting sets Reiter's algorithm constructs a finite tree where the vertices (also called nodes) are labeled with minimal conflict sets and the edges with components of the system. Since an explanation, i.e., a MIS (Definition 4.6) or a MISA (Definition 4.7) is a minimal inconsistent subset of a KB that becomes consistent if one element of the subset is removed, a minimal hitting set for the set of all MISs resp. MISAs corresponds to a repair for an inconsistent KB. Moreover, the hitting set problem can be seen as a generalization of the well-known problem of finding a minimal *vertex cover*. Given an undirected graph $G = (V, E)$, a set of vertices $V' \subseteq V$ is called a vertex cover, if each edge in E of G is incident to at least one vertex of V', i.e., $V' \cap \{v_1, v_2\} \neq \emptyset$ for every edge $\{v_1, v_2\} \in E$ of G. Both, the construction of a smallest minimal hitting set as well as the computation of a smallest minimal vertex cover are each one of Karp's NP-complete problems [Kar72].

5.2 Conflict Graph

Obviously, on determining a repair for an inconsistent federated *DL-Lite*$_\mathcal{A}$ KB $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$ the corresponding set $\mathcal{M}$ of MISAs generated by Algorithm 4.1 cannot be treated separately since a single assertion may contradicts more than one assertion, i.e., is part of several MISAs. Resolving a MISA by adding an assertion to the repair thus may affect, i.e., may resolve other MISAs as well. Moreover, the removal of an assertion may affect some other assertions according to the axioms of the TBox $\mathcal{T}$. Given for example a concept inclusion axiom of the form $C_2 \sqsubseteq C_1$ and the two assertions $\langle C_1(\sigma_I), n \rangle$ and $\langle C_2(\sigma_I), m \rangle$. If $\langle C_1(\sigma_I), n \rangle$ is now added to the repair, this would imply that the assertion $\langle C_2(\sigma_I), m \rangle$ has to be part of the repair as well due to the axiom $C_2 \sqsubseteq C_1$. We call this relationship an *entailment relation* between assertions which is defined as follows:

Definition 5.2 (Entailment Relation between Assertions)**.** *Given a federated DL-Lite*$_\mathcal{A}$ *knowledge base* $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$ *and two assertions* $\alpha_n = \langle \alpha, n \rangle$ *and* $\alpha'_m = \langle \alpha', m \rangle$ *where* $\alpha \in \mathcal{A}_n$*,* $\alpha' \in \mathcal{A}_m$ *and* $\mathcal{A}_n, \mathcal{A}_m \in \mathcal{A}_F$*. If* $\langle \mathcal{T}, \{\alpha'\} \rangle \models \alpha$ *we say that the assertion* α' *resp.* α'_m *has an entailment relation to* α *resp.* α_n *and shortly write* $\alpha'_m \models_\mathcal{T} \alpha_n$*. We denote an entailment relation by a binary tuple* (α'_m, α_n) *and the finite set of all entailment relations within* $\mathcal{K}_F$ *by* $\mathcal{E}$*.*

Obviously, given that an assertion α is stated by two source $\mathcal{A}_1$ and $\mathcal{A}_2$, i.e., $\alpha_1 = \langle \alpha, 1 \rangle$ and $\alpha_2 = \langle \alpha, 2 \rangle$, there exist the two entailment relations (α_1, α_2) and (α_2, α_1). According to the syntax of *DL-Lite*$_\mathcal{A}$ (see Definition 2.15) any subsumption axiom generally comprises only one element on the left and one element on the right hand side of the subsumption relation, or can be normalized to that form, why in *DL-Lite*$_\mathcal{A}$ all entailment relations between assertions can be sufficiently expressed as per Definition 5.2. Moreover, since a negative inclusion, i.e., a negative inclusion axiom, a value-domain axiom (i.e., attribute range) or a functionality assertion axiom for a role or an attribute, is affected only to subsumed elements in $\mathcal{T}$ (i.e., concepts, roles or attributes), we can conclude that all assertions which contradict α are also in contradiction to each assertions α' for which $\alpha' \models_\mathcal{T} \alpha$ holds. Formally, let $\mathcal{C}_\alpha = \{\alpha_\mathcal{C} \mid \{\alpha, \alpha_\mathcal{C}\} \in \mathcal{M}\}$ and $\mathcal{C}_{\alpha'} = \{\alpha'_\mathcal{C} \mid \{\alpha', \alpha'_\mathcal{C}\} \in \mathcal{M}\}$ denote the sets of conflicting assertions of α resp. of α', then $\mathcal{C}_\alpha \subseteq \mathcal{C}_{\alpha'}$ if $\alpha' \models_\mathcal{T} \alpha$. Accordingly, if an assertion α is added to the repair, each assertion α' for which $\alpha' \models_\mathcal{T} \alpha$ holds have to be added to the repair as well.

Consequently, the set $\mathcal{C} = \bigcup_{\boldsymbol{m} \in \mathcal{M}} \boldsymbol{m}$ of all *conflicting assertions*, i.e., all assertions that are part of at least one MISA, constitutes a complex network of correlated assertions. Based on Definition 4.7 of MISAs and Definition 5.2 of entailment relations, this network can be modeled as a *conflict graph* $G_\mathcal{C} = (\mathcal{C}, \mathcal{M}, \mathcal{E})$, where each conflicting assertion $\alpha \in \mathcal{C}$ is represented by a vertex and a contradiction between two assertions described by a MISA $\{\alpha, \alpha'\} \in \mathcal{M}$ is represented by an undirected edge. An unary MISA $\{\alpha\} \in \mathcal{M}$ is represented by an undirected *loop*, connecting the assertion α to itself. Moreover, an entailment relation $(\alpha', \alpha) \in \mathcal{E}$ of an assertion $\alpha' \in \mathcal{C}$ to an assertion $\alpha \in \mathcal{C}$, i.e., $\alpha' \models_\mathcal{T} \alpha$, is represented by

a directed dotted edge, indicating the direction of the corresponding entailment relation.

Example 5.1 (Conflict Graph)**.** *Referring to our running example of Section 3.3, the complete set $\mathcal{M}$ of MISAs generated by Algorithm 4.1 is given with*

$$\begin{aligned}\mathcal{M} = \big\{ &\{\alpha_1, \beta_2\}, \{\alpha_1, \beta_3\}, \{\alpha_1, \gamma_1\}, \{\alpha_2, \beta_2\}, \{\alpha_2, \beta_3\}, \{\alpha_2, \gamma_1\}, \{\alpha_3, \gamma_2\}, \\ &\{\alpha_3, \gamma_3\}, \{\alpha_4, \beta_4\}, \{\alpha_4, \gamma_4\}, \{\alpha_4, \gamma_5\}, \{\alpha_5, \beta_4\}, \{\alpha_5, \beta_5\}, \{\alpha_5, \gamma_5\}, \\ &\{\alpha_9, \beta_9\}, \{\beta_1, \beta_2\}, \{\beta_1, \beta_3\}, \{\beta_1, \gamma_1\}, \{\beta_2, \gamma_1\}, \{\beta_3, \gamma_1\}, \{\beta_4, \gamma_4\}, \\ &\{\beta_4, \gamma_5\}, \{\beta_5, \gamma_5\}, \{\beta_8, \gamma_7\}, \{\beta_8, \gamma_8\}, \{\beta_9\}, \{\gamma_7, \gamma_8\} \big\}.\end{aligned}$$

Moreover, according to Definition 5.2 the set $\mathcal{E}$ of entailment relations between conflicting assertions is determined by

$$\mathcal{E} = \big\{ (\alpha_1, \beta_1), (\alpha_2, \alpha_1), (\alpha_2, \beta_1), (\beta_1, \alpha_1), (\beta_3, \beta_2), (\beta_4, \beta_5), (\gamma_3, \gamma_2) \big\}.$$

The resulting conflict graph for our running example consists of fife independent subgraphs and is depicted in Figure 5.1.

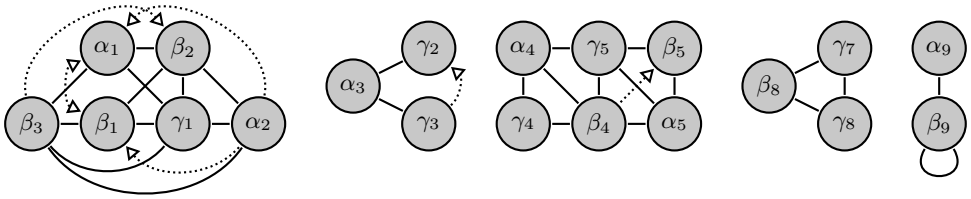

Figure 5.1: Conflict Graph

It is easy to see that while unary MISAs are simple to resolve, assertions of binary MISAs usually do have relations (conflict relations and/or entailment relations) to other assertions why the resolution of a MISA may depend on or may affect the resolution of other MISAs.

5.3 Majority Voting-Based Repair

To resolve the identified contradictions (MISAs) within a given federated KB we follow the assumption that the more data sources are integrated, the higher the likelihood that valid assertions are present more frequently. Conversely, the probability that an assertion is not valid correlates with the number of MISAs in which the assertion is involved.

Following this assumption on generating a repair for a given set $\mathcal{M}$ of MISAs, we will now propose a greedy approach that identifies those assertions within the conflict graph, that are contradicted more frequently and hence are likely to be not valid. Thus, in order to apply an appropriate majority voting heuristic, we define the *cardinality of an assertion* α denoted by $\#\alpha$ as the number of MISAs in which α is involved, i.e., formally given by $\#\alpha = \#\{\bigcup_{\boldsymbol{m} \in \mathcal{M} \mid \alpha \in \boldsymbol{m}} \boldsymbol{m}\}$. Based on

these cardinalities we can make a decision in favour of one of two contradicting assertions in order to resolve the corresponding MISA, as far as both cardinalities are different. Hence, we refer to such MISAs as *resolvable MISAs* and otherwise as *unresolvable MISAs*. However, since adding an assertion to the repair affects the cardinality of each of its contradicting assertion and hence may influence the following steps, we start with the resolution of MISAs that are less intricate, or in other words that are less connected within the conflict graph. For this purpose we define the cardinality of a MISA $\boldsymbol{m}$ as the sum of all cardinalities of its assertions, i.e., formally with $\#\boldsymbol{m} = \sum_{\alpha \in \boldsymbol{m}} \#\alpha$. Based on that, by resolving MISAs with lowest cardinality first, we are reducing the impact (of potentially wrong decisions) on subsequent decisions.

Algorithm 5.1 depicts in detail our approach for generating a majority voting-based repair. Given a set $\mathcal{M}$ of MISAs, the algorithm starts with the trivial resolution of unary MISAs (resulting from case (i) and (iii) of clash types explained in Section 4.2.1) by adding the only assertion of each unary MISA to the repair (Line 4). As the assertion α of an unary MISA might also be part of other (binary) MISAs that will also be resolved by adding α to the repair, the set $\mathcal{M}_\alpha$ of all MISAs comprising α is removed from $\mathcal{M}$ (Line 6). The subsequent part (while loop) of the algorithm is dealing with the non-trivial resolution of the remaining set $\mathcal{M}$ of binary MISAs. After the current cardinalities for all remaining assertions and MISAs in the conflict graph are calculated, the set $\mathcal{M}_{minCard}$ of all resolv-

Algorithm 5.1: GenerateMajorityVotingBasedRepair($\mathcal{M}$)

Input: set $\mathcal{M}$ of MISAs
Output: (partial) repair $\mathcal{R}$ generated by majority voting

```
1  begin
2      R ← ∅
3      foreach m ∈ M | #m = 1 do
4          R ← R ∪ m
5          M_α ← {m' ∈ M | m ∩ m' ≠ ∅}
6          M ← M \ M_α
7      while true do
8          M_minCard ← GetResolvableMISAsWithMinCardinality(M)
9          if M_minCard = ∅ then
10             break
11         foreach m ∈ M_minCard do
12             α ← GetAssertionWithMaxCardinality(m, M)
13             R ← R ∪ α
14             M_α ← {m' ∈ M | α ∈ m'}
15             M ← M \ M_α
16     return R
17 end
```

able MISAs with minimum cardinality is determined (Line 8). For each MISA in $\mathcal{M}_{minCard}$ the assertion α with higher cardinality is added to the repair (Line 13) and all resolved MISAs comprising α are removed from $\mathcal{M}$ (Line 15). This part (Line 8 to 15) of the algorithm is repeated until no resolvable MISAs are left and the algorithm terminates.

It is easy to see that Algorithm 5.1: GenerateMajorityVotingBasedRepair runs in polynomial time with respect to the number of vertices, i.e., MISAs. By applying an efficient heuristic each step in the algorithm corresponds to an eligible decision with a least possible impact to the remaining conflict graph and hence to subsequent decisions.

However, even this approach is aimed to find a good repair by trying to identify exactly those assertions that are actually wrong, from a theoretical point of view the algorithm obviously does not guarantee that the resulting (partial) repair is minimal according to Definition 5.1. Because of that, in the following we propose the additional Algorithm 5.2: MinimizeRepair minimizing the (partial) repair $\mathcal{R}$ generated by Algorithm 5.1. Given the original set $\mathcal{M}$ of MISAs and the (partial) repair $\mathcal{R}$, the algorithm removes every assertion α from the repair, if there exists no unary MISA $\{\alpha\} \in \mathcal{M}$ and the complete set $\mathcal{C}_\alpha$ of assertions that originally contradicted α is part of the repair as well.

Algorithm 5.2: MinimizeRepair$(\mathcal{M}, \mathcal{R})$

Input: set $\mathcal{M}$ of MISAs,
(partial) repair $\mathcal{R}$
Output: minimal (partial) repair $\mathcal{R}'$

```
1 begin
2     R' ← R
3     foreach α ∈ R do
4         if {α} ∉ M then
5             C_α ← ⋃_{m ∈ M | α ∈ m} m \ α
6             if C_α \ R' = ∅ then
7                 R' ← R' \ α
8     return R'
9 end
```

Proposition 5.1. *Given a set $\mathcal{M}$ of MISAs and a corresponding (partial) repair $\mathcal{R}$ generated by Algorithm 5.1:* GenerateMajorityVotingBasedRepair*, then Algorithm 5.2:* MinimizeRepair *generates a repair $\mathcal{R}'$ that is always minimal for the resolved MISAs.*

Proof. Obviously, the algorithm removes an assertion α from the repair only if the current repair comprises all assertions that originally contradicted α, i.e., each assertion $\alpha_\mathcal{C}$ for which $\{\alpha, \alpha_\mathcal{C}\} \in \mathcal{M}$ holds. Hence, given $\mathcal{R}' = \mathcal{R} \setminus \alpha$ we can conclude that $\{\boldsymbol{m} \in \mathcal{M} \mid \boldsymbol{m} \cap \mathcal{R} = \emptyset\} = \{\boldsymbol{m} \in \mathcal{M} \mid \boldsymbol{m} \cap \mathcal{R}' = \emptyset\}$. More-

over, since *DL-Lite*$_\mathcal{A}$ is monotonic[1] and Algorithm 5.2 terminates only after no more assertion can be removed from $\mathcal{R}'$, we can further conclude that after the termination of Algorithm 5.2 there exists no proper subset $\mathcal{R}'' \subset \mathcal{R}'$ for which $\{\boldsymbol{m} \in \mathcal{M} \mid \boldsymbol{m} \cap \mathcal{R}' = \emptyset\} = \{\boldsymbol{m} \in \mathcal{M} \mid \boldsymbol{m} \cap \mathcal{R}'' = \emptyset\}$ holds. □

Proposition 5.2. *Given a set $\mathcal{M}$ of MISAs and a corresponding (partial) repair $\mathcal{R}$ generated by Algorithm 5.1:* GenerateMajorityVotingBasedRepair*, then Algorithm 5.2:* MinimizeRepair *generates a repair $\mathcal{R}'$ that considers for each assertion $\alpha \in \mathcal{R}'$ every entailment relation $\alpha' \models_\mathcal{T} \alpha$, i.e., $\alpha \in \mathcal{R}' \rightarrow \alpha' \in \mathcal{R}'$ if $\alpha' \models_\mathcal{T} \alpha$, as long as the MISAs comprising α' are resolved by Algorithm 5.1.*

Proof. An entailment relation $\alpha' \models_\mathcal{T} \alpha$ is not satisfied only if the assertion α is part of the repair but not α'. Thus, given that $\alpha' \models_\mathcal{T} \alpha$ and $\alpha \in \mathcal{R}'$ but $\alpha' \notin \mathcal{R}'$, we can conclude that α' was originally involved in at least one additional MISA compared to α since otherwise α' would have been treated identically to α in Algorithm 5.1. Hence, we know that $\mathcal{C}_\alpha \subset \mathcal{C}_{\alpha'}$, where $\mathcal{C}_\alpha = \{\alpha_\mathcal{C} \mid \{\alpha, \alpha_\mathcal{C}\} \in \mathcal{M}\}$ and $\mathcal{C}_{\alpha'} = \{\alpha'_\mathcal{C} \mid \{\alpha', \alpha'_\mathcal{C}\} \in \mathcal{M}\}$ denote the sets of conflicting assertions of α resp. of α'. Due to Proposition 5.1 we can conclude that at least one assertion $\alpha_\mathcal{C} \in \mathcal{C}_\alpha$ that originally contradicted α, i.e., for which $\{\alpha, \alpha_\mathcal{C}\} \in \mathcal{M}$ holds, is not part of the repair, i.e., $\alpha_\mathcal{C} \notin \mathcal{R}'$. Moreover, since we already know that $\mathcal{C}_\alpha \subset \mathcal{C}_{\alpha'}$, we can further conclude that $\alpha_\mathcal{C}$ is also contradicting α' such that $\{\alpha', \alpha_\mathcal{C}\} \in \mathcal{M}$. As Algorithm 5.1 terminates only after no resolvable MISA is left, we can conclude that the MISA $\boldsymbol{m} = \{\alpha', \alpha_\mathcal{C}\}$ is unresolvable. Otherwise, either α' would be part of the repair which contradicts the initial assumption or $\alpha_\mathcal{C}$ would be part of the repair which contradicts Proposition 5.1. □

Hence, to ensure that entailment relations of assertions ending up in unresolvable MISAs are also considered, we have to apply an additional algorithm extending the minimal (partial) repair $\mathcal{R}'$ of Algorithm 5.2: MinimizeRepair. Given the original set $\mathcal{M}$ of MISAs and the minimal (partial) repair $\mathcal{R}'$, the following Algorithm 5.3: ExtendRepairByEntailmentRelatedAssertions determines the set $\mathcal{C}'$ of all assertions that are still involved in any conflict after $\mathcal{R}'$ would be removed. Each assertion $\alpha' \in \mathcal{C}'$ is added to the repair if there exists an entailment relation $\alpha' \models_\mathcal{T} \alpha$ to an assertion α that is already part of the repair $\mathcal{R}'$.

Since $\mathcal{R}'$ is already minimal for the resolved MISAs and $\mathcal{C}_\alpha \subseteq \mathcal{C}_{\alpha'}$ holds for each assertion added to the repair by Algorithm 5.3, it directly follows that Proposition 5.1 still holds for the extended repair $\mathcal{R}''$.

Moreover, as we already know that before the repair $\mathcal{R}''$ is applied the set $\mathcal{M}'$ of all remaining MISAs, given by $\mathcal{M}' = \bigcup_{\boldsymbol{m} \in \mathcal{M} \mid \boldsymbol{m} \cap \mathcal{R}' = \emptyset} \{\boldsymbol{m}\}$, comprises only unresolvable MISAs and hence results in a conflict graph that can be divided into independent subgraphs where each independent subgraph inevitably consists of assertions having the same cardinality. As a consequence, the application of

[1] Note that any DL being a subset of FOL is monotonic, which means that an additional statement (i.e., axioms or assertions) always leads to supplementary logical consequences, i.e., given that $\mathcal{K} \subseteq \mathcal{K}'$ then we can conclude that $Mod(\mathcal{K}') \subset Mod(\mathcal{K})$ [Sav13].

Algorithm 5.3: ExtendRepairByEntailmentRelatedAssertions($\mathcal{M}, \mathcal{R}'$)

Input: set $\mathcal{M}$ of MISAs,
minimal (partial) repair $\mathcal{R}'$
Output: extended (partial) repair $\mathcal{R}''$

```
1 begin
2     C' = ⋃_{m ∈ M | m ∩ R' = ∅} m
3     R'' ← R'
4     foreach α' ∈ C' do
5         if ⟨T, {α'}⟩ ⊨ α ∈ R' then
6             R'' ← R'' ∪ α'
7     return R''
8 end
```

Algorithm 5.3 generates a repair that, applied to the remaining conflict graph, may removes assertions from a subgraph, resulting in some of the remaining MISAs now become resolvable. In order to maximize the set of resolved MISAs, Algorithm 5.1 can again be applied to the remaining set $\mathcal{M}''$ of MISAs not yet resolved by the repair $\mathcal{R}''$, where $\mathcal{M}'' = \bigcup_{\boldsymbol{m} \in \mathcal{M} \mid \boldsymbol{m} \cap \mathcal{R}'' = \emptyset} \{\boldsymbol{m}\}$. However, a subsequent application of Algorithm 5.2: MinimizeRepair is not required, since the remaining set $\mathcal{M}'$ already comprises only unresolvable MISAs why the subsequent application of Algorithm 5.1 to $\mathcal{M}''$ will end up in a repair that is always minimal for the resolved MISAs. Moreover, given an entailment relation $\alpha' \models_{\mathcal{T}} \alpha$ between two assertions $\alpha, \alpha' \in \mathcal{C}'$ that both are still involved in some MISAs, from the fact that $\mathcal{M}'$ already comprises only unresolvable MISAs we can further conclude that $\mathcal{C}_\alpha = \mathcal{C}_{\alpha'}$ holds for $\alpha, \alpha' \in \mathcal{C}'$. Because of that, both assertions α and α' are treated identically in Algorithm 5.1, why the consideration of its entailment relation is ensured and hence a supplementary application of Algorithm 5.3 and thus of any of the previous algorithms would not be effective in order to further resolve any of the remaining MISAs $\boldsymbol{m} \in \mathcal{M}''$.

The following Algorithm 5.4: GenerateRepairForResolvableMISAs summarizes the complete approach proposed of this section.

Algorithm 5.4: GenerateRepairForResolvableMISAs($\mathcal{M}$)

Input: set $\mathcal{M}$ of MISAs
Output: minimal (partial) repair $\mathcal{R}$

```
1 begin
2     R ← GenerateMajorityVotingBasedRepair(M)
3     R ← MinimizeRepair(M, R)
4     R ← ExtendRepairByEntailmentRelatedAssertions(M, R)
5     M' ← ⋃_{m ∈ M | m ∩ R = ∅} {m}
6     R ← R ∪ GenerateMajorityVotingBasedRepair(M')
7     return R
8 end
```

As a result, we can conclude that the computational complexity of Algorithm 5.4 is in PTIME with respect to the size of the knowledge base. From a theoretical point of view both extensions, Algorithm 5.2: MinimizeRepair and Algorithm 5.3: ExtendRepairBy EntailmentRelatedAssertions, and hence the subsequent application of Algorithm 5.1: GenerateMajorityVotingBasedRepair are necessary in order to ensure that the resulting (partial) repair is minimal according to Definition 5.1 and every entailment relation is considered. However, as we will observe in our experimental evaluation (see Chapter 9) both of these algorithms do not have any effect on the generated repair why we can empirically conclude that (at least for the used dataset) the addressed cases are (mainly) artificial and usually do not occur in practice.

Example 5.2 (Majority Voting-Based Repair Generation)**.** *Figure 5.2 illustrates the application of Algorithm 5.1 to each independent subgraph of the conflict graph (see Figure 5.1) for our running example. While in the rightmost subgraph the unary MISA* $\{\beta_9\}$ *is resolved at the first step by adding the assertion* β_9 *(depicted as dashed node) to the repair, for the other subgraphs the (resolvable) MISAs with the lowest cardinality are identified. As explained at the beginning of this section, the cardinality of an assertion is defined as the number of MISAs in which the assertion is involved and the cardinality of a MISA is given by the sum of all cardinalities of its assertions. Note that the graph is only annotated with assertion cardinalities but the MISA cardinalities are omitted for reasons of clarity. For example, in the leftmost subgraph the lowest MISA cardinality is 7, why the MISAs* $\{\alpha_1, \beta_2\}$, $\{\alpha_1, \beta_3\}$, $\{\alpha_2, \beta_2\}$, $\{\alpha_2, \beta_3\}$, $\{\beta_1, \beta_2\}$ *and* $\{\beta_1, \beta_3\}$ *are treated in the first step.*

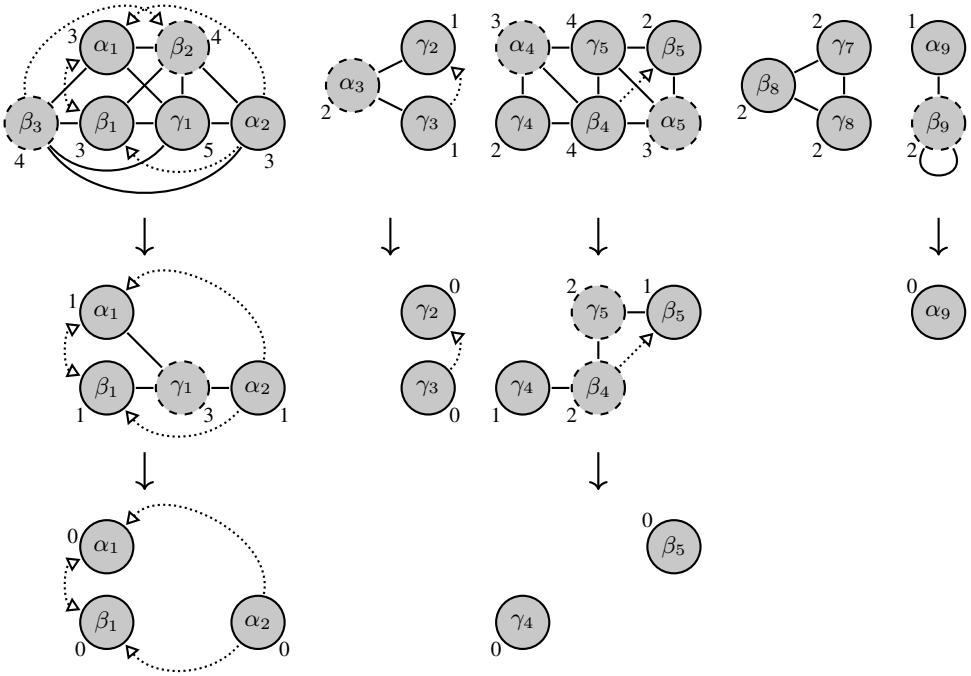

Figure 5.2: Majority Voting

For each selected MISA the assertion with the higher cardinality is added to the repair, which is in the considered case assertion β_2 and β_3 (depicted as dashed nodes). Subsequently, the modified subgraph is then processed in the next iteration and the algorithm terminates once no more resolvable MISAs can be found.

