Reasoning-Supported Quality Assurance for Knowledge Bases

Zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften (Dr. rer. pol.)
von der Fakultät für Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)
genehmigte Dissertation
von

Dipl.-Inf. Nadeschda Nikitina
aus Schachtinsk

Karlsruhe, 2012

Dekan:	Prof. Dr. Christof Weinhardt, Karlsruher Institut für Technologie
Hauptreferent:	Prof. Dr. Rudi Studer, Karlsruher Institut für Technologie
Korreferent:	Prof. Dr. Heiner Schtuckenschmidt, Universität Mannheim

Tag der mündlichen Prüfung: 28. November 2012

Dekan: Prof. Dr. Clemens Puppe, Karlsruher Institut für Technologie
Referent: Prof. Dr. Rudi Studer, Karlsruher Institut für Technologie

Tag der mündlichen Prüfung: 28. November 2012

Abstract

The increasing application of ontology reuse and automated knowledge acquisition tools in ontology engineering brings about a shift of development efforts from knowledge modeling towards quality assurance. When ontology reuse or automatic knowledge acquisition are applied, accuracy and conciseness are the two most typical quality problems. Yet, despite the high practical importance, there has been a substantial lack of support for essential quality assurance activities concerning these two quality dimensions. In this thesis, we make a significant step forward in ontology engineering by developing a support for two such essential quality assurance activities.

We develop a sophisticated framework and the corresponding tool support for partially automating the inspection of ontologies with respect to accuracy. This is a significant contribution in the field of ontology engineering, since manual inspection of ontologies, not replaceable by ontology debugging or constraint formalization in professional ontology engineering projects, is one of the most costly alternatives in quality assurance due to the high amount of required user interaction. The framework is based on the assumption that the deductive closure of the correct axioms must be disjoint from the set of incorrect axioms, which holds for all standardized ontology languages with formal semantics. Given this general assumption, we employ reasoning in order to reduce the number of decisions that have to be taken by a domain expert in order to complete the inspection. Due to its generality, the framework allows for a maximum automation achieved by reasoning for a wide range of ontology modeling languages and for a flexible choice of initial constraints applying to the ontology.

Since the order of inspection has an impact on the effectiveness of the reasoning-based support, we further propose and compare various axiom ranking techniques used to determine a beneficial order of inspection. These ranking heuristics are based on the expected accuracy ratio of an ontology and aim at choosing axioms with the highest number of subsequent automatic evaluations. In order to deliberate the user from having to provide an estimate of the accuracy ratio in advance, we show that this estimate can effectively be learned on-the-fly over the course

of the revision. Additionally, since the above reasoning-based support is computationally expensive, we elaborate on techniques ensuring a decent computational efficiency of the approach. To this end, we introduce a simple partitioning method as well as auxiliary data structures that are used for keeping track of dependencies between axioms. We provide comprehensive evaluation results demonstrating the effectiveness of the framework and the aforementioned optimizations.

Further, we address the problem of improving the conciseness of ontologies. Due to the significant impact of the ontology's size on the cost of reasoning and maintenance, this step is essential for the performance of Semantic Web applications, in particular in combination with ontology reuse. We consider the problem of general module extraction – a semantics-preserving computation of a smaller ontology given a set of relevant ontology entities. Currently, there are two concrete formal manifestations of this problem – uniform interpolation and classical module extraction. First, we complete the picture characterizing these two problems in description logics. To this end, we solve the problem of uniform interpolation for general $\mathcal{EL}$ terminologies by providing a worst-case optimal algorithm for its computation and deriving a triple-exponential bound on the size of uniform interpolants. Further, we take a critical look on these two problem manifestations in the light of the three conflicting objectives for general module extraction: reducing the size of the extracted general module, reducing the size of its signature and preserving the syntactic similarity of the general module and the initial ontology. In most application scenarios, all three objectives are important. We show that neither classical module extraction nor uniform interpolation take this into account. To overcome these shortcomings of uniform interpolation and classical module extraction, we derive an alternative problem manifestation with a more balanced prioritization of objectives. We show how a minimal module of the novel type can be computed in 2ExpTime by applying a particular normalization to the initial ontology and employing classical module extraction to the result.

Given that general module extraction in ontology engineering is of a particular interest for large ontologies and that some practically relevant application scenarios involve user interaction, also the computational feasibility has to be taken into account. The complexity results for the extraction of minimal modules based on all three of the aforementioned problem manifestations (ExpTime, 3ExpTime and 2ExpTime) are not satisfactory with respect to this aspect. To enable the application of general module extraction in practice, we further develop a tractable approach to computing modules of the novel type. We show that this tractable approach yields, in most cases, small modules, even through it does not guarantee the minimality of the result. In our evaluation, the approach outperforms classical minimal module extraction by the factor 2.0 to 2.2. In comparison with the only

alternative tractable approach to classical module extraction, the proposed novel approach even yields 12 times more concise modules.

ABSTRACT

Danksagungen

Diese Arbeit wäre nicht möglich gewesen ohne die Unterstützung vieler. Dafür möchte ich im Einzelnen danken:

Zu allererst Prof. Dr. Rudi Studer, der meine Arbeit offiziell betreut hat und Dr. Peter Haase und Dr. Johanna Völker, die das Projekt *NanOn* auf den Weg gebracht haben, aus dem diese Dissertation entstanden ist.
Ebenso der Prüfungskommission, Prof. Dr. Heiner Stuckenschmidt für die freundliche Übernahme des Korreferats sowie Prof. Dr. Detlef Seese, der sich als Prüfer zur Verfügung gestellt hat.
Der besondere Dank gilt meinem de-facto Betreuer Dr. Sebastian Rudolph, dem ich sowohl meine Arbeit betreffend als auch persönlich viel zu verdanken habe.
Meiner Co-Autorin Birte Glimm, von der ich während unserer Zusammenarbeit Vieles lernen durfte, sowohl fachlich als auch persönlich, und von der ich häufig wertvolle Unterstützung bekommen habe.
Prof. Dr. Frank Wolter und Dr. Boris Konev von der Universität Liverpool sowie Prof. Dr. Carsten Lutz von der Universität Bremen für interessante Diskussionen und Vorarbeiten, die mich bei meiner Arbeit sehr weitergebracht haben.
Karlsruhe House of Young Scientists für die finanzielle Unterstützung sowohl bei der Weiterbildung als auch dem Aufbau der Zusammenarbeit mit Kollegen an den Universitäten Oxford, Bolzano und Liverpool.
Dr. Markus Krötzsch, der mich während seiner Zeit am Institut inspiriert hat, meine Forschung in Richtung Beschreibungslogiken auszubauen.
Meinen Kollegen am Lehrstuhl, allen voran Basil Ell, die meine Zeit am Lehrstuhl zu dem Erlebnis gemacht haben, das sie war.
Und zu guter letzt meinen Eltern und meinem Verlobten für ihre Unterstützung sowie meiner Tochter, die mit ihrer Geburt bis zur Fertigstellung dieser Arbeit gewartet hat.

Vielen Dank!

Contents

List of Tables

List of Figures

Part I

Introduction

CHAPTER 1

Introduction

As the corresponding tool support for ontology engineering gains in maturity and the amount of freely accessible ontological data grows, the deployment of ontologies spreads to increasingly powerful and critical applications. The development of ontologies is, however, a highly complex and error-prone task which requires stable and reliable quality assurance methodology and tool support. This thesis addresses two particular types of quality aspects playing an important role within ontology development and reuse: accuracy and conciseness of ontologies. This first chapter motivates and outlines the work: In Section 1.1, we discuss why the recently emerged semantic technologies play an increasingly central role in information handling and give examples of the modern use of ontologies. Further, we explain why quality assurance is indispensable in most cases and elaborate on the common sources of quality problems in Section 1.2. Subsequently, in Section 1.3, we describe the background of this work and define its scope and objectives. In the last section, we give an overview of this thesis and outline its structure.

1.1 Semantics in Information Handling

Since their advent, computers have been going through a rapid evolution process and have become an indispensable part of our civilization. Over the past century, they proved to be exceptionally useful for the management, acquisition and transfer of information, largely replacing printed artifacts such as letters, paper-based files, but also newspapers and books. The current development is characterized, on the one hand, by the constantly evolving means to efficiently satisfy modern information needs, and, on the other hand, by the amount of information publicly available in digital form growing at a breath-taking rate. Internet, the most versatile source of information at the present time, supports most of our every-day activities, varying from shopping and cooking to carrying out scientific research. In the 21st century, in addition to the role of the Internet in accessing the overwhelming amount of the publicly available knowledge, the tremendous potential of the Internet as a medium for sharing personal information has been realized. A wide range of platforms for all-round personal communication such as facebook, twitter, wikis and forums have arisen, enriching the Internet with information about personal experiences, opinions and ideas.

Within industry, information technology has established itself as a major strategic asset. A considerable amount of resources is spent on investment into cutting-edge technologies supporting knowledge management. Knowledge bases, expert systems, knowledge repositories, group decision support systems, intranets, and systems for computer-supported cooperative work are nowadays state of the art even within small-size organisations [ADDICOTT et al. 2006].

The scientific interest in digitally represented knowledge goes back to the mid-1970s. Around that time, researchers in the field of Artificial Intelligence (AI) [RUSSELL and NORVIG 2002] recognized that digitalized knowledge is the key to building large and powerful AI systems. One of the key ambitions within AI research has been to enable machines to draw logical inferences from the explicitly modelled knowledge. In the 21st century, the vision of the *Semantic Web* [BERNERS-LEE et al. 2001] arose, transferring the ideas of AI to the Web. Aiming at structuring the meaningful content of Web pages and "adding logic to the Web" [BERNERS-LEE et al. 2001], researchers envisioned a world in which intelligent

agents can carry out sophisticated tasks for users by processing the additional, *semantic* content of Web pages.

The key notion behind this vision is that of an *ontology* – a formal model of a particular domain describing a set of concepts and relationships between them. Originating from philosophy [SOWA 2000], where *Ontology* refers to a subdiscipline studying the *nature* and *essential properties* of the reality, the term gained a more pragmatic meaning in Computer Science, not being restricted to reflect the reality, but seen as a means to formally describe arbitrary models. The best known definition of an ontology in Computer Science is perhaps the one by Gruber: "an ontology is an explicit specification of a conceptualization" [GRUBER 1993]. In the light of this general definition, we can say that the main idea of Semantic Web is to foster a broad usage of shared explicit conceptualizations on the Web enabling a more sophisticated automatic processing of Web content, for instance, in order to facilitate search or an automatic aggregation of information across the Web.

Inspired by the vision of AI, and later on the Semantic Web, a wide usage of ontological modeling started with Expert Systems [JACKSON 1999, MATKAR and PARAB 2011] – software systems emulating the decision-making ability of a human expert by the means of logical inference – and has later on been largely carried out independently from particular software systems. Mostly at the early stage, several ontology development endeavours aiming at a broad coverage of all-purpose knowledge have been undertaken. Amongst others, the Cyc project [LENAT and GUHA 1989] started in 1984, aiming at capturing common sense knowledge in a form of an ontology and providing a corresponding inference engine. It yielded a large ontology with hundreds of thousands of terms, along with millions of assertions relating the terms to each other. Further examples for such general-purpose ontologies are the General Formal Ontology (GFO) [HERRE 2010], and Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [GANGEMI et al. 2002]. The problematic point about such general-purpose ontologies is, however, that it is usually impossible to guarantee a decent coverage given a realistic project budget. Ontology applications requiring a high coverage are, therefore, more effective in a more narrow scope, for instance, within a particular domain. Thus, ontologies capturing various domains emerged, e.g., medicine [RECTOR et al. 1994, SIOUTOS et al. 2007, GOLBREICH et al. 2006,

SPACKMAN et al. 1997], bioinformatics [ASHBURNER 2000, SMITH et al. 2007, DEGTYARENKO et al. 2008, GKOUTOS et al. 2005, NATALE et al. 2011] and geoscience [GOODWIN 2005, RASKIN and PAN 2005, FOX et al. 2009]. An example closely connected to this work is the ontology *NanOn* developed at AIFB and covering nanotechnological terms in scientific literature.

Mainly used for research purposes at the early stage of their development, semantic technologies are becoming increasingly common in practice. The reason for this is that the corresponding methodologies and tools are constantly gaining in maturity. Perhaps the most prominent step forward was the standardization of ontology modeling languages by the World Wide Web Consortium (W3C), including the *Resource Description Framework* RDF [MANOLA and MILLER 2004] used for exchange of factual data, and the *Web Ontology Language* OWL [MCGUINNESS and VAN HARMELEN 2004] based on *description logics* (DLs) [BAADER et al. 2007]. The latter has been revised and released in 2009 under the name OWL 2 [OWL WORKING GROUP 27 October 2009]. For application scenarios where scalability of reasoning is of utmost importance, specific tractable sublanguages (the so-called *profiles* [MOTIK et al. 27 October 2009]) of OWL have been put into place, among them the OWL EL profile based on description logics of the $\mathcal{EL}$ family [BRANDT 2004, BAADER et al. 2005].

OWL brings the advantage of formal modeling languages with well-defined reasoning procedures to the Web, thereby fostering semantic interoperability. Formal semantics serves as a quasi-standard providing a declarative, implementation-independent specification of the implicit statements that can be deduced from an ontology. It establishes a common understanding of reasoning among both, tool developers and ontology engineers. Thus, by introducing these standards, W3C provided the necessary technological foundation for a globally distributed development of tools and methods by a large number of independent parties, which is crucial for the competitiveness of semantic technologies in practice.

Worth special emphasis is also the positive impact of the recent advances in knowledge representation research, fostering a constant improvement of reasoning tools and the development of logically sound tool support for tasks that are too complex to be carried out by hand, e.g., ontology debugging and generating explanations for logical consequences.

As the tool support gains in stability and reliability, ontologies become viable in increasingly critical applications. For instance, biomedical research extensively uses semantic technologies to cope with an enormous stream of complex experimental data resulting from revolutionary changes in biomedical technology enabling a quick and cheap assay of biological molecules [TIPNEY and HUNTER 2010]: PubMed – the online database of biomedical literature [WHEELER et al. 2000] – comprised in 2009 over 19 million entries and continues growing at a rate of 1.5 publications per minute. Indeed, a number of ontologies (Gene Ontology [ASHBURNER 2000], Molecular Interaction Ontology [KERRIEN et al. 2007], ChEBI [DEGTYARENKO et al. 2008], KEGG [KANEHISA and GOTO 2000], BioPAX [DEMIR et al. 2010] and Disease Ontology [SCHRIML et al. 2012]) are utilized in various ways in order to support an efficient analysis of the resulting data and a generalization of observations [TIPNEY and HUNTER 2010]. Firstly, ontologies serve as a common syntactic and semantic basis for gathering evidence and documenting experimental results. For instance, ontologies are used for a large-scale annotation of publications [BAUMGARTNER et al. 2007] and it is anticipated that future publications may explicitly include such annotations [TIPNEY and HUNTER 2010]. Secondly, ontologies come with reasoning support that, among other things can be used to increase the precision and recall of biological text mining [TIPNEY and HUNTER 2010, CARRUTHERS et al. 2002] as well as the precision of manually carried out annotations [COLOMB 2002]. Reasoning can also facilitate the construction of new hypotheses by automatically checking their consistency [LIEBMAN and MOLINARO 2011, MIREL 2009] or even automatically verifying them given a sufficient available evidence [STEVENS and LORD 2009].

Also the number of examples for the industrial usage of semantic technologies grows constantly [SMITH 2011, BEHRENDT 2008, SUREEPHONG et al. 2008, BAHRAMMIRZAEE 2010]. One of such examples for industrially deployed software based on semantic technologies is Semantic MediaWiki (SMW) [KRÖTZSCH and VRANDECIC 2011]. SMW is an extension of MediaWiki [BARRETT 2009] enabling users to annotate the wiki's contents with arbitrary semantic relations, thereby explicitly assigning attributes to described objects or establishing relationships between objects. SMW significantly reduces the effort of

information maintenance and access by enabling structured queries over the annotated information. Among other things, SMW offers the following advantages to its users:

- Queries over annotated information alleviate finding and comparing information from different pages.
- Inline queries enable editors to add dynamically created lists or tables to a page, thus avoiding redundancy and insuring the consistency of the content.

Many more examples for the application areas of ontologies can be found, e.g., recommender systems [MARTINS and SILVA 2011, MIDDLETON et al. 2009, OZDIKIS et al. 2011], question answering [UNGER et al. 2012, FU et al. 2009, FERRÁNDEZ et al. 2009, TARTIR et al. 2009, VARGAS-VERA et al. 2003, BLOEHDORN et al. 2007], query expansion [DÍAZ-GALIANO et al. 2009, BHOGAL et al. 2007, CARPINETO and ROMANO 2012, LIU et al. 2004, GRAUPMANN et al. 2005] and document classification [IFRIM and WEIKUM 2006, BECHINI et al. 2008, SONG et al. 2005].
It may be expected that, with the increasing maturity of the corresponding tool support and amount of freely accessible ontological data, the spreading of ontologies to increasingly powerful and critical applications will continue. Naturally, one of the requirements is the availability of sophisticated quality assurance methods and tools for ontologies, which is the major concern of this thesis and will be discussed in more detail in the next section.

1.2 Quality Assurance for Ontologies

In the view of the constantly evolving semantic technologies and the subsequent increasing role that ontologies play in different application areas, the quality requirements for ontologies come to the fore. Clearly, low ontology quality, in particular inaccurate statements, can have notable negative effects in most usage scenarios and need to be addressed. For instance, in case that an ontology serves as a basis for question-answering or search, frequent modeling errors are likely to have a

negative effect on the precision and recall such that the advantage of elaborate semantic technologies is nullified. In case of general-purpose ontologies meant to be widely used by the research community, biased scientific results and systems that are of a limited usefulness to the targeted group of users are the likely consequences of low ontology quality. Yet, in particular for commercially deployed ontologies playing a central role within some software system, quality assurance is crucial. In the latter context, quality problems become a substantial financial risk, potentially leading to costly delays and unfulfilled contractual obligations. For instance, if an ontology is used within a task requiring a considerable amount of human effort, e.g., large-scale data acquisition involving manual inspections, a single modeling error can cause enormous additional costs. In the view of a prospective application of ontologies in life-critical systems, e.g., those automatically controlling manned vehicles or detecting and managing life-threatening emergencies, the availability of a solid, mature methodology and tool support for assuring the quality of ontologies is of an utmost importance. Quality assurance for ontologies is, however, highly non-trivial and, in general, it is not possible to guarantee the absence of quality problems in ontologies. Even in most thoroughly carried out ontology engineering projects such a guarantee cannot be given.