As illustrated in the figure, after at most 2 iterations there remain only MISAs that are not resolvable via majority voting and Algorithm 5.1 returns a repair $\mathcal{R}$ that is given with

$$\mathcal{R} = \{\, \alpha_3, \alpha_4, \alpha_5, \beta_2, \beta_3, \beta_4, \beta_9, \gamma_1, \gamma_5 \,\}\,.$$

It is easy to see that Algorithm 5.2 and Algorithm 5.3, and hence the subsequent application of Algorithm 5.1 as defined in Algorithm 5.4 do not have any effect on the generated repair since $\mathcal{R}$ is already minimal and each entailment relation is considered.

As illustrated in Example 5.2, our majority voting-based approach proposed in this section cannot resolve all logical conflicts of our running example. Especially if the originally given set $\mathcal{M}$ of MISAs already comprises unresolvable MISAs that are not connected to any resolvable MISA, this leads to logical conflicts not addressed by the generated (partial) repair. Predestinated for such MISAs are in particular contradictory assertions of different values for functional roles or attributes. In order to get a complete repair resolving all logical conflicts we will use in the following section the statistical evidence implicitly given by the current partial repair to resolve the remaining MISAs.

5.4 Signature Accuracy

Suppose for example that, contrary to assertions of the concept C stated in $\mathcal{A}_n$, a large fraction of the assertions of concept C' in $\mathcal{A}_m$ is part of the (partial) repair. Relying on the correctness of the (partial) repair and given an unresolved MISA $\{\langle C(\sigma_I), n\rangle, \langle C'(\sigma_I), m\rangle\}$, we can place more confidence in assertion $\langle C(\sigma_I), n\rangle$ and hence add $\langle C'(\sigma_I), m\rangle$ to the repair.

For this purpose we determine for each element (concept names, role names or attribute names) of a source-specific (TBox) signature an accuracy value, called *signature accuracy*[2]. The calculation of the accuracy of a signature element with respect to a specific data source is based on the set of conflicting assertions and the set of assertions that are *correct*. Here we understand correct assertions to mean all non-conflicting assertions that comprise at least one individual not occurring in a conflicting assertion but in a non-conflicting assertion stated in at least one other data source integrated in the federated KB. Besides, the set of conflicting assertions can be further divided into the following subcategories:

- *likely false assertions* which are those assertions that are part of the repair generated by our majority voting-based approach proposed in Section 5.3

[2]Note that we intentionally avoid here the terms 'trust' or 'probability' in order to prevent any confusion with the calculated trust values in Chapter 6.

- *likely true assertions* that are all assertions that become conflict-free after the repair is removed, and
- *still conflicting assertions* denoting all assertions that are still part of some MISAs not resolved by the repair so far.

The conjunction of both, the set of conflicting assertions and the set of correct assertions, can be considered as an adequate sample of all assertions of the federated ABox $\mathcal{A}_F$.

Accordingly, we can formally define the signature accuracy as follows:

Definition 5.3 (Signature Accuracy)**.** *Given a set $\boldsymbol{\mathcal{K}} = \{\mathcal{K}_1, \ldots, \mathcal{K}_N\}$ of knowledge bases (data sources) where each $\mathcal{K}_n$ is defined with $\mathcal{K}_n = \langle \mathcal{T}_n, \mathcal{A}_n \rangle$ over a signature Σ_n and let $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$ denotes an inconsistent federated DL-Lite$_\mathcal{A}$ knowledge base over the signature $\Sigma_F = \Sigma_I \cup \bigcup_{n=1,\ldots,N} \Sigma_n$ integrating $\boldsymbol{\mathcal{K}}$, where $\mathcal{T}_F = \mathcal{T}_I \cup \bigcup_{n=1,\ldots,N} \mathcal{T}_n$ denotes the federated TBox, $\mathcal{A}_F = \bigcup_{n=1,\ldots,N} \mathcal{A}_n$ the federated ABox, $\mathcal{T}_I$ is an intermediary TBox over the signature Σ_F and Σ_I denotes an additional signature used in $\mathcal{T}_I$ for the intermediary description of the federated domain of interest. For a simplified notation of this definition we assume that every ABox $\mathcal{A}_n \in \mathcal{A}_F$ comprise source-related ABox assertions such that $\mathcal{A}_n = \bigcup_{\alpha \in \mathcal{A}_n} \langle \alpha, n \rangle$. Moreover, let $\mathcal{M}$ denotes the complete set of MISAs for $\mathcal{K}_F$ and $\mathcal{R}$ is the (partial) repair generated by the majority voting-based approach proposed in Section 5.3. Then, the set $\mathcal{C}$ of all conflicting assertions is given by $\mathcal{C} = \bigcup_{\boldsymbol{m} \in \mathcal{M}} \boldsymbol{m}$, the set of likely false assertions corresponds to the repair $\mathcal{R}$, the set of still conflicting assertions is determined with $\mathcal{C}' = \bigcup_{\boldsymbol{m} \in \mathcal{M}'} \boldsymbol{m}$, where $\mathcal{M}' = \bigcup_{\boldsymbol{m} \in \mathcal{M} \mid \boldsymbol{m} \cap \mathcal{R} = \emptyset} \{\boldsymbol{m}\}$ denotes the set of all MISAs not resolved by $\mathcal{R}$, and the set of likely true assertions is given by $\mathcal{C}_{true} = \mathcal{C} \setminus (\mathcal{R} \cup \mathcal{C}')$. As we denote a non-conflicting assertion $\alpha \in \mathcal{A}_n$ as a correct assertion if the described individual(s) also occur in a non-conflicting assertion $\alpha' \in (\mathcal{A}_F \setminus \mathcal{A}_n)$, the set $\mathcal{A}'_F$ of all correct assertions is defined with $\mathcal{A}'_F = \bigcup_{\mathcal{A}_n \in \mathcal{A}_F} \bigcup_{\alpha \in (\mathcal{A}_n \setminus \mathcal{C})} \alpha \mid ind(\alpha) \cap ind(\alpha' \in (\mathcal{A}_F \setminus (\mathcal{A}_n \cup \mathcal{C}))) \neq \emptyset$, where $ind(\alpha)$ is an external function that returns the (unary or binary) set of individuals of an assertion α.*

Given a signature element σ of data source $\mathcal{K}_n$ that is either a concept name, a role name or an attribute name such that $\sigma \in \langle \Sigma_C, \Sigma_R, \Sigma_A \rangle$, where Σ_C denotes the set of concept names, Σ_R the set of role names and Σ_A the set of attribute names in the signature $\Sigma_n = \langle \Sigma_I, \Sigma_V, \Sigma_C, \Sigma_D, \Sigma_R, \Sigma_A \rangle$ of data source $\mathcal{K}_n$. The signature accuracy for σ (with respect to data source $\mathcal{K}_n$) is defined as

$$acc(\sigma, n) = \begin{cases} 1 - \dfrac{\#\{\alpha \in \mathcal{A}_n \cap \mathcal{R} \mid \sigma(\alpha) = \sigma\} + \displaystyle\sum_{\alpha \in \mathcal{A}_n \cap \mathcal{C}' \mid \sigma(\alpha) = \sigma} \frac{\#\{\boldsymbol{m} \in \mathcal{M}' \mid \alpha \in \boldsymbol{m}\}}{1 + \#\{\boldsymbol{m} \in \mathcal{M}' \mid \alpha \in \boldsymbol{m}\}}}{\#\{\alpha \in \mathcal{A}_n \cap (\mathcal{C} \cup \mathcal{A}'_F) \mid \sigma(\alpha) = \sigma\}}, \\ \qquad\qquad \textit{if } \{\alpha \in \mathcal{A}_n \cap (\mathcal{C} \cup \mathcal{A}'_F) \mid \sigma(\alpha) = \sigma\} \neq \emptyset, \\ \varnothing, \qquad\qquad \textit{otherwise}, \end{cases} \tag{5.1}$$

where $\sigma(\alpha)$ is an external function that returns the signature element σ_α (concept name, role name or attribute name) of an assertion α. The codomain of acc is restricted to the interval $]0,1[$ such that $0 < acc(\sigma,n) < 1$. Accuracy values outside of this interval, i.e., for $acc(\sigma,n) = 0$ and $acc(\sigma,n) = 1$, the accuracy is set to the fixed value 0.001 and 0.999 respectively.

Roughly speaking, we determine the accuracy value for a signature element σ with respect to a specific data source $\mathcal{K}_n$, i.e., $\mathcal{A}_n$, with '1− the ratio of incorrect assertions on σ in $\mathcal{A}_n$ with respect to the total set of assertions on σ (conflicting assertions and correct assertions) in $\mathcal{A}_n$. The number of incorrect assertions is given by the set of likely false assertions, i.e., assertions that are part of the repair generated by our majority voting-based approach proposed in Section 5.3, and the likelihood of being false for each still conflicting assertions. This likelihood is in turn calculated based on the number of unresolved MISAs in which an assertion α is still involved.

Example 5.3 (Signature Accuracy)**.** *Let us consider, for example, the signature element $Paper$ in data source $\mathcal{K}_1$. Of the four assertions α_1, α_3, α_7 and α_8, the two assertions α_1 and α_3 are involved in conflicts, where α_1 is classified as likely true assertion and α_3 as likely false assertion. On the contrary, α_7 and α_8 are non-conflicting assertions whereas none of them can be considered to be correct. While it is obvious for α_8, α_7 cannot be treated as correct, since even if the individual* **I6** *described by α_7 is also part of the non-conflicting assertion β_6,* **I6** *is contained in the conflicting assertion γ_8 as well. As a result, the signature accuracy for $Paper$ with respect to $\mathcal{K}_1$ is calculated according to Equation 5.1 with*

$$acc(Paper, 1) = 1 - \frac{1+0}{2} = 0.5\,.$$

The complete list of signature accuracies for our running example is as follows:

$$\begin{array}{rrr} acc(Paper,1) = 0.5 & acc(Paper,2) = 0.667 & acc(SlideSet,3) = 0.667 \\ acc(publishedIn,1) = 0.5 & acc(Proceedings,2) = 0.667 & acc(Proceedings,3) = 0.667 \\ acc(\mathsf{edition},1) = 0.999 & acc(publishedIn,2) = 0.333 & acc(slideSetOf,3) = 0.444 \\ & acc(\mathsf{edition},2) = 0.001 & \end{array}$$

5.5 Learned Repair

Given the accuracies for each signature element $\sigma \in \bigcup_{n=1,\ldots,N} \Sigma_n$ that is either a concept name, a role name or an attribute, we can now use these values to resolve all remaining MISAs unresolved by the majority voting-based approach proposed in Section 5.3. As the calculation of the signature accuracies is based on the statistical evidence given by the partial repair, we call the subsequent extension as *learned repair*. The resolution of remaining MISAs based on signature accuracies is outlined in the following Algorithm 5.5: GenerateRepairForUnresolvedMISAs.

Algorithm 5.5: GenerateRepairForUnresolvedMISAs($\mathcal{M}, \mathcal{R}, \Sigma_{acc}$)

Input: set $\mathcal{M}$ of MISAs,
(partial) repair $\mathcal{R}$,
complete map Σ_{acc} of signature accuracies
Output: complete repair $\mathcal{R}'$ for $\mathcal{M}$

begin
  $\mathcal{M}' \leftarrow \bigcup_{\boldsymbol{m} \in \mathcal{M} \mid \boldsymbol{m} \cap \mathcal{R} = \emptyset} \{\boldsymbol{m}\}$
  $\mathcal{C}' \leftarrow \bigcup_{\boldsymbol{m} \in \mathcal{M}'} \boldsymbol{m}$
  $\mathcal{C}' \leftarrow \mathsf{SortDescending}(\mathcal{C}', \Sigma_{acc})$
  $\mathcal{R}' \leftarrow \mathcal{R}$
  foreach $\alpha \in \mathcal{C}'$ **do**
    $\mathcal{M}'_\alpha \leftarrow \{\boldsymbol{m}' \in \mathcal{M}' \mid \alpha \in \boldsymbol{m}'\}$
    if $\mathcal{M}'_\alpha \neq \emptyset$ **then**
      foreach $\boldsymbol{m}'_\alpha \in \mathcal{M}'_\alpha$ **do**
        $\alpha_\mathcal{C} \leftarrow \boldsymbol{m}'_\alpha \setminus \alpha$
        if $\sigma_{acc}(\alpha, \Sigma_{acc}) > \sigma_{acc}(\alpha_\mathcal{C}, \Sigma_{acc})$ **then**
          $\mathcal{R}' \leftarrow \mathcal{R}' \cup \alpha_\mathcal{C}$
          $\mathcal{M}'_{\alpha_\mathcal{C}} \leftarrow \{\boldsymbol{m}' \in \mathcal{M}' \mid \alpha_\mathcal{C} \in \boldsymbol{m}'\}$
          $\mathcal{M}' \leftarrow \mathcal{M}' \setminus \mathcal{M}'_{\alpha_\mathcal{C}}$
  $\mathcal{C}' \leftarrow \bigcup_{\boldsymbol{m} \in \mathcal{M}'} \boldsymbol{m}$
  foreach $\alpha \in \mathcal{C}'$ **do**
    if $\{\boldsymbol{m}' \in \mathcal{M}' \mid \alpha \in \boldsymbol{m}'\} \neq \emptyset$ **then**
      $\mathcal{R}_{random} \leftarrow \emptyset$
      if Random($true, false$) **then**
        $\mathcal{R}_{random} \leftarrow \{\alpha\}$
      else
        $\mathcal{R}_{random} \leftarrow \bigcup_{\boldsymbol{m}' \in \mathcal{M}' \mid \alpha \in \boldsymbol{m}'} \boldsymbol{m}' \setminus \alpha$
      $\mathcal{R}' \leftarrow \mathcal{R}' \cup \mathcal{R}_{random}$
      foreach $\alpha_\mathcal{R} \in \mathcal{R}_{random}$ **do**
        $\mathcal{M}'_{\alpha_\mathcal{R}} \leftarrow \{\boldsymbol{m}' \in \mathcal{M}' \mid \alpha_\mathcal{R} \in \boldsymbol{m}'\}$
        $\mathcal{M}' \leftarrow \mathcal{M}' \setminus \mathcal{M}'_{\alpha_\mathcal{R}}$
  $\mathcal{R}' \leftarrow \mathsf{SortDescending}(\mathcal{R}', \Sigma_{acc})$
  $\mathcal{R}' \leftarrow \mathsf{MinimizeRepair}(\mathcal{M}, \mathcal{R}')$
  return $\mathcal{R}'$
end

Given a set $\mathcal{M}$ of MISAs, a corresponding (partial) repair $\mathcal{R}$ generated by the majority voting-based approach proposed in Section 5.3 and a map Σ_{acc} comprising for each signature element $\sigma \in \Sigma_n$ of every data source $\mathcal{K}_n$ integrated in the federated KB knowledge base $\mathcal{K}_F$ the associated signature accuracy according to Definition 5.3. The algorithm starts by determining the set $\mathcal{M}'$ of MISAs not re-

solved by the repair $\mathcal{R}$. Based on that, the set $\mathcal{C}'$ of still conflicting assertions is identified and sorted by signature accuracy in descending order. Hence, by iterating over the ordered set $\mathcal{C}'$, the algorithm starts with those assertions for which the corresponding signature accuracy has the highest value. Given an assertion $\alpha \in \mathcal{C}'$, the algorithm determines the set $\mathcal{M}'_\alpha$ of MISAs in which α is involved. By using the external function $\sigma_{acc}(\alpha, \Sigma_{acc})$ that returns the signature accuracy of the signature element σ_α (concept name, role name or attribute name) of assertion α with respect to the data source $\mathcal{K}_n$, i.e., $\mathcal{A}_n$ in which α is stated, the algorithm checks for each MISA $\boldsymbol{m}'_\alpha \in \mathcal{M}'_\alpha$ if the associated signature accuracy of α is greater than the signature accuracy belonging to assertion $\alpha_\mathcal{C} \in \boldsymbol{m}'_\alpha$ that is contradicting α. If so, the contradicting assertion $\alpha_\mathcal{C}$ is added to the extended repair $\mathcal{R}'$ and hence, the set $\mathcal{M}'_{\alpha_\mathcal{C}}$ of all MISAs comprising $\alpha_\mathcal{C}$ is removed from the set $\mathcal{M}'$ of unresolved MISAs.

However, after the resolution of MISAs based on signature accuracies, there may still exist some unresolved MISAs due to equivalent signature accuracy values belonging to the conflicting assertions of a MISA. Because of that, the algorithm randomly decides for each still conflicting assertion $\alpha \in \mathcal{C}'$ whether α or all its contradicting assertions are added to the repair $\mathcal{R}'$.

Even if all MISAs will now be resolved, we are obviously not able to ensure that the resulting repair $\mathcal{R}'$ is minimal according to Definition 5.1. Because of that, Algorithm 5.2: MinimizeRepair is applied to the repair $\mathcal{R}'$ before Algorithm 5.5 terminates. Moreover, as the assertions of the repair are sorted in advance by signature accuracy in descending order, we make sure that assertions having a higher value in the associated signature accuracy are removed from the repair first.

Proposition 5.3. *Given a set $\mathcal{M}$ of MISAs, a corresponding (partial) repair $\mathcal{R}$ generated by the majority voting-based approach proposed in Section 5.3 and a map Σ_{acc} comprising for each signature element of every integrated data source the associated signature accuracy according to Definition 5.3. The repair $\mathcal{R}'$ generated by Algorithm 5.5:* GenerateRepairForUnresolvedMISAs *is complete, i.e., all MISAs are resolved such that $Mod(\mathcal{K} \setminus \mathcal{R}') \neq \emptyset$ holds.*

Proof. Obviously, the algorithm terminates only after all MISAs are resolved (at least in the part of random decisions from Line 16 to 26). Moreover, as the subsequent application of Algorithm 5.2: MinimizeRepair does not affect the set of resolved MISAs (see Proposition 5.1), we can conclude that $Mod(\mathcal{K} \setminus \mathcal{R}') \neq \emptyset$. □

Proposition 5.4. *Given a set $\mathcal{M}$ of MISAs, a corresponding (partial) repair $\mathcal{R}$ generated by the majority voting-based approach proposed in Section 5.3 and a map Σ_{acc} comprising for each signature element of every integrated data source the associated signature accuracy according to Definition 5.3. Then, the repair $\mathcal{R}'$ generated by Algorithm 5.5:* GenerateRepairForUnresolvedMISAs *is always minimal, i.e., there exists no proper subset $\mathcal{R}'' \subset \mathcal{R}'$ for which $Mod(\mathcal{K} \setminus \mathcal{R}'') \neq \emptyset$ holds.*

Proof. The proof follows directly from Proposition 5.1. □

Proposition 5.5. *Given a set $\mathcal{M}$ of MISAs, a corresponding (partial) repair $\mathcal{R}$ generated by the majority voting-based approach proposed in Section 5.3 and a map Σ_{acc} comprising for each signature element of every integrated data source the associated signature accuracy according to Definition 5.3. Then, Algorithm 5.5:* GenerateRepairForUnresolvedMISAs *generates a repair $\mathcal{R}'$ that considers for each assertion $\alpha \in \mathcal{R}'$ every entailment relation $\alpha' \models_{\mathcal{T}} \alpha$, i.e., $\alpha \in \mathcal{R}' \rightarrow \alpha' \in \mathcal{R}'$ if $\alpha' \models_{\mathcal{T}} \alpha$.*

Proof. Given an entailment relation $\alpha' \models_{\mathcal{T}} \alpha$, we already know that $\mathcal{C}_\alpha \subseteq \mathcal{C}_{\alpha'}$, where $\mathcal{C}_\alpha = \{\alpha_\mathcal{C} \mid \{\alpha, \alpha_\mathcal{C}\} \in \mathcal{M}\}$ and $\mathcal{C}_{\alpha'} = \{\alpha'_\mathcal{C} \mid \{\alpha', \alpha'_\mathcal{C}\} \in \mathcal{M}\}$ denote the sets of conflicting assertions of α resp. of α'. Moreover, given that $\alpha \in \mathcal{R}'$ we can conclude from Proposition 5.4 that at least one assertion $\alpha_\mathcal{C} \in \mathcal{C}_\alpha$ is not part of the repair, since otherwise $\mathcal{R}'$ would not be minimal. The entailment relation $\alpha' \models_{\mathcal{T}} \alpha$ is not satisfied only if $\alpha \in \mathcal{R}'$ but $\alpha' \notin \mathcal{R}'$. However, as $\alpha_\mathcal{C}$ is also a conflicting assertion of α', i.e., $\alpha_\mathcal{C} \in \mathcal{C}_\alpha \subseteq \mathcal{C}_{\alpha'}$, from Proposition 5.3 we can conclude that $\alpha' \in \mathcal{R}'$, since otherwise the MISA $\boldsymbol{m} = \{\alpha', \alpha_\mathcal{C}\}$ would not be resolved and hence the repair $\mathcal{R}'$ would not be complete. □

Hence, we can generally conclude that every entailment relation is considered by a repair that is complete and minimal.

The determination of all signature accuracies and the generation of a learned repair via Algorithm 5.5, can be computed in polynomial time with respect to the size of the ABox. Hence, the overall complexity of our federated debugging approach, starting from the generation of clash queries up to the generation of a complete repair stays in polynomial time with respect to the knowledge base size.

Example 5.4 (Learned Repair)**.** *By reference to the majority voting-based repair $\mathcal{R}$ of Example 5.2 we can observe that the MISAs $\{\beta_8, \gamma_7\}$, $\{\beta_8, \gamma_8\}$ and $\{\gamma_7, \gamma_8\}$ of our running example are not resolved. Given now the signature accuracies from Example 5.3, Algorithm 5.5 can be applied to this remaining set of MISAs. Figure 5.3 shows the remaining conflict graph and the corresponding signature accuracy of each assertion. Due to the fact that the signature accuracy of γ_8 is lower than the corresponding accuracy values of its contradicting assertions, the assertion γ_8 is added to the learned repair $\mathcal{R}'$. The still remaining MISA $\{\beta_8, \gamma_7\}$ that could not be resolved based on signature accuracies is therefore subsequently resolved randomly in Algorithm 5.5.*

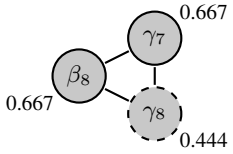

Figure 5.3: Learned Repair

5.6 Related Work

The problem of resolving inconsistency in DL KBs has already been addressed by several works. A comprehensive survey and analysis of corresponding approaches is given by Haase and Qi [HQ07].

Moreover, an approach that is applicable to a scenario similar to our setting is poposed by Bonatti et al. [Bon+11]. Based on annotated logic programs indicators of provenance and trust are tracked during the reasoning process in order to detect and repair inconsistency. However, Bonatti et al. pursue distributed implementation strategies over a cluster of commodity hardware while our approach is applied to a federated setting of loosely coupled KBs. Another important difference to our approach is the origin of the trust values (in our approach called signature accuracies). While the authors of [Bon+11] apply a well-known page rank algorithm that does not at all consider logical dependencies, in our approach the signature accuracies are calculated based on the statistical evidence gathered by applying a majority voting to explicit and implicit conflicts.

Another strongly related work has been published by Chortis and Flouris [CF15] and in which a framework for detecting and repairing inconsistency in *DL-Lite*$_{\mathcal{A}}$ KBs is proposed. Based on a similar idea the logical conflicts are identified by querying and a repair is generated based on the computation of a vertex cover. However, this approach does not consider any entailment relations between conflicting assertion and is not designed for a federated setting. Moreover, by generating the vertex cover any conflict that cannot be resolved based on the vertex cardinalities is decided randomly instead of first trying to resolve other conflicts or taking any statistical evidence into account.

As opposed to the repair of inconsistency Lembo et al. [Lem+11] propose different variants of inconsistency-tolerant semantics for (U)CQ answering over *DL-Lite* KBs. Based on this work, Masotti et al. [MRR11] generate a corresponding repair for an inconsistent *DL-Lite*$_{\mathcal{A}}$ KB that comprises every ABox assertion that is involved in any conflict. However, even related to our work, the resulting repair is not minimal according to our definition and the proposed approach is only evaluated with very small datasets.

Lambrix and Ivanova [LI13] presented a debugging approach for a network of TBoxes comprising the detection and repair of missing and wrong is-a statements defined by the TBoxes and their mappings. Even this approach deals with a network of TBoxes and hence is related to our work, its focus is rather to detect and repair incoherence instead of inconsistency.

In the context of data integration the subject of data cleansing has also become a significant problem and is strongly related to KB debugging. Data cleansing typically comprises the detection and fixing of imprecise or corrupt data by commonly applying data mining techniques such as in [NLK09; Kha+15]. However, the focus of this topic is more on syntactical errors (such as wrong data types or varying data values) and data anomalies which are not concerned by our work.

Even also addressing a different setting, the problem of KB evolution is very

similar to KB debugging as well. Given a consistent KB the objective is the incorporation new knowledge during evolution by preserving coherency and consistency. Some works in this context under the consideration of the *DL-Lite* family are for example [Cal+10; Qi+15].

As we can see, depending on the specifics of the setting there exist several approaches in order to deal with inconsistent KBs. However, to the best of our knowledge, none of them considers a federated setting of loosely coupled KBs such as LOD sources or addresses the consideration of entailment relations on generating a minimal repair.

5.7 Summary

By relying on the approach of the previous Chapter 4 for an efficient inconsistency detection in federated *DL-Lite*$_\mathcal{A}$ KBs and by proposing an approach for resolving inconsistency in federated *DL-Lite*$_\mathcal{A}$ KBs in this chapter we have mainly addressed research question Q2**:** How can the process of debugging federated knowledge bases be designed in a convenient, efficient, and eligible way? Based on the definition of source-related ABox assertions and the generation of MISAs in the previous chapter we have introduced the definition of entailment relations between assertions and described the representation of a conflict graph modeling the network of conflicting assertions and their relationships. Subsequently, we have described the first phase of the repair generation that is based on the application of a majority voting scheme. In principle, the algorithm is based on a heuristics that selects MISAs (represented by undirected edges in the conflict graph) with minimal cardinality and removes from those MISAs the assertion (represented by vertices) that is involved in more conflicts. Obviously, this approach does not aim at generating a complete repair, but applies an efficient heuristic where each step in the algorithm corresponds to an eligible decision and ensures the consideration of all entailment relations. Moreover, especially the definition of source-related ABox assertions facilitates the exploitation of explicit but also implicit redundancies caused by federating different KBs in order to verify or disprove assertions that are involved in logical conflicts. As a result we are able to show that the debugging process does indeed benefit from the characteristics of a federated KB. Not only the number of identified conflicts can be increased but also the repair is improved with respect to validity and completeness (amount of MISAs that could be resolved). In the second phase of our repair generation, we determine the degree of accuracy for signature elements with respect to a specific data source by analyzing the partial repair of the first phase. With the help of this approach we are able to extend the repair generated in the first phase in order to provide a complete and minimal repair resolving the inconsistency of a federated *DL-Lite*$_\mathcal{A}$ KB.

By introducing signature accuracies which can also be interpreted as a certain kind of trust values, we gave also a first hint to research question Q3 that is asking for the feasibility of assessing the trustworthiness of individual assertions with re-

spect to certain data sources based on the debugging results and which is addressed in the next chapter.

Chapter 6

Fine-grained Trust Assessment

Following the desired property of minimality for repairs by which the logical conflicts are resolved with a minimal impact on the KB, i.e., its ABox, the resulting KB is a maximal sub-KB that is consistent. However, applying a repair, i.e., resolving conflicts by removing (or ignoring) a subset of assertions may result in loss of information. Moreover, especially in context of data or information integration a growing number and thus an increasing variety of data sources integrated in a federated KB likely lead to cases where "the actual truth is a matter of perspective" [Gol+17]. Hence, it would be detrimental to remove some assertions of a data source as there might exists another federated KB integrating the same data source but for which exactly those assertions are actually correct. Furthermore, conditioned by the LOD cloud (or more generally the Semantic Web) where each source is managed and maintained autonomously without any influence on its further development, there is no realistic option to apply any changes, why each data source needs to be considered as fixed.

To come up these obstacles, we tackle the challenge of determining the trustworthiness of the information provided by the federated KB in order to propose an alternative approach for handling inconsistency in federated KBs. In this regard we are relying on the definition of *trust* as "the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context" by Grandison and Sloman [GS00]. In our work we are concerned with distributed data sources (entities) and their competences to provide reliably information with respect to specific individuals. From this point of view our notion of trust additionally bears reference to context dependency, by considering the federated KB integrating various data sources as the specified context for which the trust values hold. Hence, given the context of a federated KB, the measure of trust essentially indicates the probability of an assertion (or a set of assertions) to be true. [1]

We start this chapter by giving a brief introduction to Markov networks in Sec-

[1] In the relevant literature (such as in [YHY08]), the term *trust* is rather used on the level of data sources, whereas the term *probability* is used more frequently in relation to a particular assertion. However, note that we use both terms interchangeably within the scope of this thesis.

tion 6.1 and to Gibbs sampling, a Markov chain Monte Carlo algorithm that can be used to approximate probabilistic inference in Markov networks, in Section 6.2. Subsequently, we propose in Section 6.3 an automated approach for fine-grained trust assessment in federated KBs at different levels of granularity. By considering a conflict graph as a Markov network graph that models all probabilistic dependencies between the conflicting assertions and applying an appropriate Gibbs sampling we start in Section 6.3.1 with the assessment of trust values at the level of assertions. Based on that, we continue with the assessment of signature trusts in Section 6.3.2 and data source trusts in Section 6.3.3. Finally we give an overview of related works in Section 6.4 and summarize this chapter in Section 6.5.

While we published in [Nol+17] the initial version of our approach for an automated fine-grained trust assessment in federated KBs, this chapter comprises an advancement of that approach ensuring the consideration of entailment relations between conflicting assertions, which has been neglected so far.

6.1 Markov Networks

As a result of assigning probabilities to the assertions, each assertion can be considered as a random variable. Due to the deterministic dependencies given by the KB we use only *Boolean random variables* (also called *propositional variables*) with values $\{0, 1\}$ representing the *state* of an assertion, which is *false* ($x = 0$) if the assertion is part of the repair, or *true* ($x = 1$) otherwise. The *(marginal) probability* of a random variable x with respect to a specific value x is denoted by $p(x = \mathrm{x})$ and is simply a real number between 0 and 1 describing a degree of belief (or trust) in $x = \mathrm{x}$ such that the probabilities of both values $\{0, 1\}$ for x sum up to 1, i.e., $p(x = 0) + p(x = 1) = 1$.

Let $X = \{x_1, x_2, ..., x_n\}$ be an ordered set of (Boolean) random variables. An assignment of truth values to all variables $x_i \in X$ is represented by a vector $\mathbf{x}$ and is referred to as a *possible world*. The set of all possible worlds is denoted by $\mathcal{X}$. Similarly as for random variables, every possible world $\mathbf{x}$ can be associated with a corresponding *joint probability* $p(X = \mathbf{x})$ such that the probabilities of all possible worlds (i.e., a set of exhaustive and mutually exclusive combinations of truth value assignments to X) sum up to 1, i.e., $\sum_{\mathbf{x} \in \mathcal{X}} p(X = \mathbf{x}) = 1$. As there may exist some truth value assignments $\mathbf{x}$ to X that are contradictory (i.e., inconsistent) with respect to the deterministic dependencies given by the KB and hence result in $p(X = \mathbf{x}) = 0$, we consider the set $\mathcal{X}$ of possible worlds as a set that only comprises truth value assignments for which $p(X = \mathbf{x}) > 0$ holds. Conversely, we call an assignment $\mathbf{x}$ of truth values to X with $p(X = \mathbf{x}) = 0$ an *impossible world*. [RD06; Dom+08; GM10; Kli11]

By considering a specific truth value assignment $\mathbf{x}_E$ to an ordered set X_E of random variables as evidence, based on the *prior* probabilities we can now determine the *posterior* probability of $x = \mathrm{x}$ given $X_E = \mathbf{x}_E$. This so called *conditional probability* of $x = \mathrm{x}$ is denoted by $p(x = \mathrm{x} \mid X_E = \mathbf{x}_E)$ and can be defined

with

$$p(x = \mathrm{x} \mid X_E = \mathbf{x}_E) = \frac{p(x = \mathrm{x}, X_E = \mathbf{x}_E)}{p(X_E = \mathbf{x}_E)}\,. \tag{6.1}$$

Obviously, $p(x = \mathrm{x} \mid X_E = \mathbf{x}_E)$ is defined only if $p(X_E = \mathbf{x}_E) \neq 0$.

However, given a set $X = \{x_1, x_2, ..., x_n\}$ of (Boolean) random variables, in principle there exist 2^n possible truth value assignments to X. In order to compactly represent all the assignments of truth values and to describe the complex structure of conditional dependencies between the random variables the formalism of graphical models can be exploited. Probabilistic graphical models enable to efficiently handle and reason on uncertain information, i.e., probabilistic knowledge, by flexibly simulating possible worlds (i.e., assignments of truth values) in proportion to its likelihoods. Besides Bayesian networks, one of the most significant graphical models is the graph-based formalism of *Markov networks* (also called *Markov random fields*) [KF09]. While in a Bayesian network the dependencies of random variables are described by a directed acyclic graph, the modeling with Markov networks is based on undirected graphs that can be cyclic. As the conflict relation described by a MISA is undirected and possibly result in cyclic parts of the conflict graph, we will use in the following the formalism of Markov networks as probabilistic graphical model. Moreover, even an entailment relation is represented in a conflict graph by a directed edge (see Section 5.2), the relation has to be considered as bidirectional why the resulting graph is still undirected. The bidirectionality follows from the fact that, given an entailment relation $\alpha' \models_{\mathcal{T}} \alpha$, if α is $false$ then we can conclude that α' has to be $false$ as well. On the other hand, if α' is $true$ it directly follows that α has also to be $true$.

The core of a Markov network is a so called *Markov network graph*, which is an undirected graph $G_{\mathcal{H}} = (X, E)$, where its vertices correspond to the random variables $X = \{x_1, x_2, ..., x_n\}$ and its edges E represent direct probabilistic dependencies (or interactions) between two connected vertices. A subset X_c of X in $G_{\mathcal{H}}$ is called a *clique*, if all vertices in X_c are fully connected, i.e., each pair of vertices in X_c is connected by an edge. Formally, a Markov network can be defined as follows:

Definition 6.1 (Markov Network)**.** *Let $G_{\mathcal{H}} = (X, E)$ be a Markov network graph over a set $X = \{x_1, x_2, ..., x_n\}$ of random variables. A Markov network can be expressed in terms of a* log-linear model *which is composed of the Markov network graph $G_{\mathcal{H}}$ and a set of feature functions $F = \{f_1(X_1), f_2(X_2), ..., f_m(X_m)\}$. Each feature function (in short feature) $f_c(X_c) \in F$ defines a (non-negative) numerical value for each possible assignment $\mathbf{x}_c$ of truth values to some clique X_c. The joint probability distribution is then defined as*

$$p(X = \mathbf{x}) = \frac{1}{Z} \exp\Bigg(\sum_{f_c(X_c) \in F} w_c f_c(X_c = \mathbf{x}_c) \Bigg)\,, \tag{6.2}$$

where Z is a normalization constant, called partition function, which is ensuring that $\sum_{\mathbf{x}\in\mathcal{X}} p(X = \mathbf{x}) = 1$, and the real-valued weights $W = \{w_c : f_c(X_c) \in F\}$ are the parameters of the log-linear model.

Note that, as a feature defines a numerical value for each possible assignment $\mathbf{x}_c$ to a set X_c of random variables and we are using only Boolean random variables with values $\{0, 1\}$, we focus within the scope of this thesis on binary features such that $f(X_c = \mathbf{x}_c) \in \{0, 1\}$.

The *Markov blanket* B_x of a vertex (random variable) x is defined as the minimal set of vertices rendering x independent from the remaining graph, which is in a Markov network graph simply the set of all neighboring vertices of x. Hence, the Markov blanket B_x comprises each vertex for which in $G_{\mathcal{H}}$ there exists a direct probabilistic dependency to x and consequently represents the union of all cliques in which x is contained in. Based on a given assignment to each vertex $b_x \in B_x$, the conditional probability of a value assignment x to vertex x can be determined according to the following Definition 6.2.

Definition 6.2 (Conditional Probability)**.** *Given the assignment $\mathbf{b}_x$ of truth values to the Markov blanket B_x of a vertex x in a Markov network graph $G_{\mathcal{H}}$, the conditional probability of $x = \mathrm{x}$ is given with*

$$p(x = \mathrm{x} \mid B_x = \mathbf{b}_x) \tag{6.3}$$
$$= \frac{\exp\Big(\sum\limits_{f_x \in F_x} w_x f_x(x = \mathrm{x}, B_x = \mathbf{b}_x)\Big)}{\exp\Big(\sum\limits_{f_x \in F_x} w_x f_x(x = 0, B_x = \mathbf{b}_x)\Big) + \exp\Big(\sum\limits_{f_x \in F_x} w_x f_x(x = 1, B_x = \mathbf{b}_x)\Big)},$$

where $F_x \subseteq F$ denotes the subset of features in which x appears and $w_x \in W$ is the associated real-valued weight. According to Definition 6.1, F represents the set of features in the corresponding Markov network and $W = \{w_c : f_c(X_c) \in F\}$ are the parameters of the log-linear model.

6.2 Approximate Probabilistic Inference

Given a Markov network modeling all probabilistic dependencies between the vertices, we are now interested in the marginal posterior probability of each vertex (Boolean random variable) for being *true* ($x = 1$), taking into consideration the joint probability of each possible world. However, the complexity of computing these probabilities, which is also called *marginal inference*, is time exponential in the number of vertices. To approximate marginal inference in Markov networks, one of the most commonly used methods is *Markov chain Monte Carlo* (MCMC) and particularly *Gibbs sampling* [Kol+07; KF09].

Generally, MCMC algorithms walk through the space $\mathcal{X}$ of possible worlds by *sampling* each vertex $x \in X$ in random order according to a defined random *transition model*. Given the current state (assignment of truth values) $\mathbf{x} \in \mathcal{X}$

of X, by sampling vertex x a random value in $[0, 1]$ is generated and the state x of vertex x is changed (*flipped*) to $\neg\mathrm{x}$ in the subsequent state $\mathbf{x}'$ of X if the random value is less than or equal to the *transition probability* $T_x(\mathbf{x} \to \mathbf{x}')$ defined by the transition model, where $\mathbf{x}' = \{\neg\mathrm{x}\} \cup (\mathbf{x}_t \setminus \{\mathrm{x}\})$. Until all vertices are sampled, we obtain again a possible world (assignment of truth values) $\mathbf{x}_{t+1} \in \mathcal{X}$ that can be considered as a sample of the full joint probability distribution, and hence can be treated as a '*transition*' (or '*jump*') between different possible worlds, i.e., from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$. By starting with an initial possible world $\mathbf{x}_0$ and repeating the sampling process of all vertices K times we obtain a sequence $\mathbf{x}_0, \ldots, \mathbf{x}_K$ of possible worlds, called *Markov chain*, where each possible world $\mathbf{x}_t$ only depends on the preceding possible world $\mathbf{x}_{t-1}$.

Gibbs sampling [GM10] is a special type of MCMC methods where the transition probability for a vertex x corresponds to its conditional probability given by Equation (6.3), i.e., $T_x(\mathbf{x} \to \mathbf{x}') = p(x = \neg\mathrm{x} \mid B_x = \mathbf{b}_x)$. Hence, the probability of transition does not depend on the current assigned truth value x of x but only on the state $\mathbf{b}_x$ of the corresponding Markov blanket B_x. As a flipped state of a vertex is considered in all subsequent vertex samplings it is ensured by Equation (6.3) that a sampling process of all vertices always ends up in a possible world. Hence, MCMC methods respectively Gibbs sampling will never step into an impossible world and thus result in a significantly faster probabilistic inference.

Intuitively, after a sufficiently large number K of samplings the process converges to the desired posterior probability distribution and the marginal posterior probability of each vertex for being $true$ can be approximated by simply determine the fraction of the number of states in the Markov chain in which $x = 1$. Moreover, in order to ensure that the generated Markov chain converges to a unique stationary distribution that is reached from any initial possible world $\mathbf{x}_0 \in \mathcal{X}$, the Markov chain hast to fulfill the condition of being *regular* [Kol+07; KF09]. A Markov chain is called regular if there exists a number L such that for any pair $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ we can get from $\mathbf{x}$ to $\mathbf{x}'$ in L steps with a probability greater than 0.

6.3 Trust Levels

Having introduced Markov networks and sampling methods to approximate marginal inference we can now define in the following Section 6.3.1 an appropriate Markov network for a conflict graph and a corresponding Gibbs sampling in order to assess trust values for any assertion within the Markov network graph. Based on that, we subsequently tackle the assessment of signature trusts in Section 6.3.2 as well as the assessment of data source trusts in Section 6.3.3.

6.3.1 Assertion Trusts

By considering the vertices of a conflict graph as Boolean random variables, we obtain a corresponding Markov network graph that models all probabilistic depen-

dencies between the random variables. In order to formulate now an appropriate Markov network we have to define the corresponding set of features and the associated set of weights. Formally, given an inconsistent federated *DL-Lite*$_\mathcal{A}$ knowledge base $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$, a corresponding conflict graph $G_\mathcal{C} = (\mathcal{C}, \mathcal{M}, \mathcal{E})$ and a repair $\mathcal{R}$ that is complete and minimal. As we consider each assertion $\alpha_i \in \mathcal{C}$ of $G_\mathcal{C}$ as a Boolean random variable $\alpha_i \in \{0, 1\}$, which is either *false* ($\alpha_i = 0$) if $\alpha_i \in \mathcal{R}$, or *true* ($\alpha_i = 1$) otherwise, we define for each assertion α_i a feature $f_a(\alpha_i) \in F$ that is simply reflecting the state of the assertion and hence is given by

$$f_a(\alpha_i = \mathrm{x}) = \mathrm{x}\,, \tag{6.4}$$

such that $f_a(\alpha_i = 0) = 0$ and $f_a(\alpha_i = 1) = 1$. In order to consider the calculated signature accuracies (Section 5.4) as (marginal) prior probabilities, we define the associated weight $w_a(\alpha_i)$ of a feature $f_a(\alpha_i)$ as the log odds between a world in which the assertion α_i is *true* and a world in which it is *false*, based on the statistical evidence gathered by the majority voting-based approach proposed in Section 5.3. Given the complete map Σ_{acc} comprising for each signature element $\sigma \in \Sigma_n$ of every data source $\mathcal{K}_n$ integrated in the federated KB knowledge base $\mathcal{K}_F$ the associated signature accuracy according to Definition 5.3 and let $\sigma_{acc}(\alpha, \Sigma_{acc})$ be as described in Section 5.5 an external function that returns the signature accuracy of the signature element σ_α (concept name, role name or attribute name) of assertion α with respect to the data source $\mathcal{K}_n$, i.e., $\mathcal{A}_n$ in which α is stated. Then, the weight $w_a \in W$ of feature $f_a(\alpha_i) \in F$ is defined with

$$w_a = \ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right). \tag{6.5}$$

Moreover, in order to consider the conflict relations that are described by the set $\mathcal{M}$ of MISAs, the condition $\neg(\bigwedge_{\alpha \in \boldsymbol{m}} \alpha)$ has to be satisfied for each MISA $\boldsymbol{m} \in \mathcal{M}$. Hence, we define for each assertion α_i a feature $f_c(\alpha_i, B^\mathcal{M}_{\alpha_i}) \in F$ ensuring that a sampling will not result in an assignment of truth values that comprises any logical conflict. Formally, $f_c(\alpha_i, B^\mathcal{M}_{\alpha_i})$ is given with

$$f_c(\alpha_i = \mathrm{x}, B^\mathcal{M}_{\alpha_i} = \mathbf{b}^\mathcal{M}_{\alpha_i}) = \mathrm{x} \wedge \left(\bigvee\nolimits_{\mathrm{b}^\mathcal{M}_{\alpha_i} \in \mathbf{b}^\mathcal{M}_{\alpha_i}} \mathrm{b}^\mathcal{M}_{\alpha_i}\right), \tag{6.6}$$

where $B^\mathcal{M}_{\alpha_i} \subseteq B_{\alpha_i}$ denotes a subset of the Markov blanket B_{α_i} of α_i comprising all conflicting assertions of α_i, such that $\boldsymbol{m} \cap B^\mathcal{M}_{\alpha_i} \neq \emptyset$ holds for every MISA $\boldsymbol{m} \in \mathcal{M} \mid \alpha_i \in \boldsymbol{m}$. As each feature $f_c(\alpha_i, B^\mathcal{M}_{\alpha_i}) \in F$ describes a hard constraint that has to be satisfied by every possible world $\mathbf{x} \in \mathcal{X}$, the corresponding weight $w_c \in W$ of f_c is defined with

$$w_c \to -\infty\,. \tag{6.7}$$

As a consequence, if any vertex $b^\mathcal{M}_{\alpha_i} \in B^\mathcal{M}_{\alpha_i}$ is *true* (i.e., $\mathrm{b}^\mathcal{M}_{\alpha_i} = 1$), the conditional probability (given by Equation (6.3)) of α_i is $\lim_{w_c \to -\infty} p(\alpha_i = 0 \mid B^\mathcal{M}_{\alpha_i} = \mathbf{b}^\mathcal{M}_{\alpha_i}) = 1$ and $\lim_{w_c \to -\infty} p(\alpha_i = 1 \mid B^\mathcal{M}_{\alpha_i} = \mathbf{b}^\mathcal{M}_{\alpha_i}) = 0$, such that a vertex α_i can only be flipped to *true* iff all its conflicting vertices $B^\mathcal{M}_{\alpha_i}$ are *false* and hence

result in no inconsistency. Moreover, given an assignment $\mathbf{x}'$ of truth values representing an impossible world, i.e., $\mathbf{x}' \notin \mathcal{X}$, that violates any constraint f_c, the joint probability distribution (given by Equation (6.2)) is $\lim_{w_c \to -\infty} p(\mathcal{C} = \mathbf{x}') = 0$.

Provided that there exists no entailment relation in a conflict graph $G_{\mathcal{C}} = (\mathcal{C}, \mathcal{M}, \mathcal{E})$, i.e., $\mathcal{E} = \emptyset$, then the set F of features according to Equation (6.4) and Equation (6.6) and the corresponding set W of weights as per Equation (6.5) and Equation (6.7) are sufficient to model an appropriate Markov network for $G_{\mathcal{C}}$.

Proposition 6.1. *Given an inconsistent federated DL-Lite$_{\mathcal{A}}$ knowledge base $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$, a corresponding conflict graph $G_{\mathcal{C}} = (\mathcal{C}, \mathcal{M}, \mathcal{E})$ with $\mathcal{E} = \emptyset$ and a repair $\mathcal{R}$ that is complete and minimal. A corresponding Markov network is composed by $G_{\mathcal{C}}$ considered as Markov network graph, the set F of features according to Equation (6.4) and Equation (6.6) and the corresponding set W of weights as per Equation (6.5) and Equation (6.7). By sampling from the resulting Markov network, we get a Markov chain that is regular and hence converges to the desired posterior probability distribution.*

Proof. By considering each assertion $\alpha_i \in \mathcal{C}$ of $G_{\mathcal{C}}$ as a Boolean random variable $\alpha_i \in \{0, 1\}$, that is either *false* ($\alpha_i = 0$) if $\alpha_i \in \mathcal{R}$, or *true* ($\alpha_i = 1$) otherwise, based on the given repair $\mathcal{R}$ the initial truth value assignment $\mathbf{x}_0$ to $\mathcal{C}$ is determined and, by Definition 5.1, represents a possible world, i.e., $\mathbf{x}_0 \in \mathcal{X}$. Moreover, since in case of $\mathcal{E} = \emptyset$ the conditional probability (given by Equation (6.3)) of a vertex α_i only dependents on the states of its conflicting neighbors in $B^{\mathcal{M}}_{\alpha_i} \subseteq B_{\alpha_i}$, by flipping the state of α_i there only exist the following two cases:

(i) Given that $\forall b^{\mathcal{M}}_{\alpha_i} \in B^{\mathcal{M}}_{\alpha_i} \mid b^{\mathcal{M}}_{\alpha_i} = 0$, i.e., all contradicting neighbors of α_i are *false* which is denoted by $B^{\mathcal{M}}_{\alpha_i} = 0$ for short, then the conditional probability that α_i is *true* in the next possible world is given with

$$\begin{aligned} p(\alpha_i = 1 \mid B^{\mathcal{M}}_{\alpha_i} = 0) &= \frac{\exp\Big(\ln\Big(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\Big)\Big)}{\exp(0) + \exp\Big(\ln\Big(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\Big)\Big)} \\ &= \sigma_{acc}(\alpha, \Sigma_{acc})\,, \end{aligned} \tag{6.8}$$

and conversely the probability that α_i is *false* in the next possible world is

$$\begin{aligned} p(\alpha_i = 0 \mid B^{\mathcal{M}}_{\alpha_i} = 0) &= \frac{\exp(0)}{\exp(0) + \exp\Big(\ln\Big(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\Big)\Big)} \\ &= 1 - \sigma_{acc}(\alpha, \Sigma_{acc})\,. \end{aligned} \tag{6.9}$$

(ii) Given that $\exists b^{\mathcal{M}}_{\alpha_i} \in B^{\mathcal{M}}_{\alpha_i} \mid b^{\mathcal{M}}_{\alpha_i} = 1$, i.e., at least one conflicting neighbor of α_i is *true* which is denoted by $B^{\mathcal{M}}_{\alpha_i} = 1$ for short, then the conditional

probability that α_i is $true$ in the next possible world is defined by

$$p(\alpha_i = 1 \mid B^{\mathcal{M}}_{\alpha_i} = 1) = \lim_{w_c \to -\infty} \frac{\exp\left(\ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right) + w_c\right)}{\exp(0) + \exp\left(\ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right) + w_c\right)}$$
$$= 0\,, \tag{6.10}$$

and hence the probability that α_i is $false$ in the next possible world is given by

$$p(\alpha_i = 0 \mid B^{\mathcal{M}}_{\alpha_i} = 1) = \lim_{w_c \to -\infty} \frac{\exp(0)}{\exp(0) + \exp\left(\ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right) + w_c\right)}$$
$$= 1\,. \tag{6.11}$$

Given that at least one conflicting neighbor of α_i is $true$, we can conclude that the current state of α_i has to be $false$, as we provided a possible world $\mathbf{x}_0$ (or more generally $\mathbf{x}_t$), and hence remains unchanged as shown in Case (ii). Consequently, due to $f_c(\alpha_i, B^{\mathcal{M}}_{\alpha_i})$ as defined in Equation (6.6) it is ensured that any state flip and hence any transition (comprising a random ordered sampling of all vertices) will always end up with a possible world (consistent state) $\mathbf{x}_{t+1} \in \mathcal{X}$. Moreover, given that the flip of a state will not end up in an impossible world (Case (i)), in Equation (6.3) the conflict feature $f_c(\alpha_i, B^{\mathcal{M}}_{\alpha_i})$ (Equation (6.6)) of α_i is $false$, i.e., $f_c(\alpha_i, B^{\mathcal{M}}_{\alpha_i}) = 0$, and only the feature $f_a(\alpha_i = \mathrm{x})$ (Equation (6.4)) remains. As a result, the conditional (prior) probability of an assertion α_i, i.e., for $\alpha_i = 1$, corresponds exactly to the signature accuracy (Definition 5.3) that is assigned to the signature element σ_{α_i} (concept name, role name or attribute name) of α_i with respect to the corresponding data source $\mathcal{K}_n$. As by definition the signature accuracy value is restricted to the interval $]0, 1[$, both states $\alpha_i = 0$ and $\alpha_i = 1$ of any assertion $\alpha_i \in \mathcal{C}$ have a positive probability given an assignment of truth values to all the remaining variables in $\mathcal{C} \setminus \{\alpha_i\}$. Thus, we can get from any possible world $\mathbf{x} \in \mathcal{X}$ to any possible world $\mathbf{x}' \in \mathcal{X}$ in at most $L = |\mathcal{C}|$ steps of state flips with a probability that is greater than 0, which proves that the resulting Markov chain is regular. □

Before the state of an assertion can be flipped to $true$, all its contradicting assertions in $B^{\mathcal{M}}_{\alpha_i} \subseteq B_{\alpha_i}$ hast to be $false$. Obviously, this implies an intermediate state representing a possible world that is not maximal; or, in other words, the repair implicitly given by all assertions that are $false$ is not minimal. However, this is absolutely legitimate and even desired in particular with regard to the computation of adequate probabilities for assertions.

So far, we have disregarded the entailment relations in $G_{\mathcal{C}} = (\mathcal{C}, \mathcal{M}, \mathcal{E})$ as we assumed that $\mathcal{E} = \emptyset$. However, given that $\mathcal{E} \neq \emptyset$, a sampling transition may leads to inconsistency in the given knowledge base $\mathcal{K}_F$, i.e., an impossible world.