Quality problems can originate from many different sources: ontology engineers can be inexperienced in using the modeling language or tool or simply distracted by their environment. The problem can also arise from vaguely stated requirements or from a difference in perceptions of ontology engineers. In particular the latter has been actively discussed within the Semantic Web community, mostly in connection with the Cyc ontology [LENAT and GUHA 1989]. Such a difference in perceptions is likely to result in a conceptually inconsistent model and can be caused by various factors. On the one hand, this difficulty arises if a domain term does not have an established meaning or the common perception is a simplification. In fact, for many rules that could be stated about a domain concept, counterexamples can be found. For instance, the statement "all birds can fly" is an assumption that works for most birds and can be appropriate in a particular context, where counterexamples do not occur, but is not true in general. This problem has been investigated by John McCarthy since 1970 under the name *qualification problem* [MCCARTHY and HAYES 1987]. On the other hand, the boundaries of the relevant

domain part are usually fuzzy. This is particularly likely for upper level ontologies or ontologies capturing cross-domain knowledge. All in all, it is rather common for ontology development projects that the requirements are initially underspecified and are elaborated during the modeling, ideally resulting in an explicit documentation of numerous assumptions. One of the objectives of quality assurance is to identify the cases, where an agreement is necessary and, thereby, to bring about the explicity of assumptions guaranteeing the conceptual consistency of the model.
Despite the above discussed potential error sources, manual ontology modeling remains the most accurate and most expensive alternative. In the recent years, a large number of automatic ontology construction and extension techniques have emerged, most of which are based on heuristics. We can give the following examples:

- Extracting ontology elements and dependencies between them from natural language texts [VÖLKER 2009, BUITELAAR et al. 2005, SUCHANEK et al. 2008, ALFONSECA et al. 2010];
- Predicting dependencies between ontology elements using machine learning [RETTINGER et al. 2012];
- Semi- or fully automatic matching and merging of ontologies [GRUBER 2007, CROSS and HU 2011, EUZENAT et al. 2011].

The idea behind such heuristic approaches is to partially automate ontology development such that the expert part of the task largely consists of ensuring the quality of the results, in particular accuracy. Despite the required extensive quality assurance, the overall effort is usually significantly lower than that of a purely manual modeling. For this reason, such heuristic approaches are increasingly popular and are, in some cases, even the only alternative, e.g. in case of annotations on PubMed [BAUMGARTNER et al. 2007].
Another popular alternative to a purely manual ontology modeling bringing about an intensified need for quality assurance is partial reuse of existing ontologies. Ontology reuse is a common practice, since for most domains there exists a range of ontologies developed for related purposes and overlapping in their scopes with the

scope of the developed ontology. Similarly to the ontological data obtained using heuristic knowledge acquisition methods, the knowledge reused from external sources does not come for free. In addition to the uncertainty about the accuracy of the content, external ontologies are usually only partially relevant within a new application context. Due to the high complexity of reasoning (up to N2EXPTIME for established modeling languages), the cost of automatic inferencing increases significantly with the size of the ontology. Therefore, special attention has to be paid to the potentially high amount of irrelevant information.

Up to now, we have been discussing quality on a rather abstract level. The concrete definition of quality and the corresponding requirements for its assurance depend on the requirements of the application in question. In literature, various quality criteria catalogs are suggested [PAK and ZHOU 2011, RADULOVIC and GARCIA-CASTRO 2011, BURTON-JONES et al. 2005, STVILIA 2007, BOLOTNIKOVA et al. 2011, FERRÁNDEZ et al. 2009, COLOMB 2002, GÓMEZ-PÉREZ 2004, GRUBER 1995, GRÜNINGER and FOX 1995, OBRST et al. 2007, LEI et al. 2007, HUANG et al. 2012]. Recently, most of the criteria proposed in the literature have been unified by Vrandecic [VRANDEČIĆ 2010], to *accuracy, adaptability, clarity, completeness, computational efficiency, conciseness, consistency* and *organizational fitness*. Within this work, we consider two quality dimensions, *semantic accuracy* and *semantic conciseness*, each of which overlaps with several of the above criteria. The origin and the scope of the two quality dimensions is discussed in the next section.

1.3 Thesis Background, Scope and Objectives

This work has its roots within the ontology development project *NanOn*. The aim of NanOn is to facilitate search in scientific literature within the domain of nanotechnology. To this end, the NanOn ontology contains, on the one hand, concepts and relations covering the major terms of the domain. On the other hand, it contains lexical patterns for identifying occurrences of concepts and relations in natural language texts in order to enable an ontology-based indexing [NIKITINA 2012]. Given

such an indexing, scientific publications can be explored by the means of structured queries using NanOn concepts and relations. This allows for a more detailed specification of information needs, and, thereby, provides a more effective support for complex search tasks. For instance, a search for a particular property of *Indium Tin Oxide layers* by the means of a structured instead of non-structured query over our evaluation corpus increases the precision from around 50% to almost 100%. The reason for that is that many texts refer to both terms, *Indium Tin Oxide* and *layers*, without mentioning *Indium Tin Oxide layers*. In contrast to non-structured queries, which do not consider relations between terms, a structured query only retrieves results, where *Indium Tin Oxide* and *layer* are related to each other by the means of the relation *hasMicrostructure*. A similar improvement of literature search has been reported within the medical domain [GIACOMELLI et al. 2012].

To realize the above ontology-based support, the ontology has been developed according to the corresponding application requirements, determined, on the one hand, by the set of competency questions – generic queries defining the scope of an ontology on a high-level – and, on the other hand, by the characteristics of the underlying text corpus. First, the high-level structure of the ontology consisting of general concepts and relations has been modelled by the domain experts according to the informally stated competency questions. Subsequently, different methods have been applied in order to achieve a decent literature coverage by ontology terms within the scope of the high-level ontology structure. The manual annotation of literature, even though yielding results of the highest quality, was not feasible on the large-scale and did not significantly contribute to the overall coverage. A reuse of existing large ontologies (Chebi [DEGTYARENKO et al. 2008], Chemical Methods Ontology (CMO) [ROYAL SOCIETY OF CHEMISTRY 2012]) covering closely related fields as well as an application of heuristics based on natural language processing yielded a notable amount of new axioms, however, in both cases, at the cost of lower quality. For instance, in case of reused ontologies, the domain experts noticed several incorrect subclass relationships indicating that the external ontologies require a manual inspection. Moreover, a large proportion of the reused ontologies did not fit into the scope of the ontology and had to be excluded in order to increase the efficiency of automatic processing. On the whole, we can categorize the quality problems that had to be addressed within the NanOn project, in particular

within the context of ontology reuse, into *semantic accuracy, semantic conciseness* and *lexico-syntactic accuracy and completeness*. The latter, lexico-syntactic criteria capture factors that determine the readability of the ontology, i.e., the extent to which the intended meaning can be understood by an external person. This includes the comprehensiveness and unambiguity of the labels and natural language definitions. Since the focus of this thesis are the semantic aspects of ontologies, we do not elaborate on these quality criteria, but focus on the remaining two:

Semantic accuracy

An axiom or, accordingly, an ontology is semantically accurate, if it complies with the knowledge about the domain seen from the perspective of the particular application context. Within NanOn, this quality dimension has been addressed with the highest priority, and, with respect to this point, NanOn is not a special case. For instance, Huang et al. [HUANG et al. 2012] conduct a study about the priorities of knowledge engineers on data quality dimensions in genome annotation work and come to the same conclusion.

In most application contexts, semantic accuracy reflects, among other things, objectively measurable aspects such as the absence of logical contradictions within the ontology. However, as the definition suggests, an exact judgement about the semantic accuracy of an axiom requires taking into account the application context. For instance, a very narrow definition of a term can be suitable or even required within a particular application, while it does not reflect reality in general. For this reason, an evaluation of an ontology with respect to the semantic accuracy can be challenging.

Within the NanOn project, various quality assurance techniques have been applied to ensure semantic accuracy. Among others, manual inspection has been carried out, e.g., in order to detect modeling errors or assess the quality of automatically annotated concept and relation instances within literature. Despite a plethora of research on quality assurance with respect to semantic accuracy, we were not able to find a suitable tool support that would allow us to reduce the manual effort of such an inspection. In this work, we consider such an inspection process, in which a domain expert inspects a set of candidate axioms and decides for each of them

whether it is semantically correct with respect to the application in question. The first objective of this thesis is to investigate, how we can reduce the manual effort of such an exhaustive manual inspection, called *ontology revision*, by employing automated reasoning. Based on assumptions underlying standardized ontology languages, e.g., the assumption that the deductive closure of the approved axioms must be disjoint from the set of declined axioms, the aim is to partially automate the above process and thereby to reduce the number of decisions that have to be taken by a domain expert in order to complete the inspection.

Semantic conciseness

An ontology is considered as semantically concise, if the information that can be inferred from it does not exceed the required scope and detailedness of modeling. Semantic conciseness is determined, on the one hand, by the competency questions or a similar informal specification, and, on the other hand, by the general application goals, e.g., high precision and recall of a semantic search engine. This quality problem is usually addressed in case of ontology reuse, where the proportion of irrelevant information is likely to be high and have a notable effect on the performance of ontology engineering tools or the final application itself. But also in case of upcoming manual maintenance procedures, the corresponding quality assurance with respect to the ontology's conciseness can save a significant amount of effort. Within the project NanOn, conciseness was an important aspect due to the reuse of large, only partially relevant ontologies.

Since the set of statements that can be inferred from an ontology is usually infinite, there is no straightforward way of measuring semantic conciseness. There exist, however, good approximations, e.g., the proportion of irrelevant entities within the ontology's vocabulary. The above approximation was, indeed, used within NanOn. Such an evaluation on the vocabulary level yields two subsets, the set of relevant and the set of irrelevant vocabulary elements. While vocabulary separation is arguably simple, the more challenging task is ensuring that no relevant information is lost when the irrelevant information is eliminated from the ontology based on the aforementioned vocabulary separation.

The above task is highly complex already for rather simple logics: even checking if a particular subontology preserves all relevant consequences for a given ontology and relevant vocabulary requires exponential time. Therefore, it is usually not possible to carry out such a "safe" elimination of irrelevant information from an ontology by hand. The difficulty with the above task is that the complexity-optimal solution depends, among other things, on the logic in which the original ontology is expressed. For most representatives of description logics underlying the standardized ontology languages and the corresponding profiles, the problem has been solved. Only for the lightweight logic $\mathcal{EL}$ underlying the OWL EL profile, the problem remains open. The second objective of this thesis is to provide an approach to separating relevant and irrelevant information within an ontology given a particular vocabulary separation for the above logic. To this end, the aim is, on the one hand, to investigate the theoretical problem, i.e., to determine the complexity and identify the bound on the output size, and, on the other hand, to provide practically feasible algorithms and an implementation.

To sum up, the objective of this thesis is to investigate the possibilities to reduce the effort of the corresponding quality assurance with respect to the above introduced dimensions – semantic accuracy and conciseness – by applying automated deduction methods. In the next section, we give an overview of the contributions within this work and outline its structure.

1.4 Guide to the Reader

This work consists of three parts, an introductory part, a main part and a concluding part. The introductory part motivates this work, introduces the necessary foundations and provides an overview of the state of the art. The main part contains the contributions of this thesis, structured according to the corresponding quality dimensions. The last, concluding part summarizes this work and provides an outlook. In the following, we outline each part.

Part I: Introduction

Chapter 1 The first chapter motivates this work, discusses its background and outlines its objectives.

Chapter 2 This chapter formally introduces description logics and further necessary logical foundations. First, we briefly review the syntax and semantics of the description logic $\mathcal{SROIQ}$ (and our notation for it), since it is the logical underpinning of the commonly used standardized ontology language OWL 2 DL. After introducing the fundamental logical notions such as unsatisfiability and entailment, we introduce frequently used fragments of $\mathcal{SROIQ}$ and give an overview of the description logics nomenclature. Since one of the approaches presented within this thesis is not specific to a particular logic, but rather relies on particular properties of logics such as monotonicity, we further discuss some abstract properties of logics relevant within this work.

Chapter 3 This chapter provides an overview of the state of the art for quality assurance with respect to semantic accuracy and conciseness. The approaches to quality assurance with respect to semantic accuracy are divided into four categories according to the high-level strategy being followed into ontology debugging, formalizing constraints, comparing with external sources and inspecting ontologies manually. The approaches for quality assurance with respect to semantic conciseness are mostly theoretical. Here, we give an overview of solved subproblems and the corresponding complexity/decidability results.

Part II: Reasoning Support for Ensuring Accuracy and Conciseness

Chapter 4 In this chapter, we elaborate on the idea to partially automate quality assurance with respect to semantic accuracy using reasoning. First, given the assumption that the deductive closure of the approved axioms must be disjoint from the set of declined axioms, we develop a general framework for the corresponding reasoning support based on the basic notions of revision states and closure. To ensure a decent effectiveness of the reasoning-based support, we propose and compare various axiom ranking techniques

based on the notion of axiom impacts. Since the above reasoning support is computationally expensive, we further introduce decision spaces – auxiliary data structures for organizing dependencies between axioms and allowing for an efficient realization of the above framework. Finally, we present the user front end of our implementation and a detailed comparison to related approaches. The content of this chapter has been peer-reviewed and published at various venues [NIKITINA et al. 2012, NIKITINA et al. 2011a, NIKITINA et al. 2011b, NIKITINA et al. 2011c, NIKITINA 2010].

Chapter 5 In this chapter, we address quality assurance with respect to semantic conciseness, namely separating relevant and irrelevant information within an ontology given a particular vocabulary separation. On the one hand, we solve a theoretical problem of uniform interpolation or forgetting for the lightweight description logic $\mathcal{EL}$, that has been investigated by leading researchers since 2008. We provide a worst-case optimal algorithm and derive the corresponding tight bounds on the output size. On the other hand, we revise the requirements for the corresponding task and show that the existing approaches choose a prioritization of requirements, which is rarely beneficial in practice. We derive two further problem definitions for the above task of separating relevant and irrelevant information within an ontology aiming at a more balanced prioritization of requirements. Further, we provide a practical approximation for solving the redefined problem in polynomial time with the corresponding polynomial bounds on the size of the result. The content of this chapter has been peer-reviewed and published at various venues [NIKITINA and GLIMM 2012, NIKITINA and RUDOLPH 2012, NIKITINA 2011].

Part III: Conclusions

Chapter 6 This chapter summarizes the contributions of this work and draws a conclusion.

Chapter 7 This last chapter points out the limitations of this work and discusses some ideas for future work.

CHAPTER 2

Description Logics

With the wide-spread adoption of the W3C-specified OWL Web Ontology Language [OWL WORKING GROUP 27 October 2009] and its profiles [MOTIK et al. 27 October 2009], description logics [BAADER et al. 2007] have developed into one of the most popular family of formalisms employed for knowledge representation and reasoning. DLs are characterized by the well-understood model-theoretic semantics and computational properties. In this chapter, we introduce the description logic $\mathcal{SROIQ}$ [HORROCKS et al. 2006], which provides the logical underpinning for OWL 2 DL – the most expressive representant of the OWL family. We discuss core formal notions such as satisfiability, entailment, knowledge base equivalence and emulation, and briefly recall the standard reasoning tasks. In addition, we introduce the tractable fragment of $\mathcal{SROIQ}$, $\mathcal{EL}$ [BAADER et al. 2010], which is the formal basis for the OWL EL profile. Since a part of this thesis is not restricted to any concrete logic, but rather requires a set of general properties to hold for the used representation formalism, we further discuss some abstract properties of logics.

2.1 The Description Logic $\mathcal{SROIQ}$

Like description logic knowledge bases in general, a $\mathcal{SROIQ}$ knowledge base defines *concepts*, *roles*, *individuals* (called altogether *entities*) and the relationships between them. As the name suggests, individuals are used to represent concrete objects, e.g., the country *germany*, the chemical element *gold*, or the person *steve-jobs*, while concepts represent groups of individuals with some common properties, e.g., *EuropeanCountry*, *Metal*, *ChiefExecutiveOfficer*[1]. Roles are binary relations that may hold between individuals. The relationships between entities, e.g., *germany* is-a *EuropeanCountry* or *Metal* is-a *ChemicalElement*, are specified by the means of *axioms*. A $\mathcal{SROIQ}$ knowledge base can consist of three parts: *TBox*, *RBox* and *ABox*, containing different types of axioms. While TBox and RBox specify relationships between concepts and roles and their properties, e.g., the transitivity of the part-of role, the ABox states the relationships of individuals to concepts or other individuals via roles, e.g., *steve-jobs* is-founder-of *apple-inc*. In the following, we introduce the syntax of $\mathcal{SROIQ}$, showing how the axioms of the above types can be specified, followed by the semantics of $\mathcal{SROIQ}$, which is defined in a model-theoretic way by the means of interpretations. Finally, we discuss how logical conclusions are drawn in $\mathcal{SROIQ}$.

2.1.1 Syntax

For the definition of $\mathcal{SROIQ}$ syntax, we assume three countably infinite and mutually disjoint sets of names: the set of concept names N_C, the set of role names N_R and the set of individual names N_I. The (usually finite) subsets of these three sets used in a particular knowledge base $\mathcal{O}$ are called the *signature* of $\mathcal{O}$. We denote the latter by $\text{sig}(\mathcal{O})$. In addition to the above introduced sets, description logics make use of two special concepts, namely $\top$ (*top concept*) – the concept that comprises all individuals – and $\bot$ (*bottom concept*) – the concept that has no associated individuals. Moreover, $\mathcal{SROIQ}$ uses a special role, namely the *universal role* u, which connects each pair of individuals with each other. In other words, u is the counterpart of $\top$ for roles. For a role r, the notation r^- is used to

[1] We adhere to the widely used convention to capitalize concept names while using lower case names for individuals and roles.

refer to the inverse role of r. Examples for inverse roles are, for instance *parent-of* and *child-of*. $\mathcal{SROIQ}$ axioms can consist of concept and role expressions, which build upon *atomic* concepts and roles, i.e., elements of the above sets N_C and N_R. The *signature* of a concept expression C, a role expression r or an axiom α is the set of entity names occurring in it and is denoted herein by sig(C), sig(r) and sig(α), respectively. To refer to the subsets of sig consisting only of concept names, role names and individual names, we further use the notations sig$_C$, sig$_R$ and sig$_I$, respectively.

RBox Syntax

The set of $\mathcal{SROIQ}$ role expressions $\mathbf{R}$ can be specified using the following grammar:

$$\mathbf{R} ::= u | N_R | N_R^{-}.$$

In this thesis, we use the symbols r, s, possibly with subscripts, to denote roles. An *RBox* consists, among other things, of *role inclusions* $r_1 \circ ... \circ r_n \dot{\sqsubseteq} s$ with $s \in N_R$ and $r_i \in \mathbf{R}$ for all $1 \leq i \leq n$. A finite set of such role inclusions is also called a *role hierarchy*. Given a role hierarchy with the relation $\dot{\sqsubseteq}$, we denote the corresponding the transitive-reflexive closure of $\dot{\sqsubseteq}$ over $\mathbf{R}$ by $\dot{\sqsubseteq}^*$. A role r is called a *sub-role* (respectively, *super-role*) of a role s, if $r \dot{\sqsubseteq}^* s$ (respectively, $s \dot{\sqsubseteq}^* r$).
In order to ensure that the standard reasoning problems remain decidable, the role hierarchy in $\mathcal{SROIQ}$ is required to be *regular*. Intuitively, the aim of regularity is to prevent a role hierarchy from containing cyclic $\dot{\sqsubseteq}^*$-dependencies, i.e., regular role hierarchies are not allowed to contain equivalent roles (roles r, s such that $r \dot{\sqsubseteq}^* s$ and $s \dot{\sqsubseteq}^* r$). The regularity can be defined by distinguishing between *simple* and *non-simple* roles and ensuring the existence of a strict partial order on the latter in order to avoid $\dot{\sqsubseteq}^*$-cycles. Later on, we will further impose restrictions on the usage of non-simple roles within concept expressions and particular RBox axioms, e.g., those expressing the irreflexivity and asymmetry of roles. The set $\mathbf{R}_n$ of non-simple roles can be specified by the means of an inductive definition as the smallest set fulfilling the following properties:

- For every role $r \in \mathbf{R}$ occurring in a role inclusion axiom of the form $r_1 \circ ... \circ r_n \dot{\sqsubseteq} r$ with $n \geq 2$ holds $r \in \mathbf{R}_n$;

- For every role $r \in \mathbf{R}$ occurring in a role inclusion axiom $s \dot{\sqsubseteq} r$ with some $s \in \mathbf{R}_n$ holds $r \in \mathbf{R}_n$;
- For every role r with $r \in \mathbf{R}_n$ holds $(r)^- \in \mathbf{R}_n$.