Example 6.1 (Sampling of an Impossible World)**.** *Suppose that we have a KB* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ *with* $\mathcal{T} = \{A \sqsubseteq B \sqsubseteq \neg C\}$ *and* $\mathcal{A} = \{A(\mathbf{I}), B(\mathbf{I}), C(\mathbf{I})\}$*. Obviously, there exists the entailment relation* $A(\mathbf{I}) \models_{\mathcal{T}} B(\mathbf{I})$ *and both assertions* $A(\mathbf{I})$ *and* $B(\mathbf{I})$ *are contradicting* $C(\mathbf{I})$*. If we now assume that the state of the assertion* $C(\mathbf{I})$ *is* $false$*, then the state of the assertion* $C(\mathbf{I})$ *remains unchanged due to the conflict feature given by Equation (6.6), but the assertion* $A(\mathbf{I})$ *as well as the assertion* $B(\mathbf{I})$ *are sampled independently according to its corresponding signature accuracies. However, in consequence of the independent sampling the assertion* $A(\mathbf{I})$ *may remain in the state* $true$ *whereas the state of assertion* $B(\mathbf{I})$ *may be flipped to* $false$*, and hence by contradicting* $A(\mathbf{I}) \models_{\mathcal{T}} B(\mathbf{I})$ *results in the impossible world as depicted in the following Figure 6.1.*

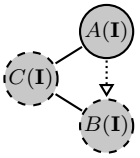

Figure 6.1: Sampling of an Impossible World

In order to ensure that entailment relations are considered by sampling transitions, we will now define an additional feature such that the condition $\neg\alpha' \vee \alpha$ is satisfied for each entailment relation $\boldsymbol{e} = (\alpha', \alpha) \in \mathcal{E}$. Therefore, we now consider the complete Markov blanket B_{α_i} of an assertion α_i, which is given by $B_{\alpha_i} = B^{\mathcal{M}}_{\alpha_i} \cup B^{\mathcal{E}}_{\alpha_i}$, where $B^{\mathcal{E}}_{\alpha_i}$ denotes the set of all assertions to which α_i is connected by an entailment relation. More precisely, we subdivide the set $B^{\mathcal{E}}_{\alpha_i}$ into $B^{\mathcal{E}}_{\alpha_i} = B^{\mathcal{E}_{\models}}_{\alpha_i} \cup B^{\mathcal{E}_{\models}}_{\alpha_i}$ by differentiating between the set $B^{\mathcal{E}_{\models}}_{\alpha_i}$ that is composed of all assertions $b^{\mathcal{E}_{\models}}_{\alpha_i}$ for which $b^{\mathcal{E}_{\models}}_{\alpha_i} \models_{\mathcal{T}} \alpha_i$ holds and the set $B^{\mathcal{E}_{\models}}_{\alpha_i}$ comprising every assertion $b^{\mathcal{E}_{\models}}_{\alpha_i}$ for which $\alpha_i \models_{\mathcal{T}} b^{\mathcal{E}_{\models}}_{\alpha_i}$ holds. For each assertion α_i we can now define a feature $f_e(\alpha_i, B^{\mathcal{E}}_{\alpha_i}) \in F$ that is given by

$$f_e(\alpha_i = \mathrm{x}, B^{\mathcal{E}}_{\alpha_i} = \mathbf{b}^{\mathcal{E}}_{\alpha_i}) = \neg\mathrm{x} \wedge \left(\bigvee_{\mathrm{b}^{\mathcal{E}_{\models}}_{\alpha_i} \in \mathbf{b}^{\mathcal{E}_{\models}}_{\alpha_i}} \mathrm{b}^{\mathcal{E}_{\models}}_{\alpha_i} \right) \vee \mathrm{x} \wedge \neg \left(\bigwedge_{\mathrm{b}^{\mathcal{E}_{\models}}_{\alpha_i} \in \mathbf{b}^{\mathcal{E}_{\models}}_{\alpha_i}} \mathrm{b}^{\mathcal{E}_{\models}}_{\alpha_i} \right). \tag{6.12}$$

As already for conflict features, every entailment feature $f_e(\alpha_i, B^{\mathcal{E}}_{\alpha_i}) \in F$ also describes a hard constraint that has to be satisfied by each possible world $\mathbf{x} \in \mathcal{X}$, why the corresponding weight $w_e \in W$ is again defined with

$$w_e \to -\infty\,. \tag{6.13}$$

Provided that $\mathbf{x}_0 \in \mathcal{X}$ and there exist only non-cyclic entailment relations, due to the combination of both, the conflict feature $f_c(\alpha_i, B^{\mathcal{M}}_{\alpha_i}) \in F$ and the entailment feature $f_e(\alpha_i, B^{\mathcal{E}}_{\alpha_i}) \in F$, the state of an assertion may only be flipped iff the sampling transition does not result in an impossible world.

Proposition 6.2. *Given a Markov network as in Proposition 6.1 but with $\mathcal{E} \neq \emptyset$ and an additional entailment feature $f_e(\alpha_i, B^{\mathcal{E}}_{\alpha_i}) \in F$ for every assertion $\alpha_i \in \mathcal{C}$. Provided that the initial truth value assignment $\mathbf{x}_0$ to $\mathcal{C}$ represents a possible world, i.e., $\mathbf{x}_0 \in \mathcal{X}$, then the regularity of a Markov chain generated via Gibbs sampling is still ensured as long as there exists no cyclic entailment relations, i.e., $B^{\mathcal{E}_{\models}}_{\alpha_i} \cap B^{\mathcal{E}_{\models\!\mid}}_{\alpha_i} = \emptyset$ holds for any α_i,*

Proof. Since $B^{\mathcal{M}}_{\alpha_i} \subseteq B_{b^{\mathcal{M}}_{\alpha_i}}$ holds for every $b_{\alpha_i} \in B^{\mathcal{E}_{\models\!\mid}}_{\alpha_i}$ (as already shown in Section 5.3) we can conclude that if at least one contradicting neighbor $b^{\mathcal{M}}_{\alpha_i} \in B^{\mathcal{M}}_{\alpha_i}$ of α_i is *true* ($B^{\mathcal{M}}_{\alpha_i} = 1$) the state of α_i and every vertex $b_{\alpha_i} \in B^{\mathcal{E}_{\models\!\mid}}_{\alpha_i}$ has to be *false* as we provided that the initial truth value assignment $\mathbf{x}_0$ to $\mathcal{C}$ represents a possible world. Hence, given $B^{\mathcal{M}}_{\alpha_i} = 1$ and $B^{\mathcal{E}_{\models\!\mid}}_{\alpha_i} = 0$, the entailment feature $f_e(\alpha_i, B^{\mathcal{E}}_{\alpha_i}) \in F$ of α_i can only be *true* if the state of α_i is *true*, However, since $\alpha_i = 1, B^{\mathcal{M}}_{\alpha_i} = 1$ also causes that $f_c(\alpha_i, B^{\mathcal{M}}_{\alpha_i})$ will be *true* as well, the regularity for the case of $B^{\mathcal{M}}_{\alpha_i} = 1$ follows directly from Proposition 6.1.

Moreover, given that $\alpha'' \models_{\mathcal{T}} \alpha' \models_{\mathcal{T}} \alpha$, according to Definition 5.2 we can easily conclude that $\alpha'' \models_{\mathcal{T}} \alpha$ follows directly. Hence, this implies that $B^{\mathcal{E}_{\models\!\mid}}_{\alpha_i} \subseteq B^{\mathcal{E}_{\models\!\mid}}_{b_{\alpha_i}}$ holds for every $b_{\alpha_i} \in B^{\mathcal{E}_{\models}}_{\alpha_i}$ and conversely that $B^{\mathcal{E}_{\models}}_{\alpha_i} \subseteq B^{\mathcal{E}_{\models}}_{b_{\alpha_i}}$ holds for every $b_{\alpha_i} \in B^{\mathcal{E}_{\models\!\mid}}_{\alpha_i}$. Hence, by providing that we originate from a possible world and there exists no cyclic entailment relations such that $B^{\mathcal{E}_{\models}}_{\alpha_i} \cap B^{\mathcal{E}_{\models\!\mid}}_{\alpha_i} = \emptyset$ holds for any α_i, it could never be the case that any $b^{\mathcal{E}_{\models}}_{\alpha_i} \in B^{\mathcal{E}_{\models}}_{\alpha_i}$ is *false* (i.e., $\mathrm{b}^{\mathcal{E}_{\models}}_{\alpha_i} = 0$) if there exists a vertex $b^{\mathcal{E}_{\models\!\mid}}_{\alpha_i} \in B^{\mathcal{E}_{\models\!\mid}}_{\alpha_i}$ that is *true* (i.e., $\mathrm{b}^{\mathcal{E}_{\models\!\mid}}_{\alpha_i} = 1$) and vice versa. As a consequence, given that $B^{\mathcal{M}}_{\alpha_i} = 0$ there exists only the following cases for flipping the state of α_i according to its conditional probability (given by Equation (6.3)):

(i) Given that all assertions that are connected with α_i by an entailment relation are *false* such that $B_{\alpha_i} = 0$, then the probability that α_i is *true* in the next possible world is given with

$$p(\alpha_i = 1 \mid B_{\alpha_i} = 0) = \lim_{w_e \to -\infty} \frac{\exp\left(\ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right)\right) + w_e}{\exp(0) + \exp\left(\ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right) + w_e\right)} = 0\,, \tag{6.14}$$

and conversely the conditional probability that α_i is *false* in the next possible world is

$$p(\alpha_i = 0 \mid B_{\alpha_i} = 0) = \lim_{w_e \to -\infty} \frac{\exp(0)}{\exp(0) + \exp\left(\ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right)\right) + w_e} = 1\,. \tag{6.15}$$

(ii) Given again that $B^{\mathcal{E}_{\models\!\mid}}_{\alpha_i} = 0$ and at least one but not all assertions $b^{\mathcal{E}_{\models}}_{\alpha_i} \in B^{\mathcal{E}_{\models}}_{\alpha_i}$ are *true*, i.e., $\exists b_{\alpha_i}, b'_{\alpha_i} \in B^{\mathcal{E}_{\models}}_{\alpha_i} \mid b_{\alpha_i} = 0 \wedge b'_{\alpha_i} = 1$, then the conditional

probabilities for $\alpha_i = 1$ and $\alpha_i = 0$ are equivalent to Equation (6.14) and Equation (6.15), respectively.

(iii) However, given that every assertion $b_{\alpha_i}^{\mathcal{E}_{\models}} \in B_{\alpha_i}^{\mathcal{E}_{\models}}$ is *true*, which is denoted by $B_{\alpha_i}^{\mathcal{E}_{\models}} = \mathbf{1}$ for short, while $B_{\alpha_i}^{\mathcal{E}_{\models\!\!|}} = 0$ still holds, then the conditional probability that α_i is *true* in the next possible world is defined by

$$p(\alpha_i = 1 \mid B_{\alpha_i}^{\mathcal{M}}, B_{\alpha_i}^{\mathcal{E}_{\models\!\!|}} = 0, B_{\alpha_i}^{\mathcal{E}_{\models}} = \mathbf{1}) = \frac{\exp\left(\ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right)\right)}{\exp(0) + \exp\left(\ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right)\right)} = \sigma_{acc}(\alpha, \Sigma_{acc}) \,, \tag{6.16}$$

and hence the probability for $\alpha_i = 0$ is given by

$$p(\alpha_i = 0 \mid B_{\alpha_i}^{\mathcal{M}}, B_{\alpha_i}^{\mathcal{E}_{\models\!\!|}} = 0, B_{\alpha_i}^{\mathcal{E}_{\models}} = \mathbf{1}) = \frac{\exp(0)}{\exp(0) + \exp\left(\ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right)\right)} = 1 - \sigma_{acc}(\alpha, \Sigma_{acc}) \,. \tag{6.17}$$

(iv) Given, on the other hand, that at least one assertion $b_{\alpha_i}^{\mathcal{E}_{\models\!\!|}} \in B_{\alpha_i}^{\mathcal{E}_{\models\!\!|}}$ is *true* and what implies that $B_{\alpha_i}^{\mathcal{E}_{\models}} = \mathbf{1}$ holds as we provided to originate from a possible world, then the conditional probability for $\alpha_i = 1$ is

$$p(\alpha_i = 1 \mid B_{\alpha_i}^{\mathcal{M}} = 0, B_{\alpha_i}^{\mathcal{E}_{\models\!\!|}} = 1, B_{\alpha_i}^{\mathcal{E}_{\models}} = \mathbf{1}) = \lim_{w_e \to -\infty} \frac{\exp\left(\ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right)\right)}{\exp(w_e) + \exp\left(\ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right)\right)} = 1 \,, \tag{6.18}$$

and conversely the probability for $\alpha_i = 0$ is given with

$$p(\alpha_i = 0 \mid B_{\alpha_i}^{\mathcal{M}} = 0, B_{\alpha_i}^{\mathcal{E}_{\models\!\!|}} = 1, B_{\alpha_i}^{\mathcal{E}_{\models}} = \mathbf{1}) = \lim_{w_e \to -\infty} \frac{\exp(w_e)}{\exp(w_e) + \exp\left(\ln\left(\frac{\sigma_{acc}(\alpha, \Sigma_{acc})}{1 - \sigma_{acc}(\alpha, \Sigma_{acc})}\right)\right)} = 0 \,. \tag{6.19}$$

Hence, a state flip of an assertion α_i is only performed according to the corresponding signature accuracy (Definition 5.3), iff each assertion $b_{\alpha_i}^{\mathcal{E}_{\models}} \in B_{\alpha_i}^{\mathcal{E}_{\models}}$ is *true* and every assertion $b_{\alpha_i}^{\mathcal{E}_{\models\!\!|}} \in B_{\alpha_i}^{\mathcal{E}_{\models\!\!|}}$ is *false*. Consequently, given a possible world $\mathbf{x}_t \in \mathcal{X}$,

due to $f_e(\alpha_i, B^{\mathcal{E}}_{\alpha_i})$ as defined in Equation (6.12) it is ensured that any state flip and hence any transition will always end up with a possible world (consistent state) $\mathbf{x}_{t+1} \in \mathcal{X}$, as long as $B^{\mathcal{E}_{\models}}_{\alpha_i} \cap B^{\mathcal{E}_{\models}}_{\alpha_i} = \emptyset$ holds for any assertion $\alpha_i \in \mathcal{C}$. As we supposed that there exist no cyclic entailment relations, for all assertions $\alpha_i \in \mathcal{C}$ we can define an order $\prec$ such that no assertion α_i is lower in the order than every assertion $b^{\mathcal{E}_{\models}}_{\alpha_i} \in B^{\mathcal{E}_{\models}}_{\alpha_i}$, i.e., $b^{\mathcal{E}_{\models}}_{\alpha_i} \prec \alpha_i$ holds for every $b^{\mathcal{E}_{\models}}_{\alpha_i} \in B^{\mathcal{E}_{\models}}_{\alpha_i}$. Hence, since we again can get from any possible world $\mathbf{x} \in \mathcal{X}$ to any possible world $\mathbf{x}' \in \mathcal{X}$ in at most $L = |\mathcal{C}|$ steps of state flips with a probability that is greater than 0, the regularity of the resulting Markov chain is still given. □

However, let us now assume that there exists some cyclic entailment relations such that $B^{\mathcal{E}_{\models}}_{\alpha_i} \cap B^{\mathcal{E}_{\models}}_{\alpha_i} \neq \emptyset$ holds for any α_i. The simplest case for this is given with $\alpha_i \models_{\mathcal{T}} b_{\alpha_i}$ and $b_{\alpha_i} \models_{\mathcal{T}} \alpha_i$ in order that $B^{\mathcal{E}_{\models}}_{\alpha_i} = B^{\mathcal{E}_{\models}}_{\alpha_i} = \{b_{\alpha_i}\}$. Hence, given the two Boolean random variables α_i and b_{α_i} the corresponding state graph of sampling transitions is depicted in Figure 6.2.

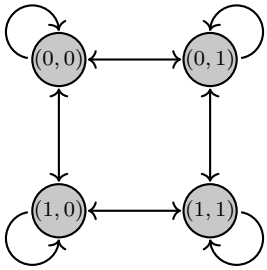

Figure 6.2: State Transition Graph of Two Variables [GM10]

As we require a possible world $\mathbf{x}_0 \in \mathcal{X}$, the initial truth value assignment of b_{α_i} necessarily has to be equivalent to α_i in order that we may only start in state $(0, 0)$ or state $(1, 1)$. However, given that $\alpha_i = b_{\alpha_i} = 0$ the conditional probability that α_i is flipped to *true* is $p(\alpha_i = 1 \mid B_{\alpha_i} = 0) = 0$ and, on the contrary, given that $\alpha_i = b_{\alpha_i} = 1$ then the conditional probability for $\alpha_i = 0$ is given with $p(\alpha_i = 0 \mid B^{\mathcal{M}}_{\alpha_i} = 0, B^{\mathcal{E}_{\models}}_{\alpha_i}, B^{\mathcal{E}_{\models}}_{\alpha_i} = \mathbf{1}) = 0$. Hence, in both cases the state of α_i will never be flipped so that the state $(1, 1)$ can never be reached from state $(0, 0)$ or vice versa, which is why the regularity of the Markov chain is no longer given.

In order to be able to handle cyclic entailment relations in Gibbs sampling, we have to ensure that states like $(0, 1)$ and $(0, 1)$ of the example shown in Figure 6.2 representing an impossible world are skipped over during sampling transitions. However, since the emergence of disconnected sampling states cannot be prevented by defining an additional feature, we have to modify the sampling process accordingly. For that we redefine the set $B^{\mathcal{E}}_{\alpha_i}$ of all assertions that are connected with assertion α_i by an entailment relation by further subdividing this set into $B^{\mathcal{E}}_{\alpha_i} = B^{\mathcal{E}_{\models}}_{\alpha_i} \cup B^{\mathcal{E}_{\models}}_{\alpha_i} \cup B^{\mathcal{E}_{\models\models}}_{\alpha_i}$. In this way we differentiate between the set $B^{\mathcal{E}_{\models}}_{\alpha_i}$ that is given with $B^{\mathcal{E}_{\models}}_{\alpha_i} = \{b^{\mathcal{E}_{\models}}_{\alpha_i} \mid b^{\mathcal{E}_{\models}}_{\alpha_i} \models_{\mathcal{T}} \alpha_i \wedge \alpha_i \not\models_{\mathcal{T}} b^{\mathcal{E}_{\models}}_{\alpha_i}\}$, the set $B^{\mathcal{E}_{\models}}_{\alpha_i}$ which

is defined by $B^{\mathcal{E}_{\models}}_{\alpha_i} = \{b^{\mathcal{E}_{\models}}_{\alpha_i} \mid \alpha_i \models_{\mathcal{T}} b^{\mathcal{E}_{\models}}_{\alpha_i} \wedge b^{\mathcal{E}_{\models}}_{\alpha_i} \not\models_{\mathcal{T}} \alpha_i\}$, and the set $B^{\mathcal{E}_{\models\!\models}}_{\alpha_i}$ comprising every assertion b_{α_i} for which both, $b_{\alpha_i} \models_{\mathcal{T}} \alpha_i$ and $\alpha_i \models_{\mathcal{T}} b_{\alpha_i}$, hold, i.e., $B^{\mathcal{E}_{\models\!\models}}_{\alpha_i} = \{b_{\alpha_i} \mid b_{\alpha_i} \models_{\mathcal{T}} \alpha_i \wedge \alpha_i \models_{\mathcal{T}} b_{\alpha_i}\}$. As a consequence, only the set $B^{\mathcal{E}_{\models\!\models}}_{\alpha_i}$ may comprise assertions for which there exists a cyclic entailment relation to α_i. Hence, by separating all assertions $b^{\mathcal{E}_{\models\!\models}}_{\alpha_i} \in B^{\mathcal{E}_{\models\!\models}}_{\alpha_i}$ from $B^{\mathcal{E}_{\models}}_{\alpha_i}$ and $B^{\mathcal{E}_{\models}}_{\alpha_i}$, any cyclic entailment relation to α_i is leave out of consideration when α_i is sampled. Subsequently, after the sampling of α_i every assertion $b^{\mathcal{E}_{\models\!\models}}_{\alpha_i} \in B^{\mathcal{E}_{\models\!\models}}_{\alpha_i}$ is set to the same state of α_i and thus may only be sampled again in the next sampling process. Hence, by skipping over the impossible worlds this modification of the Gibbs sampling ensures that all cyclic entailment relations are considered such that the sampling transition ends up again with a possible world.

Proposition 6.3. *Given a Markov network as in Proposition 6.2 but where $\mathcal{E}$ may comprise cyclic entailment relations, such that $B^{\mathcal{E}_{\models\!\models}}_{\alpha_i} \neq \emptyset$. Provided that the initial truth value assignment $\mathbf{x}_0$ to $\mathcal{C}$ represents again a possible world, i.e., $\mathbf{x}_0 \in \mathcal{X}$, then an application of the modified Gibbs sampling where every assertion $b^{\mathcal{E}_{\models\!\models}}_{\alpha_i} \in B^{\mathcal{E}_{\models\!\models}}_{\alpha_i}$ is set to the sampled state x' of α_i results in a Markov chain that is still regular.*

Proof. Due to our limitation on *DL-Lite*$_{\mathcal{A}}$, an entailment relation according to Definition 5.2 is inherently restricted to exactly two assertions. Moreover, given that $\alpha_i \models_{\mathcal{T}} b_{\alpha_i}$ and $b_{\alpha_i} \models_{\mathcal{T}} \alpha_i$ we can easily conclude that the sets of neighboring assertions are equivalent for both assertions, α_i and b_{α_i}, i.e., $B^{\mathcal{M}}_{\alpha_i} = B^{\mathcal{M}}_{b_{\alpha_i}}$, $B^{\mathcal{E}_{\models}}_{\alpha_i} = B^{\mathcal{E}_{\models}}_{b_{\alpha_i}}$, $B^{\mathcal{E}_{\models}}_{\alpha_i} = B^{\mathcal{E}_{\models}}_{b_{\alpha_i}}$, and $B^{\mathcal{E}_{\models\!\models}}_{\alpha_i} \setminus \{b_{\alpha_i}\} = B^{\mathcal{E}_{\models\!\models}}_{b_{\alpha_i}} \setminus \{\alpha_i\}$. Hence, in order to get an assignment of truth values that represents a possible world $\mathbf{x} \in \mathcal{X}$ all assertions in $\{\alpha_i\} \cup B^{\mathcal{E}_{\models\!\models}}_{\alpha_i}$ must necessarily have the same state. As a consequence, instead of considering each assertion as a separate Boolean random variable we could alternatively define a Boolean random variable $\boldsymbol{\alpha}^{\models\!\models}_i$ representing the state of all assertions in $\{\alpha_i\} \cup B^{\mathcal{E}_{\models\!\models}}_{\alpha_i}$. Since in Gibbs sampling the assertions are processed in random order, every assertion in $\boldsymbol{\alpha}^{\models\!\models}_i$ is selected with the same probability. Because of that, the signature accuracy that is assigned to $\boldsymbol{\alpha}^{\models\!\models}_i$ is simply the average of all corresponding signature accuracies of every assertion in $\boldsymbol{\alpha}^{\models\!\models}_i = \{\alpha_i\} \cup B^{\mathcal{E}_{\models\!\models}}_{\alpha_i}$ and hence is given with

$$\sigma_{acc}(\boldsymbol{\alpha}^{\models\!\models}_i, \Sigma_{acc}) = \frac{\sum_{\alpha \in \boldsymbol{\alpha}^{\models\!\models}_i} \sigma_{acc}(\alpha, \Sigma_{acc})}{\#\boldsymbol{\alpha}^{\models\!\models}_i}\,. \tag{6.20}$$

As a result, by substituting in $G_{\mathcal{C}} = (\mathcal{C}, \mathcal{M}, \mathcal{E})$ for every set $\boldsymbol{\alpha}^{\models\!\models}_i$ any assertion of $\{\alpha_i\} \cup B^{\mathcal{E}_{\models\!\models}}_{\alpha_i}$ by $\boldsymbol{\alpha}^{\models\!\models}_i$ we will get a Markov network graph $G'_{\mathcal{C}} = (\mathcal{C}', \mathcal{M}', \mathcal{E}')$ that still models exactly the same probabilistic dependencies but without any cyclic entailment relations. Thus, the regularity for a Markov chain generated via the modified Gibbs sampling follows directly from Proposition 6.2. □

As a consequence, despite the existence of cyclic entailment relations the modified Gibbs sampling converges to the desired posterior probability distribution.

Moreover, it is easy to see that the computational complexity of the Gibbs sampling is linear with respect to the number of conflicting assertions. Nevertheless, in terms of performance optimization, the Gibbs sampling can be applied in parallel to each independent subgraph of the conflict graph (i.e., the Markov network graph) as there exist no probabilistic dependencies.

Based on the resulting Markov chain $\mathbf{x}_0, \ldots, \mathbf{x}_K$ we can now calculate the (approximated) marginal probability (trust) of each assertion according to the following definition:

Definition 6.3 (Assertion Trusts)**.** *Given an inconsistent federated DL-Lite$_\mathcal{A}$ knowledge base $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$, and let $\mathcal{C}$ denotes the set of all conflicting assertions in $\mathcal{K}_F$, $\mathcal{A}'_F \subseteq \mathcal{A}_F$ represents the set of all correct assertions (according to Definition 5.3), and $\mathbf{x}_0, \ldots, \mathbf{x}_K$ is a Markov chain generated via the proposed Gibbs sampling. Then, the assertion trust $p(\alpha_i)$ for a federated assertion $\alpha_i \in \mathcal{A}_F$ of $\mathcal{K}_F$ is given by*

$$p(\alpha_i) = \begin{cases} 1.0\,, & \text{if } \alpha_i \in \mathcal{A}'_F\,, \\ \dfrac{\sum\limits_{\mathbf{x}_j \in \{\mathbf{x}_0,\ldots,\mathbf{x}_K\}} \mathbf{x}_j(\alpha_i)}{K}\,, & \text{if } \alpha_i \in \mathcal{C}\,, \\ \varnothing\,, & \text{otherwise}\,, \end{cases} \tag{6.21}$$

where $\mathbf{x}_j(\alpha_i)$ returns the truth value assignment $\mathrm{x}_i \in \mathbf{x}_j$ for α_i and $\varnothing$ denotes undefined.

It is easy to see that the assertion trust for an assertion α that is part of an unary MISA, i.e., $\{\alpha\} \in \mathcal{M}$, will always result in $p(\alpha) = 0$, since α is part of its Markov blanked, i.e., $\alpha \in B_\alpha$, why the conflict feature with $f_c(\alpha = 1, B_\alpha^\mathcal{M} = 1) = 1$ prevents in any case that α will be flipped to $true$.

The assessment of assertion trusts using the proposed Gibbs sampling is outlined in Algorithm 6.1: AssessAssertionTrusts. Given for an inconsistent federated *DL-Lite$_\mathcal{A}$* knowledge base $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$ the set $\mathcal{M}$ of MISAs, a corresponding repair $\mathcal{R}$ that is complete and minimal, the set $\mathcal{E}$ of entailment relations within $\mathcal{K}$, a map Σ_{acc} comprising for each signature element $\sigma \in \Sigma_n$ of every data source $\mathcal{K}_n$ integrated in the federated KB knowledge base $\mathcal{K}_F$ the associated signature accuracy according to Definition 5.3, a set $\mathcal{A}'_F$ that consists of all correct assertions (according to Definition 5.3) in $\mathcal{K}_F$, and a number K of samplings. The algorithm starts by initializing the set $\mathcal{C}$ of conflicting assertions based on the given set $\mathcal{M}$ of MISAs and subsequently iterates over the number of samples. In each sampling, the set $\mathcal{C}$ of conflicting assertions is shuffled such that the assertions are processed in random order. For each conflicting assertions $\alpha \in \mathcal{C}$ the corresponding sets of neighboring assertions, i.e., $B_\alpha^\mathcal{M}$, $B_\alpha^{\mathcal{E}_=}$, $B_\alpha^{\mathcal{E}_\models}$, $B_\alpha^{\mathcal{E}_\models}$, are determined. By using the external function $\sigma_{acc}(\alpha, \Sigma_{acc})$ that returns the signature accuracy of the signature element σ_α (concept name, role name or attribute name) of α with respect to the data source $\mathcal{K}_n$, i.e., $\mathcal{A}_n$ in which α is stated, it is decided according to assigned signature accuracy if the state of α may be set to $true$ or $false$ in the next possible

Algorithm 6.1: AssessAssertionTrusts$(\mathcal{M}, \mathcal{R}, \mathcal{E}, \Sigma_{acc}, \mathcal{A}'_F, K)$

Input: set $\mathcal{M}$ of MISAs,
(complete) repair $\mathcal{R}$ for $\mathcal{M}$,
set $\mathcal{E}$ of entailment relations,
map Σ_{acc} of signature accuracies,
set $\mathcal{A}'_F$ of correct assertions,
number of samplings K

Output: set $\mathcal{P}_{\mathcal{A}_F}$ of assertion trusts

begin
  $\mathcal{C} \leftarrow \bigcup_{\boldsymbol{m} \in \mathcal{M}} \boldsymbol{m}$
  $S_{true}, \mathcal{P}_{\mathcal{A}_F} \leftarrow \emptyset$
  $\forall \alpha \in \mathcal{C} : S_{true}[\alpha] \leftarrow 0$
  for $k \leftarrow 1$ *to* K **do**
    $\mathcal{C}_{t+1} \leftarrow \emptyset$
    $\mathcal{C} \leftarrow \mathsf{Shuffle}(\mathcal{C})$
    foreach $\alpha \in \mathcal{C}$ **do**
      if $\alpha \notin \mathcal{C}_{t+1}$ **then**
        $B^{\mathcal{M}}_{\alpha} \leftarrow \{b^{\mathcal{M}}_{\alpha} \mid (\alpha, b^{\mathcal{M}}_{\alpha}) \in \mathcal{M}\}$
        $B^{\mathcal{E}_{\models}}_{\alpha} \leftarrow \{b^{\mathcal{E}_{\models}}_{\alpha_i} \mid (b^{\mathcal{E}_{\models}}_{\alpha}, \alpha) \in \mathcal{E} \wedge (\alpha, b^{\mathcal{E}_{\models}}_{\alpha}) \notin \mathcal{E}\}$
        $B^{\mathcal{E}_{\vDash}}_{\alpha} \leftarrow \{b^{\mathcal{E}_{\vDash}}_{\alpha_i} \mid (\alpha, b^{\mathcal{E}_{\vDash}}_{\alpha}) \in \mathcal{E} \wedge (b^{\mathcal{E}_{\vDash}}_{\alpha}, \alpha) \notin \mathcal{E}\}$
        $B^{\mathcal{E}_{\models\models}}_{\alpha} \leftarrow \{b^{\mathcal{E}_{\models\models}}_{\alpha_i} \mid (\alpha, b^{\mathcal{E}_{\models\models}}_{\alpha}) \in \mathcal{E} \wedge (b^{\mathcal{E}_{\models\models}}_{\alpha}, \alpha) \in \mathcal{E}\}$
        if $\mathsf{Random}(0,1) \leq \sigma_{acc}(\alpha, \Sigma_{acc})$ **then**
          if $B^{\mathcal{M}}_{\alpha} \setminus \mathcal{R} = \emptyset \wedge \{\alpha\} \notin \mathcal{M} \wedge B^{\mathcal{E}_{\vDash}}_{\alpha} \cap \mathcal{R} = \emptyset$ **then**
            $\mathcal{R} \leftarrow \mathcal{R} \setminus (\{\alpha\} \cup B^{\mathcal{E}_{\models\models}}_{\alpha})$
            $S_{true}[\alpha] \leftarrow S_{true}[\alpha] + 1$
            $\forall b^{\mathcal{E}_{\models\models}}_{\alpha} \in B^{\mathcal{E}_{\models\models}}_{\alpha} : S_{true}[b^{\mathcal{E}_{\models\models}}_{\alpha}] \leftarrow S_{true}[b^{\mathcal{E}_{\models\models}}_{\alpha}] + 1$
          else
            if $B^{\mathcal{E}_{\models}}_{\alpha} \setminus \mathcal{R} = \emptyset$ **then**
              $\mathcal{R} \leftarrow \mathcal{R} \cup (\{\alpha\} \cup B^{\mathcal{E}_{\models\models}}_{\alpha})$
            else
              $S_{true}[\alpha] \leftarrow S_{true}[\alpha] + 1$
              $\forall b^{\mathcal{E}_{\models\models}}_{\alpha} \in B^{\mathcal{E}_{\models\models}}_{\alpha} : S_{true}[b^{\mathcal{E}_{\models\models}}_{\alpha}] \leftarrow S_{true}[b^{\mathcal{E}_{\models\models}}_{\alpha}] + 1$
          $\mathcal{C}_{t+1} \leftarrow \mathcal{C}_{t+1} \cup \{\alpha\} \cup B^{\mathcal{E}_{\models\models}}_{\alpha}$
  $\forall \alpha \in \mathcal{C} : \mathcal{P}_{\mathcal{A}_F}[\alpha] \leftarrow S_{true}[\alpha]/K$
  $\forall \alpha \in \mathcal{A}'_F : \mathcal{P}_{\mathcal{A}_F}[\alpha] \leftarrow 1.0$
  return $\mathcal{P}_{\mathcal{A}_F}$
end

world. In order to avoid that the algorithm does not step into an impossible world, both conditions described by the conflict feature $f_c(\alpha, B_\alpha^{\mathcal{M}})$ (Equation (6.6)) and the entailment feature $f_e(\alpha, B_\alpha^{\mathcal{E}})$ (Equation (6.12)) have to be satisfied before the state of α is actually flipped. Given that the state of α is set to $false$, α and every assertion in $B_\alpha^{\mathcal{E}\models}$ are added to the repair $\mathcal{R}$. If, on the other hand, the state of α is set to $true$, α and every assertion in $B_\alpha^{\mathcal{E}\models}$ are removed from the repair $\mathcal{R}$ and for any assertion $\alpha_i \in \{\alpha\} \cup B_\alpha^{\mathcal{E}\models}$ the corresponding counter $S_{true}[\alpha_i]$ representing the number of states in which the assertion α_i is $true$ is incremented by 1. Before the next assertion is selected, α and every assertion in $B_\alpha^{\mathcal{E}\models}$ are added to the set $\mathcal{C}_{t+1}$ in order to ensure that no assertion of $\mathcal{C}$ is sampled twice during one sampling process. Finally, after K samplings according to Equation 6.21 the probability for each assertion $\alpha \in \mathcal{C}$ is calculated based on the number $S_{true}[\alpha]$ of states in which $\alpha = true$ and for all assertions $\alpha \in \mathcal{A}'_F$ (correct assertions) the trust value 1.0 is assigned.

Obviously, the assessment of probabilities cannot be done for all assertions in $\mathcal{A}_F$ of $\mathcal{K}_F$, i.e., for any assertion in $\mathcal{A}_F \setminus (\mathcal{C} \cup \mathcal{A}'_F)$ that is neither correct nor involved in any MISA. Because of that, for those assertions we determine in the following Section 6.3.2 trust values for individual signature elements with respect to a specific data source, called *signature trusts*.

Example 6.2 (Assertion Trusts)**.** *By using the signature accuracies of Example 5.3 and applying Algorithm 6.1 with $K = 10{,}000$ to the conflict graph (Figure 5.1) of our running example we are getting the following assertion trusts for the conflicting assertions:*

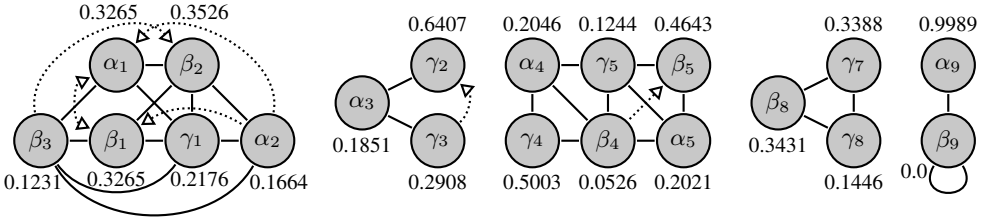

Figure 6.3: Assertion Trusts

Moreover, β_7 and γ_6 are considered as correct assertions, why the assessed trust value for each of those assertions is 1.0. As we can observe, α_6, α_7, α_8 and β_6 are the only assertions for which no trust value has been assessed since all of them are neither involved in any conflict nor regarded as correct.

6.3.2 Signature Trusts

Based on the assessed assertion trusts we can now easily define a trust value for a signature element $\sigma \in \Sigma_n$ of a data source $\mathcal{K}_n$ integrated in the federated KB knowledge base $\mathcal{K}_F$ as follows:

Definition 6.4 (Signature Trust)**.** *Given an inconsistent federated DL-Lite$_\mathcal{A}$ knowledge base $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$, and let $\mathcal{C}$ denotes the set of all conflicting assertions in $\mathcal{K}_F$ and $\mathcal{A}'_F \subseteq \mathcal{A}_F$ represents the set of all correct assertions (according to Definition 5.3). Then, the signature trust $p(\sigma_i, n)$ for a signature element $\sigma_i \in \Sigma_n$ of data source $\mathcal{K}_n$ in $\mathcal{K}_F$ is defined with*

$$p(\sigma, n) = \begin{cases} \dfrac{\sum\limits_{\alpha \in \mathcal{A}_n \cap (\mathcal{C} \cup \mathcal{A}'_F) \mid \sigma(\alpha) = \sigma} p(\alpha)}{\#\{\alpha \in \mathcal{A}_n \cap (\mathcal{C} \cup \mathcal{A}'_F) \mid \sigma(\alpha) = \sigma\}}, & \text{if } \{\alpha \in \mathcal{A}_n \cap \mathcal{C} \mid \sigma(\alpha) = \sigma\} \neq \emptyset, \\ \varnothing, & \text{otherwise}, \end{cases} \tag{6.22}$$

where $\sigma(\alpha)$ is an external function that returns the signature element σ_α (concept name, role name or attribute name) of an assertion α, $p(\alpha)$ is the assessed trust value for assertion α according to Definition 6.3, and $\varnothing$ denotes undefined.

Intuitively, the signature trust for a signature element σ with respect to data source $\mathcal{K}_n$ is given by the average of all trust values for assertions on σ in $\mathcal{A}_n$ of $\mathcal{K}_n$. As a result, for any assertion which has an undefined assertion trust value we can now use as probability the calculated signature trust value that is assessed for the respective signature element with respect to the corresponding data source.

Example 6.3 (Signature Trusts)**.** *According to Definition 6.4, the trust value of the signature element $Paper$ in data source $\mathcal{K}_1$ is calculated based on the assertion trusts of α_1 and α_3 and hence given with $p(Paper, 1) = \frac{0.3265+0.1851}{2} = 0.2558$. Since e.g. for α_7 and α_8 no assertion trust could be assessed, the corresponding signature trust of $Paper$ with respect to data source $\mathcal{K}_1$ is used as probability for those assertions. The complete list of calculated signature trusts for our running example is given below:*

$p(Paper, 1) = 0.2558$	$p(Paper, 2) = 0.3348$	$p(SlideSet, 3) = 0.4529$
$p(publishedIn, 1) = 0.3933$	$p(Proceedings, 2) = 0.6056$	$p(Proceedings, 3) = 0.6694$
$p(\mathsf{edition}, 1) = 0.9989$	$p(publishedIn, 2) = 0.3919$	$p(slideSetOf, 3) = 0.1866$
	$p(\mathsf{edition}, 2) = 0.0$	

6.3.3 Data Source Trusts

However, it is easy to see that a signature trust can only be assessed for a signature element σ of a data source $\mathcal{K}_n$ if there exist any assertion α for σ in $\mathcal{A}_n$ that is involved in any MISA, i.e., $\alpha \in \mathcal{C}$. Because of that, we in turn assess a trust value, called *data source trust*, for each data source $\mathcal{K}_n$ that is integrated in the federated KB knowledge base $\mathcal{K}_F$. Based on the calculated signature trusts, the trust value for a specific data source $\mathcal{K}_n$ can be formally defined as:

Definition 6.5 (Data Source Trust)**.** *Given again an inconsistent federated DL-Lite$_\mathcal{A}$ knowledge base $\mathcal{K}_F = \langle \mathcal{T}_F, \mathcal{A}_F \rangle$, and let Σ_n denotes the signature of data source*

$\mathcal{K}_n$ in $\mathcal{K}_F$, and $\mathcal{C}$ represents the set of all conflicting assertions in $\mathcal{K}_F$. The data source trust $p(n)$ for data source $\mathcal{K}_n$ in $\mathcal{K}_F$ is then given with

$$p(n) = \begin{cases} \dfrac{\sum\limits_{\sigma \in \Sigma_n \mid \{\alpha \in \mathcal{A}_n \cap \mathcal{C} \mid \sigma(\alpha) = \sigma\} \neq \emptyset} p(\sigma, n) * \#\{\alpha \in \mathcal{A}_n \mid \sigma(\alpha) = \sigma\}}{\sum\limits_{\sigma \in \Sigma_n \mid \{\alpha \in \mathcal{A}_n \cap \mathcal{C} \mid \sigma(\alpha) = \sigma\} \neq \emptyset} \#\{\alpha \in \mathcal{A}_n \mid \sigma(\alpha) = \sigma\}}, & \text{if } \mathcal{A}_n \cap \mathcal{C} \neq \emptyset, \\ \varnothing\,, & \text{otherwise}, \end{cases} \tag{6.23}$$

where $\sigma(\alpha)$ is again an external function returning the signature element σ_α of an assertion α, $p(\sigma, n)$ is the assessed trust value for signature element σ with respect to data source $\mathcal{K}_n$ according to Definition 6.4, and $\varnothing$ denotes undefined.

Roughly speaking, the trust value for a data source $\mathcal{K}_n$ is calculated as the average of the weighted sum of signature trusts, where each trust value of a sigature element $\sigma \in \Sigma_n$ is weighted by the number of assertions on σ in $\mathcal{A}_n$ of $\mathcal{K}_n$. As there still might exist some signature elements and hence some assertions without an assessed probability, we can now use instead the trust value of the respective data source. However, if a data source does not comprise any conflicting assertions such that $\mathcal{A}_n \cap \mathcal{C} = \emptyset$, a corresponding trust value for such a data source can obviously not be computed. In this exceptional case we might use a default or user-defined trust value in order to ensure that we get a federated KB in which a trust value is assessed for every signature element and every assertion.

Example 6.4 (Data Source Trusts)**.** *Based on the calculated signature trusts of Example 6.3 we can now determine the trust values for each data source according to Definition 6.5. Correspondingly, the data source trust of $\mathcal{K}_1$ is calculated with $p(1) = \frac{0.2558*4+0.3933*4+0.9989*1}{9} = 0.3995$. All data source trusts for our running are as below:*

$$p(1) = 0.3995 \qquad p(2) = 0.4069 \qquad p(3) = 0.4071$$

Let us suppose, for example, that data source $\mathcal{K}_1$ would contain an additional assertion $SlideSet(\mathbf{I8})$. Since for this assertion as well as for the signature element $SlideSet$ with respect to $\mathcal{K}_1$ no trust value can be assessed, the corresponding data source trust value $p(1) = 0.3995$ is used instead.

6.4 Related Work

By proposing in this chapter an automated approach for fine-grained trust assessment, we address both, an alternative handling of inconsistency as well as the assessment of trust values. Because of that, we divide this section about related work into the subject areas of paraconsistent & approximate logics (Section 6.4.1) and trust & quality assessment (Section 6.4.2).

6.4.1 Paraconsistent & Approximate Logics

As we already know from Section 2.3 answering (U)CQs over an inconsistent KB makes no sense. However, in the context of OBII resp. of federated KBs the requirement for consistency can often not be met and the application of a repair may not be realizable especially when the data sources are managed and maintained autonomously. An alternative strategy is to keep the inconsistency but refine the semantics of query answering, called paraconsistent or inconsistency-tolerant query answering. Some works in this direction are for example [Ber06; Lem+11; Lem+12; Sav13] as already mentioned in Section 4.4. Moreover, in order to perform general reasoning tasks over inconsistent KBs there exist several works following the subject of paraconsistent logics. A survey of such approaches can be found in [BCG07; Ngu10].

In addition to paraconsistent logics there exist approaches addressing the representation and reasoning of knowledge with uncertainty or vagueness [LS08; Jia+09]. Proposed extensions of DLs for uncertainty and vagueness can be classified in probabilistic DLs [Ram+12; Gut+17], possibilistic DLs [BB13; Zho+09], fuzzy DLs [Str15] and rough DLs [SKP07; Kee11]. Approaches exploiting probabilistic graphical models, i.e., Bayesian networks or Markov networks, for probabilistic knowledge representation and reasoning are, e.g., [dFL08; MC15] and [NNS11], respectively.

However, even those approaches are closely related to our work, under paraconsistent semantics contradictory assertions are just excluded from any reasoning and approximate logics mainly focus on uncertain or vague knowledge representation and reasoning but typically do not address the upstream process of determining precise degrees for the uncertainty of individual assertions.

6.4.2 Trust & Quality Assessment

The notion of trust has been used in a heterogeneous way within the semantic web community (surveyed in [AG07]) and in order to consider the quality of different data sources, varying truth discovery techniques are proposed, such as in [Gal+10; Li+12; Liu+17; YHY08; ZH12; Don+15]. The principle of truth discovery is to assess the reliability of individual data sources such that the more frequently true information is provided, the higher the trust in a data source. Hence, the information that is provided by a reliable source is considered to be trustworthy. A comprehensive survey of methods for truth discovery is given by Li et al. [Li+16]. However, since such approaches typically assess only the reliability of a data source but not the quality of the provided information, the fact that trust values for assertions or signature elements may differ widely from the assessed data source trust is just neglected. Instead of assessing trust values for assertions by an assumed data source trust (top-down strategy), in our approach we follow a bottom-up strategy by determining a data source trust based on signature trusts and consequently on individual assertion trusts.

In order to consider the varying reliability of sources among different topics Ma et al. [Ma+15] proposed an approach that automatically assigns topics to a question and estimates the topic-specific expertise of a source. By using a Bayesian model and Gibbs sampling in order to calculate probabilistic values on facts (assertions) Zhao et al. [Zha+12] proposed an approach that is more closer to our work. However, the authors base their notion of conflicting facts on direct contradictions that origin from a closed world assumption, while we are using a TBox in order to find both, explicit and implicit conflicts. Moreover, as we are relying on an open world assumption, non-stated facts do not correspond to the claim of their negation.

Instead of estimating source reliability only based on the accuracy of the provided information, there further exist frameworks and methodologies for assessing the quality of data sources and its provided information by considering diverse quality dimensions and metrics, such as accessibility, performance, reputation, timeliness and others. A systematic review of such approaches assessing the quality of LOD sources and a comprehensive list of dimensions and metrics under a common classification scheme is proposed by Zaveri et al. [Zav+16]. Besides, a more recent survey in this context is given also by Mountantonakis and Tzitzikas [MT19].

Obviously, the referred works are typically assess trust values based on external criteria or expect some initial trust estimations. In contrast, our approach is different from the mentioned approaches in two aspects. First, we assume for each data source and every assertion the same level of trust prior to the majority voting. By analyzing and leveraging the conflict graph that is constructed based on a well-defined semantics, as well as by exploiting the statistical evidence gathered by inconsistency resolution, we compute individual probabilities for all conflicting assertions. The resulting trust values aids a domain expert or an application to individually discard assertions with low probabilities. Moreover, our approach can be easily extended such that the majority voting starts with varying trust values based on an analysis of data provenance. Second, the intention of our approach is not to use the calculated probabilities for truth discovery but to represent uncertain knowledge and thus the application of probabilistic reasoning and paraconsistent logics. To the best of our knowledge there is currently no other approach in this direction.

6.5 Summary

In this chapter we proposed an automated approach for fine-grained trust assessment at different levels of granularity. By considering the conflict graph as a Markov network graph and exploiting the statistical evidence gathered by the repair generation via majority voting, we facilitate the application of Gibbs sampling in order to calculate trust values for any conflicting assertion. Based on these assertion trusts we further assess trust values at the level of signature elements as well as for each individual data source. As we could show the assessment of adequate trust

values based on our debugging results, we gave an answer to research question Q3.

Moreover, by contributing a fine-grained trust assessment at different levels of granularity we further provide a fully automated approach to transform a conventional federated KB into a probabilistic one as what is asked for in research question Q4. To sum up, we can emphasize that our approach finally enables

- the representation and reasoning on uncertain (i.e., imprecise) knowledge with fine-grained probabilities,
- the application of paraconsistent (inconsistency-tolerant) logics taking the probabilities into account,
- the computation of the most probable consistent federated KB.

Part III

Experimental Evaluation

Chapter 7

LOD Dataset and Experimental Settings

So far we have shown that the proposed approaches in theory deliver remarkably good results, i.e., a complete inconsistency detection including the generation of corresponding explanations (MISAs), a complete repair that considers any entailment relations, and the assessment of assertion trust values based on a regular Markov chain. However, we currently do not know how the implementation of the provided algorithms do perform in practice and which quality the results do have. In order to be able to evaluate these criteria and hence to answer research question Q5 and Q6 we have set up a large distributed LOD dataset from the domain of library science that has already been used for the experimental evaluation of our previous works [Nol+16; Nol+17].

In particular, for our experimental dataset we have selected the following four LOD sources:

$\mathcal{K}_1$: FacetedDBLP[1],
$\mathcal{K}_2$: Bibsonomy[2],
$\mathcal{K}_3$: RKB Explorer ePrints Open Archives[3],
$\mathcal{K}_4$: RKB Explorer DBLP[4].

Our approach for inconsistency detection relies solely on query answering and hence does not impose any additional requirements on the data sources expect of a querying interface. However, since a SPARQL interface is not provided by each of those sources respectively in order to bypass any downtime and to avoid any bottleneck we have loaded the dump of each data source into a separate Virtuoso[5] 7.2.2 instance (Open-Source Edition), an RDF triple store that is providing a SPARQL interface. All four Virtuoso instances are hosted in an Ubuntu 14.04 LTS virtual

[1] `http://dblp.l3s.de`
[2] `https://www.bibsonomy.org`
[3] `http://foreign.rkbexplorer.com`
[4] `http://dblp.rkbexplorer.com`
[5] `http://vos.openlinksw.com`

machine with 6x Intel Xeon CPUs (à 4 cores @ 2.50 GHz) and 96 GB of RAM (16 GB of RAM are assigned to each Virtuoso instance).

For providing a generalized description of the considered domain in terms of a federated *DL-Lite*$_\mathcal{A}$ TBox according to Definition 3.1, we have defined an intermediary TBox by using the OWL 2 QL profile as specification language. Besides axioms aligning TBox expressions of the integrated sources we have further added negative inclusions and functionality assertions in the intermediary TBox as there is a lack of such axioms in the source-specific TBoxes. Moreover, in order to ensure that the federated TBox is coherent, we additionally had to apply some minor modifications to scattered TBoxes of the integrated sources. The collection of our federated TBox as well as the referenced TBoxes is available online[6].

As opposed to *DL-Lite*$_\mathcal{A}$ (see Section 2.2), the OWL 2 QL profile, i.e., OWL 2 in general, does not rely on the UNA but provides the explicit object property `owl:sameAs` for expressing that two different IRIs are denoting the same entity. Like already mentioned in Section 4.1, Calvanese et al. [Cal+15] proposed an approach that takes under a set of restrictions individual equality statements into account on query answering but retains the FOL-rewritability. However, as especially in data sources of the LOD cloud this object property is extensively used but for the sake of simplicity and without loss of generality we imposed the UNA for our approaches, we had to modify the dataset, i.e., the ABoxes of the integrated sources, accordingly. Hence, to resolve every `owl:sameAs` relation in accordance to/for the purpose of/in view of Hence, for the purpose of satisfying the UNA all IRIs that are (directly or indirectly) linked by `owl:sameAs` and thus denoting the same entity are amended to the same IRI. Moreover, by using unique attributes (such as ISBN or ISSN) we have additionally detected duplicate representations of an entity and changed the respective identifier to the same IRI in order to gain a higher overlapping of the integrated sources. Finally, since the integrated LOD sources also comprise blank nodes and IRIs that are not well-formed, we have also performed an additional preprocessing of the dataset in order to eliminate any representation of anonymous resources and to fix all syntactic errors. Unfortunately, we are currently not allowed to publish the final dataset of our experimental evaluation due to legal restrictions.

In order to evaluate the implementation of our approaches (written in Java) against our experimental LOD dataset, we have further set up a CentOS 6.7 virtual machine consisting of 6x Intel Xeon CPUs (à 4 cores @ 2.50 GHz) and 174 GB of RAM. Note that the high memory size is especially required for the (naïve) federated querying algorithm (see Definition 4.4) since the results of all query atoms are merged centralized. All subsequent algorithms of our approaches occupy considerably less than the half of the memory.

[6] `https://www.researchgate.net/publication/299852903`

Chapter 8

Federated Inconsistency Detection and Explanation

Concerning research question Q5, in this chapter we discuss the results of applying our approach proposed in Chapter 4 for detecting and explaining inconsistency in federated *DL-Lite*$_{\mathcal{A}}$ KBs to our experimental LOD dataset. Besides analyzing the runtime performance in Section 8.1, we also evaluate the resulting set of generated MISAs (explanations) in Section 8.2. More precisely, in order to answer research question Q6 with respect to inconsistency detection we examine how the generated set of MISAs is effected by integrating an additional data source into the federated KB and further compare the results of the federated settings with the local ones, i.e., the results that are generated by applying our approach to each integrated data source independently. Finally, in Section 8.3 we use an artificially generated dataset to compare our approach with two standard reasoners.

The experimental results of Section 8.3 have already published in [Nol+14]. Moreover, the results of Section 8.1 and Section 8.2 are part of our experimental evaluation described in [Nol+16]. However, in the evaluation of our previous work [Nol+16] we did not consider unary MISAs, i.e., MISAs that result especially from case (iii) (incorrect datatypes) of Definition 4.1, as their resolution is trivial and not crucial in federated settings. For the sake of completeness we now take those MISAs also into account why some of the measured values are different to the values already presented. Moreover, we additionally consider the number of conflicting assertions in order to evaluate the connectivity of the resulting conflict graph.

8.1 Runtime Performance

Based on the federated TBox of our experimental dataset that comprises and extends the semantics of all four integrated data sources our approach generates 496 clash queries, where 74 of which result from value-domain axioms, 8 are due to functionality assertion axioms and 414 stem from negative inclusion axioms. Since

some of the clash queries can be implicitly derived by another clash query and hence are part of its expansion, the number of the first type of clash queries can be reduced from 74 to 67 and from 414 to 281 for the last kind of clash queries.

The subsequent application of the TreeWitness algorithm constituted by Kikot et al. [KKZ12] by using the open-source OBDA framework –ontop–[1] results in a set of expanded clash queries comprising 44,175 CQs.

Afterwards, according to Definition 4.4 in Section 4.2.4 and Algorithm 4.1 in Section 4.3, the complete set of expanded clash queries is evaluated through a naïve federated querying algorithm where each query atom is simply evaluated at each integrated data source and the corresponding answers are merged according to the logical operators in the query. For optimizing the querying runtime the query atoms are evaluated in parallel, where the number of simultaneous threads is limited to 64.

The runtime for the complete inconsistency detection over all four data sources and the generation of appropriate MISAs takes 242.21 min (minutes), where 0.84 min of which are required for the clash query generation and expansion, 170.93 min are taken purely for evaluating the clash query atoms at all data sources, and the subsequent joining including the generation of corresponding MISAs last 70.44 min. Despite a parallelization of the federated querying, the complete runtime strongly depends on the latency of the network and the performance of the machines hosting the data sources.