The definition of regular role hierarchy is based on a certain strict partial ordering, i.e., irreflexive and transitive, on non-simple roles. A strict partial order $\prec$ on the set of roles $\mathbf{R}$ is called a *regular order*, if $\prec$ satisfies additionally

$$s \prec r \Longleftrightarrow s^- \prec r,$$

for all roles r and s. Assuming such a regular order $\prec$, a role inclusion $w \dot{\sqsubseteq} r$ with $r \in N_R$ is said to be $\prec$*-regular*, if

1. $w = r \circ r$, or
2. $w = r^-$, or
3. $w = r_1 \circ ... \circ r_n$ and $r_i \prec r$, for all $1 \leq i \leq n$, or
4. $w = r \circ r_1 \circ ... \circ r_n$ and $r_i \prec r$, for all $1 \leq i \leq n$, or
5. $w = r_1 \circ ... \circ r_n \circ r$ and $r_i \prec r$, for all $1 \leq i \leq n$.

We say that a role hierarchy is *regular*, if there is exists a regular order $\prec$ on non-simple roles such that each role inclusion is $\prec$-regular.
In addition to role inclusions, a $\mathcal{SROIQ}$ RBox can contain *role characteristics* – statements of the form $\mathsf{Trans}(r)$ (transitivity), $\mathsf{Ref}(r)$ (reflexivity), $\mathsf{Irr}(s)$ (irreflexivity), $\mathsf{Sym}(r)$ (symmetry), $\mathsf{Asy}(s)$ (asymmetry), or $\mathsf{Dis}(s_1, s_2)$ (role disjointness), where s, s_1 and s_2 are simple roles and r may be simple or non-simple. We say that a $\mathcal{SROIQ}$ RBox is regular if its role hierarchy is regular.

TBox Syntax

The set $\mathbf{C}$ of $\mathcal{SROIQ}$ concept expressions can be defined by the means of the following grammar:

$$\mathbf{C} ::= N_C | \top | \bot | \{N_I\} | (\mathbf{C} \sqcap \mathbf{C}) | (\mathbf{C} \sqcup \mathbf{C}) | \neg \mathbf{C} | \exists r.\mathbf{C} | \forall r.\mathbf{C} | \exists s.\mathsf{Self} | \leqslant ns.\mathbf{C} | \geqslant ns.\mathbf{C}$$

where n is a natural number, $r \in N_R$ and $s \in \mathbf{R}$. We use symbols A, B to denote atomic concepts and C, D to denote arbitrary concepts, i.e., concept expressions. A *general terminology* or *general TBox* consists of *general concept inclusions* – axioms of the form $C \sqsubseteq D$. Additionally, *concept equivalence* axioms – statements of the form $C \equiv D$ – can be used as a shorthand for $C \sqsubseteq D$ and $D \sqsubseteq C$. We say that C is a *subsumer* of D (respectively, *subsumee*), if $D \sqsubseteq C$ (respectively, $C \sqsubseteq D$).

ABox Syntax

Knowledge bases can also include a set of axioms about individuals, called *ABox*. The latter consists of *assertions* that can be one of the following:

- $C(a)$ (concept assertion) ,
- $r(a, b)$ (role assertion),
- $\neg r(a, b)$ (negative role assertion),
- $a \approx b$ (equality assertion),
- $a \not\approx b$ (inequality assertion),

where $a, b \in N_I$, $r \in \mathbf{R}$ and $C \in \mathbf{C}$. We use the symbols a, b to denote individuals.

2.1.2 Semantics

The semantics in description logics is defined in a model-theoretic way, i.e., by the means of interpretations. An interpretation $\mathcal{I}$ is given by the *domain* – a non-empty, possibly countably infinite set $\Delta^{\mathcal{I}}$ of individuals – and an *interpretation function* – a function $\cdot^{\mathcal{I}}$ assigning each individual $a \in N_I$ an element $a^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each concept $A \in N_C$ a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and each role $r \in N_R$ a subset $r^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Intuitively, this means that each domain element $a \in \Delta^{\mathcal{I}}$ with $a \in A^{\mathcal{I}}$ is an instance of the concept A, and, each pair a, b of domain elements with $\langle a, b\rangle \in r^{\mathcal{I}}$ for some role r are connected by this role. The interpretation of the special concepts and roles, namely $\top, \bot, u$, is fixed to $\Delta^{\mathcal{I}}, \emptyset$ and $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, respectively.

Syntax	Semantics
$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\{a\}$	$\{a^{\mathcal{I}}\}$
$\forall r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \langle x, y\rangle \in r^{\mathcal{I}}$ implies $y \in C^{\mathcal{I}}\}$
$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \langle x, y\rangle \in r^{\mathcal{I}}$ for some $y \in C^{\mathcal{I}}\}$
$\exists s.\mathsf{Self}$	$\{x \in \Delta^{\mathcal{I}} \mid \langle x, x\rangle \in s^{\mathcal{I}}\}$
$\leqslant ns.C$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid \langle x, y\rangle \in s^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\} \leq n\}$
$\geqslant ns.C$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid \langle x, y\rangle \in s^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\} \geq n\}$
r^-	$\{\langle x, y\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \langle y, x\rangle \in r^{\mathcal{I}}\}$

Table 2.1: *Inductive definition of an interpretation $\mathcal{I}$ in $\mathcal{SROIQ}$.*

The interpretation for arbitrary concepts and roles is defined inductively as shown in Table 2.1.

A particular type of interpretations, namely *models*, play a central role in inferencing. The property of being a model is defined based on the notion of *satisfiability*. An axiom α is *satisfied* by an interpretation $\mathcal{I}$, in symbols $\mathcal{I} \models \alpha$, if the corresponding condition in Table 2.2 holds for $\mathcal{I}$ and α. An interpretation is a *model* of a knowledge base $\mathcal{O}$, in symbols $\mathcal{I} \models \mathcal{O}$, if it satisfies all of its axioms. Moreover, a knowledge base $\mathcal{O}$ is called *satisfiable* or *consistent* if it has a model, and it is called *unsatisfiable* or *inconsistent*, otherwise.

We say that a knowledge base $\mathcal{O}$ *entails* an axiom α (or that α is a consequence of $\mathcal{O}$), in symbols, $\mathcal{O} \models \alpha$, if α is satisfied by all models of $\mathcal{O}$. For instance, $\mathcal{O} = \{\mathit{Human} \sqsubseteq \mathit{Mammal}, \mathit{Mammal} \sqsubseteq \mathit{Animal}\}$ entails the axiom $\mathit{Human} \sqsubseteq \mathit{Animal}$, since all models of $\mathcal{O}$ satisfy the condition $\mathit{Human}^{\mathcal{I}} \subseteq \mathit{Animal}^{\mathcal{I}}$. Moreover, we can say that $\bot \sqsubseteq \top$ is a consequence of any knowledge base, while no consistent knowledge base entails $\top \sqsubseteq \bot$. In fact, an inconsistent knowledge base entails any axiom, since the set of models that have to satisfy the axiom is empty. A set of all axioms entailed by a knowledge base $\mathcal{O}$ is called the *deductive closure* of $\mathcal{O}$.

Axiom α	Conditions for $\mathcal{I} \models \alpha$
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$r_1 \circ ... \circ r_n \dot{\sqsubseteq} r$	$r_1^{\mathcal{I}} \circ ... \circ r_n^{\mathcal{I}} \subseteq r^{\mathcal{I}}$
$\mathsf{Trans}(r)$	$r^{\mathcal{I}} \circ r^{\mathcal{I}} \subseteq r^{\mathcal{I}}$
$\mathsf{Ref}(r)$	$\langle x, x \rangle \in r^{\mathcal{I}}$ for all $x \in \Delta^{\mathcal{I}}$
$\mathsf{Irr}(s)$	$\langle x, x \rangle \notin s^{\mathcal{I}}$ for all $x \in \Delta^{\mathcal{I}}$
$\mathsf{Dis}(s_1, s_2)$	$s_1^{\mathcal{I}} \cap s_2^{\mathcal{I}} = \emptyset$
$\mathsf{Sym}(r)$	if $\langle x, y \rangle \in r^{\mathcal{I}}$ then $\langle y, x \rangle \in r^{\mathcal{I}}$ for all $x, y \in \Delta^{\mathcal{I}}$
$\mathsf{Asy}(s)$	if $\langle x, y \rangle \in s^{\mathcal{I}}$ then $\langle y, x \rangle \notin s^{\mathcal{I}}$ for all $x, y \in \Delta^{\mathcal{I}}$
$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
$r(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in r^{\mathcal{I}}$
$\neg r(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \notin r^{\mathcal{I}}$
$a \approx b$	$a^{\mathcal{I}} = b^{\mathcal{I}}$
$a \not\approx b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$

Table 2.2: *Conditions for satisfiability of an axiom α by an interpretation $\mathcal{I}$.*

Based on the above definition of entailment, we can generalize the definition to the level of knowledge bases as follows: A knowledge base $\mathcal{O}_1$ *entails* a knowledge base $\mathcal{O}_2$, in symbols, $\mathcal{O}_1 \models \mathcal{O}_2$, if for any $\alpha \in \mathcal{O}_2$ holds $\mathcal{O}_1 \models \alpha$. Two knowledge bases, $\mathcal{O}_1, \mathcal{O}_2$, are *equivalent* ($\mathcal{O}_1 \equiv \mathcal{O}_2$), if they mutually entail each other, i.e., $\mathcal{O}_1 \models \mathcal{O}_2$ and $\mathcal{O}_2 \models \mathcal{O}_1$.

2.2 Knowledge Base Emulation

As the constructs in $\mathcal{SROIQ}$ are rather redundant, i.e., there are several ways to express semantically equivalent statements, many procedures employ normalization – a syntactic, meaning-preserving transformation of a knowledge base. Also in this thesis, a particular normalization is employed in order to simplify the subsequent processing. Within the context of normalization, in addition to the above introduced equivalence relation on knowledge bases, the notion of *emulation* plays a central role. The latter is a weaker form of knowledge base equivalence, reflect-

ing that one of the two knowledge bases introduces some additional vocabulary, which, however, does not change the semantics of the common vocabulary elements. Formally, a knowledge base $\mathcal{O}_1$ *emulates* a knowledge base $\mathcal{O}_2$, if the following conditions are true:

1. For any interpretation $\mathcal{I}$ holds $\mathcal{I} \models \mathcal{O}_1 \Rightarrow \mathcal{I} \models \mathcal{O}_2$;

2. For any model $\mathcal{I}_2$ of $\mathcal{O}_2$ there is a model $\mathcal{I}_1$ of $\mathcal{O}_1$ with the same domain as $\mathcal{I}_2$ such that for any $\Theta \in \text{sig}_C(\mathcal{O}_2) \cup \text{sig}_R(\mathcal{O}_2) \cup \text{sig}_I(\mathcal{O}_2)$ holds $\Theta^{\mathcal{I}_1} = \Theta^{\mathcal{I}_2}$.

2.3 Standard Reasoning Tasks

One of the aims of formal semantics for knowledge bases is to enable automatic inferencing – deriving logical consequences of a knowledge base. Of course, there are many particular questions that can be answered by applying inferencing. A range of such questions is very common in practice and, therefore, the corresponding reasoning tasks computing the answers are implemented by most of the modern reasoners. The latter reasoning tasks, called *standard reasoning tasks*, are the following:

Inconsistency checking: Is $\mathcal{O}$ inconsistent?

Concept satisfiability checking: Given a concept C, is there a model $\mathcal{I}$ of $\mathcal{O}$ with $C^{\mathcal{I}} \neq \emptyset$?

General concept subsumption checking: Given two concepts C, D, does $\mathcal{O}$ entail $C \sqsubseteq D$?

Instance checking: Given a concept C and individual name a, does $\mathcal{O}$ entail $C(a)$?

Other commonly used tasks that can be seen as an aggregation of the above given ones are *classification* – deriving of all subsumption relations between atomic concepts – and *instance retrieval* – deriving of all instances for a given concept. The latter tasks are also very common and considered by some authors as standard.

2.4 Common Fragments of the Logic $\mathcal{SROIQ}$

The cost of reasoning in $\mathcal{SROIQ}$ are fairly high (N2EXPTIME [HORROCKS et al. 2006]). For this reason, ontologies in practice usually do not exhaust its expressivity, but are specified using less expressive fragments with lower computational complexity. Among them is the liteweight family of description logics DL Lite used for ontology-based data access as well as the tractable description logic $\mathcal{EL}$, being the logical underpinning of the OWL EL profile. Within this thesis, $\mathcal{EL}$ is considered in detail in a rather technical context in Chapter 5, while DL-Lite is used within the evaluation in the same chapter. Further, various fragments are mentioned within the next chapter. Here, we explicitly recall the syntax of the fragments relevant within this work. The semantics of the corresponding constructs has already been given in Section 2.1.

2.4.1 The Fragment $\mathcal{EL}$

In contrast to $\mathcal{SROIQ}$, $\mathcal{EL}$ knowledge bases do not include an RBox, but can only consist of a TBox and an ABox. An $\mathcal{EL}$ TBox is a set of concept inclusions formed using the set $\mathbf{C}_{\mathcal{EL}}$ of general $\mathcal{EL}$ concepts given by the following grammar:

$$\mathbf{C}_{\mathcal{EL}} ::= N_C | \top | (\mathbf{C}_{\mathcal{EL}} \sqcap \mathbf{C}_{\mathcal{EL}}) | \exists r.\mathbf{C}_{\mathcal{EL}}$$

with $r \in N_R$. An $\mathcal{EL}$ ABox can consist of the following types of assertions:

- $C(a)$ (concept assertion) ,
- $r(a, b)$ (role assertion),
- $a \approx b$ (equality assertion),

where $a, b \in N_I$, $r \in \mathbf{R}$ and $C \in \mathbf{C}$.
The reason why $\mathcal{EL}$ is tractable is that it has the *finite minimal model property*, i.e., for any $\mathcal{EL}$ knowledge base, there exists a finite minimal model, and, at the same time, it is *closed under simulation* [LUTZ et al. 2010]. The latter means that checking for the existence of simulations on minimal models (that can be done in polynomial time) is sufficient in $\mathcal{EL}$ to solve reasoning problems such as subsumption checking. None of these two properties hold for $\mathcal{SROIQ}$.

2.4.2 DL-Lite

The family of description logics DL-Lite has been designed to enable the usage of ontologies as a conceptual view over data repositories. The data complexity of query answering is within LOGSPACE for most members of the family. Another particularity is that queries over DL-Lite ontologies can be rewritten as SQL queries so that standard database query engines can be used. The syntax of DL-Lite$_{\texttt{bool}}$, the most expressive language of the family, is given as follows:

$$\mathbf{R} ::= N_R \mid N_R^-,$$

$$\mathbf{B} ::= \bot \mid \top \mid N_C \mid \exists s \mid \leqslant n s \mid \geqslant n s, \quad \mathbf{C} ::= \mathbf{B} \mid \neg \mathbf{C} \mid (\mathbf{C} \sqcap \mathbf{C}) \mid (\mathbf{C} \sqcup \mathbf{C}),$$

where n is a natural number and $s \in \mathbf{R}$. A concept inclusion in DL-Lite$_{\texttt{bool}}$ has the form $C_1 \sqsubseteq C_2$ with $C_i \in \mathbf{C}$. Concept inclusions in DL-Lite$_{\texttt{horn}}$, a popular, Horn fragment of DL-Lite$_{\texttt{bool}}$, are restricted to $\bigsqcap_{0 \leq i \leq n} D_i \sqsubseteq E$ for some natural number n with $D_i, E \in \mathbf{B}$.

The complexity of DL-Lite$_{\texttt{horn}}$ is lower due to the Horn property: While checking subsumption is CONP-complete in DL-Lite$_{\texttt{bool}}$, it is PTIME-complete in DL-Lite$_{\texttt{horn}}$. The data complexity of the query answering problem for DL-Lite$_{\texttt{bool}}$ is CONP-complete, while for DL-Lite$_{\texttt{horn}}$ knowledge bases it is in LOGSPACE [ARTALE et al. 2007]. The underlying models of DL-Lite knowledge bases can be infinite as neither DL-Lite$_{\texttt{bool}}$ nor DL-Lite$_{\texttt{horn}}$ has the finite model property [CALVANESE et al. 2005].

2.4.3 The Fragment $\mathcal{ALC}$ and Some Extensions

Attributive Concept Language with Complements, $\mathcal{ALC}$, is the least expressive description logic that comprises all Boolean concept constructors [LUTZ et al. 2010]. $\mathcal{ALC}$ is seen as the basic DL from which more expressive DLs are derived by adding further constructs. Like $\mathcal{EL}$ and DL-Lite, it does not allow for role inclusions. The syntax is given by the following grammar:

$$\mathbf{C} ::= N_C \mid \top \mid \bot \mid (\mathbf{C} \sqcap \mathbf{C}) \mid (\mathbf{C} \sqcup \mathbf{C}) \mid \neg \mathbf{C} \mid \exists r.\mathbf{C} \mid \forall r.\mathbf{C},$$

where $r \in N_R$. There is a range of extensions, all of which are comprised by $\mathcal{SROIQ}$ and some of which are referenced throughout this work in various combinations. The extensions of concept constructors referenced in this context are nominals and qualified cardinality restrictions, denoted by symbols $\mathcal{O}$ and $\mathcal{Q}$, respectively.[2] Further extensions are inverse roles ($\mathcal{I}$) and role subsumptions (not allowing for role chains) ($\mathcal{H}$). $\mathcal{S}$ is an abbreviation for $\mathcal{ALC}$ with transitive roles. For instance, $\mathcal{SHOIQ}$, the logical underpinning of OWL 1, is an extension of $\mathcal{ALC}$ with all of the above constructs. The complexity of reasoning varies depending on the extensions from EXPTIME-complete in case of $\mathcal{ALC}$ [DONINI and MASSACCI 2000] to NEXPTIME-complete in case of $\mathcal{SHOIQ}$ [TOBIES 2001].