8.2 Explanation Analysis

Table 8.1 summarizes the characteristics of each data source and depicts the experimental evaluation results of our approach for detecting and explaining inconsistency in a federated *DL-Lite*$_{\mathcal{A}}$ KB. Besides displaying statistics of each data source ($\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_4$) individually, the table depicts two federated settings on which we have applied the implementation of our Algorithm 4.1: DetectInconsistency. The federated KB $\mathcal{K}'_F$ comprises data source $\mathcal{K}_1$, $\mathcal{K}_2$ and $\mathcal{K}_3$, whereas $\mathcal{K}_F$ integrates all four data sources and hence represents the setting that is mainly used for our experimental evaluations (as already in the previous Section 8.1). However, as discussed in Section 5.3, our majority voting-based repair generation approach relies on the assumption that the probability of an assertion for being not valid correlates with the number of MISAs in which the assertion is involved. In order to evaluate this assumption and to analyze the impact of an additional data source on the debugging results, we use the setting of the federated KB $\mathcal{K}'_F$ and compare the debugging results with those of $\mathcal{K}_F$. Moreover, we also compare both federated settings with the local setting where our approach is applied to each data source independently. For this purpose the numbers of data source $\mathcal{K}_1$, $\mathcal{K}_2$ and $\mathcal{K}_3$ and of all data sources are sum up in an additional table row headed with the Σ' and Σ, respectively.

[1] http://ontop.inf.unibz.it

Table 8.1: Results of Federated Inconsistency Detection and Explanation

		$\#\mathcal{A}$	$\#\mathcal{M}$	$\#\mathcal{C}$
Setting	$\mathcal{K}_1$	72,372,256	3,266,765	2,690,206
	$\mathcal{K}_2$	17,765,873	1,096,337	580,106
	$\mathcal{K}_3$	166,320,474	12,016,541	7,993,721
	$\mathcal{K}_4$	27,897,291	27,392	21,765
	Σ'	256,458,603	16,379,643	11,264,033
	Σ	284,355,894	16,407,035	11,285,798
	$\mathcal{K}'_F$	256,458,603	16.605.548	11,358,902
	$\mathcal{K}_F$	284,355,894	18,147,988	12,405,328

The first table column depicts the ABox size (number of triples, i.e., assertions) of each data source ($\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3$ and $\mathcal{K}_4$) respectively of each federated KB $\mathcal{K}'_F$ and $\mathcal{K}_F$. In the second column we can find the number of generated MISAs and in the third column the number of conflicting assertions. As we can see, $\mathcal{K}_3$: RKB Explorer ePrints Open Archives is the largest data source with more than 166 million triples and has also the highest number of local conflicts. By comparing the sum of all local settings (row Σ' and Σ) with the numbers of the federated settings we can observe that the number of MISAs is dominated by local conflicts. However, we can also observe that due to the federation the number of MISAs increases from $\approx$16.3 (Σ') to $\approx$16.6 ($\mathcal{K}'_F$) and from $\approx$16.4 (Σ) to $\approx$18.1 million MISAs ($\mathcal{K}_F$). It is also interesting to see that according to our initial assumption the number of (federated) MISAs is significantly influenced by the integration of an additional data source ($\mathcal{K}'_F$ versus $\mathcal{K}_F$). Furthermore, by taking a look at the number of conflicting assertions ($\#\mathcal{C}$), we can also notice that the more data sources are integrated the higher connected the conflict graph.

A closer look at the set of MISAs for $\mathcal{K}_F$ shows that 1.038 MISAs are unary and hence result from incorrect data types of data values. Besides, $\approx$67.3% (i.e., 12,209,235) of the MISAs result from functionality assertion axioms where 0.5% of them are federated. All 5,937,715 other MISAs are caused by negative inclusion axioms with a rate of 28.3% federated explanations.

Figure 8.1 depicts the set of all federated MISAs and aggregated by the two data sources from which the conflicting assertions stem from. As we can note, every possible combination of data sources results in a set with more than 55,000 MISAs. Even though the both data sources $\mathcal{K}_1$ and $\mathcal{K}_4$ are originally based upon the Digital Bibliography & Library Project (DBLP[2]), an interesting fact is that these

[2] https://dblp.org/

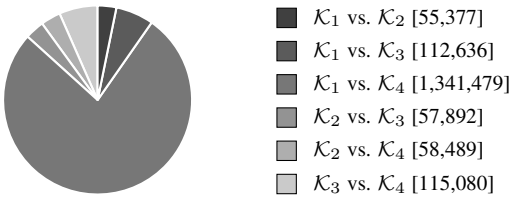

Figure 8.1: Distribution of Federated MISAs

two sources cause 77% of all federated MISAs. One possible reason for this could be a different preprocessing and mapping to distinct TBoxes of the underlying DBLP dataset.

In order to gain some further insights into the set of MISAs generated for $\mathcal{K}_F$, Figure 8.2 illustrates the number of explanations for each axiom causing inconsistency. As already mentioned before, the majority of the explanations are caused by functionality assertion axioms and most of them can be traced back to the axiom $(\mathsf{funct}\ title)$. On the other hand, MISAs that result from negative inclusion axioms are mainly caused by the axiom $ScientificEssay \sqsubseteq \neg Book$.

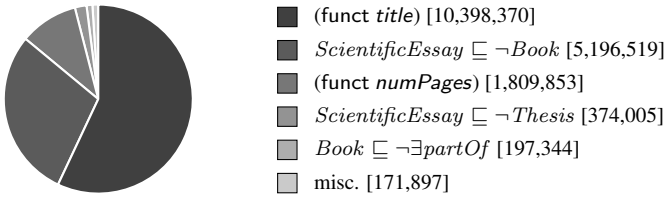

Figure 8.2: Axioms Causing Inconsistency

8.3 Comparison to Related Approaches

In order to further evaluate the performance of our approach we compare the implementation of Algorithm 4.1 with two standard reasoners. On the one hand, we use the reasoning system Pellet[3] [Sir+07] which offers a specific service for computing inconsistency explanations. On the other hand, the second one is HermiT[4] [MSH09; HMW12] that is employed as underlying reasoner by the black box algorithm of the OWL API[5] for computing explanations. Due to the huge size of our experimental LOD dataset we have artificially generated five RDF datasets comprising 500, 5,000, 10,000, 50,000 and 100,000 ABox assertions according to the TBox definition of the running example in our previous work [Nol+14]. For each dataset the assertion are generated randomly such that they comprise a rate

[3] Version 2.3.0
[4] Version 1.3.8
[5] `https://owlcs.github.io/owlapi/`, Version 3.4.3

of about 2% of conflicting assertions and every possible clash type mentioned in Section 4.2.1 may occur. The collection of these datasets and the corresponding TBox is available online[6].

Since both of the standard reasoners are only applicable to non-distributed datasets, we have run our implementation of Algorithm 4.1 in both, a local and a federated setting. While for the local setting the dataset is completely loaded into a central repository (implemented as a local Virtuoso instance), for the federated setting all ABox assertions are randomly distributed over four Virtuoso instances. In contrast to the hardware setting described in Chapter 7, we have run our implementation and the two standard reasoners on an Intel Core i7 CPU quad core CPU @ 2.50 GHz and an assigned memory size of 4 GB. The results of this evaluation are illustrated in Table 8.2, where $\mathcal{K}_L$ denotes the local setting and $\mathcal{K}_F$ the distributed setting of our approach.

Table 8.2: Comparison with Standard Reasoner

		$\#\mathcal{A}$				
		500	5,000	10,000	50,000	100,000
Pellet	runtime (ms)	1,713,901	>18,000,000	–	–	–
	$\#\mathcal{M}$	20 (13)	–	–	–	–
HermiT	runtime (ms)	[Error]	–	–	–	–
	$\#\mathcal{M}$	–	–	–	–	–
$\mathcal{K}_L$	runtime (ms)	359	563	718	1,313	2,171
	$\#\mathcal{M}$	13	113	191	1,050	2,121
$\mathcal{K}_F$	runtime (ms)	29,360	234,878	464,973	2,227,466	4,541,140
	$\#\mathcal{M}$	13	113	191	1,050	2,121

What attracts attention is that the runtimes of the local and the federated setting of our approach differ significiantly. Moreover, the runtimes of the federated setting increase linearly with respect to the size of the ABox, while the number of assertions only has a minor impact in the local setting. Besides the latency in the network, the main reason for this is that our early implementation described in [Nol+14] was based on ARQ[7], a query engine for Apache Jena. However, in ARQ the evaluation of federated queries was handled very inefficently and that is why we have introduced and implemented our own (naïve) federated querying algorithm (see Definition 4.4) in the subsequent work [Nol+16].

More surprising, however, are the performance results of Pellet and HermiT. Even for the smallest dataset the reasoning system Pellet takes considerably more time for computing all explanations than our approach in both settings. For the

[6] `https://www.researchgate.net/publication/263051841`

[7] `https://jena.apache.org/documentation/query/`, Version 2.11.1

dataset with $\#\mathcal{A} = 5{,}000$ Pellet did not end up with any result after five hours, why we stopped the execution after that time. On the contrary to that, HermiT did not produce any result for the smallest dataset but ends up with an `OutOfMemoryError`, despite of an assigned memory of 4 GB. Unfortunately, we could not find an explanation for that and were not able to solve that problem either.

Given for the dataset with $\#\mathcal{A} = 500$ the result of Pellet and the result our approach (which is consequentially the same in both settings), we can observe that the set $\mathcal{M}$ of explanations generated by Pellet comprises seven more elements. By manually analyzing both explanation sets it can be ascertain that both algorithms generate the same explanations. However, the result of Pellet comprises some additional explanations that are not minimal (in particular for conflicting attribute assertions) why Pellet produces a higher number of explanations.

Chapter 9

Repair Plan Generation

Based on the previous chapter we discuss the results of applying our approach for generating a repair plan as proposed in Chapter 5 to the given complete set $\mathcal{M}$ of MISAs for our experimental LOD dataset and hence provide an answer to research question Q5 with respect to repair generation. After briefly analyzing the runtime performance in Section 9.1, we are evaluating in Section 9.2 the repair generated by Algorithm 5.4 based on majority voting and the learned repair computed by Algorithm 5.5 based on signature accuracies. As already in the previous section we again address research question Q6 by evaluating the effects on the results of integrating an additional data source into the federated KB and further compare the generated repairs of the federated settings with the results that can be achieved by considering each data source separately. In the last Section 9.3 we eventually evaluate the quality of the generated repairs.

We have already published some experimental results for our approach of repair plan generation in [Nol+16]. However, the following results are not exactly the same as unary MISAs are now completely taken into consideration and we have refined the calculation of signature accuracies in [Nol+17]. Moreover, as described in Section 5.3, we have extended our approach by a repair minimization (Algorithm 5.2) and the consideration of entailment relations (Algorithm 5.3) and are therefore now also evaluating the effect of these algorithms on the generated repair of Algorithm 5.1.

9.1 Runtime Performance

For our federated KB $\mathcal{K}_F$ the resulting conflict graph consists of 12,405,328 vertices, 18,147,988 conflicting edges, 1,429,931 entailment relations and 31,256 equivalence relations (bidirectional entailment relations). In order to generate a majority voting-based repair the while loop of Algorithm 5.1 repeats 413 times until no more MISAs can be resolved and takes 4.87 min. The subsequent application of Algorithm 5.5) takes 3.47 min for the repair generation based on signature accuracies. For further optimizing the runtime of the repair generation, the conflict graph

could be divided into independent subgraphs such that both, Algorithm 5.1 and Algorithm 5.5 can be applied to each subgraph in parallel. However, since we are more interested in evaluating the generated repair than in optimizing the runtime, in the following we will focus solely on the analysis of the repair.

9.2 Repair Analysis

By extending Table 8.1, Table 9.1 depicts the size of the generated repairs for each setting, i.e., the repair $\mathcal{R}$ resulting from Algorithm 5.4 and the repair $\mathcal{R}'$ that is generated based on signature accuracies and representing the intermediate result at Line 15 of Algorithm 5.5. The values in parenthesis of these two columns represent the numbers of MISAs that are resolved by the respective repair. The last column shows the rate of MISAs that are not resolved by $\mathcal{R} \cup \mathcal{R}'$ and hence are addressed by the random repair generated by Algorithm 5.5 (Line 15 et seqq.).

Table 9.1: Results of Repair Generation[1]

Setting	$\#\mathcal{A}$	$\#\mathcal{M}$	$\#\mathcal{C}$	$\#\mathcal{R}$	$\#\mathcal{R}'$	remaining MISA rate
$\mathcal{K}_1$	72,372,256	3,266,765	2,690,206	46,128 (291,025)	1,187,461 (1,188,115)	54.72%
$\mathcal{K}_2$	17,765,873	1,096,337	580,106	4,654 (15,525)	246,289 (247,180)	76.04%
$\mathcal{K}_3$	166,320,474	12,016,541	7,993,721	1,024,564 (2,057,957)	2,513,198 (5,258,733)	39.11%
$\mathcal{K}_4$	27,897,291	27,392	21,765	1,409 (27,388)	4 (4)	0%
Σ'	256,458,603	16,379,643	11,264,033	1,075,346 (2,364,507)	3,946,948 (6,694,028)	44.69%
Σ	284,355,894	16,407,035	11,285,798	1,076,755 (2,391,895)	3,946,952 (6,694,032)	44.62%
$\mathcal{K}'_F$	256,458,603	16,605,548	11,358,902	1,109,674 (4,770,707)	3,928,942 (5,538,586)	37.92%
$\mathcal{K}_F$	284,355,894	18,147,988	12,405,328	1,994,174 (7,170,124)	3,073,489 (4,681,609)	34.69%

As we can observe, especially in column $\#\mathcal{R}$ the number of resolved MISAs is significantly higher than the size of the repair and hence indicates again that

[1] Note that the values in column $\#\mathcal{R}'$ differ from those published in [Nol+16] due to a measuring error in our previous work.

the conflict graph is highly connected. Moreover, one of the most considerable results is that we could ascertain that in each of our settings the generated repair of Algorithm 5.1 and the repair resulting from Algorithm 5.4 are equivalent. This confirms our claim in Section 5.3 that in theory Algorithm 5.2 and Algorithm 5.3, and hence the subsequent application of Algorithm 5.1 are necessary but (at least for our experimental LOD dataset) these algorithms do not have any effect on the generated repair and hence address (mainly) artificial cases that usually do not occur in practice. It is also interesting to see that the rate of remaining MISAs is significantly higher in the local setting than in both federated settings. Due to the additional data source in $\mathcal{K}_F$ the number of MISAs resolved by $\mathcal{R}$ is increased by 2,399,417 but for $\mathcal{R}'$ the number of resolved MISAs is reduced by 856,977. Hence, in $\mathcal{K}_F$ more MISAs are resolvable by our majority voting-based approach while in $\mathcal{K}'_F$ those MISAs could not be resolved or are resolved merely based on signature accuracies. As a consequence, in $\mathcal{K}_F$ the rate of MISAs not resolved by $\mathcal{R} \cup \mathcal{R}'$ is further decreased from 37.92% to 34.69%. Both effects support our assumption that the integration an additional data source into a federated KB has a positive impact on the number of MISAs not resolved randomly.

In order to highlight the impact of our federated debugging approach, we compare in Table 9.2 for each data source $\mathcal{K}_n$ the number of MISAs resolved in the local setting with the number of local MISAs that are resolved by the repairs $\mathcal{R}$ and $\mathcal{R}'$ generated for both of our federated settings. Besides the percentage differences of the resolved MISAs (Column "$\Delta\mathcal{M}$ res. by $\mathcal{R}$" and "$\Delta\mathcal{M}$ res. by $\mathcal{R}'$"), the last column of each federated setting $\mathcal{K}'_F$ and $\mathcal{K}_F$ depicts the rate of local MISAs that are not resolved by the repairs $\mathcal{R} \cup \mathcal{R}'$ of the respective federated setting. If, for instance, we consider data source $\mathcal{K}_1$ the number of local MISAs resolved by the repair $\mathcal{R}$ generated via majority voting is increased by 341.32% in the federated setting $\mathcal{K}'_F$. On the contrary, by applying the repair $\mathcal{R}$ of $\mathcal{K}_F$ to the set of local MISAs in $\mathcal{K}_1$ the number of resolved MISAs is further increased by 637.72%. By summing up the (local) results for all data sources in the last row, we can see that

Table 9.2: Impact of Federated Knowledge Base Debugging

	$\mathcal{K}'_F$			$\mathcal{K}_F$		
	$\Delta\mathcal{M}$ res. by $\mathcal{R}$	$\Delta\mathcal{M}$ res. by $\mathcal{R}'$	rem. local MISA rate	$\Delta\mathcal{M}$ res. by $\mathcal{R}$	$\Delta\mathcal{M}$ res. by $\mathcal{R}'$	rem. local MISA rate
$\mathcal{K}_1$	+341.32%	-0.14%	24.37%	+637.72%	-72.74%	24.37%
$\mathcal{K}_2$	+158.98%	-0.14%	73.82%	+173.84%	-1.08%	73.82%
$\mathcal{K}_3$	+57.08%	-22.17%	39.04%	+57.09%	-22.17%	39.04%
$\mathcal{K}_4$	–	–	–	+0.01%	-100.0%	0%
Σ	+92.74%	-17.45%	38.44%	+127.84%	-30.37%	38.38%

due to the federated setting $\mathcal{K}_F$ the number of local MISAs resolved by repairs $\mathcal{R}$ is expanded by a total of 127.84% and further results in an decrease of 30.37% for MISAs addressed by $\mathcal{R}'$. As a consequence, this in turn yield a reduction of remaining local MISAs from originally 44.62% (see Table 9.1) to 38.38%, which facilitates the achievement of a significantly higher recall rate in identifying wrong assertions. Moreover, while the number of remaining MISAs is not changed from $\mathcal{K}'_F$ to $\mathcal{K}_F$, the additional data source $\mathcal{K}_4$ in $\mathcal{K}_F$ further causes that for each data source significantly more local MISAs can be resolved via majority voting instead of solely based on signature accuracies. Hence, we can conclude that the effect of both, the federated setting as well as the additional data source $\mathcal{K}_4$ in $\mathcal{K}_F$ is clearly in evidence.

Taking a look at the majority voting (Algorithm 5.1) applied to the federated setting $\mathcal{K}_F$, the highest cardinality value that is found amounts to 18,189. This extraordinary high value is caused by the fact that in data source $\mathcal{K}_1$ and $\mathcal{K}_4$ a bunch of articles are assigned to the journal series "*Bioinformatics*"[2], but in $\mathcal{K}_3$ the journal series is wrongly defined as an article.

Based on the results of our repair generation via majority voting the signature accuracies are calculated according to Definition 5.3. The distribution of the resulting signature accuracies for our federated setting $\mathcal{K}_F$ is depicted in Figure 9.1.

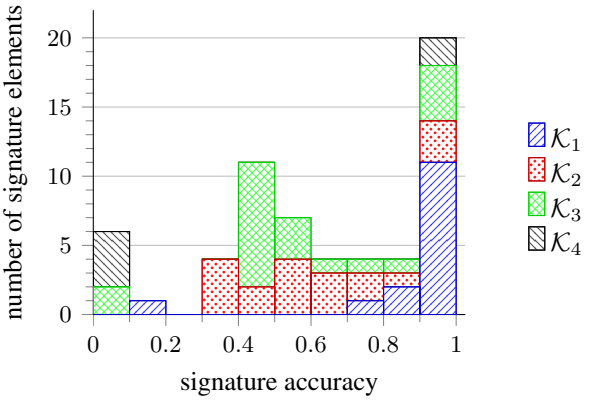

Figure 9.1: Distribution of Signature Accuracies

For a more detailed analysis Table 9.3 depicts the top 10 of lowest signature accuracies. Especially the very low accuracies of the first seven signature elements suggest the assumption that assertions on these elements are probably misapplied in the respective data source. A glance into those sources confirms this assumption. All assertions on the attribute *description* in $\mathcal{K}_3$ as well as the attributes *description*, *month−of* and *year−of* in $\mathcal{K}_4$ comprise data values with incorrect data types. In contrast, the data types of the values in assertions on attribute *title* of $\mathcal{K}_4$ are indeed correct, the data values, however, do not correspond to the actual titles but rather to some kind of internal identifier, why the low signature accuracy is also justified.

[2] `http://bioinformatics.oxfordjournals.org/`

Table 9.3: Top 10 of Lowest Signature Accuracies

signature accuracy	data source	$\sigma \in \Sigma_F$
0.001	$\mathcal{K}_3$	*http://purl.org/dc/terms/description*
0.001	$\mathcal{K}_4$	*http://purl.org/dc/elements/1.1/title*
0.001	$\mathcal{K}_4$	*http://purl.org/dc/elements/1.1/description*
0.001	$\mathcal{K}_4$	*http://www.aktors.org/ontology/support#month-of*
0.001	$\mathcal{K}_4$	*http://www.aktors.org/ontology/support#year-of*
0.0556	$\mathcal{K}_3$	*http://purl.org/ontology/bibo/volume*
0.1433	$\mathcal{K}_1$	*http://swrc.ontoware.org/ontology#volume*
0.3220	$\mathcal{K}_2$	*http://swrc.ontoware.org/ontology#Unpublished*
0.3513	$\mathcal{K}_2$	*http://swrc.ontoware.org/ontology#TechnicalReport*
0.3781	$\mathcal{K}_2$	*http://swrc.ontoware.org/ontology#Booklet*

Besides, if we examine assertions on attribute *volume* in data source $\mathcal{K}_1$ and $\mathcal{K}_3$ (row 6 and 7) we can observe that the attribute is used in these sources for articles published in collections, but the domain of *volume* are collections like proceedings, journals or books. Moreover, the low signature accuracies for the attribute *volume* are also reflected by the fact that the axiom $ScientificEssay \sqsubseteq \neg Book$ is the second of the top 5 axioms causing inconsistency (as depicted in Figure 8.2) since $\exists$*volume* is part of the expansion of $Book$.

Finally, we have analyzed the set of remaining MISAs not resolved by $\mathcal{R} \cup \mathcal{R}'$ and hence addressed by the random repair generated by Algorithm 5.5. All of those MISAs are only local ones and are exclusively caused by functionality assertion axioms. As depicted in Figure 9.2, about 85.6% of the MISAs result form the axiom (funct *title*), up to 14.3% from the axiom (funct *numPages*) and a negligible quantity of MISAs are caused by the axiom (funct *issn*).

Figure 9.2: Axioms of Remaining MISAs

9.3 Qualitative Analysis

As the repair $\mathcal{R}$ generated via majority voting seems more reliable than the repair $\mathcal{R}'$ that is based on signature accuracies, we expect that the federated setting $\mathcal{K}'_F$ as well as the additional data source in $\mathcal{K}_F$ do not have only a positive impact on the quantity (as already shown in Table 9.1 and 9.2) but also on the quality of the generated repairs. In order to evaluate the repair quality 100 resolved MISAs are randomly selected for each repair ($\mathcal{R}$ and $\mathcal{R}'$) in both federated settings ($\mathcal{K}'_F$ and $\mathcal{K}_F$) and are manually evaluated by three persons. The resulting precision values of this evaluation are presented in Table 9.4. The first two column shows the percentage of MISAs resolved correctly or incorrectly by the corresponding repair. If, however, an URI of an assertion is not accessible or at least two persons did not come to the same decision, the resolution of that MISA is annotated as uncertain.

Table 9.4: Quality of Repairs

		correct	incorrect	uncertain
$\mathcal{R}$	$\mathcal{K}'_F$	93%	2%	5%
	$\mathcal{K}_F$	97%	0%	3%
$\mathcal{R}'$	$\mathcal{K}'_F$	86%	11%	3%
	$\mathcal{K}_F$	81%	9%	10%

Overall, the evaluation indicates a high precision of our approach for repair generation and substantiate that our algorithms are based up on an eligible heuristics. Moreover, the measured precision scores also confirms that the repair generated by our majority voting approach is a valid basis to gain again a high precision for the repair based on signature accuracies. By comparing the results of the federated settings we can observe on the one hand an increase from 93% in $\mathcal{K}'_F$ to 97% in $\mathcal{K}_F$ with respect to $\mathcal{R}$. On the other hand, for both repairs we can also find a marginal drop of 2% in the rate of assertions annotated to be incorrect. Despite the fact that the analyzed samples are comparatively small, the findings indicate a positive impact of integrating an additional data source. Besides, the evaluation results presented in this and the previous section also clearly show the positive impact on recall of the federated settings.

Chapter 10

Fine-grained Trust Assessment

In this chapter we eventually discuss the results of applying our proposed approach for fine-grained trust assessment of Chapter 6 on our experimental LOD dataset and hence complete our answer to research question Q5. We start in Section 10.1 with briefly analyzing the runtime performance of our implementation for assessing trust values at different levels of granularity. Moreover, we examine the convergence of the assertion trusts determined by the sampling in Algorithm 6.1 and continue in Section 10.2 with an analysis of the assessed trusts values. Finally, by evaluating the quality of the calculated assertions we close this chapter with Section 10.3.

The evaluation already described in [Nol+17] is similar to evaluation method of this chapter. However, the results that we discuss in the following sections are different from [Nol+17] since in the initial version of our approach we did not consider entailment relations between conflicting assertions.

10.1 Runtime and Convergence Performance

For optimizing the runtime of the assertion trusts assessment via sampling, the conflict graph is divided into independent subgraphs that are handled by Algorithm 6.1 in parallel, where the number of simultaneous threads is limited to 256. Figure 10.1 depicts the runtime as well as the corresponding convergence of the assertion trusts resulting from an increasing number of samples K with a step size of 200. As we can see, after a short burn-in period of 400 samples, in which the desired distribution may not be exactly represented by the state of the Markov network graph, the runtime increases linearly with the number of samples. Moreover, as K increases the sampling converges to the desired posterior probability distribution such that after 10,000 samplings the maximal deviation of an assertion trust value compared to the trust value in the previous sample is 0.019.

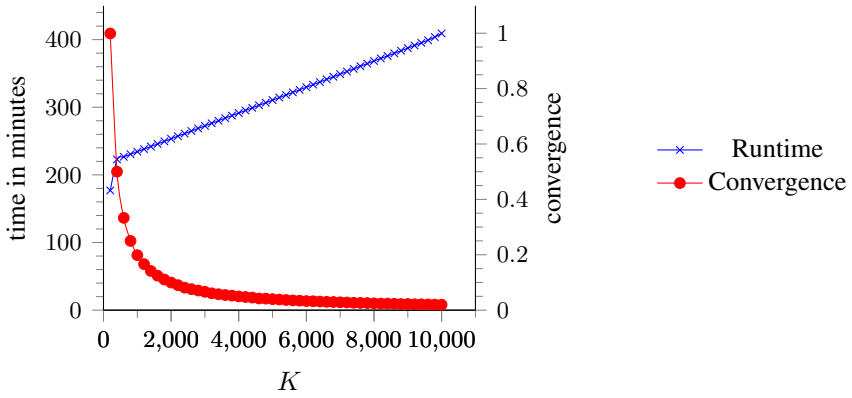

Figure 10.1: Runtime and Convergence Performance

10.2 Accuracy and Trust Value Analysis

Given the conflict graph for our running example, the corresponding Markov network graph is initialized according to the repair generated by our approach proposed in Chapter 5. Moreover, the signature accuracies (see Definition 5.3) are used as (marginal) prior probabilities within the sampling by Algorithm 6.1 in order to assess trust values for all conflicting assertions. The subsequent application of our approach for fine-grained trust assessment proposed in Chapter 6 yields the distribution of assertion trusts depicted in Figure 10.2. Moreover, the resulting distribution of signature trusts as well as the individual data source trusts are shown in Figure 10.3 and Figure 10.4, respectively.