2.5 Abstract Properties of Knowledge Representation Languages

Some approaches discussed in this thesis are not restricted to a particular logic, but only require that taking all consequences in this logic is a closure operation. The latter requirement is fulfilled, if the underlying entailment relation $\models$ has the following properties:

Extensitivity: any statement logically follows from itself: $\{\alpha\} \models \alpha$,

Monotonicity: adding further statements does not invalidate previous consequences: $\mathcal{O} \models \alpha$ implies $\mathcal{O} \cup \mathcal{O}' \models \alpha$,

Idempotency: extending an ontology with an entailed axiom does not yield new consequences: $\mathcal{O} \models \alpha$ and $\mathcal{O} \cup \{\alpha\} \models \beta$ imply $\mathcal{O} \models \beta$.

[2] Extension symbols are usually simply added to the name of the extended logic, e.g., $\mathcal{ALCO}$.

CHAPTER 3

State of the Art

The area of research on quality assurance applicable to semantic technologies is very broad. Among others, ontology engineering can draw from the long experience in database and software engineering communities. For instance, the principles of code, schema and data reviews as well as test case design can analogously be applied in order to identify quality problems in ontologies. In addition, there are many ways to indirectly evaluate the quality of ontologies, as it is the case in general for artifacts combined to complex products. For instance, the quality of an ontology used within a semantic search engine can indirectly be evaluated in terms of the established precision and recall measures. In other words, the dimensionality of possibilities to detect and resolve quality problems in ontologies is far too high to fit into a scope of a thesis. Here, we discuss existing approaches that are tailored towards ontologies as a semantic resource and address the two quality dimensions central in this thesis, namely semantic accuracy (Section 3.1) and semantic conciseness (Section 3.2). For existing work on the other two quality dimensions discussed in Section 1.2, see, for instance, [GOOCH 2012, ROGOZAN and PAQUETTE 2005, VERSPOOR et al. 2009, MAEDCHE and STAAB 2002, PARK et al. 2011, CORCHO et al. 2009] (lexico-syntactic accuracy) and [BREWSTER et al. 2004, RECTOR et al. 2011, ELHADAD et al. 2009, OUYANG et al. 2011] (lexico-syntactic completeness).

A wide range of proposed techniques for enhancing the quality of ontologies approach the problem on a rather general level in the sense that they are not specific to a particular quality dimension given in Section 1.2. Instead, they aim at fully exploiting the possibilities of the underlying techniques or resources, and, based on these possibilities, define the scope of applicability. For instance, tagging ontologies with feedback provided by users is not specific to any particular quality aspect and can easily span over various dimensions. Such high-level, dimension-spanning approaches are discussed in Section 3.3.

3.1 Semantic Accuracy

With an exception of the rare cases in which there exists a formalized gold standard, detecting and resolving semantic accuracy problems is a non-trivial task. Perhaps, the most well-known and most frequently addressed semantic accuracy problem is the logical inconsistency or incoherence of knowledge bases. Indeed, logical inconsistency is severe, since it excludes any usage scenario involving standard reasoning: any statement can be inferred from an inconsistent ontology, and, therefore, standard reasoning does not yield any meaningful results in the latter case.

There is a plethora of research on how to support ontology engineers in *debugging ontologies* (Subsection 3.1.1), i.e., identifying the source of logical inconsistency or incoherence, applied, for instance, in [SCHNOES et al. 2009, FELFERNIG et al. 2009, FELDMAN et al. 2009, DUONG et al. 2010] to enhance the quality of ontologies. Being an important, potentially frequently applied procedure during ontology development, the debugging of ontologies only yields a very weak guarantee in terms of semantic accuracy. Most modeling errors do not result in inconsistency or incoherence and, therefore, would remain undetected. A broader spectrum of quality problems can be detected by formalizing particular application requirements in the form of automatically processable constraints that the ontology has to satisfy (Subsection 3.1.2). The existing approaches of the latter category vary in the formalisms used for expressing the corresponding constraints as well as the amount of manual effort required to enable automatic checking. A special, but well-represented case within the above category are approaches com-

paring ontologies to some external sources, e.g., other ontologies or structured data sets, in order to find indications for quality problems (Subsection 3.1.3). On the whole, it can be said for representatives of this category that the detectable problems are limited to those explicitly anticipated by the ontology engineers or taken into account within the corresponding external resource. Given that most modeling errors are difficult to foresee, as reported, for instance, in the study by Ceusters et al. [CEUSTERS et al. 2004], in addition to such an automated constraint checking, further quality assurance is necessary in order to sufficiently address applications underlying strict quality requirements. The arguably most general approach to detecting semantic accuracy problems is manual inspection of the ontology's axioms (Subsection 3.1.4), which can reveal problems not being anticipated by the ontology engineers. Due to the high amount of required user interaction, manual inspection is one of the most costly alternatives. Thus, it is usually applied to ontology fragments with a high estimated probability of quality problems that do not match any known, formalized constraints. For instance, the output of automatic knowledge acquisition tools can contain a large proportion of problematic data that is difficult to quality-assure automatically. But also in case of ontology fragments, for which the quality in general is highly critical due to their central role within an application, such an inspection yields a necessary additional level of certainty. In this section, we discuss the existing approaches of the four aforementioned categories – ontology debugging, constraint formalization, comparison to other ontologies and manual inspection.

3.1.1 Ontology Debugging

As discussed above, inconsistency and incoherence of ontologies are a strong indication of modeling errors and can substantially hinder the application of reasoning. Therefore, it is, in most cases, necessary to identify axioms causing the corresponding problem and remove or correct the cause as quickly as possible. The former task is truly non-trivial for large and complex ontologies and usually can not be carried out by hand. The corresponding tool support is usually based on a particular non-standard reasoning task, namely computing *justifications* of inconsistency and incoherence, i.e., minimal subsets of the knowledge base that cause them. Yet,

even with the help of reasoning, the corresponding diagnosis remains difficult. One reason is that, in general, there can be a large number of different justifications for a single consequence of a knowledge base, and only a human expert can decide, which of the justifications is the true cause of the problem. Therefore, the aim of the tool support is not only to determine a small set of potentially incorrect axioms, but also to ensure that the manual effort required for identifying the cause is as low as possible.

Existing approaches vary in granularity of diagnosis, the extent to which the structure of the knowledge base is modified, the type of problem they address (inconsistency, incoherence or both), the considered part of the knowledge base (TBox and RBox or also ABox), supported logic and algorithm complexity. For an overview of early approaches, see, for instance, [HAASE and QI 2007, BELL et al. 2007]. More recent research concentrates on optimizations for special cases, e.g. exploitation of some additional information, or novel techniques to combine user interaction with reasoning, e.g., interactive exclusion of diagnoses based on user decisions.

An example for a diagnosis and repair tool within the context of AIFB is the system RaDON [JI et al. 2009]. Among other things, RaDON is capable of computing a set of minimal inconsistent subontologies for any given ontology. Additionally, it implements a simple heuristic-based support for fully automated repair, which iteratively removes axioms from minimal inconsistent subontologies until the inconsistency is eliminated. The latter is, however, only applicable in cases where a loss of potentially relevant and correct axioms is acceptable.

Examples for approaches exploiting additional information are stated by Meilicke et al. [MEILICKE et al. 2007] and Du et al. [DU and SHEN 2008]. The authors of the latter work propose an approach to automatically computing minimum cost diagnoses for ABoxes assuming that each removable ABox assertion is given a removal cost. Meilicke et al. apply this strategy in the context of ontology mapping diagnosis based on the confidence values assigned to mappings during the matching. While such approaches do not require user interaction, they do not guarantee that the best computed diagnosis is the one the user is looking for.

A very different, logic- and syntax-based approach to automating the diagnosis using additional information is presented by Schockaert et al.

[SCHOCKAERT and PRADE 2010]. The authors investigate how to appropriately relax axioms by the means of merging operators that are based on possibilistic logic and background knowledge indicating the extent, to which axioms are similar to each other.

An example for an approach to combining user interaction with reasoning is, for instance, the debugging approach by Shchekotykhin et al. [SHCHEKOTYKHIN et al. 2012]. The authors support the user in finding the right diagnosis by asking him questions and, based on his decisions, excluding diagnoses by the means of reasoning.

All in all, we can say that ontology debugging is an important means to enhance the quality of ontologies. However, the spectrum of quality problems that can be detected in this way is very narrow, since many modeling errors do not result in inconsistency or incoherence. Thus, in case of professional ontology development, more general quality assurance methods are required in order to provide a stronger guarantee of semantic accuracy.

3.1.2 Formalized Constraints

Even though inconsistency and incoherence are the most frequently addressed problems, substantial disadvantages can already be caused by errors in consistent and coherent ontologies, leading to an unpredictable or inefficient behaviour of the corresponding software system relying on it. In case that it is known a-priori, which kind of quality problems are likely to occur, it can be possible to automate their detection by formalizing the corresponding constraints. Similarly to ontology debugging, such automatic constraint checks are suitable for a frequent application during the development of the ontology.

The constraints can be specific for the particular application, or context-independent, e.g., those representing good and bad ontology modeling practices. Metamodeling based on notions from philosophy is a popular example for the use of the application-independent kind of constraints. The best known framework for applying philosophy for quality assurance in ontology engineering is *OntoClean*, proposed by Guarino and Welty [GUARINO and WELTY 2002]. The authors introduce several philosophical notions (essentiality, rigidity, unity, etc.) that char-

acterize ontology entities. Based on these notions, the framework provides a set of constraints that prohibit particular relationships between entities based on their characteristics. In this way, the framework can be applied during modeling in order to discover potentially problematic design decisions such as the use of subsumption relationships for expressing a part-whole relation or some meta-level characteristics of concepts.

Various tools supporting the application of OntoClean have been developed by the research community. In order to minimize the manual effort of "tagging" ontology entities with the corresponding characteristics, Völker et al. propose the tool *AEON* [VÖLKER et al. 2005] for automatic ontology tagging with the characteristics introduced in OntoClean. A complementary contribution has been done by Glimm et al. [GLIMM et al. 2010]. The authors introduce a metamodeling encoding scheme with full reasoning support through standard OWL 2 reasoning systems. The capabilities of the latter metamodeling scheme is not limited to the OntoClean framework, but allows for a formalization of a broad spectrum of constraints.

Another dimension, according to which we can classify the existing approaches is the required expressivity for the constraints in question. The latter determines, which formalisms and tools are necessary for the corresponding formalization and checks. One possibility is to use the standardized ontology and query languages, e.g., [VRANDEČIĆ and GANGEMI 2006, VRANDEČIĆ 2010, FELLMANN et al. 2011]. Vrandecic et al. [VRANDEČIĆ and GANGEMI 2006] point out the analogy between such constraint-based ontology evaluation and unit tests commonly used in software development by coining an entailment-based checking of constraints expressed in OWL as *ontology unit tests*. Another approach [VRANDEČIĆ 2010] uses the SPARQL query language [PRUD'HOMMEAUX and SEABORNE 2008] over the ontology graph in order to discover *anti-patterns* – strong indicators for problems in an ontology. This is yet another analogy to software engineering, where anti-patterns have been introduced already in 1995 by Koenig [KOENIG 1995]. Another example of constraints formulated as queries is the approach by Fellmann et al. [FELLMANN et al. 2011]. The authors check the semantic accuracy of business process models using queries in conjunction with an ontology-based process representation.

Yet, constraints do not need to be bound by the expressivity of the standardized ontology and query languages. They can as well be based on more expressive logical formalisms such as Semantic Web Rule Language (SWRL) [HORROCKS et al. 2004] or autoepistemic constructs, e.g., K- and A-operators introduced by Grimm et al. [GRIMM and MOTIK 2005] and used for constraint formulation. The latter formalism allows, for instance, for imposing restrictions underlying closed world assumption, e.g., specify that individuals with particular properties have to be present in the knowledge base. In this way, we can, for instance, ensure that every bank account instance in a knowledge base consists of an account number and a bank code. Such a constraint can not be formulated using OWL.

Arpinar et al. [ARPINAR et al. 2006] propose an approach to specifying constraints as rules in RuleML [BOLEY et al. 2010] – a rule-based formalism for ontology specification. Fürber et al. [FÜRBER and HEPP 2011] introduce *data quality rules* – executable definitions that allow the identification and measurement of particular semantic accuracy problems. Yeh et al. [YEH et al. 2011] describes an approach based on *Multilayered extended semantic Networks* (MultiNets) [HELBIG 2005] – a language for meaning representation of natural language expressions – and an automated theorem prover.

Also concerning constraint formalization, we can say in conclusion that the spectrum of quality problems that can be detected is not broad enough to completely cover the needs for quality assurance in professional ontology development. Since the extent to which the requirements can be formalized by the ontology engineers is usually very limited due to the rather informal nature of application requirements and the high dimensionality of the errors' origin, many quality problems remain undetected.

3.1.3 Comparing to Other Ontologies

Another frequently applied and relatively low-cost strategy of identifying semantic accuracy problems is comparing formalized sources, e.g., two different ontologies, with each other. The assumption is that disagreements are a potential indication for semantic accuracy problems. The idea is rather simple and

has been applied for quality assurance in ontologies already in the past century. For instance, Rogers et al. [ROGERS et al. 1998] cross-validate two ontologies, Read Thesaurus and GALEN. After an integration, the inferred relationships in GALEN are compared with those manually created in Read Thesaurus to identify disagreements. Ceusters et al. [CEUSTERS et al. 2004] integrate SNOMED with LinkKBase, a medical ontology containing a large number of formalized constraints, and check the results for indications of modeling errors in SNOMED. Brewster et al. [BREWSTER et al. 2004] suggest to measure the degree of structural fit between an ontology and a corpus of documents by comparing the ontology with the ontology of hidden "topics" generated from the corpus based on clustering. Köhler et al. [KÖHLER et al. 2011] propose a tool, *GULO* (Getting an Understanding of LOgical definitions), for integrating an ontology with an external ontology that is assumed to be a gold standard. The integration is done by the means of definitions connecting the ontologies with each other. After the integration, the inferred relationships can be compared. Park et al. [PARK et al. 2011] propose the system *GOChase* that uses the hierarchical structure of the Gene Ontology and the NCBI taxonomy as well as twenty seven different biological databases. The aim of the system is to detect inconsistencies in ontology-based annotations by the means of reasoning. Another example of using external ontologies for identifying quality problems is stated by Legg et al. [LEGG and SARJANT 2012]. The authors use the ontological structure of Cyc for accuracy-checking while learning an ontology from Wikipedia.

To summarize the pros and contras, a comparison of an ontology with some external sources is an arguably low-cost alternative for detecting semantic accuracy problems. However, this strategy is only applicable in case that such sources exist and it is questionable whether the identified disagreements indeed indicate quality problems. Also, the proportion of semantic accuracy problems identified in this way is limited by the characteristics of the corresponding external resource.

3.1.4 Manual Inspection

The approaches of the above discussed categories either assume the existence of an a-priori known schema for detecting the corresponding semantic accuracy prob-

lems, or the existence of a formalized information source that can serve as a reference. In other words, they assume the availability of a formal specification of the semantic accuracy for the corresponding ontology. Moreover, while these strategies allow for a low-cost error detection due to the possible automation, in general, it is unlikely that all semantic accuracy problems will be detected in this way. Semantic accuracy problems are as multifaceted as their sources and are determined by the corresponding application context. They are difficult to foresee, not to mention formalize to the full extent. For this reason, semantic accuracy is usually not or only partially formally specified and cannot be verified completely by the means of an automatic procedure. For the same reason, also the extent, to which external formal specifications used for a comparison comply with the particular application requirements is unlikely to be known before a detailed manual inspection. In contrast to that, manual inspection does not require an availability of fully formalized application requirements, but can be carried out by the ontology engineers based on their understanding of the domain and the application context. This allows for a detection of a broader range of errors and yields valuable detailed insights into the characteristics of the current semantic accuracy problems that can serve, among others, as a basis for a partial automation of the corresponding quality assurance. However, in particular in case of large and complex ontologies, manual inspection is costly in terms of human effort, and requires an appropriate tool support.

The research on supporting manual inspection of ontological data is still at an early stage. We are aware of three approaches [MEILICKE et al. 2008, JIMÉNEZ-RUIZ et al. 2009a, JIMÉNEZ-RUIZ et al. 2009b] that aim at supporting manual inspection of ontologies, two of which are applied in the context of ontology mapping revision. All three approaches define a set of logic-based criteria that are used to automatically detect incorrect statements based on decisions taken by the expert.

Meilicke et al. [MEILICKE et al. 2008] aim at reducing the manual effort of mapping revision by propagating implications of expert decisions on the accuracy of a mapping based on incoherence and entailment. To this end, the authors interpret mapping correspondences as bridge rules [SERAFINI and TAMILIN 2005], for which the corresponding logical notions are introduced. The decision-taking is then partially automated such that any bridge rule entailed by the set of approved

bridge rules is automatically approved and each bridge rule that would make the set of approved bridge rules together with the two ontologies incoherent is automatically marked as incorrect. This work is very related to one of the approaches introduced within this thesis. In fact, the idea to propagate the logical consequences of expert decisions originates from the latter work. In this thesis, we elaborate on this idea and generalize it in such a way that the capabilities of reasoning are used to the full extent given the assumption that axioms marked as incorrect should not be logical consequences of the approved axioms. In contrast to that, within the approach by Meilicke et al. only some particular logical implications (based on incoherence and entailment) of expert decisions are propagated, i.e., the possibilities of automation are not used to the full extent. We discuss further differences between the two approaches in detail in Chapter 4.

In contrast to the above discussed approach to mapping revision, the two other approaches, *ContentMap* [JIMÉNEZ-RUIZ et al. 2009b] (applied in the context of mapping revision) and *ContentCVS* [JIMÉNEZ-RUIZ et al. 2009a] (supporting integration of changes into an evolving ontology) focus on the visualization of consequences and user guidance in case of difficult evaluation decisions. The authors do not aim at reducing the number of decisions that have to be taken during a manual evaluation. In the contrary, these approaches selectively materialize and visualize the logical consequences of the axioms under investigation and support the revision of those consequences. Subsequently, the approved and declined axioms are determined in correspondence with the revision of the consequences.

3.2 Semantic Conciseness

The task of improving the conciseness of an ontology while preserving the relevant information highly depends on the underlying formalism and the definition of relevance. In case that the relevance is defined in a syntactic way, e.g., the atomic subsumers of a particular concept explicitly stated within the ontology are considered as the only relevant information, the task of separating relevant and irrelevant information can be carried out by the means of simple syntactic transformations, e.g., [NOY and MUSEN 2003, SEIDENBERG and RECTOR 2006]. An overview of syntax-based approaches can be found, for instance, in

[STUCKENSCHMIDT et al. 2009]. If, however, the relevance is defined based on semantics, the above task boils down to separating logical consequences of the knowledge base into the set of relevant and irrelevant ones and subsequently determining a knowledge base that ideally entails only the relevant consequences. Unfortunately, such a separation of the original knowledge base into the corresponding sets of relevant and irrelevant axioms is not always possible. Each of the existing approaches follows one of the two following strategies to resolve this situation:

- The first strategy is to exclude as much irrelevant information as possible by determining a minimal subset of the knowledge base required to preserve all relevant consequences. Such a computation of a subontology is an established non-standard reasoning task called *module extraction*. If we follow this strategy, we have to tolerate the presence of some irrelevant information within the resulting knowledge base.

- The second strategy is to change the syntactic structure of the knowledge base by exchanging the explicitly given axioms by some logical consequences from the deductive closure in such a way that the new knowledge base allows for an exact separation of relevant and irrelevant axioms. The latter transformation is referred to as *forgetting* or *uniform interpolation* .

In both of these cases, the intricacy of the problem and the corresponding complexity-optimal algorithms highly differ depending on the underlying logic. A further important factor determining the problem complexity is the concrete, application-specific definition of relevance. For instance, if a knowledge base is used for answering conjunctive queries, the ontology language and the query language, i.e., the logic used for inferring the consequences of the knowledge base, do not coincide in case of $\mathcal{SROIQ}$ knowledge bases. In the following, we discuss for both categories, module extraction and forgetting-based approaches, the constellations of ontology and query languages, for which the corresponding problem has been investigated so far, and give the corresponding complexity results.

3.2.1 Module Extraction Approaches

Deciding whether a subset of a knowledge base preserves all consequences (expressed in the corresponding query language) over a given relevant signature is usually harder than standard reasoning tasks for the corresponding ontology language, even if the query and the ontology languages coincide. We now consider the different definitions of modules (basically determined by the chosen query language) that have arisen over the last two decades and give the obtained complexity results.

To the best of our knowledge, the first definition of a module for ontologies goes back to Garson [GARSON 1989]. His definition of modules corresponds up to a slight difference to the commonly used definition based on deductive conservative extensions: for two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ formulated in a DL L, and a signature $\Gamma \subseteq \text{sig}(\mathcal{T}_1)$, $\mathcal{T}_1 \cup \mathcal{T}_2$ is a Γ-*conservative extension* of $\mathcal{T}_1$ iff for all axioms α expressed in L with $\text{sig}(\alpha) \in \Gamma$, we have $\mathcal{T}_1 \models \alpha$ iff $\mathcal{T}_1 \cup \mathcal{T}_2 \models \alpha$. $\mathcal{T}_2$ is then said to be a module of $\mathcal{T}_1 \cup \mathcal{T}_2$ with respect to the signature Γ. Note that the ontology and the query languages coincide within the above definition.

Ghilardi, Lutz and Wolter [GHILARDI et al. 2006] adopt this notion for the definition of modules, however restrict the considered axioms of the query language explicitly to concept subsumptions. They show that deciding if a subontology is a module in the description logic $\mathcal{ALC}$ is 2EXPTIME-complete. In a subsequent work, Lutz, Walter and Wolter [LUTZ et al. 2007] show that the same problem is 2EXPTIME-complete for $\mathcal{ALCQI}$, but undecidable for $\mathcal{ALCQIO}$. The authors also investigate a stronger definition of modules defined directly on models instead of entailed consequences: given two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$, $\mathcal{T}_1 \cup \mathcal{T}_2$ is a *model-conservative extension* of $\mathcal{T}_1$ iff for every model $\mathcal{I}$ of $\mathcal{T}_1$, there exists a model of $\mathcal{T}_1 \cup \mathcal{T}_2$ which can be obtained from $\mathcal{I}$ by modifying the interpretation of symbols in $\text{sig}(\mathcal{T}_2) \setminus \text{sig}(\mathcal{T}_1)$ while leaving the interpretation of symbols in $\text{sig}(\mathcal{T}_1)$ fixed. In the above definition, the query language corresponds to Second Order Logic. The authors show that the corresponding problem based on the latter notion is undecidable for $\mathcal{ALC}$.

In a more recent work, Konev, Lutz, Walter and Wolter [KONEV et al. 2008] consider the decidability of the above problem based on model-conservative extensions

for $\mathcal{ALC}$ under different additional restrictions, e.g., restriction of the relevant signature to concept names, and obtain complexity results ranging from Π_2^p to undecidable. Further, the authors consider the problem for acyclic $\mathcal{EL}$ terminologies. It is interesting that, in contrary to acyclic $\mathcal{ALC}$ terminologies, for which the problem remains undecidable, for acyclic $\mathcal{EL}$ terminologies the complexity goes down to PTIME. In a later work [KONEV et al. 2009a], the above authors present a full complexity picture for $\mathcal{ALC}$ and its common extensions. Instead of considering only the two notions of conservative extensions, they investigate a broad range of query languages, starting with the language allowing for expressing inconsistency only and ending with Second Order Logic. More recently, Lutz and Wolter [LUTZ and WOLTER 2010] show that the above notion of model-conservative extensions is undecidable also for such a lightweight logic as $\mathcal{EL}$.

In parallel, Kontchakov, Wolter and Zakharyaschev [KONTCHAKOV et al. 2008] investigate the above decision problem for two representatives of the DL-Lite family of description logics as ontology languages and existential Σ-queries as a query language. They show that, for DL-Lite$_{\texttt{horn}}$, the problem is CONP-complete, and for DL-Lite$_{\texttt{bool}}$ Π_2^p-complete.

A different definition of modules in the way how the relevant signature Σ is interpreted is followed by Cuenca Grau, Parsia, Sirin and Kalyanpur [GRAU et al. 2006]. The authors argue that it is not sufficient to preserve only consequences expressed using Σ, but the elements of Σ only indicate the core of the relevant subontology. The concrete assumption of the latter work is that the ontology has to additionally preserve all atomic subsumees and subsumers of relevant concepts, even if the latter are not part of the relevant signature. Thus, within their definition of modules, atomic subsumees and subsumers are included into the module recursively until all reachable concepts are covered. Despite a semantics-based definition, the approach has clear syntactic features due to a partially syntactic specification of requirements for modules. The above assumption significantly simplifies the task of module extraction, which can be done in PTIME for OWL DL by the means of a graph partitioning algorithm.

In a more recent work, Cuenca Grau, Horocks, Kazakov and Sattler [GRAU et al. 2007c] investigate the problem of module extraction from a different perspective: for a signature Σ, and an ontology $\mathcal{T}_2$, the problem is to decide

whether, for any module $\mathcal{T}_1$ with $\text{sig}(\mathcal{T}_1) \cap \text{sig}(\mathcal{T}_2) \subseteq \Sigma$, the extended ontology $\mathcal{T}_1 \cup \mathcal{T}_2$ is a deductive-conservative extension of $\mathcal{T}_1$. Thus, the decision does not depend on the module $\mathcal{T}_1$, but only on the usage of Σ within the extension $\mathcal{T}_2$. Note that the latter definition is more restrictive, i.e., any module in the latter sense is also a module according to the previous definition based on deductive-conservative extensions. An important difference from the practical point of view , in particular in the context of semantic conciseness, is that modules in the sense of the latter definition are likely to be larger than modules according to the previous definition. The reason for this is that axioms are moved from $\mathcal{T}_2$ to $\mathcal{T}_1$ until the usage restrictions for Σ are fulfilled disregarding whether or not $\mathcal{T}_1$ contains sufficient axioms to entail all important Σ-consequences. This is motivated by the aim of the latter work to provide a guarantee that the meaning of Σ-entities is not modified within $\mathcal{T}_2$. Notwithstanding this deviating objective and the resulting suboptimality in terms of module size, in a follow-up work [GRAU et al. 2007a], the above authors propose a tractable algorithm for computing modules from OWL DL ontologies based on an approximation for the above definition of modules. The approximation is based on a notion of *syntactic locality* [GRAU et al. 2007c] that defines the locality of an axiom on the syntactic level. It is guaranteed that the extracted module is a module in the sense of the above definition, however the tractability comes at the cost of a further suboptimality in terms of module size.

The research results for module extraction are largely theoretic. However, two of the above discussed approaches, the one proposed by Kontchakov et al. [KONTCHAKOV et al. 2008] and the one by Cuenca Grau et al. [GRAU et al. 2007a], come with an implementation, which we will use within this thesis for an empirical evaluation.

As we will show in this thesis, with respect to the objective of improving the conciseness of ontologies, module extraction is not the most optimal method. In many application scenarios it is not necessary that the resulting ontology is a subset of the initial one. In fact, the comprehensiveness of the resulting ontology can also be achieved by imposing weaker restrictions on its syntactic structure. We will show that such a relaxation of restrictions on the syntactic structure in most cases allows for a significant improvement in terms of semantic conciseness.

3.2.2 Forgetting-Based Approaches

The problem of forgetting or uniform interpolation is based on the notion of *inseparability*, which is defined analogously to the notion of conservative extensions with the difference that no subset relation between the two knowledge bases is required, i.e., they are allowed to be syntactically very different from each other. For this reason, the notion of inseparability is symmetric. Also the inseparability can be defined with respect to to different query languages, i.e., the expressivity of Σ-consequences is not necessarily the same as the expressivity of any of the two ontologies. As in case of module extraction, also in case of forgetting the problem complexity significantly varies depending on the corresponding ontology and query language. In most existing approaches, it is assumed that the ontology and the query languages coincide.

Additionally to the inseparability, the result of forgetting or uniform interpolation is required to use only entities from the relevant signature. The latter restriction makes the problem very intricate. Among other things, due to the latter restriction, it can happen that no finite result exists for a particular knowledge base and signature.

The investigated problem definitions vary not only in the choice of ontology and query languages, but also in the way the irrelevant signature is defined (only concepts or also roles) and the scope of application (concept expressions, TBoxes or knowledge bases). We discuss the existing approaches according to the expressivity of ontology languages.

For $\mathcal{ALC}$ and its various extensions, the foundations of forgetting are well-understood both, on the level of concepts [CATE and CONRADIE 2006, WANG et al. 2009b] as well as on the level of TBoxes. The problem on the level of TBoxes turned out to be very difficult and has been investigated by various authors over the past years. The first observation on the TBox level was that there are very simple TBoxes and signatures Σ such that the uniform interpolant with respect to Σ cannot be expressed in $\mathcal{ALC}$ (nor in first-order predicate logic) [GHILARDI et al. 2006]. Wang et al. [WANG et al. 2009a] devise an algorithm for approximating interpolants of $\mathcal{ALC}$-TBoxes (for both, existing or non-existing interpolants). In a later work, Wang et al. [WANG et al. 2010] attempt to give an

algorithm that computes uniform interpolants of $\mathcal{ALC}$-TBoxes in an exact way, and also decides their existence in $\mathcal{ALC}$. Unfortunately, that algorithm turns out to be incorrect as shown by Lutz and Wolter [LUTZ and WOLTER 2011]. The latter authors propose an approach for computing uniform interpolants with respect to $\mathcal{ALC}$ terminologies. Additionally, they show the tight triple-exponential bound on their size and the 2EXPTIME-completeness for deciding their existence.

Forgetting of concepts in DL-Lite is investigated by Wang et al. [WANG et al. 2008]. Forgetting is based on the notion of model-inseparability (counterpart of model-conservative extensions) for a particular signature, i.e. inseparability defined in terms of interpretation extensions. This is a rather strong notion with the query language being Second Order Logic. The authors propose a polynomial time forgetting algorithm for both, DL-Lite TBoxes and ABoxes, and DL-Lite knowledge bases.

For the lightweight logic $\mathcal{EL}$, a general solution for computing uniform interpolants had not been proposed before this thesis. Also the bound on the size of uniform interpolants in $\mathcal{EL}$ remained unknown. A procedure for deciding the existence of uniform interpolants in $\mathcal{EL}$ has been proposed by Lutz and Wolter [LUTZ et al. 2012]. The latter decision problem has been shown to be EXPTIME-complete. Konev et al. [KONEV et al. 2009b] have proposed an EXPTIME algorithm for computing uniform interpolants which, however, does not allow for general concept inclusions in the corresponding TBox and relies on sufficient but not necessary acyclicity conditions. In this thesis, we close this gap by deriving a worst-case optimal algorithm for computing uniform interpolants for general $\mathcal{EL}$ terminologies and determining the corresponding tight, triple-exponential bound on the output size.

Further, we will show later on that also uniform interpolation is not optimally suited for improving the semantic conciseness of ontologies. In addition to their triple-exponential size in the worst-case, uniform interpolants can be highly difficult to read for ontology engineers due to the double-exponential size of concept expressions in the worst case. Thus, the problem of improving the conciseness of ontologies has not yet been addressed sufficiently considering typical application requirements, i.e., maintaining the ontology's comprehensiveness while achieving the highest possible conciseness.

3.3 Criteria-Independent Approaches

In this section, we consider two well represented categories of generic quality assurance techniques that are not specific to a particular quality dimension in general, but can, among other things, be applied to address semantic accuracy and conciseness problems.

3.3.1 Approaches Based on Feedback Provided by Users

One of the most effective quality assurance mechanisms for online markets is the principle of online reviews provided by users for commercial products. The idea is, of course, easily transferable to online information resources. So far, several approaches have been proposed to transfer the latter idea to ontologies.

For instance, Supekar et al. [SUPEKAR 2005] proposes to extend ontologies with a default set of metadata documenting its design policy, how it is being used by others, as well as "peer reviews" provided by its users.

Lewen et al. [LEWEN and D'AQUIN 2010] employ user ratings to determine the user-perceived quality of ontologies. The combination of an Open Rating System (ORS), user ratings, and information on trust between users is combined to compute a personalized ranking of ontologies.

Xie et al. [XIE and BURSTEIN 2011] propose a framework for predicting values of quality attributes based on previous value judgments of users encoded in resource metadata descriptions.

Pierkot et al. [PIERKOT et al. 2011] propose a method for resource selection in the context of GIS-related resources that takes into account the information about the user profile, the application domain, the requirements, and the intended use, which is assumed to be encoded as metadata and made available to the resource search engine. This metadata encoding the intended usage of the resource is then mapped onto the metadata encoding the properties of the corresponding resources in order to assess the quality of the resources within the particular usage context.

Despite its high potential effectiveness, the idea has not yet been adopted by the ontology engineering community. On the one hand, there are no standards or established tools supporting an extensive exchange of feedback information with the required detail. On the other hand, ontology users do not seem to be willing to invest

a high amount of effort into quality assurance of ontologies that are developed by others. Hence, it is not reasonable to expect that the feedback provided by the users of an ontology would help to detect the majority of quality problems. Additionally, the feedback information can be difficult to interpret, since an ontology can be used in different applications with different underlying requirements. Therefore, the quality feedback from one particular context is not necessarily relevant within a further application context. To sum up, professional ontology development aiming at guaranteeing a high quality of the ontology requires more reliable means of quality assurance.

3.3.2 Structure-Based Approaches

There is a wide range of attempts to quantify different quality criteria by defining metrics based on the structure of ontologies. The goal is to enable an automatic evaluation of ontologies by measuring the corresponding quality aspects and identifying indications for potential quality problems. The numeric values are often used for computing an overall score as a weighted sum of its per-criterion scores. Such an ordering allows the users to automatically compare ontologies with each other and sort them according to this score in order to support the choice of suitable resources for a particular application context. The latter support is called *ontology ranking*.

There exists a large number of structure-based measurement frameworks, e.g., [AMIRHOSSEINI and SALIM 2011, BEYDOUN et al. 2011, BACHIR BOUIADJRA and BENSLIMANE 2011, JANOWICZ et al. 2008, TARTIR et al. 2005, LEI et al. 2007, SUPEKAR 2004, GANGEMI et al. 2006, STVILIA 2007]. To name one concrete example, Gomez-Perez et al. [LOZANO-TELLO and GÓMEZ-PÉREZ 2004] propose a hierarchical framework of metrics called OntoMetric. It consists of 160 characteristics spread across five quality dimensions. The hope is that the latter can be used in order to automate the assessment of quality and suitability of ontologies to users' system requirements. Such a measurement of ontology attributes can provide a useful overview of an external ontology. However, the common difficulty with the structure-based metrics as a means for quality assessment is that they are

usually only weak indications for the corresponding quality problems, limited to measurable aspects of ontologies and not sufficiently taking into account the information about the particular application requirements.

Part II

Reasoning Support for Ensuring Accuracy and Conciseness

CHAPTER 4

Accuracy-Based Revision

Since the introduction of ontology modeling languages, in particular the standardized language OWL, a variety of large ontologies has been developed and made publicly available in order to facilitate the development of knowledge-intensive applications. Moreover, a wide range of heuristic ontology management tools have arisen, e.g., ontology matching or learning tools, aiming at a reduction of ontology engineering cost. In both cases, quality assurance plays an essential role. Manual inspection of ontologies is one of the most reliable, but also most expensive quality assurance methods in terms of manual effort.

Currently, there is very limited support available for reducing the effort of such a manual inspection aiming at ensuring the semantic accuracy of ontologies. So far, knowledge representation (KR) research has been focusing on restoring the consistency of knowledge bases enriched with new axioms as done in various belief revision and repair approaches, see, e.g., [SATOH 1988, SCHLOBACH and CORNET 2003, QI and YANG 2008]. Thereby, new axioms not causing an inconsistency are accepted as valid facts not requiring further inspection.

To close the gap, we address the scenarios requiring a more restrictive quality assurance: we consider a revision process in which a domain expert inspects a set

of candidate axioms and decides for each of them whether it is a desired logical consequence (approval) or not (decline). The third possibility would be to exclude an axiom from the inspection, e.g. due to its unclear meaning. We call this exhaustive manual inspection of the acquired data *accuracy-based interactive ontology revision*, or simply *interactive ontology revision*. If we assume that the deductive closure of correct statements must be disjoint from the set of incorrect statements, then this revision process can be partially automated: based on the decisions taken by the expert, we can automatically decline or approve yet unevaluated axioms depending on their logical relationships with the already evaluated axioms. On the one hand, we can automatically approve axioms that are entailed by the already confirmed statements, since declining them would violate the above given assumption. On the other hand, we can automatically decline axioms that would cause any of the already declined axioms to become a consequence of the confirmed ones, since accepting them would also violate our assumption.
Throughout this chapter, we use the following running example.