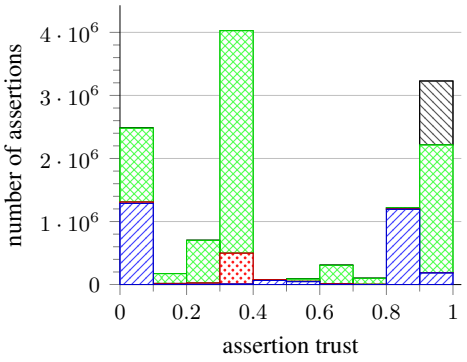

Figure 10.2: Assertion Trusts

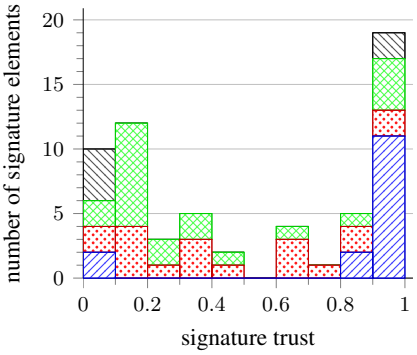

Figure 10.3: Signature Trusts

$\mathcal{K}_1 = 0.8889$ $\mathcal{K}_2 = 0.4843$ $\mathcal{K}_3 = 0.5087$ $\mathcal{K}_4 = 0.9978$

Figure 10.4: Data Source Trusts

If, for example, we consider the calculated trust values relating to data source $\mathcal{K}_4$, we can see in Figure 10.2 that the assessed trusts for most of the assertions which are involved in any MISA are > 0.9 and hence result in a data source trust close to 1.0 (see Figure 10.4). However, in Figure 10.3 we can notice that for $\mathcal{K}_4$ there exist two signature elements (i.e., the role $article\text{-}of\text{-}journal$ and the concept $Journal$) with an assessed trust value > 0.9 but four elements (i.e., the attributes *title*, *description*, *month−of* and *year−of* as already listed in Table 9.3) with a signature trust of < 0.1. Since in $\mathcal{K}_4$ there exist only a negligible number of assertions on those attributes the data source trust is not affected by the low trust values for these signature elements. As a consequence, it can therefore be concluded that we can basically rely on assertions of $\mathcal{K}_4$ but explicitly not with regard to assertions on the attributes *title*, *description*, *month−of* and *year−of*. On the contrary, with an assessed trust value of 0.4843, $\mathcal{K}_2$ represents the least reliable data source in $\mathcal{K}_F$. However, since there exists four signature elements, i.e., the concept $Misc$ and the attributes *edition*, *chapter*, and *series*, of $\mathcal{K}_2$ with a trust value > 0.8 (see Figure 10.3), we can rely on assertions of $\mathcal{K}_2$ on these signature elements even the data source is rather considered to be not trustworthy.

In order to gain a deeper insight into the assessed trust values, we further examine the trust values that are assessed to assertions which are part of the majority voting-based repair $\mathcal{R}$ generated for our federated setting $\mathcal{K}_F$. Figure 10.5 depicts the trust value distribution of repair assertions, where it should be noted that for purposes of presentation the y-axis is scaled logarithmically.

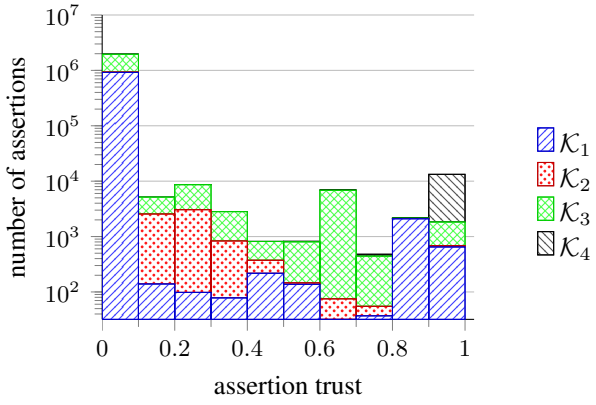

Figure 10.5: Repair Assertion Trusts

As indicated by the figure, merely for a small fraction, i.e., 23,649 (1.18%) of the assertions the assessed trust is ≥ 0.5. This reflects the high precision (97%) of repair $\mathcal{R}$ with respect to $\mathcal{K}_F$ as already shown in Table 9.4 and hence verifies that our approach for an automated fine-grained trust assessment produces good results.

In addition to the trust values of repair assertions, let us also compare the signature accuracies used as (marginal) prior probabilities (Figure 9.1) with the assessed signature trust values (Figure 10.3). As we can see from these figures, 22 signature

Table 10.1: Top 5 of Signature Elements with High Deviation

data source	$\sigma \in \Sigma$	**Signature Accuracy**	**Signature Trust**
$\mathcal{K}_1$	*http://swrc.ontoware.org/ontology#Proceedings*	0.7634	0.0384
$\mathcal{K}_2$	*http://swrc.ontoware.org/ontology#Proceedings*	0.5768	0.1669
$\mathcal{K}_2$	*http://swrc.ontoware.org/ontology#PhDThesis*	0.4862	0.1070
$\mathcal{K}_3$	*http://purl.org/ontology/bibo/EditedBook*	0.5912	0.2241
$\mathcal{K}_3$	*http://purl.org/ontology/bibo/Book*	0.5345	0.1855

elements have a signature accuracy of ≤ 0.5 whereas this is the case with regard to the trust values for 32 signature elements. The top 5 signature elements with the highest deviation between the signature accuracy and the signature trust are listed in Table 10.1. By analyzing, for instance, conflicting assertions in $\mathcal{K}_F$ on signature element $Proceedings$ of data source $\mathcal{K}_1$ and $\mathcal{K}_2$ it turns out that most of the corresponding MISAs are not resolved by the majority voting-based repair $\mathcal{R}$. Moreover, since the accuracy of signature element $Proceedings$ is frequently less than the signature accuracy corresponding to the conflicting assertions, this in turn results in low trust values for assertions on $Proceedings$. Due to the fact that the signature elements of conflicting assertions in MISAs resolved by $\mathcal{R}$ differ in many cases from the signature elements of assertions that are part of MISAs not addressed by $\mathcal{R}$, similar observations can be made for the remaining rows of Table 10.1.

10.3 Qualitative Analysis of Trust Values

For evaluating the quality of assessed assertion trusts with respect to our federated setting $\mathcal{K}_F$, we randomly selected 100 assertions out of the majority voting-based repair $\mathcal{R}$ with an assessed trust value ≥ 0.8. Despite the established high precision of $\mathcal{R}$ (see Table 9.4), this sample represents a set of assertions that seems likely to be mistakenly selected to be part of the repair.

Similar to our qualitative analysis of the repairs described in Section 9.3, the selected sample of assertions is again manually evaluated by three persons. The qualitative analysis shows that 80% of the assertions are annotated to be correct and only 17% are identified as wrong assertions. Hence, this precision scores indicate a high precision of the assessed trust values. Besides, this precision scores also substantiate that the calculation of signature accuracy values used as prior probability is a valid basis and that our approach for an automated fine-grained trust assessment is eligible.

Part IV

Conclusion

Chapter 11

Summary

We have started this Thesis in Chapter 1 with a discussion about the motivation and the covered research questions. By introducing in the subsequent Chapter 2 the foundations of Description Logics and Semantic Web technologies we have provided a formal framework required to describe the addressed research problem of federated KB debugging as well as the proposed approaches. Chapter 3 comprises a clarification of the motivating context for this work followed by a discussion about the general problem of ontology-based information integration. This includes a formal definition of federated KBs as well as a specification of the addressed research problem. At the end of this chapter a running example is introduced.

By reflecting in the following on our given answers to the six research questions stated in Section 1.2, we summarize the essential part, i.e., Part II and Part III, of this thesis.

Research question Q1 asks for a formal description of the problem of inconsistency management in federated KBs and its peculiarities. Related to this question, in Section 3.2 we we have discussed the general problem of OBII, formally defined federated KBs and clarified the research problem addressed by this work. Moreover, by further analyze in Chapter 4 the peculiarities of federated KBs we have argued that reasoning over a federated KB is a challenging problem. This is particularly due to the loosely coupled network of data sources in context of OBII and the resulting large amounts of extensional knowledge that have to be handled. To tackle this problem we have identified FOL-rewritability as an appropriate property of the underlying DL language and correspondingly justified why *DL-Lite*$_{\mathcal{A}}$ is sufficient for our purpose.

The call for a convenient, efficient, and eligible debugging process for federated KBs is made by research question Q2. By proposing in Chapter 4 an approach of inconsistency detection in federated *DL-Lite*$_{\mathcal{A}}$ KBs in consideration of the identified peculiarities and requirements we gave an answer to Q2 with respect to the first part of the debugging process. Our approach consists of a generation of clash queries and its federated evaluation, followed by a generation of the correspond-

ing MISAs composed of source-related ABox assertions. In order to address the second part of the debugging process we proposed in Chapter 5 an approach for resolving inconsistency in federated *DL-Lite*$_\mathcal{A}$ KBs. Based on the representation of the identified conflicts and their relationships as a conflict graph the approach comprises the application of a majority voting scheme. Moreover, the approach also includes a subsequent calculation and application of signature accuracies in order to generate an appropriate repair. By exploiting explicit but also implicit redundancies caused by federating different KBs during the verification of conflicting assertions we were able to show that the debugging process does indeed benefit from the characteristics of a federated KB with respect to the number of identified conflicts as well as the amount and validity of resolved MISAs.

Research question Q3 considers the feasibility of assessing the trustworthiness of individual assertions with respect to certain data sources based on the debugging results. We have answered this question by providing in Chapter 6 an alternative strategy for handling inconsistency in federated knowledge bases. More specifically, in the proposed approach we consider the conflict graph as a Markov network graph in which the statistical evidence (i.e., the signature accuracies) gathered by the repair generation via majority voting are used as prior probabilities. This facilitates the application of Gibbs sampling for approximating the trust value of each conflicting assertion.

Furthermore, based on these trust values of conflicting assertions we could also assess adequate trust values at the level of signature elements as well as for each individual data source. Thus, by providing a fully automated approach for assessing a fine-grained trust assessment at different levels of granularity we have further provided an answer to research question Q4, that is asking for an automated transformation of a conventional (inconsistent) federated KB into a probabilistic one.

In Part III of this thesis we have discussed the results of our experimental evaluation of the proposed approaches against a set of large distributed LOD sources out of the domain of library science. By setting up different federated settings we could empirically evaluate the scalability and the runtime of our proposed approaches as well as the quality of the corresponding results which is asked by research question Q5. More precisely, in our largest federated *DL-Lite*$_\mathcal{A}$ KB with almost 285 million assertions more than 18 million clashes are detected with a runtime of approximately four hours, where nearly 3 hours are required purely for the evaluation of clash queries. The subsequent repair generation based on majority voting and signature accuracies took 8.34 minutes. The runtime for the fine-grained trust assessment strongly depends on the number of samplings and amounts 6.8 hours for sampling each of the 12.4 million conflicting assertions 10.000 times. In respect to the quality of the generated repair that consists of nearly two million assertions we could measure a precision of up to 97%. Moreover, by evaluating assertions that are part of the repair but an assigned trust value of ≥ 0.8, we could also measure a high precision of 80% for the results of our fine-grained trust assessment approach.

Besides answering research question Q5 we have further addressed research question Q6 which considers the impacts of an additional data sources integrated into a federated KB. For our experimental dataset it could be shown that by integrating an additional data source the number of detected clashes is increased by 1.5 million and the resulting conflict graph also becomes higher connected. Moreover, due to the additional source the rate of remaining MISAs not resolved by the repair is decreased by 3.23% and the precision of the repair generated via majority voting is improved by 4%. Hence, we could show that the integration of an additional data source does not have only a positive impact on the quantity but also on the quality of the generated repairs, which again positively affects the fineness of the assessed trust values.

Chapter 12

Discussion

In this chapter we end this thesis with some closing remarks and a discussion of some potential directions of future work. Overall, we have provided an efficient debugging process for inconsistent federated *DL-Lite*$_\mathcal{A}$ KBs and further proposed an alternative approach to the resolution of conflicts by transforming an inconsistent federated *DL-Lite*$_\mathcal{A}$ KB into a probabilistic one. But indeed, LOD sources are so far primarily used to supplement existing data with some additional information and thus are commonly not considered as full-featured KBs why for this case of application a verification of consistency is not a crucial requirement. However, following the vision of the Semantic Web the federation of distributed KBs and hence the handling of inconsistency in federated KBs becomes essential. Moreover, this will also become more important if semantic technologies are used more frequently in enterprises.

Besides, the results of this thesis can also be transferred to federated databases since the expressivity of *DL-Lite*$_\mathcal{A}$ and of relational databases are equivalent. As a consequence, OBDA systems integrating more than one database could be expanded such that the information about the originating source is preserved and could be reused in any subsequent reasoning task.

Due to the high quality values measured for the generated repairs and assessed trust values with respect to the real-world dataset of our evaluation, an application of the proposed approaches to practical scenarios is indeed conceivable. However, as the debugging results may cause that some data will just be ignored or even get lost, a fully automated application to scenarios with critical data such as for example in the business environment or the public sector is not advisable. Hence, in such cases a manual review of the debugging results is essential.

Moreover, the evaluated runtimes of our algorithms are contradicting an application on data sources with dynamic content or a debugging of federated KBs on demand. In practical scenarios, however, with rather static data the performance of our approaches is indeed sufficient.

As we have used a representative and huge real-world dataset for our evaluation, it is to be expected that similar good results can be achieved for other datasets.

Obviously, an additional evaluation with a further dataset would support this conclusion. However, as the data preprocessing (i.e., ontology matching, data interlinking, etc.) for the federation of some data sources is not trivial and a very time consuming process, for this work we have waived an evaluation of our approaches against another dataset.

Within the scope of this thesis we have focused on the lightweight DL language *DL-Lite*$_{\mathcal{A}}$ but did not study the problem of federated KB debugging with respect to more expressible DLs. However, the expressivity of *DL-Lite*$_{\mathcal{A}}$ is sufficient in order to perform a complete reasoning (like inconsistency detection) for numerous practical scenarios. Moreover, the approaches of this thesis can be adapted to other DLs, provided that the language is FOL-rewritable.

For this work, we have assumed that the TBox of the federated KB is free from any modeling errors why our proposed approaches for repair generation and trust assessment are only based on the definition of MISAs. However, by extending the approaches with respect to the definition of MISs some of the debugging results could be used for detecting modeling errors. As a consequence, a combination of our approaches with works for identifying and repairing incoherent alignments (such as [Mei11]) or with ontology matching and data interlinking approaches could be mutually beneficial.

What is also left to some future work is the consideration of KB evolution. More precisely, given the case that some of the integrated sources of a federated KB are modified, the changes may affect the previously generated debugging results and should therefore be taken into consideration accordingly.

Bibliography

[ABC99] Marcelo Arenas, Leopoldo Bertossi, and Jan Chomicki. "Consistent Query Answers in Inconsistent Databases". In: *Proceedings of the 18th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*. Vol. 99. Philadelphia, PA, USA: ACM, 1999, pp. 68–79.

[AG07] Donovan Artz and Yolanda Gil. "A Survey of Trust in Computer Science and the Semantic Web". In: *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 5.2 (2007), pp. 58–71.

[Ahm+15] Shqiponja Ahmetaj, Wolfgang Fischl, Reinhard Pichler, Mantas Šimkus, and Sebastian Skritek. "Towards Reconciling SPARQL and Certain Answers". In: *Proceedings of the 24th International Conference on World Wide Web (WWW 2015)*. WWW '15. Florence, Italy: ACM, 2015, pp. 23–33.

[Art+09] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. "The *DL-Lite* Family and Relations". In: *Journal of Artificial Intelligence Research* 36.1 (2009), pp. 1–69.

[Aue+07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. "DBpedia: A Nucleus for a Web of Open Data". In: *The Semantic Web – Proceedings of the 6th International Semantic Web Conference (ISWC 2007) and the 2nd Asian Semantic Web Conference (ASWC 2007)*. Busan, Korea: Springer Berlin Heidelberg, 2007, pp. 722–735.

[Baa+10] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications (Second Edition)*. Cambridge University Press, 2010. ISBN: 9780521150118.

[BB13] Salem Benferhat and Zied Bouraoui. "Possibilistic *DL-Lite*". In: *Proceedings of the 7th International Conference on Scalable Uncertainty Management (SUM 2013)*. Washington, DC, USA: Springer Berlin Heidelberg, 2013, pp. 346–359.

[BBL05] Franz Baader, Sebastian Brandt, and Carsten Lutz. "Pushing the $\mathcal{EL}$ Envelope". In: *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*. Edinburgh, Scotland: Morgan Kaufmann, 2005, pp. 364–369.

[BBL08] Franz Baader, Sebastian Brandt, and Carsten Lutz. "Pushing the EL Envelope Further". In: *Proceedings of the 4th International Workshop on OWL: Experiences and Directions (OWLED 2008 DC)*. Vol. 4. Washington, DC, USA: CEUR Electronic Workshop Proceedings, 2008.

[BCG07] Jean-Yves Béziau, Walter A. Carnielli, and Dov M. Gabbay. *Handbook of Paraconsistency*. College Publications, 2007.

[Ber06] Leopoldo Bertossi. "Consistent Query Answering in Databases". In: *ACM SIGMOD Record* 35.2 (2006), pp. 68–76.

[BHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. "The Semantic Web". In: *Scientific American* 284.5 (2001), pp. 34–43.

[BO15] Meghyn Bienvenu and Magdalena Ortiz. "Ontology-Mediated Query Answering with Data-Tractable Description Logics". In: *Reasoning Web. Web Logic Rules: Tutorial Lectures of the 11th International Summer School 2015 (RW 2015)*. Berlin, Germany: Springer International Publishing, 2015, pp. 218–307.

[Bon+11] Piero A. Bonatti, Aidan Hogan, Axel Polleres, and Luigi Sauro. "Robust and Scalable Linked Data Reasoning Incorporating Provenance and Trust Annotations". In: *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 9.2 (2011), pp. 165–201.

[Bor96] Alex Borgida. "On the Relative Expressiveness of Description Logics and Predicate Logics". In: *Artificial intelligence* 82.1–2 (1996), pp. 353–367.

[Bui+13] Carlos Buil-Aranda, Olivier Corby, Souripriya Das, Lee Feigenbaum, Paula Gearon, Birte Glimm, Steve Harris, Sandro Hawke, Ivan Herman, Nicholas Humfrey, Nico Michaelis, Chimezie Ogbuji, Matthew Perry, Alexandre Passant, Axel Polleres, Eric Prud'hommeaux, Andy Seaborne, and Gregory Todd Williams. *SPARQL 1.1 Overview*. W3C Recommendation. W3C, Mar. 2013. URL: `https://www.w3.org/TR/2013/REC-sparql11-overview-20130321/`.

[Cal+07a] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, and Riccardo Rosati. "Ontology-Based Database Access". In: *Proceedings of the 15th Italian Symposium on Advanced Database Systems (SEBD 2007)*. SEBD 2007. Torre Canne, Italy, 2007, pp. 324–331.

[Cal+07b] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. "Tractable Reasoning and Efficient Query Answering in Description Logics: The *DL-Lite* Family". In: *Journal of Automated Reasoning* 39.3 (2007), pp. 385–429.

[Cal+09] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodríguez-Muro, and Riccardo Rosati. "Ontologies and Databases: The *DL-Lite* Approach". In: *Reasoning Web. Semantic Technologies for Information Systems: Tutorial Lectures of the 5th International Summer School 2009 (RW 2009)*. Brixen-Bressanone, Italy: Springer Berlin Heidelberg, 2009, pp. 255–356.

[Cal+10] Diego Calvanese, Evgeny Kharlamov, Werner Nutt, and Dmitriy Zheleznyakov. "Evolution of *DL-Lite* Knowledge Bases". In: *The Semantic Web – Proceedings of the 9th International Semantic Web Conference (ISWC 2010)*. Shanghai, China: Springer Berlin Heidelberg, 2010, pp. 112–128.

[Cal+13] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. "Data Complexity of Query Answering in Description Logics". In: *Artificial Intelligence* 195 (2013), pp. 335–360.

[Cal+15] Diego Calvanese, Martin Giese, Dag Hovland, and Martin Rezk. "Ontology-Based Integration of Cross-Linked Datasets". In: *The Semantic Web – Proceedings of the 14th International Semantic Web Conference (ISWC 2015)*. Bethlehem, PA, USA: Springer International Publishing, 2015, pp. 199–216.

[Cal+18] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. "Ontology-Based Data Access and Integration". In: *Encyclopedia of Database Systems*. Springer New York, 2018, pp. 2590–2596.

[CDL02] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. "Description Logics for Information Integration". In: *Computational Logic: Logic Programming and Beyond: Essays in Honour of Robert A. Kowalski Part II*. Springer Berlin Heidelberg, 2002, pp. 41–60.

[CF15] Michalis Chortis and Giorgos Flouris. "A Diagnosis and Repair Framework for *DL-Lite*$_{\mathcal{A}}$ KBs". In: *The Semantic Web: Research and Applications – Proceedings of the Satellite Events of the 12th European Conference on the Semantic Web (ESWC 2015)*. Portorož, Slovenia: Springer International Publishing, 2015, pp. 199–214.

[CGT89] Stefano Ceri, Georg Gottlob, and Letizia Tanca. “What You Always Wanted to Know About Datalog (And Never Dared to Ask)”. In: *IEEE Transactions on Knowledge and Data Engineering* 1.1 (1989), pp. 146–166.

[CGT90] Stefano Ceri, Georg Gottlob, and Letizia Tanca. *Logic Programming and Databases*. Springer Berlin Heidelberg, 1990.

[CWL14] Richard Cyganiak, David Wood, and Markus Lenthaler. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation. W3C, Feb. 2014. URL: https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[Dam+10] Mariana Damova, Atanas Kiryakov, Kiril Simov, and Svetoslav Petrov. “Mapping the Central LOD Ontologies to PROTON Upper-Level Ontology”. In: *Proceedings of the 5th International Workshop on Ontology Matching (OM-2010) collocated with the 9th International Semantic Web Conference (ISWC-2010)* (2010), pp. 61–72.

[dFL08] Claudia d’Amato, Nicola Fanizzi, and Thomas Lukasiewicz. “Tractable Reasoning with Bayesian Description Logics”. In: *Proceedings of the 2nd International Conference on Scalable Uncertainty Management (SUM 2008)*. Naples, Italy: Springer Berlin Heidelberg, 2008, pp. 146–159.

[Dom+08] Pedro Domingos, Stanley Kok, Daniel Lowd, Hoifung Poon, Matthew Richardson, and Parag Singla. “Markov Logic”. In: *Probabilistic Inductive Logic Programming: Theory and Applications*. Springer Berlin Heidelberg, 2008, pp. 92–117.

[Don+15] Xin Luna Dong, Evgeniy Gabrilovich, Kevin Murphy, Van Dang, Wilko Horn, Camillo Lugaresi, Shaohua Sun, and Wei Zhang. “Knowledge-Based Trust: Estimating the Trustworthiness of Web Sources”. In: *Proceedings of the 41st International Conference on Very Large Data Bases (VLDB 2015)* 8.9 (2015), pp. 938–949.

[Don+94] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. “Deduction in Concept Languages: from Subsumption to Instance Checking”. In: *Journal of Logic and Computation* 4.4 (1994), pp. 423–452.

[ES13] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer Berlin Heidelberg, 2013.

[Flo+06] Giorgos Flouris, Zhisheng Huang, Jeff Z. Pan, Dimitris Plexousakis, and Holger Wache. “Inconsistencies, Negations and Changes in Ontologies”. In: *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006) - Volume 2*. Boston, MA, USA: AAAI Press, 2006, pp. 1295–1300.

[FNS11] Alfio Ferrara, Andriy Nikolov, and François Scharffe. “Data Linking for the Semantic Web”. In: *International Journal on Semantic Web and Information Systems (IJSWIS)* 7.3 (2011), pp. 46–76.

[Gal+10] Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart. “Corroborating Information from Disagreeing Views”. In: *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM 2010)*. New York, NY, USA: ACM, 2010, pp. 131–140.

[GM10] Oliver Gries and Ralf Möller. “Gibbs Sampling in Probabilistic Description Logics with Deterministic Dependencies”. In: *Proceedings of the 1st International Workshop on Uncertainty in Description Logics*. CEUR Electronic Workshop Proceedings, 2010, pp. 42–51.

[Gol+17] Behzad Golshan, Alon Halevy, George Mihaila, and Wang-Chiew Tan. “Data Integration: After the Teenage Years”. In: *Proceedings of the Thirty-sixth ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*. PODS ’17. Chicago, IL, USA: ACM, 2017, pp. 101–106.

[Gro+03] Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker. “Description Logic Programs: Combining Logic Programs with Description Logic”. In: *Proceedings of the 12th International Conference on World Wide Web (WWW ’03)*. Budapest, Hungary: ACM, 2003, pp. 48–57.

[GS00] Tyrone Grandison and Morris Sloman. “A Survey of Trust in Internet Applications”. In: *IEEE Communications Surveys & Tutorials* 3.4 (2000), pp. 2–16.

[Gut+15] Víctor Gutiérrez-Basulto, Yazmín Ibáñez-García, Roman Kontchakov, and Egor V. Kostylev. “Queries with Negation and Inequalities over Lightweight Ontologies”. In: *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 35.4 (2015), pp. 184–202.

[Gut+17] Victor Gutierrez-Basulto, Jean Christoph Jung, Carsten Lutz, and Lutz Schröder. “Probabilistic Description Logics for Subjective Uncertainty”. In: *Journal of Artificial Intelligence Research* 58 (2017), pp. 1–66.

[Haa+05] Peter Haase, Frank van Harmelen, Zhisheng Huang, Heiner Stuckenschmidt, and York Sure. “A Framework for Handling Inconsistency in Changing Ontologies”. In: *The Semantic Web – Proceedings of the 4th International Semantic Web Conference (ISWC 2005)*. Galway, Ireland: Springer Berlin Heidelberg, 2005, pp. 353–367.

[HH01] Jeff Heflin and James Hendler. “A Portrait of the Semantic Web in Action”. In: *IEEE Intelligent Systems* 16.2 (2001), pp. 54–59.

[HKR09] Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 2009.

[HKS06] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. "The Even More Irresistible $\mathcal{SROIQ}$". In: *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*. KR'06. Lake District, UK: AAAI Press, 2006, pp. 57–67.

[HM01] Volker Haarslev and Ralf Möller. "RACER System Description". In: *Automated Reasoning: Proceedings of the 1rd International Joint Conference (IJCAR 2001)*. Siena, Italy: Springer Berlin Heidelberg, 2001, pp. 701–705.

[HMU13] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation: Pearson New International Edition*. Pearson Education Limited, 2013.

[HMW12] Ian Horrocks, Boris Motik, and Zhe Wang. "The HermiT OWL Reasoner". In: *Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE 2012) at the 6th International Joint Conference on Automated Reasoning (IJCAR 2012)*. Manchester, UK: CEUR Electronic Workshop Proceedings, 2012.