Example 1. *Consider the ontology in Figure 4.1. Let us assume that we have already confirmed that the axioms in the upper part consisting of concept inclusions belong to the desired consequences. We further assume that Axiom* (4.1) *to Axiom* (4.8) *in the lower part, which define several different types for the individual* nanotube1*, are still to be evaluated. If Axiom* (4.8) *is declined, we can immediately also decline Axioms* (4.1) *to* (4.6) *assuming OWL or RDFS reasoning since accepting the axioms would implicitly lead to the undesired consequence* (4.8)*. Note that no automatic decision is possible for Axiom* (4.7) *since it is not a consequence of Axiom* (4.8) *and the already approved subsumption axioms. Similarly, if Axiom* (4.1) *is approved, Axioms* (4.2) *to* (4.8) *are implicit consequences, which can be approved automatically. If we start, however, with declining Axiom* (4.1)*, no automatic evaluation can be performed.*

In the previous example, we only made decisions about class assertion axioms. This is, however, not a restriction of the approach. In general, the presented approach is independent from a particular logical formalism, but assumes monotonicity, idempotency and extensivity, introduced in Section 2.5, as well as the existence of a sound and complete reasoning procedure for checking entailment. The follow-

$$\begin{aligned}
\mathsf{AluminiumNitrideNanotube} &\sqsubseteq \mathsf{AluminiumNitride}\\
\mathsf{AluminiumNitride} &\sqsubseteq \mathsf{NonOxideCeramics}\\
\mathsf{NonOxideCeramics} &\sqsubseteq \mathsf{Ceramics}\\
\mathsf{Ceramics} &\sqsubseteq \mathsf{MaterialByMaterialClass}\\
\mathsf{MaterialByMaterialClass} &\sqsubseteq \mathsf{Material}\\
\mathsf{Material} &\sqsubseteq \mathsf{PortionOfMaterial}\\
\mathsf{Material} &\sqsubseteq \mathsf{TangibleObject}
\end{aligned}$$

$$\mathsf{AluminiumNitrideNanotube}(\,\mathsf{nanotube1}) \tag{4.1}$$
$$\mathsf{AluminiumNitride}(\,\mathsf{nanotube1}) \tag{4.2}$$
$$\mathsf{NonOxideCeramics}(\,\mathsf{nanotube1}) \tag{4.3}$$
$$\mathsf{Ceramics}(\,\mathsf{nanotube1}) \tag{4.4}$$
$$\mathsf{MaterialByMaterialClass}(\,\mathsf{nanotube1}) \tag{4.5}$$
$$\mathsf{Material}(\,\mathsf{nanotube1}) \tag{4.6}$$
$$\mathsf{PortionOfMaterial}(\,\mathsf{nanotube1}) \tag{4.7}$$
$$\mathsf{TangibleObject}(\,\mathsf{nanotube1}) \tag{4.8}$$

Figure 4.1: *An example ontology from the nanotechnology domain*

$$\begin{aligned}
\mathsf{a{:}Ordinary} &\sqsubseteq \mathsf{a{:}Employee}\\
\mathsf{a{:}Employee} &\sqsubseteq \mathsf{a{:}Person}\\
\mathsf{b{:}Ordinary} &\sqsubseteq \mathsf{b{:}Lecture}\\
\mathsf{b{:}Lecture} &\sqsubseteq \mathsf{b{:}Event}
\end{aligned}$$

$$\mathsf{a{:}Person} \sqcap \mathsf{b{:}Event} \sqsubseteq \bot \tag{4.9}$$
$$\mathsf{a{:}Employee} \sqcap \mathsf{b{:}Lecture} \sqsubseteq \bot \tag{4.10}$$
$$\mathsf{a{:}Ordinary} \equiv \mathsf{b{:}Ordinary} \tag{4.11}$$

Figure 4.2: *An example ontology from the enterprise domain*

ing example demonstrates the case where we make decisions about terminological axioms during the revision of an ontology. We use imaginary prefixes a and b to abbreviate IRIs of two different ontologies.

Example 2. *Let us assume that we have already approved the axioms in the upper part of Figure 4.2, while incoherency has been stated to be an undesired consequence by adding the corresponding axioms expressing it to the set of declined axioms. We further assume that Axiom* (4.9) *to Axiom* (4.11) *in the lower part of Figure 4.2 are still to be evaluated. If Axiom* (4.9) *is approved, we can immediately also approve Axiom* (4.10) *since it is already a consequence of the approved axioms:* a:Employee *is interpreted as a subset of the extension of* a:Person *and* b:Lecture *is interpreted as a subset of the extension of* b:Event*, but if* a:Person *and* b:Event *are disjoint due the just approved Axiom* (4.9) *then so are* a:Employee *and* b:Lecture*. Moreover, we can decline Axiom* (4.11)*, since approving this axiom would implicitly lead to incoherency, again since* a:Ordinary *and* b:Ordinary *have to be interpreted as subsets of disjoint sets and can, therefore, not be equivalent.*

From the above examples, we see that a single expert decision can predetermine several further evaluation decisions, thus allowing for automation. To capture this effect, in the following, we introduce and elaborate on the notion of *revision states* as formal foundations of our method and the notion of *revision closure* summarizing such predetermined decisions. It can further be observed that

- a high grade of automation requires a good evaluation order and
- approval and decline of an axiom have a different impact on the automatic evaluation of further axioms.

Which axioms have the highest impact on decline or approval and which axioms can be automatically evaluated once a particular decision has been made can be determined with the help of algorithms for automated reasoning, e.g., those for RDFS or OWL. For this purpose, in the following we introduce the notion of *axiom impact* capturing the number of automatically evaluated axioms upon an approval or decline of an axiom. Based on the impact, we can in theory determine a beneficial order, in which axioms are presented to the expert. One of the difficulties is, however, that it is not known in advance, which of the two decisions the domain expert takes. We show that, in some cases, a realistic prediction about the decision of the user can be made: if the proportion of accurate axioms is fairly high, also the probability of an approval is high. Hence, axioms that have a high impact on

approval (approval impact) should be evaluated with higher priority. For data with low average accuracy, the situation is reversed, i.e., axioms that have a high impact on decline (decline impact) should be considered first. We measure the average accuracy of a dataset by means of the *validity ratio*, i.e., the proportion of (manually and automatically) accepted axioms, and show that, depending on the validity ratio of a dataset, different impact measures used for axiom ranking are beneficial.

While approval and decline impact measures yield fairly good results for validity ratios close to 100% or 0%, the optimality of results is left to chance in case of validity ratios close to 50%. To close this gap, the initial notion of axiom impact is refined to take a more precise estimation of the validity ratio into account.We introduce an advanced ranking function that is based on these simple impact measures and, additionally, parametrized by an estimated validity ratio. In our evaluation, we show that the revision based on the novel ranking function almost achieves the maximum possible automation. In particular the parametrized ranking functions achieve very good results for arbitrary validity ratios.

Further, since the expected validity ratio is usually not known in advance, we suggest a ranking function where the validity ratio is learned on-the-fly during the revision. We show that, even for small datasets (50-100 axioms), it is worthwhile to rank axioms based on this learned validity ratio instead of evaluating them in a random order. Furthermore, we show that, in case of larger datasets (e.g., 5,000 axioms and more) with an unknown validity ratio, learning the validity ratio is particularly effective (with only 0.3% loss of effectiveness) due to the law of large numbers, thereby making the assumption of a known or expected validity ratio unnecessary. For such datasets, our experiments show that the proportion of automatically evaluated axioms when learning the validity ratio is nearly the same as in case where the validity ratio is known in advance.

Even for light-weight knowledge representation formalisms, reasoning is often comparably expensive and in an interactive setting it is crucial to minimize the number of reasoning tasks while maximizing the number of automated decisions. Inspired by the techniques used to optimize ontology classification [SHEARER and HORROCKS 2009], we reduce the number of reasoning tasks by introducing the notion of *decision spaces* – auxiliary data structures that allow for storing the results of reasoning and reading-off the impact that an axiom will have

upon approval or decline. Decision spaces exploit the characteristics of the logical entailment relation between axioms to maximize the amount of information gained by reasoning, and, therefore, in particular in the case of logics for which entailment checking is not tractable, decision spaces reduce the computational effort. In addition to the performance gain achieved by using decision spaces, we show that partitioning – dividing the datasets under revision into logically independent subsets – further decreases the number of required reasoning calls.
We implemented the proposed techniques in the tool *revision helper*, which even for expressive OWL reasoning and our dataset of 25,000 axioms requires on average only 0.84 seconds (7.4 reasoning calls) per expert decision, where the automatic evaluation significantly reduces the number of expert decisions.
From our evaluation, it can be observed that, on the one hand, a considerable proportion (up to 80%) of axioms can be evaluated automatically by our revision support, and, on the other hand, the application of decision spaces and partitioning significantly reduces the number of required reasoning operations, resulting in a considerable performance gain – 83% of reasoning calls could be avoided.

4.1 Revision States and Closure

Above, we described ontology revision as a process in which a domain expert inspects a set of candidate axioms and decides for each of them whether to approve or to decline it or whether to exclude it from the inspection. While the latter option is an important feature for a practical implementation (see Section 4.5), such an exclusion does not allow any conclusions about the remaining, unevaluated axioms, and, therefore, does not have any effect on the further process of the revision. Thus, in the following we abstract from the the option to exclude axioms from the revision without accepting or declining them. Then, the revision of an ontology $\mathcal{O}$ can be seen as a separation of its axioms (i.e., logical statements) into two disjoint sets: the set of wanted consequences $\mathcal{O}^{\models}$ and the set of unwanted consequences $\mathcal{O}^{\not\models}$. For convenience, we assume that axioms added to the above two sets are not removed from $\mathcal{O}$ over the course of revision. To be able to formulate assertions about the revision process, we introduce the notion of *revision states* capturing the two above sets at a particular stage of the revision.

Definition 1 (Revision State)**.** *A* revision state *is defined as a tuple* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *of ontologies with* $\mathcal{O}^{\models} \subseteq \mathcal{O}, \mathcal{O}^{\not\models} \subseteq \mathcal{O}$*, and* $\mathcal{O}^{\models} \cap \mathcal{O}^{\not\models} = \emptyset$*. Given two revision states* $(\mathcal{O}, \mathcal{O}_1^{\models}, \mathcal{O}_1^{\not\models})$ *and* $(\mathcal{O}, \mathcal{O}_2^{\models}, \mathcal{O}_2^{\not\models})$*, we call* $(\mathcal{O}, \mathcal{O}_2^{\models}, \mathcal{O}_2^{\not\models})$ *a* refinement *of* $(\mathcal{O}, \mathcal{O}_1^{\models}, \mathcal{O}_1^{\not\models})$*, if* $\mathcal{O}_1^{\models} \subseteq \mathcal{O}_2^{\models}$ *and* $\mathcal{O}_1^{\not\models} \subseteq \mathcal{O}_2^{\not\models}$*. A revision state is* complete*, if* $\mathcal{O} = \mathcal{O}^{\models} \cup \mathcal{O}^{\not\models}$*, and* incomplete *otherwise. An incomplete revision state* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *can be refined by evaluating a further axiom* $\alpha \in \mathcal{O} \setminus (\mathcal{O}^{\models} \cup \mathcal{O}^{\not\models})$*, obtaining* $(\mathcal{O}, \mathcal{O}^{\models} \cup \{\alpha\}, \mathcal{O}^{\not\models})$ *or* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models} \cup \{\alpha\})$*. We call the resulting revision state an* elementary refinement *of* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$.

We introduce the notion of *consistency* of revision states to express the condition that the deductive closure of the wanted consequences in $\mathcal{O}^{\models}$ must not contain unwanted consequences. If we want to maintain consistency, a single evaluation decision can predetermine the decision for several yet unevaluated axioms. These implicit consequences of a refinement are captured by the means of the *revision closure*.

Definition 2 (Revision State Consistency & Closure)**.** *A (complete or incomplete) revision state* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *is* consistent *if there is no* $\alpha \in \mathcal{O}^{\not\models}$ *such that* $\mathcal{O}^{\models} \models \alpha$*. The* revision closure $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *of* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *is* $(\mathcal{O}, \mathcal{O}_c^{\models}, \mathcal{O}_c^{\not\models})$ *with* $\mathcal{O}_c^{\models} := \{\alpha \in \mathcal{O} \mid \mathcal{O}^{\models} \models \alpha\}$ *and* $\mathcal{O}_c^{\not\models} := \{\alpha \in \mathcal{O} \mid \mathcal{O}^{\models} \cup \{\alpha\} \models \beta$ *for some* $\beta \in \mathcal{O}^{\not\models}\}$.

Example 3. *We consider again Example 2. Let* $\mathcal{O}^{\models}$ *be the axioms in the upper part of Fig. 4.2,* $\mathcal{O}^{\not\models}$ *be the set of axioms expressing inconsistency and incoherence of axioms in Fig. 4.2 and let* $\mathcal{O}$ *consist of* $\mathcal{O}^{\models} \cup \mathcal{O}^{\not\models}$ *and additionally Axioms* (4.9) *to* (4.11)*. Approving or declining an arbitrary axiom from* $\mathcal{O} \setminus (\mathcal{O}^{\models} \cup \mathcal{O}^{\not\models})$ *yields an elementary refinement of the revision state* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$*. Approving Axiom* (4.9) *denoted by* α *yields a consistent revision state* $(\mathcal{O}, \mathcal{O}^{\models} \cup \{\alpha\}, \mathcal{O}^{\not\models})$*, which can be further refined to obtain an inconsistent revision state by declining Axiom* (4.10) *or approving Axiom* (4.11)*, since the former axiom is entailed by* $\mathcal{O}^{\models} \cup \{\alpha\}$ *and the latter one will cause incoherence upon its approval. If we, however, compute a revision closure of* $(\mathcal{O}, \mathcal{O}^{\models} \cup \{\alpha\}, \mathcal{O}^{\not\models})$*, we obtain* $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models} \cup \{\alpha\}, \mathcal{O}^{\not\models}) = (\mathcal{O}, \mathcal{O}^{\models} \cup \{\alpha, \beta\}, \mathcal{O}^{\not\models} \cup \{\gamma\})$*, where* β *denotes Axiom* (4.10) *and* γ *denotes Axiom* (4.11).

Note that, in order to be able to maintain the consistency of a revision state, $\mathcal{O}_c^{\models}$ must contain all axioms that, in case of an accept, would lead to an entailment of any unwanted consequences. We can show the following useful properties of the closure of consistent revision states:

Lemma 1. *For* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *a consistent revision state,*

1. $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *is consistent,*
2. *every elementary refinement of* $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *is consistent,*
3. *every consistent and complete refinement of* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *is a refinement of* $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$.

Proof. We start with the first claim. By definition of closure, we have $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models}) = (\mathcal{O}, \{\alpha \in \mathcal{O} \mid \mathcal{O}^{\models} \models \alpha\}, \{\alpha \in \mathcal{O} \mid \mathcal{O}^{\models} \cup \{\alpha\} \models \beta \text{ for some } \beta \in \mathcal{O}^{\not\models}\})$. Since $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is consistent, for any $\alpha \in \mathcal{O}$ with $\mathcal{O}^{\models} \cup \{\alpha\} \models \beta$ for some $\beta \in \mathcal{O}^{\not\models}$ holds $\mathcal{O}^{\models} \not\models \alpha$, otherwise would hold $\mathcal{O}^{\models} \models \beta$. Therefore, for the set $\{\alpha \in \mathcal{O} \mid \mathcal{O}^{\models} \models \alpha\}$ holds that there is not $\beta \in \mathcal{O}^{\not\models}$ entailed by it. Thus, $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is consistent. For the second claim, $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is consistent by assumption and $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is then consistent (by the first claim). Since $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is a closure of $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$, we have $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models}) = (\mathcal{O}, \{\alpha \in \mathcal{O} \mid \mathcal{O}^{\models} \models \alpha\}, \{\alpha \in \mathcal{O} \mid \mathcal{O}^{\models} \cup \{\alpha\} \models \beta \text{ for some } \beta \in \mathcal{O}^{\not\models}\})$. Since an elementary revision of $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ has to be for an axiom $\alpha \in \mathcal{O} \setminus (\{\beta \mid \mathcal{O}^{\models} \models \beta\} \cup \{\beta \mid \mathcal{O}^{\not\models} \cup \beta \models \gamma \text{ for some } \gamma \in \mathcal{O}^{\not\models}\})$, we immediately get that the elementary refinement is consistent. For the last claim, if $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is already complete, the claim trivially holds. Otherwise, since $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is consistent, we cannot make elementary refinements that add an axiom $\alpha \in \{\beta \mid \mathcal{O}^{\models} \models \beta\}$ to $\mathcal{O}^{\not\models}$ since this would result in an inconsistent refinement, neither can we add an axiom $\alpha \in \{\beta \mid \mathcal{O}^{\not\models} \cup \beta \models \gamma \text{ for some } \gamma \in \mathcal{O}^{\not\models}\}$ to $\mathcal{O}^{\models}$. Thus, a complete and consistent refinement of $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is a refinement of $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$. □

Algorithm 1 employs the above properties to implement a general methodology for interactive ontology revision. Instead of starting with empty $\mathcal{O}_0^{\models}$ and $\mathcal{O}_0^{\not\models}$, we can initialize these sets with approved and declined axioms from a previous revision or

Algorithm 1: Interactive Ontology Revision

Input: $(\mathcal{O}, \mathcal{O}_0^{\models}, \mathcal{O}_0^{\not\models})$ a consistent revision state
Output: $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ a complete and consistent revision state

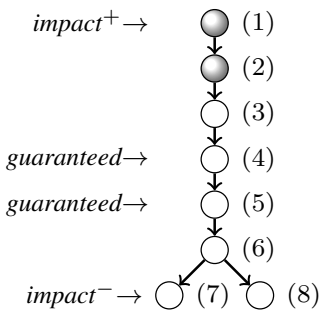
1: $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models}) \leftarrow \mathrm{clos}(\mathcal{O}, \mathcal{O}_0^{\models}, \mathcal{O}_0^{\not\models})$
2: **while** $\mathcal{O}^{\models} \cup \mathcal{O}^{\not\models} \neq \mathcal{O}$ **do**
3: choose $\alpha \in \mathcal{O} \setminus (\mathcal{O}^{\models} \cup \mathcal{O}^{\not\models})$
4: **if** expert confirms α **then**
5: $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models}) \leftarrow \mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models} \cup \{\alpha\}, \mathcal{O}^{\not\models})$
6: **else**
7: $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models}) \leftarrow \mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models} \cup \{\alpha\})$
8: **end if**
9: **end while**

add axioms of the ontology that is being developed to $\mathcal{O}_0^{\models}$. We can further initialize $\mathcal{O}_0^{\not\models}$ with axioms expressing inconsistency and unsatisfiability of predicates (i.e. of classes or relations) in $\mathcal{O}$, which we assume to be unwanted.

The above algorithm is very generic. In particular, it does not specify how to choose the next axiom for the evaluation in line 3. As mentioned earlier, choosing randomly can have a detrimental effect on the number of manual decisions needed. In the following, we discuss strategies that aim at achieving a high number of consequential automatic decisions.

4.2 Axiom Ranking

As demonstrated by Examples 1 and 2, a high grade of automation requires a good evaluation order. In this section, we discuss techniques aiming at determining a beneficial order of axiom evaluation. Ideally, we want to rank the axioms under revision and, at each evaluation step, choose one that allows for a high number of consequential automatic decisions. In what follows, we introduce a notion of *axiom impact* capturing the effect of an axiom evaluation in terms of consequential automatic decisions. The initial notion of impact is then further refined to take different validity ratios – the proportion of valid statements within a dataset – into account.



Axiom	$impact^{+a}$	$impact^{+d}$	$impact^{-}$	*guaranteed*
(1)	7	0	0	0
(2)	6	0	1	1
(3)	5	0	2	2
(4)	4	0	3	3
(5)	3	0	4	3
(6)	2	0	5	2
(7)	0	0	6	0
(8)	0	0	6	0

Table 4.1: *Example dependency graph showing axioms and entailment relationships between them and the corresponding ranking values*

4.2.1 Axiom Impacts

From the introductory examples, we could observe that approval and decline of an axiom has a different impact. Thus, we introduce two notions of *axiom impact*: the *approval impact* of an axiom refers to the number of axioms that can be automatically evaluated upon its approval, while the *decline impact* refers to the number of axioms that can be automatically evaluated upon its decline. Additionally, we define the *guaranteed impact* as the guaranteed number of axioms that can be automatically evaluated in any of the two cases after the evaluation of the corresponding axiom. Note that, after an approval, the closure might extend both $\mathcal{O}^{\models}$ and $\mathcal{O}^{\not\models}$, whereas after a decline only $\mathcal{O}^{\not\models}$ can be extended. We further define $?(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ as the number of yet unevaluated axioms and write $|S|$ to denote the cardinality of a set S:

Definition 3 (Impact)**.** *Let* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *be a consistent revision state and* $?(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models}) := |\mathcal{O} \setminus (\mathcal{O}^{\models} \cup \mathcal{O}^{\not\models})|$*. For an axiom* $\alpha \in \mathcal{O} \setminus (\mathcal{O}^{\models} \cup \mathcal{O}^{\not\models})$*, we define its* approval impact*,* $impact^{+}(\alpha)$*, its* decline impact*,* $impact^{-}(\alpha)$*, and its*

guaranteed impact *guaranteed*(α)*:*

$$\textit{impact}^{+}(\alpha) = ?(\mathcal{O}, \mathcal{O}^{\models} \cup \{\alpha\}, \mathcal{O}^{\not\models}) - ?(\text{clos}(\mathcal{O}, \mathcal{O}^{\models} \cup \{\alpha\}, \mathcal{O}^{\not\models})), \quad (4.12)$$

$$\textit{impact}^{-}(\alpha) = ?(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models} \cup \{\alpha\}) - ?(\text{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models} \cup \{\alpha\})), \quad (4.13)$$

$$\textit{guaranteed}(\alpha) = \min(\textit{impact}^{+}(\alpha), \textit{impact}^{-}(\alpha)). \quad (4.14)$$

We separate impact$^{+}(\alpha)$ *into the number of automatic approvals, impact*$^{+a}(\alpha)$*, and the number of automatic declines, impact*$^{+d}(\alpha)$*:*

$$\textit{impact}^{+a}(\alpha) = |\{\beta \in \mathcal{O} \setminus \{\alpha\} \mid \mathcal{O}^{\models} \cup \{\alpha\} \models \beta\}|, \quad (4.15)$$

$$\textit{impact}^{+d}(\alpha) = |\{\beta \in \mathcal{O} \mid \mathcal{O}^{\models} \cup \{\alpha, \beta\} \models \gamma, \textit{ for some } \gamma \in \mathcal{O}^{\not\models}\}|. \quad (4.16)$$

Note that impact$^{+}(\alpha)$ = impact$^{+a}(\alpha)$ + impact$^{+d}(\alpha)$. Ranking axioms by *impact*$^{+}$ privileges axioms for which the number of automatically evaluated axioms in case of an accept is high. Going back to our running example (Example 1), Axiom (4.1), which yields 7 automatically accepted axioms in case it is approved, will be ranked highest. The situation is the opposite for *impact*$^{-}$, which privileges axioms for which the number of automatically evaluated axioms in case of a decline is high (Axioms (4.7) and (4.8)). Ranking by *guaranteed* privileges axioms with the highest guaranteed impact, i.e., axioms with the highest number of automatically evaluated axioms in the worst-case (Axioms (4.4) and (4.5)). Table 4.1 lists the values for all ranking functions for the axioms from Example 1.

We can observe that, depending on the validity ratio of a dataset, different impact measures used for axiom ranking would be beneficial. In the following, we show that, while approval and decline impact measures yield fairly good results for validity ratios close to 100% or 0%, for validity ratios closer to 50%, revision based on a more detailed estimation of the validity ratio achieves a higher effectiveness. Therefore, we introduce an advanced ranking function based on these simple impact measures but parametrized by an estimated validity ratio.

4.2.2 Parametrized Ranking

In the last subsection, we discussed how we can increase the effectiveness of the revision by ranking axioms according to the estimation, whether the ontology under revision is expected to be of a high or a low quality. By the means of our running example, we can demonstrate that such a binary estimation of the validity ratio (100% or 0%) does not allow for achieving the optimum in case of validity ratios closer to 50%. Let us assume that Axioms (4.1) and (4.2) are incorrect, i.e., the validity ratio is 75%. Given that all axioms entailing an incorrect axiom will be declined and all axioms entailed by a correct axiom will be approved, it is easy to manually chose an optimal order of evaluation based on the validity ratio. Intuitively, we would start with Axioms 2 and 3, since this would allow us to evaluate all axioms with only two decisions.

Given the validity ratio of 75%, the previously introduced ranking functions based on the binary estimation of the validity ratio are less effective: if we use *impact*$^+$, which shows the highest value of 7 for Axiom (4.1), then the expert would decline the axiom and no subsequent automatic decisions would be possible. Next, Axiom (4.2) is highest ranked, but again declined without any automatic decisions. Finally, when Axiom (4.3) is presented to the expert and approved, all remaining axioms are approved automatically. The ranking function *impact*$^-$ even takes 7 steps, whereas *guaranteed* performs slightly better with (theoretically) 2.8 expert decisions. This is an average for the different possible choices among the highest ranked axioms assuming that these have the same probability of being chosen.

The reason for the lower effectiveness of *impact*$^+$ in the above example is that axioms are chosen according to their approval impact, even if an approval is not probable for that particular axiom. For instance, Axiom (4.1) is presented to the expert in the hope of an approval despite the fact that this could only happen if the validity ratio was 100% (due to the automatic approval of all remaining axioms). To address this issue, we now discuss the ranking function $norm_R$ that can exploit more accurate estimations of the validity ratio. $norm_R$ minimizes the deviation of the fraction of accepted and declined axioms from the expected overall ratios of desired and undesired consequences. To determine this deviation for each axiom α, we first have to compute the fraction of accepted and declined axioms by

normalizing impacts of α to values between 0 and 1. For this purpose, we define functions impact_N^+ and impact_N^-. Since in the case of an approval, we can possibly both accept and decline axioms automatically, an approval influences both, the ratio of accepted and declined axioms. To take both influences into account, in Definition 3 we split the approval impact accordingly into impact^{+a} and impact^{+d}. Along the same lines, we obtain impact_N^{+a} and impact_N^{+a} by normalizing these two components with respect to the expected validity ratio. In contrast to that, in the case of a decline, we can only decline axioms automatically. Therefore, we do not split impact^-.

Definition 4. *Let $\mathcal{O}^?$ be a connected component of the axiom dependency graph (consisting of unevaluated axioms with entailment relationships between them) and let R be the expected validity ratio. The* normalized impact functions *are given by:*

$$\begin{aligned}
\mathit{impact}_N^{+a}(\alpha) &= \frac{1 + \mathit{impact}^{+a}(\alpha)}{|\mathcal{O}^?|}, \\
\mathit{impact}_N^{+d}(\alpha) &= \frac{\mathit{impact}^{+d}(\alpha)}{|\mathcal{O}^?|}, \\
\mathit{impact}_N^{-}(\alpha) &= \frac{1 + \mathit{impact}^{-}(\alpha)}{|\mathcal{O}^?|}.
\end{aligned}$$

The ranking functions $norm_R^{+a}$, $norm_R^{+d}$ and $norm_R^-$ are then defined by

$$\begin{aligned}
norm_R^{+a}(\alpha) &= -|R - \mathit{impact}_N^{+a}(\alpha)|, \\
norm_R^{+d}(\alpha) &= -|1 - R - \mathit{impact}_N^{+d}(\alpha)|, \\
norm_R^{-}(\alpha) &= -|1 - R - \mathit{impact}_N^{-}(\alpha)|.
\end{aligned}$$

Finally, the ranking function $norm_R$ for an axiom α is defined as

$$\max(norm_R^{+a}(\alpha), norm_R^{+d}(\alpha), norm_R^{-}(\alpha)).$$

When computing impact_N^{+a}, we increment it by 1, since we are interested in the overall fraction of accepted axioms, and, α itself is one of the accepted axioms. For the same reason, we also increment impact_N^- by 1, but not impact_N^{+d}, where α itself is accepted and does not increment the number of declined axioms.

Axiom	$impact_N^{+a}$	$impact_N^{+d}$	$impact_N^{-}$	$norm_{0.75}^{+a}$	$norm_{0.75}^{+d}$	$norm_{0.75}^{-}$	$norm_{0.75}$
(1)	100.0%	0.0%	12.5%	-25.0%	-25.0%	-12.5%	-12.5%
(2)	87.5%	0.0%	25.0%	-12.5%	-25.0%	0.0%	0.0%
(3)	75.0%	0.0%	37.5%	0.0%	-25.0%	-12.5%	0.0%
(4)	62.5%	0.0%	50.0%	-12.5%	-25.0%	-25.0%	-12.5%
(5)	50.0%	0.0%	62.5%	-25.0%	-25.0%	-37.5%	-25.0%
(6)	37.5%	0.0%	75.0%	-37.5%	-25.0%	-50.0%	-25.0%
(7)	12.5%	0.0%	87.5%	-62.5%	-25.0%	-62.5%	-25.0%
(8)	12.5%	0.0%	87.5%	-62.5%	-25.0%	-62.5%	-25.0%

Table 4.2: *The values for $norm_{0.75}$ and the intermediate functions (shown in percentage)*

Table 4.2 shows the computation of $norm_{0.75}$ for Example 1. The function $norm_R^{+a}$ captures how the fraction of automatically accepted axioms deviates from the expected overall ratio of wanted consequences, e.g., accepting Axioms (4.2) or (4.4) yields a deviation of 12.5%: for the former axiom we have automatically accepted too many axioms, while for the latter we do not yet have accepted enough under the premise that the validity ratio is indeed 75%. Since Example 1 does not contain any conflicting axioms, the case of an automatic decline after an approval does not occur, i.e., $impact_N^{+d} = 0$. Therefore, the function $norm_R^{+d}$ shows that for each accept, we still deviate 25% from the expected ratio of *invalid* axioms, which is $1 - R$, i.e., 25%. The function $norm_R^{-}$ works analogously for declines. Hence, $norm_R$ is defined in a way that it takes the greatest value if the chance that all wanted (unwanted) axioms are accepted (declined) at once is maximal. Ranking based on the above idea is most effective in case that most of the connected axiom dependency graph components have a chain-like structure. The connected axiom dependency graph component presented in Table 4.2 is almost a chain, since only the bottom-most axioms (Axioms (4.7) and (4.8)) have a common predecessor. In cases analogous to Example 1, where a connected axiom dependency graph component consists only of class assertions about a simple instance, the structure is chain-line, if the corresponding TBox contains none or only few subsumptions realizing multiple inheritance.

Note that the expected validity ratio within the corresponding connected axiom dependency graph component needs to be adjusted after each expert decision, to reflect the expected validity ratio of the remaining unevaluated axioms. For instance, after Axiom (4.2) has been declined, $norm_{1.00}$ needs to be applied to rank the remaining axioms. If, however, Axiom (4.3) has been accepted, $norm_{0.00}$ is required.

Further, it is interesting to observe that employing $norm_{0.00}$ for ranking yields the same behavior as *impact*$^-$. On the other hand, $norm_{1.00}$ corresponds to *impact*$^+$ in case no conflicting axioms are involved, which is in fact very probable if R is close to 100%. Therefore, $norm$ represents a generalization of the earlier introduced impact functions *impact*$^+$ and *impact*$^-$.

Since the validity ratio is generally not known a priori, in the following, we show how one can work with an estimate that is continuously improved over the course of the revision process.

4.2.3 Learning the Validity Ratio

Users might only have a rough idea or even no idea at all of the validity ratio of a dataset in advance of the revision. Hence, it might be difficult or impossible to decide upfront which R to use for $norm_R$. To address this problem, we investigate how efficiently we can "learn" the validity ratio on the fly. In this setting, the user gives an a prior estimate for R (or we use 50% as default) and with each revision of another connected axiom dependency graph component, R is adjusted to reflect exactly the actual validity ratio at the current stage– the proportion of (manually and automatically) approved axioms within the total set of the evaluated axioms so far. Thus, the algorithm tunes itself towards an optimal ranking function, which relieves the user from choosing a validity ratio. We call the according ranking function $dynnorm$ as it dynamically adapts the estimated validity ratio over the course of the revision.

In our experiments, we show that, already for small datasets, $dynnorm$ outperforms random ordering and, in case of sufficiently large datasets, the estimate converges towards the actual validity ratio, thereby making the assumption of a known validity ratio obsolete.

4.3 Computational Effort

Since revision is an interactive process, also the computational effort required for the proposed reasoning support has to be taken into account. Computing the closure together with axiom impacts after each evaluation (Algorithm 1 lines 1, 5, and 7) can be considered very expensive. According to our profiling measurements, the reasoner methods take over 99% of the computation time. Therefore, computational effort is mostly determined by the number of reasoner calls. In this section, we introduce *decision spaces*, auxiliary data structures which significantly reduce the cost of computing the closure upon elementary revisions and provide an elegant way of determining high impact axioms. Subsequently, we combine decision spaces with a straightforward partitioning approach.

4.3.1 Decision Spaces

Examining Definitions 2 and 3 in detail, we notice that there are two binary relations on unevaluated axioms that are required in order to determine both, revision closure and axiom impacts given a particular revision state $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$:

$$\mathcal{O}^{\vDash} \cup \{\alpha\} \models \beta \qquad (\alpha E \beta)$$

$$\mathcal{O}^{\vDash} \cup \{\alpha, \beta\} \models \gamma \text{ for some } \gamma \in \mathcal{O}^{\nvDash} \qquad (\alpha C \beta)$$

In fact, in order to compute all axiom impacts for a revision state, we require complete knowledge about the relations E and C. A naive approach would require $n^2 + m \cdot n^2$ reasoner calls at each revision step, where n is the number of unevaluated axioms and $m = |\mathcal{O}^{\nvDash}|$. In what follows, we discuss a more efficient alternative realized by *decision spaces*. Intuitively, the purpose of decision spaces is to keep track of the dependencies between the axioms in such a way, that we can read-off the consequences of revision state refinements upon an approval or a decline of an axiom without calling the reasoner. Thereby, on the one hand, we avoid checking the same entailments several times, and, on the other hand, can exploit particular properties of the relations E and C in order to partially complete the relations without calling the reasoner. Example 4 demonstrates the main idea.

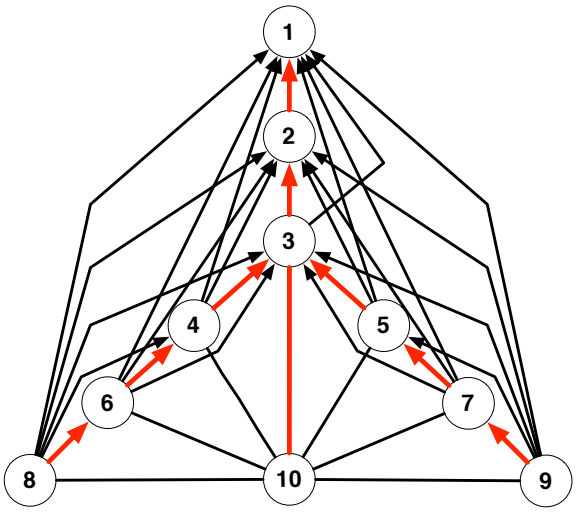


Figure 4.3: *Decision space for Example 4.*

Example 4. *Consider a revision state* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *after an application of revision closure with the set* $\mathcal{O}^{?} \subseteq \mathcal{O}$ *of unevaluated axioms given in Fig.4.3. The directed edges represent the relation* E *and undirected edges the relation* C *holding between axioms in* $\mathcal{O}^{?}$*. Assume that we checked the relations shown by red edges by the means of a reasoner. Then, we can deduce the relations represented by black edges without calling the reasoner, since, on the one hand,* E *is transitive, and, on the other hand,* $\alpha E \beta$ *and* $\beta C \gamma$ *imply* $\alpha C \gamma$ *for all* $\alpha, \beta, \gamma \in \mathcal{O}^{?}$*.*

In addition to the reduction of reasoning calls discussed above, we will show that we can reuse the information given by a decision space when computing the corresponding decision space for the next revision step, thereby avoiding many costly recomputations. We define a decision space as follows.

Definition 5 (Decision Space)**.** *Given a revision state* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *with* $\mathcal{O}^{\not\models} \neq \emptyset$*, let*

$$\mathcal{O}^{?} := \mathcal{O} \setminus (\{\alpha \mid \mathcal{O}^{\models} \models \alpha\} \cup \{\alpha \mid \mathcal{O}^{\models} \cup \{\alpha\} \models \beta, \beta \in \mathcal{O}^{\not\models}\}).$$

The according decision space *is defined by* $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})} = (\mathcal{O}^{?}, E, C)$.

The requirement that $\mathcal{O}^{\models} \neq \emptyset$ is without loss of generality since we can always add an axiom that expresses an inconsistency, which is clearly undesired. On the other hand, the non-emptiness condition ensures that two axioms which together lead to an inconsistency are indeed recognized as conflicting. For example, consider the following two axioms:

$$\mathsf{SameIndividual(\ a\ b)} \tag{4.17}$$

$$\mathsf{DifferentIndividuals(\ a\ b)} \tag{4.18}$$

We assume that Axiom (4.17) has just been approved and belongs, therefore, to $\mathcal{O}^{\models}$, whereas Axiom (4.18) is a not yet evaluated axiom. Clearly, Axiom (4.17) and (4.18) cannot be true at the same time and, consequently, the inconsistent ontology $\mathcal{O}^{\models} \cup \{(4.18)\}$ entails any axiom, but, unless we have some axiom β in $\mathcal{O}^{\models}$, this will not be recognized.