[Hor05] Herman J. ter Horst. "Completeness, Decidability and Complexity of Entailment for RDF Schema and a Semantic Extension Involving the OWL Vocabulary". In: *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 3.2–3 (2005), pp. 79–115.

[HPH03] Ian Horrocks, Peter Patel-Schneider, and Frank van Harmelen. "From $\mathcal{SHIQ}$ and RDF to OWL: The Making of a Web Ontology Language". In: *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 1.1 (2003), pp. 7–26.

[HPS09] Matthew Horridge, Bijan Parsia, and Ulrike Sattler. "Explaining Inconsistencies in OWL Ontologies". In: *Proceedings of the 3rd International Conference on Scalable Uncertainty Management (SUM 2009)*. Springer Berlin Heidelberg. Washington, DC, USA, 2009, 124–137.

[HQ07] Peter Haase and Guilin Qi. "An Analysis of Approaches to Resolving Inconsistencies in DL-based Ontologies". In: *Proceedings of the International Workshop on Ontology Dynamics (IWOD 2007) at the 4th European Semantic Web Conference (ESWC 2007)*. Innsbruck, Austria, 2007, pp. 97–109.

[HS01] Ian Horrocks and Ulrike Sattler. "Ontology Reasoning in the $\mathcal{SHOQ}(\mathbf{D})$ Description Logic". In: *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*. Vol. 1. Seattle, WA, USA: Morgan Kaufmann, 2001, pp. 199–204.

[HS13] Steve Harris and Andy Seaborne. *SPARQL 1.1 Query Language*. W3C Recommendation. W3C, Mar. 2013. URL: https://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[HVT05] Zhisheng Huang, Frank Van Harmelen, and Annette Ten Teije. "Reasoning with Inconsistent Ontologies". In: *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*. Edinburgh, Scotland: Morgan Kaufmann, 2005, pp. 454–459.

[JC11] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. "LogMap: Logic-Based and Scalable Ontology Matching". In: *The Semantic Web – Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*. Bonn, Germany: Springer Berlin Heidelberg, 2011, pp. 273–288.

[Ji+09] Qiu Ji, Peter Haase, Guilin Qi, Pascal Hitzler, and Steffen Stadtmüller. "RaDON – Repair and Diagnosis in Ontology Networks". In: *The Semantic Web: Research and Applications – Proceedings of the 6th European Semantic Web Conference (ESWC 2009)*. Crete, Greece: Springer Berlin Heidelberg, 2009, pp. 863–867.

[Jia+09] Yuncheng Jiang, Ju Wang, Suqin Tang, and Bao Xiao. "Reasoning with Rough Description Logics: An Approximate Concepts Approach". In: *Information Sciences* 179.5 (2009), pp. 600–612.

[Kal+06] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and Bernardo Cuenca Grau. "Repairing Unsatisfiable Concepts in OWL Ontologies". In: *The Semantic Web: Research and Applications – Proceedings of the 3rd European Semantic Web Conference (ESWC 2006)*. Springer Berlin Heidelberg. Budva, Montenegro, 2006, pp. 170–184.

[Kal+07] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin. "Finding All Justifications of OWL DL Entailments". In: *The Semantic Web – Proceedings of the 6th International Semantic Web Conference (ISWC 2007) and the 2nd Asian Semantic Web Conference (ASWC 2007)*. Busan, Korea: Springer Berlin Heidelberg, 2007, pp. 267–280.

[Kal06] Aditya Kalyanpur. "Debugging and Repair of OWL Ontologies". PhD thesis. College Park, MD, USA: University of Maryland, 2006.

[Kar72] Richard M. Karp. "Reducibility Among Combinatorial Problems". In: *Proceedings of a Symposium on the Complexity of Computer Computations*. Yorktown Heights, NY, USA: Springer US, 1972, pp. 85–103.

[Kaz08] Yevgeny Kazakov. "$\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$". In: *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*. KR'08. Sydney, Australia: AAAI Press, 2008, pp. 274–284.

[Kaz09] Yevgeny Kazakov. "Consequence-Driven Reasoning for Horn $\mathcal{SHIQ}$ Ontologies". In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*. Pasadena, CA, USA: AAAI Press, 2009, pp. 2040–2045.

[KD11] Atanas Kiryakov and Mariana Damova. "Storing the Semantic Web: Repositories". In: *Handbook of Semantic Web Technologies*. Springer Berlin Heidelberg, 2011, pp. 231–297.

[Kee11] C. Maria Keet. "Rough Subsumption Reasoning with rOWL". In: *Proceedings of the South African Institute of Computer Scientists and Information Technologists Conference on Knowledge, Innovation and Leadership in a Diverse, Multidisciplinary Environment (SAICSIT 2011)*. Cape Town, South Africa: ACM, 2011, pp. 133–140.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[Kha+15] Zuhair Khayyat, Ihab F. Ilyas, Alekh Jindal, Samuel Madden, Mourad Ouzzani, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Nan Tang, and Si Yin. "BigDansing: A System for Big Data Cleansing". In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. Melbourne, Victoria, Australia: ACM, 2015, pp. 1215–1230.

[Kha+17] Evgeny Kharlamov, Dag Hovland, Martin G. Skjæveland, Dimitris Bilidas, Ernesto Jiménez-Ruiz, Guohui Xiao, Ahmet Soylu, Davide Lanti, Martin Rezk, Dmitriy Zheleznyakov, Martin Giese, Hallstein Lie, Yannis Ioannidis, Yannis Kotidis, Manolis Koubarakis, and Arild Waaler. "Ontology Based Data Access in Statoil". In: *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 44 (2017), pp. 3–36.

[KKZ12] Stanislav Kikot, Roman Kontchakov, and Michael Zakharyaschev. "Conjunctive Query Answering with OWL 2 QL". In: *Proceedings of the 13th International Conference on the Principles of Knowledge Representation and Reasoning (KR'12)*. Rome, Italy: AAAI Press, 2012, pp. 275–285.

[Kli11] Pavel Klinov. "Practical Reasoning in Probabilistic Description Logic". PhD thesis. Manchester, United Kingdom: University of Manchester, 2011.

[KM08] Yevgeny Kazakov and Boris Motik. "A Resolution-Based Decision Procedure for $\mathcal{SHOIQ}$". In: *Journal of Automated Reasoning* 40 (2008), pp. 89–116.

[Kol+07] Daphne Koller, Nir Friedman, Lise Getoor, and Ben Taskar. "Graphical Models in a Nutshell". In: *Introduction to Statistical Relational Learning* (2007), pp. 13–55.

[Kon+14] Roman Kontchakov, Martin Rezk, Mariano Rodríguez-Muro, Guohui Xiao, and Michael Zakharyaschev. "Answering SPARQL Queries over Databases under OWL 2 QL Entailment Regime". In: *The Semantic Web – Proceedings of the 13th International Semantic Web Conference (ISWC 2014)*. Trentino, Italy: Springer Berlin Heidelberg, 2014, pp. 552–567.

[Koz06] Dexter C. Kozen. *Theory of Computation*. Springer London, 2006.

[KR10] Hanna Köpcke and Erhard Rahm. "Frameworks for Entity Matching: A Comparison". In: *Data & Knowledge Engineering* 69.2 (2010), pp. 197–210.

[Krö12] Markus Krötzsch. "OWL 2 Profiles: An Introduction to Lightweight Ontology Languages". In: *Reasoning Web. Semantic Technologies for Advanced Query Answering: Proceedings of the 8th International Summer School 2012 (RW 2012)*. Vienna, Austria: Springer Berlin Heidelberg, 2012, pp. 112–183.

[KSH14] Markus Krötzsch, František Simančík, and Ian Horrocks. "Description Logics". In: *IEEE Intelligent Systems* 29.1 (2014), pp. 12–19.

[KW87] Johan de Kleer and Brian C. Williams. "Diagnosing multiple faults". In: *Artificial Intelligence* 32.1 (1987), pp. 97–130.

[Lem+11] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. "Query Rewriting for Inconsistent *DL-Lite* Ontologies". In: *Proceedings of the 5th International Conference on Web Reasoning and Rule Systems (RR 2011)*. Galway, Ireland: Springer Berlin Heidelberg, 2011, pp. 155–169.

[Lem+12] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. "Inconsistency-Tolerant First-Order Rewritability of *DL-Lite* with Identification and Denial Assertions". In: *Proceedings of the 25th International Workshop on Description Logics (DL 2012)*. Rome, Italy: CEUR Electronic Workshop Proceedings, 2012.

[Len02] Maurizio Lenzerini. "Data Integration: A Theoretical Perspective". In: *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*. PODS '02. Madison, WI, USA: ACM, 2002, pp. 233–246.

[Li+12] Xian Li, Xin Luna Dong, Kenneth Lyons, Weiyi Meng, and Divesh Srivastava. "Truth Finding on the Deep Web: Is the Problem Solved?" In: *Proceedings of the 39th International Conference on Very Large Data Bases (VLDB 2013)* 6.2 (2012), pp. 97–108.

[Li+16] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. "A Survey on Truth Discovery". In: *SIGKDD Explorations* 17.2 (2016), pp. 1–16.

[LI13] Patrick Lambrix and Valentina Ivanova. “A unified approach for debugging is-a structure and mappings in networked taxonomies”. In: *Journal of Biomedical Semantics* 4 (2013), pp. 1–19.

[Li13] Yingjie Li. “A Federated Query Answering System for Semantic Web Data”. PhD thesis. Bethlehem, PA, USA: Lehigh University, 2013.

[Liu+17] Wenqiang Liu, Jun Liu, Haimeng Duan, Xie He, and Bifan Wei. “Exploiting Source-Object Network to Resolve Object Conflicts in Linked Data”. In: *The Semantic Web – Proceedings of the 14th European Semantic Web Conference (ESWC 2017)*. Portorož, Slovenia: Springer International Publishing, 2017, pp. 53–67.

[LNZ14] Xuejin Li, Zhendong Niu, and Chunxia Zhang. “Towards Efficient Distributed SPARQL Queries on Linked Data”. In: *Proceedings of the 14th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*. Dalian, China: Springer International Publishing, 2014, pp. 259–272.

[LS08] Thomas Lukasiewicz and Umberto Straccia. “Managing Uncertainty and Vagueness in Description Logics for the Semantic Web”. In: *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 6.4 (2008), pp. 291–308.

[Ma+15] Fenglong Ma, Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. “Faitcrowd: Fine Grained Truth Discovery for Crowdsourced Data Aggregation”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2015)*. Sydney, Australia: ACM, 2015, pp. 745–754.

[MC15] Denis D. Mauá and Fabio G. Cozman. “*DL-Lite* Bayesian Networks: A Tractable Probabilistic Graphical Model”. In: *Proceedings of the 9th International Conference on Scalable Uncertainty Management (SUM 2015)*. Québec City, QC, Canada: Springer International Publishing, 2015, pp. 50–64.

[McC] John P. McCrae. *The Linked Open Data Cloud Diagram*. URL: `https://www.lod-cloud.net/` (visited on 02/22/2019).

[Mei11] Christian Meilicke. “Alignment Incoherence in Ontology Matching”. PhD thesis. Mannheim, Germany: University of Mannheim, 2011.

[MH08] Boris Motik and Ian Horrocks. “OWL Datatypes: Design and Implementation”. In: *The Semantic Web – Proceedings of the 7th International Semantic Web Conference (ISWC 2008)*. Karlsruhe, Germany: Springer Berlin Heidelberg, 2008, pp. 307–322.

[Moo10] Kodylan Moodley. “Debugging and Repair of Description Logic Ontologies”. MA thesis. Durban, South Africa: University of KwaZulu-Natal, 2010.

[Mot+12] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz. *OWL 2 Web Ontology Language Profiles (Second Edition)*. W3C Recommendation. W3C, Dec. 2012. URL: https://www.w3.org/TR/2012/REC-owl2-profiles-20121211/.

[MPP12] Boris Motik, Peter F. Patel-Schneider, and Bijan Parsia. *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)*. W3C Recommendation. W3C, Dec. 2012. URL: https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/.

[MRR11] Giulia Masotti, Riccardo Rosati, and Marco Ruzzi. "Practical ABox Cleaning in *DL-Lite* (Progress Report)". In: *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*. Barcelona, Spain: CEUR Electronic Workshop Proceedings, 2011.

[MS06] Boris Motik and Ulrike Sattler. "A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes". In: *Proceedings of the 13th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2006)*. Phnom Penh, Cambodia: Springer Berlin Heidelberg, 2006, pp. 227–241.

[MSH09] Boris Motik, Rob Shearer, and Ian Horrocks. "Hypertableau Reasoning for Description Logics". In: *Journal of Artificial Intelligence Research* 36.1 (2009), pp. 165–228.

[MT19] Michalis Mountantonakis and Yannis Tzitzikas. "Large-scale Semantic Integration of Linked Data: A Survey". In: *ACM Computing Surveys (CSUR)* 52.5 (2019), pp. 1–40.

[Nen+17] Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm. "A Survey of Current Link Discovery Frameworks". In: *Semantic Web* 8.3 (2017), pp. 419–436.

[Ngu10] Linh Anh Nguyen. "Paraconsistent and Approximate Semantics for the OWL 2 Web Ontology Language". In: *Proceedings of the 7th International Conference on Rough Sets and Current Trends in Computing (RSCTC 2010)*. Warsaw, Poland: Springer Berlin Heidelberg, 2010, pp. 710–720.

[NLK09] Kalaivany Natarajan, Jiuyong Li, and Andy Koronios. "Data Mining Techniques for Data Cleaning". In: *Proceedings of the 4th World Congress on Engineering Asset Management (WCEAM 2009)*. Athens, Greece: Springer London, 2009, pp. 796–804.

[NN13] Andreas Nolle and German Nemirovski. “ELITE: An Entailment-Based Federated Query Engine for Complete and Transparent Semantic Data Integration”. In: *Proceedings of the 26th International Workshop on Description Logics (DL 2013)*. Ulm, Germany: CEUR Electronic Workshop Proceedings, 2013, pp. 854–867.

[NNS11] Mathias Niepert, Jan Noessner, and Heiner Stuckenschmidt. “Log-Linear Description Logics”. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*. Barcelona, Spain: AAAI Press, 2011, pp. 2153–2158.

[Nol+14] Andreas Nolle, Christian Meilicke, Heiner Stuckenschmidt, and German Nemirovski. “Efficient Federated Debugging of Lightweight Ontologies”. In: *Proceedings of the 8th International Conference on Web Reasoning and Rule Systems (RR 2014)*. Athens, Greece: Springer International Publishing, 2014, pp. 206–215.

[Nol+16] Andreas Nolle, Christian Meilicke, Melisachew Wudage Chekol, German Nemirovski, and Heiner Stuckenschmidt. “Schema-Based Debugging of Federated Data Sources”. In: *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*. The Hague, Netherlands: IOS Press, 2016, pp. 381–389.

[Nol+17] Andreas Nolle, Melisachew Wudage Chekol, Christian Meilicke, German Nemirovski, and Heiner Stuckenschmidt. “Automated Fine-Grained Trust Assessment in Federated Knowledge Bases”. In: *The Semantic Web – Proceedings of the 16th International Semantic Web Conference (ISWC 2017)*. Vienna, Austria: Springer International Publishing, 2017, pp. 490–506.

[NS16] Axel-Cyrille Ngonga Ngomo and Muhammad Saleem. “Federated Query Processing: Challenges and Opportunities.” In: *Proceedings of the 3rd International Workshop on Dataset PROFIling and fEderated Search for Linked Data (PROFILES 2016) co-located with the 13th European Semantic Web Conference (ESWC 2016)*. Anissaras, Greece: CEUR Electronic Workshop Proceedings, 2016.

[OK18] Peter Ochieng and Swaib Kyanda. “Large-Scale Ontology Matching: State-of-the-Art Analysis”. In: *ACM Computing Surveys (CSUR)* 51.4 (2018), p. 75.

[ORG15] Lorena Otero-Cerdeira, Francisco J. Rodríguez-Martínez, and Alma Gómez-Rodríguez. “Ontology Matching: A Literature Review”. In: *Expert Systems with Applications* 42.2 (2015), pp. 949–971.

[OŠ12] Magdalena Ortiz and Mantas Šimkus. “Reasoning and Query Answering in Description Logics”. In: *Reasoning Web. Semantic Technologies for Advanced Query Answering: Proceedings of the 8th In-*

ternational Summer School 2012 (RW 2012). Vienna, Austria: Springer Berlin Heidelberg, 2012, pp. 1–53.

[PAG06] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. "Semantics and Complexity of SPARQL". In: *The Semantic Web – Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*. Athens, GA, USA: Springer Berlin Heidelberg, 2006, pp. 30–43.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison Wesley Publ. Co., 1994.

[Pog+08] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. "Linking Data to Ontologies". In: *Journal on Data Semantics X* (2008), pp. 133–173.

[Pog16] Antonella Poggi. "On the SPARQL Direct Semantics Entailment Regime for OWL 2 QL". In: *Proceedings of the 29th International Workshop on Description Logics (DL 2016)*. Cape Town, South Africa: CEUR Electronic Workshop Proceedings, 2016.

[Qi+15] Guilin Qi, Zhe Wang, Kewen Wang, Xuefeng Fu, and Zhiqiang Zhuang. "Approximating Model-Based ABox Revision in *DL-Lite*: Theory and Practice". In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*. Austin, TX, USA: AAAI Press, 2015, pp. 254–260.

[QJH09] Guilin Qi, Qiu Ji, and Peter Haase. "A Conflict-Based Operator for Mapping Revision". In: *The Semantic Web – Proceedings of the 8th International Semantic Web Conference (ISWC 2009)*. Chantilly, VA, USA: Springer Berlin Heidelberg, 2009, pp. 521–536.

[QL08] Bastian Quilitz and Ulf Leser. "Querying Distributed RDF Data Sources with SPARQL". In: *The Semantic Web: Research and Applications – Proceedings of the 5th European Semantic Web Conference (ESWC 2008)*. Tenerife, Spain: Springer Berlin Heidelberg, 2008, pp. 524–538.

[Rak+13a] Nur Aini Rakhmawati, Jürgen Umbrich, Marcel Karnstedt, Ali Hasnain, and Michael Hausenblas. "A Comparison of Federation over SPARQL Endpoints Frameworks". In: *Proceedings of the International Conference on Knowledge Engineering and the Semantic Web (KESW 2013)*. Springer. St. Petersburg, Russia, 2013, pp. 132–146.

[Rak+13b] Nur Aini Rakhmawati, Jürgen Umbrich, Marcel Karnstedt, Ali Hasnain, and Michael Hausenblas. "Querying over Federated SPARQL Endpoints – A State of the Art Survey". In: *arXiv:1306.1723* (2013).

[Ram+12] Raghav Ramachandran, Guilin Qi, Kewen Wang, Junhu Wang, and John Thornton. "Probabilistic Reasoning in *DL-Lite*". In: *Proceedings of the 12th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2012)*. Kuching, Malaysia: Springer Berlin Heidelberg, 2012, pp. 480–491.

[RC11] Mariano Rodrıguez-Muro and Diego Calvanese. "Dependencies: Making Ontology Based Data Access Work in Practice". In: *Proceedings of the 5th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW 2011)*. Vol. 749. Santiago, Chile: CEUR Electronic Workshop Proceedings, 2011.

[RD06] Matthew Richardson and Pedro Domingos. "Markov Logic Networks". In: *Machine Learning* 62 (2006), pp. 107–136.

[Rei87] Raymond Reiter. "A Theory of Diagnosis from First Principles". In: *Artificial Intelligence* 32.1 (1987), pp. 57–95.

[RKZ13] Mariano Rodríguez-Muro, Roman Kontchakov, and Michael Zakharyaschev. "Ontology-Based Data Access: *Ontop* of Databases". In: *The Semantic Web – Proceedings of the 12th International Semantic Web Conference (ISWC 2013)*. Sydney, NSW, Australia: Springer Berlin Heidelberg, 2013, pp. 558–573.

[Rod10] Mariano Rodrıguez-Muro. "Tools and Techniques for Ontology Based Data Access in Lightweight Description Logics". PhD thesis. Bolzano, Italy: KRDB Research Centre for Knowledge and Data, Free University of Bozen-Bolzano, 2010.

[Ros07] Riccardo Rosati. "The Limits of Querying Ontologies". In: *Proceedings of the 11th International Conference on Database Theory (ICDT 2007)*. Barcelona, Spain: Springer Berlin Heidelberg, 2007, pp. 164–178.

[Rud11] Sebastian Rudolph. "Foundations of Description Logics". In: *Reasoning Web. Semantic Technologies for the Web of Data: Tutorial Lectures of the 7th International Summer School 2011 (RW 2011)*. Galway, Ireland: Springer Berlin Heidelberg, 2011, pp. 76–136.

[Sal+16] Muhammad Saleem, Yasar Khan, Ali Hasnain, Ivan Ermilov, and Axel-Cyrille Ngonga Ngomo. "A Fine-Grained Evaluation of SPARQL Endpoint Federation Systems". In: *Semantic Web* 7.5 (2016), pp. 493–518.

[San+15] Emanuel Santos, Daniel Faria, Catia Pesquita, and Francisco Couto. "Ontology Alignment Repair through Modularization and Confidence-Based Heuristics". In: *PloS ONE* 10.12 (2015), pp. 1–19.

[Sav13] Domenico Fabio Savo. "Dealing with Inconsistencies and Updates in Description Logic Knowledge Bases". PhD thesis. Rome, Italy: Sapienza University of Rome, 2013.

[SC03] Stefan Schlobach and Ronald Cornet. "Non-standard Reasoning Services for the Debugging of Description Logic Terminologies". In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*. Acapulco, Mexico: Morgan Kaufmann, 2003, pp. 355–362.

[Sch+07] Stefan Schlobach, Zhisheng Huang, Ronald Cornet, and Frank van Harmelen. "Debugging Incoherent Terminologies". In: *Journal of Automated Reasoning* 39.3 (2007), pp. 317–349.

[SE11] François Scharffe and Jérôme Euzenat. "Linked Data Meets Ontology Matching: Enhancing Data Linking Through Ontology Alignments". In: *Proceedings of the 3rd International Conference on Knowledge Engineering and Ontology Development (KEOD 2011)*. Paris, France: SciTePress, 2011, pp. 279–284.

[SE13] Pavel Shvaiko and Jérôme Euzenat. "Ontology Matching: State of the Art and Future Challenges". In: *IEEE Transactions on Knowledge and Data Engineering* 25.1 (2013), pp. 158–176.

[SH05] Heiner Stuckenschmidt and Frank van Harmelen. *Information Sharing on the Semantic Web*. Springer Berlin Heidelberg, 2005.

[Sir+07] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. "Pellet: A Practical OWL-DL Reasoner". In: *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 5.2 (2007), pp. 51–53.

[SKH11] František Simančík, Yevgeny Kazakov, and Ian Horrocks. "Consequence-Based Reasoning beyond Horn Ontologies". In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*. Barcelona, Spain: AAAI Press, 2011, pp. 1093–1098.

[SKP07] Stefan Schlobach, Michel Klein, and Linda Peelen. "Description Logics with Approximate Definitions-Precise Modeling of Vague Concepts". In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*. Hyderabad, India: AAAI Press, 2007, pp. 557–562.

[ST04] Luciano Serafini and Andrei Tamilin. "Local Tableaux for Reasoning in Distributed Description Logics". In: *Proceedings of the 17th International Workshop on Description Logics (DL 2004)*. Whistler, BC, Canada: CEUR Electronic Workshop Proceedings, 2004, pp. 100–109.

[Str15] Umberto Straccia. "All About Fuzzy Description Logics and Applications". In: *Reasoning Web. Web Logic Rules: Proceedings of the 11th International Summer School 2015 Reasoning Web International Summer School (RW 2015)*. Berlin, Germany: Springer International Publishing, 2015, pp. 1–31.

[Stu08] Heiner Stuckenschmidt. “Debugging OWL Ontologies – A Reality Check”. In: *Proceedings of the 6th International Workshop on Evaluation of Ontology-based Tools and the Semantic Web Service Challenge (EON-SWSC 2008)*. Vol. 359. Tenerife, Spain: CEUR Electronic Workshop Proceedings, 2008.

[Stu13] Heiner Stuckenschmidt. “Debugging Weighted Ontologies”. In: *Proceedings of the 2nd International Workshop on Debugging Ontologies and Ontology Mappings (WoDOOM 2013)*. Montpellier, France: CEUR Electronic Workshop Proceedings, 2013, pp. 1–8.

[TH06] Dmitry Tsarkov and Ian Horrocks. “FaCT++ Description Logic Reasoner: System Description”. In: *Automated Reasoning: Proceedings of the 3rd International Joint Conference (IJCAR 2006)*. Seattle, WA, USA: Springer Berlin Heidelberg, 2006, pp. 292–297.

[Var82] Moshe Y. Vardi. “The Complexity of Relational Query Languages”. In: *Proceedings of the 14th ACM Symposium on Theory of Computing*. San Francisco, CA, USA: ACM, 1982, pp. 137–146.

[VN11] Johanna Völker and Mathias Niepert. “Statistical Schema Induction”. In: *The Semantic Web: Research and Applications – Proceedings of the 8th Extended Semantic Web Conference (ESWC 2011)*. Springer Berlin Heidelberg, 2011, pp. 124–138.

[Vol99] Heribert Vollmer. *Introduction to Circuit Complexity: A Uniform Approach*. Springer Berlin Heidelberg, 1999.

[Wac+01] Holger Wache, Thomas Vögele, Ubbo Visser, Heiner Stuckenschmidt, Gerhard Schuster, Holger Neumann, and Sebastian Hübner. “Ontology-Based Integration of Information – A Survey of Existing Approaches”. In: *Proceedings of the International Workshop on Ontologies and Information Sharing at the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*. Seattle, WA, USA: CEUR Electronic Workshop Proceedings, 2001, pp. 108–117.

[Wöl+11] Stephan Wölger, Katharina Siorpaes, Tobias Bürger, Elena Simperl, Stefan Thaler, and Christian Hofer. *A Survey on Data Interlinking Methods*. Technical Report. STI Innsbruck, Mar. 2011.

[YHY08] Xiaoxin Yin, Jiawei Han, and Philip S. Yu. “Truth Discovery with Multiple Conflicting Information Providers on the Web”. In: *IEEE Transactions on Knowledge and Data Engineering* 20.6 (2008), 796–808.

[Zav+16] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. “Quality Assessment for Linked Data: A Survey”. In: *Semantic Web* 7.1 (2016), pp. 63–93.

[ZH12] Bo Zhao and Jiawei Han. "A Probabilistic Model for Estimating Real-valued Truth from Conflicting Sources". In: *Proceedings of the VLDB Workshop on Quality in Databases (QDB 2012)* (2012).

[Zha+12] Bo Zhao, Benjamin Rubinstein, Jim Gemmell, and Jiawei Han. "A Bayesian Approach to Discovering Truth from Conflicting Sources for Data Integration". In: *Proceedings of the 38th International Conference on Very Large Data Bases (VLDB 2012)* 5.6 (2012), pp. 550–561.

[Zho+09] Liping Zhou, Houkuan Huang, Guilin Qi, Yue Ma, Zhisheng Huang, and Youli Qu. "Measuring Inconsistency in *DL-Lite* Ontologies". In: *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*. Milan, Italy: IEEE, 2009, pp. 349–356.