As a direct consequence of this definition, we have $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\models})} = \mathbb{D}_{\mathrm{clos}(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\models})}$. The following properties follow immediately from the above definition:

Lemma 2. *For any decision space $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\models})} = (\mathcal{O}^{?}, E, C)$, the following hold:*

- P1 *$(\mathcal{O}^{?}, E)$ is a quasi-order (i.e., reflexive and transitive),*
- P2 *C is symmetric,*
- P3 *$\alpha E \beta$ and $\beta C \gamma$ imply $\alpha C \gamma$ for all $\alpha, \beta, \gamma \in \mathcal{O}^{?}$, and*
- P4 *if $\alpha E \beta$ then $\alpha C \beta$ does not hold.*

Proof. For P1, due to the assumed properties (monotonicity, extensivity and idempotency) of the underlying logic we have $\{\alpha\} \models \alpha$ (extensivity) and $\mathcal{O}^{\models} \cup \{\alpha\} \models \alpha$ (monotonicity) and it follows that E is reflexive. Given $\mathcal{O}^{\models} \cup \{\alpha\} \models \beta$ and $\mathcal{O}^{\models} \cup \{\beta\} \models \gamma$, idempotency ensures $\mathcal{O}^{\models} \cup \{\alpha\} \models \gamma$, hence E is transitive. For P2, symmetry of C is an immediate consequence from its definition. For showing P3, suppose $\mathcal{O}^{\models} \cup \{\alpha\} \models \beta$ and $\mathcal{O}^{\models} \cup \{\beta, \gamma\} \models \delta$ for some $\delta \in \mathcal{O}^{\models}$. Monotonicity allows to get $\mathcal{O}^{\models} \cup \{\alpha, \gamma\} \models \beta$ from the former and $\mathcal{O}^{\models} \cup \{\alpha, \beta, \gamma\} \models \delta$ from the latter, whence $\mathcal{O}^{\models} \cup \{\alpha, \gamma\} \models \delta$ follows via idempotency. To see that E and C are mutually exclusive (P4), assume the contrary, i.e., $\mathcal{O}^{\models} \cup \{\alpha\} \models \beta$ and $\mathcal{O}^{\models} \cup \{\alpha, \beta\} \models \gamma$ for some $\gamma \in \mathcal{O}^{\models}$ hold simultaneously. Yet, idempotency

allows to conclude $\mathcal{O}^{\models} \cup \{\alpha\} \models \gamma$. However then α cannot be contained in $\mathcal{O}^{?}$ by definition, which gives a contradiction and proves the claim. □

In fact, the properties established in Lemma 2 are characteristic. This means that any structure satisfying these properties can be seen as the decision space for an appropriate revision state:[1]

Lemma 3. *Let V be finite set and let $\underline{E}, \underline{C} \subseteq V \times V$ be relations for which $(V, \underline{E})$ is a quasi-order, $\underline{C} = \underline{C}^{-}$, $\underline{E} \circ \underline{C} \subseteq \underline{C}$ and $\underline{E} \cap \underline{C} = \emptyset$. Then there is a decision space $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\not\models})}$ isomorphic to $(V, \underline{E}, \underline{C})$.*

Proof. As a very basic formalism, we choose propositional logic as KR language. Let $\mathcal{O}$ contain one atomic proposition p_v for every $v \in V$, let $\mathcal{O}^{\models} = \{p_v \rightarrow p_{v'} \mid v\underline{E}v'\} \cup \{\neg p_v \vee \neg p_{v'} \mid v\underline{C}v'\}$ and let $\mathcal{O}^{\not\models} = \{false\}$. First observe that $\mathcal{O}^{?} = \{p_v \mid v \in V\}$. Next, we claim that the function $f : V \rightarrow \mathcal{O}$ with $v \mapsto p_v$ is an isomorphism between $(V, \underline{E}, \underline{C})$ and $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\not\models})}$. Clearly, f is a bijection. Moreover, $v\underline{E}v'$ implies $p_v E p_{v'}$ by modus ponens since $p_v \rightarrow p_{v'} \in \mathcal{O}^{\models}$. Likewise, $v\underline{C}v'$ implies $p_v C p_{v'}$ due to $\neg p_v \vee \neg p_{v'} \in \mathcal{O}^{\models}$. The two other directions are shown indirectly.

Let $\uparrow v = \{\tilde{v} \mid v\underline{E}\tilde{v}\}$. To show that $p_v E p_{v'}$ implies $v\underline{E}v'$ assume there are $p_v, p_{v'}$ with $p_v E p_{v'}$, but $v\underline{E}v'$ does not hold. Now, consider the propositional interpretation mapping $p_{\tilde{v}}$ to *true* whenever $\tilde{v} \in \uparrow v$ and to *false* otherwise. It can be verified that this interpretation is a model of $\mathcal{O}^{\models}$ and satisfies p_v as well as $\neg p_{v'}$, hence $\mathcal{O}^{\models} \cup \{p_v\} \not\models p_{v'}$ and consequently $p_v E p_{v'}$ cannot hold, so we have a contradiction.

To show that $p_v C p_{v'}$ implies $v\underline{C}v'$ assume there are $p_v, p_{v'}$ with $p_v C p_{v'}$, but $v\underline{C}v'$ does not hold. Now, consider the propositional interpretation mapping $p_{\tilde{v}}$ to *true* whenever $\tilde{v} \in \uparrow v \cup \uparrow v'$ and to *false* otherwise. It can be verified that this interpretation is a model of $\mathcal{O}^{\models}$ and satisfies p_v as well as $p_{v'}$, hence $\mathcal{O}^{\models} \cup \{p_v, p_{v'}\} \not\models$ *false* and consequently $p_v C p_{v'}$ cannot hold, so we have a contradiction. □

The following lemma shows how decision spaces can be used for computing the closure of an updated revision state and the corresponding updated impacts of

[1] As usual, we let $R^{-} = \{(y, x) \mid (x, y) \in R\}$ as well as $R \circ S = \{(x, z) \mid (x, y) \in R, (y, z) \in S \text{ for some } y\}$.

axioms upon an elementary refinement of a given revision state. As usual for (quasi)orders, we define $\uparrow\alpha = \{\beta \mid \alpha E \beta\}$ and $\downarrow\alpha = \{\beta \mid \beta E \alpha\}$. Moreover, we let $\wr\alpha = \{\beta \mid \alpha C \beta\}$.

Lemma 4. *Given* $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\vDash},\mathcal{O}^{\nvDash})} = (\mathcal{O}^{?}, E, C)$ *for a revision state* $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$ *such that* $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash}) = \mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$ *with* $\mathcal{O}^{\nvDash} \neq \emptyset$ *and* $\alpha \in \mathcal{O}^{?}$*, then the following statements hold:*

1. $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash}) = (\mathcal{O}, \mathcal{O}^{\vDash} \cup \uparrow\alpha, \mathcal{O}^{\nvDash} \cup \wr\alpha)$,
2. $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \{\alpha\}) = (\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \downarrow\alpha)$,
3. $\mathit{impact}^{+}(\alpha) = |\uparrow\alpha| + |\wr\alpha| - 1$, *and*
4. $\mathit{impact}^{-}(\alpha) = |\downarrow\alpha| - 1$.

Proof. 1. By definition of closures, we have that $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash})$ is $(\mathcal{O}, \mathcal{O}_c^{\vDash}, \mathcal{O}_c^{\nvDash})$ for $\mathcal{O}_c^{\vDash} = \{\beta \in \mathcal{O} \mid \mathcal{O}^{\vDash} \cup \{\alpha\} \models \beta\}$ and $\mathcal{O}_c^{\nvDash} = \{\beta \in \mathcal{O} \mid \mathcal{O}^{\vDash} \cup \{\alpha, \beta\} \models \gamma, \gamma \in \mathcal{O}^{\nvDash}\}$.

By definition of the entails and conflicts relation we obtain $\mathcal{O}_c^{\vDash} = \mathcal{O}^{\vDash} \cup \{\beta \in \mathcal{O}^{?} \mid \alpha E \beta\}$ and $\mathcal{O}_c^{\nvDash} = \mathcal{O}^{\nvDash} \cup \{\beta \in \mathcal{O}^{?} \mid \alpha C \beta\}$.

By definition of $\uparrow\alpha$ and $\wr\alpha$ follows $\mathcal{O}_c^{\vDash} = \mathcal{O}^{\vDash} \cup \uparrow\alpha$ and $\mathcal{O}_c^{\nvDash} = \mathcal{O}^{\nvDash} \cup \wr\alpha$. Thus we obtain $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash}) = (\mathcal{O}, \mathcal{O}^{\vDash} \cup \uparrow\alpha, \mathcal{O}^{\nvDash} \cup \wr\alpha)$ as claimed.

2. Since $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$ is already closed, $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \{\alpha\})$ is $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}_c^{\nvDash})$ with $\mathcal{O}_c^{\nvDash} = \{\beta \in \mathcal{O} \mid \mathcal{O}^{\vDash} \cup \{\beta\} \models \gamma \text{ for some } \gamma \in (\mathcal{O}^{\nvDash} \cup \{\alpha\})\}$. Due to the prior closedness, α is the only possible γ that will yield some β, hence $\mathcal{O}_c^{\nvDash} = \mathcal{O}^{\nvDash} \cup \{\beta \in \mathcal{O}^{?} \mid \mathcal{O}^{\vDash} \cup \{\beta\} \models \alpha\}$. By definition of the entails relation, this implies $\mathcal{O}_c^{\nvDash} = \mathcal{O}^{\nvDash} \cup \{\beta \in \mathcal{O}^{?} \mid \beta E \alpha\})$, whence by definition of $\downarrow\alpha$ follows $\mathcal{O}_c^{\nvDash} = \mathcal{O}^{\nvDash} \cup \downarrow\alpha)$. Therefore $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \{\alpha\}) = (\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \downarrow\alpha)$

3. By Definition 3, $\mathrm{impact}^{+}(\alpha) = ?(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash}) - ?(\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash}))$. By Definition 2, $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash})$ equals

$$(\mathcal{O}, \{\beta \in \mathcal{O} \mid \mathcal{O}^{\vDash} \cup \{\alpha\} \models \beta\}, \{\beta \in \mathcal{O} \mid \mathcal{O}^{\vDash} \cup \{\alpha, \beta\} \models \gamma, \gamma \in \mathcal{O}^{\nvDash}\}).$$

By the definition of $?(\cdot)$ (Definition 3), $\text{impact}^{+}(\alpha) = |\mathcal{O} \setminus (\mathcal{O}^{\models} \cup \{\alpha\} \cup \mathcal{O}^{\not\models})| - |\mathcal{O} \setminus (\mathcal{O}_{\alpha}^{\models} \cup \mathcal{O}_{\alpha}^{\not\models})|$ where $\mathcal{O}_{\alpha}^{\models} = \{\beta \in \mathcal{O} \mid \mathcal{O}^{\models} \cup \{\alpha\} \models \beta\}$ and $\mathcal{O}_{\alpha}^{\not\models} = \{\beta \in \mathcal{O} \mid \mathcal{O}^{\models} \cup \{\alpha, \beta\} \models \gamma, \gamma \in \mathcal{O}^{\not\models}\}$.

By definition of the entails and conflicts relations, the term $|\mathcal{O} \setminus (\mathcal{O}_{\alpha}^{\models} \cup \mathcal{O}_{\alpha}^{\not\models})|$ equals

$$|\mathcal{O} \setminus (\mathcal{O}^{\models} \cup \{\beta \in \mathcal{O}^{?} \mid \alpha E \beta\} \cup \mathcal{O}^{\not\models} \cup \{\beta \in \mathcal{O}^{?} \mid \alpha C \beta\})|,$$

which, by definition of $\uparrow$ and $\wr$, is $|\mathcal{O} \setminus (\mathcal{O}^{\models} \cup \uparrow\alpha \cup \mathcal{O}^{\not\models} \cup \wr\alpha)|$. Overall we then have $\text{impact}^{+}(\alpha) = |\mathcal{O}| - (|\mathcal{O}^{\models}| + 1 + |\mathcal{O}^{\not\models}|) - (|\mathcal{O}| - (|\mathcal{O}^{\models}| + |\uparrow\alpha| + |\mathcal{O}^{\not\models}| + |\wr\alpha|))$, which is $|\uparrow\alpha| + |\wr\alpha| - 1$.

4. By Definition 3, $\text{impact}^{-}(\alpha) = ?(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models} \cup \{\alpha\}) - ?(\text{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models} \cup \{\alpha\}))$. By Definition 2, the latter is $?(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models} \cup \{\beta \in \mathcal{O} \mid \mathcal{O}^{\models} \cup \{\beta\} \models \alpha\})$. Using Definition 3, $\text{impact}^{-}(\alpha)$ then is:

$$|\mathcal{O} \setminus (\mathcal{O}^{\models} \cup \mathcal{O}^{\not\models} \cup \{\alpha\})| - |\mathcal{O} \setminus (\mathcal{O}^{\models} \cup \mathcal{O}^{\not\models} \cup \{\beta \in \mathcal{O} \mid \mathcal{O}^{\models} \cup \{\beta\} \models \alpha\})|$$

By definition of the entails relation the latter is $|\mathcal{O} \setminus (\mathcal{O}^{\models} \cup \mathcal{O}^{\not\models} \cup \{\beta \in \mathcal{O}^{?} \mid \beta E \alpha\})|$, which, by definition of $\downarrow$, is $|\mathcal{O} \setminus (\mathcal{O}^{\models} \cup \mathcal{O}^{\not\models} \cup \downarrow\alpha)|$. Thus $\text{impact}^{-}(\alpha) = |\mathcal{O}| - (|\mathcal{O}^{\models}| + |\mathcal{O}^{\not\models}| + 1) - (|\mathcal{O}| - (|\mathcal{O}^{\models}| + |\mathcal{O}^{\not\models}| + |\downarrow\alpha|)) = |\downarrow\alpha| - 1$. □

Hence, the computation of the revision closure (lines 5 and 7 of Algorithm 1) and axiom impacts does not require any entailment checks if the according decision space is available. For the computation of decision spaces, we exploit the structural properties established in Lemmas 2 and 3 in order to reduce the number of required entailment checks in cases where the relations E and C are partially known. For this purpose, we define the rules R0 to R9 displayed in Table 4.3, which describe the interplay between the relations E and C and their complements $\overline{E}$ and $\overline{C}$. The rules can serve as production rules to derive new instances of these relations thereby minimizing calls to costly reasoning procedures. By virtue of Lemma 3, we also have the guarantee that no further rules of this kind can be created, i.e., the rule set is complete for decision spaces.

R0		$\rightarrow$	$E(x,x)$	reflexivity of E
R1	$E(x,y) \wedge E(y,z)$	$\rightarrow$	$E(x,z)$	transitivity of E
R2	$E(x,y) \wedge C(y,z)$	$\rightarrow$	$C(x,z)$	(P3)
R3	$C(x,y)$	$\rightarrow$	$C(y,x)$	symmetry of C
R4	$E(x,y)$	$\rightarrow$	$\overline{C}(x,y)$	disjointness of E and C
R5	$\overline{C}(x,y)$	$\rightarrow$	$\overline{C}(y,x)$	symmetry of C
R6	$E(x,y) \wedge \overline{C}(x,z)$	$\rightarrow$	$\overline{C}(y,z)$	(P3)
R7	$C(x,y)$	$\rightarrow$	$\overline{E}(x,y)$	disjointness of E and C
R8	$\overline{C}(x,y) \wedge C(y,z)$	$\rightarrow$	$\overline{E}(x,z)$	(P3)
R9	$E(x,y) \wedge \overline{E}(x,z)$	$\rightarrow$	$\overline{E}(y,z)$	transitivity of E

Table 4.3: *Completion rules for partially known decision spaces*

An analysis of the dependencies between the rules R0 to R9 reveals an acyclic structure (indicated by the order of the rules). Therefore E,C,$\overline{C}$, and $\overline{E}$ can be saturated one after another. Moreover, the exhaustive application of the rules R0 to R9 can be condensed into the following operations:

$$\begin{aligned} E &\leftarrow E^* \\ C &\leftarrow E \circ (C \cup C^-) \circ E^- \\ \overline{C} &\leftarrow E^- \circ (\overline{C} \cup Id \cup \overline{C}^-) \circ E \\ \overline{E} &\leftarrow E^- \circ (\overline{C} \circ C \cup \overline{E}) \circ E^- \end{aligned}$$

The correctness of the first operation (where $(\cdot)^*$ denotes the reflexive and transitive closure) is a direct consequence of R0 and R1. For the second operation, we exploit the relationships

$$E \circ C \circ E^- \overset{\text{R2}}{\subseteq} C \circ E^- \overset{\text{R3}}{\subseteq} C^- \circ E^- = (E \circ C)^- \overset{\text{R2}}{\subseteq} C^- \overset{\text{R3}}{\subseteq} C$$

$$E \circ C^- \circ E^- = E \circ (E \circ C)^- \overset{\text{R2}}{\subseteq} E \circ C^- \overset{\text{R3}}{\subseteq} E \circ C \overset{\text{R2}}{\subseteq} C$$

that can be further composed into

$$E \circ C \circ E^- \cup E \circ C^- \circ E^- = E \circ (C \cup C^-) \circ E^- \subseteq C$$

Conversely, iterated backward chaining for C with respect to R2 and R3 yields $E \circ (C \cup C^-) \circ E^-$ as a fixpoint, under the assumption $E = E^*$. The correctness of the last two operations can be shown accordingly.

Algorithm 2 realizes the cost-saving identification of the complete entailment and conflict relations of a decision space. Maintaining sets of known entailments (E), non-entailments ($\overline{E}$), conflicts (C) and non-conflicts ($\overline{C}$), the algorithm always closes these sets under the above operations before it cautiously executes expensive deduction checks to clarify missing cases. First, the initially known (non-) entailments and (non-) 0 conflicts are closed in the aforementioned way (lines 1–7). There and in the subsequent lines, we split computations into several ones where appropriate in order to minimize the size of sets subject to the join operation ($\circ$). Lines 8–26 describe the successive clarification of the entailment relation (for cases where neither entailment nor non-entailment is known yet) via deduction checks. After each such clarification step, the sets $E, \overline{E}, C$, and $\overline{C}$ are closed. Thereby, we exploit known properties of intermediate results such as already being transitive or symmetric to avoid redoing the according closure operations unnecessarily (`transupdatediff` computes, for a relation R and a pair of elements (α, β), the difference between the reflexive transitive closure of R extended with (α, β) and R^*, i.e., $(R \cup \{(\alpha, \beta)\})^* \setminus R^*)$). Likewise, we also avoid redundant computations and reduce the size of the input sets for the join operations by explicitly bookkeeping sets E',C',$\overline{C}'$, and $\overline{E}'$ containing only the instances newly added in the current step. Lines 27–38 proceed in an analogous way with the stepwise clarification of the conflicts relation.

Now we consider the complexity of the above given decision space completion. Since the complexity of entailment checking will almost always outweigh the complexity of the other operations in Algorithm 2, we first analyze the complexity of the algorithm under the assumption that entailment checking is done by a constant time oracle. We then show how entailment checking can be factored in.

Lemma 5. *Let $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ be a revision state with $\mathcal{O}^{\not\models} \neq \emptyset$ and $E, \overline{E}, C, \overline{C}$ (possibly empty) subsets of the entailment and conflicts relations. We denote the size $|\mathcal{O}|$ of $\mathcal{O}$ with n. Given $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ and $E, \overline{E}, C, \overline{C}$ as input, Algorithm 2*

Algorithm 2: Decision Space Completion

Input: $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ a consistent revision state; $E, \overline{E}, C, \overline{C}$ subsets of the entailment and conflict relations and their complements

Output: $(\mathcal{O}^?, E, C)$ the corresponding decision space

$E \leftarrow E^*$
$C \leftarrow E \circ C \circ E^-$
$C \leftarrow C \cup C^-$
$\overline{C} \leftarrow E^- \circ \overline{C} \cup Id_{\mathcal{O}^?} \circ E$
$\overline{C} \leftarrow \overline{C} \cup \overline{C}^-$
$\overline{E} \leftarrow (\overline{C} \circ C) \cup \overline{E}$
$\overline{E} \leftarrow E^- \circ \overline{E} \circ E^-$
while $E \cup \overline{E} \neq \mathcal{O}^? \times \mathcal{O}^?$ **do**
  pick one $(\alpha, \beta) \in \mathcal{O}^? \times \mathcal{O}^? \setminus (E \cup \overline{E})$
  if $\mathcal{O}^{\models} \cup \{\alpha\} \models \beta$ **then**
    $E' \leftarrow \texttt{transupdatediff}(E, (\alpha, \beta))$
    $E \leftarrow E \cup E'$
    $C' \leftarrow (E' \circ C) \setminus C$
    $C' \leftarrow C' \cup (C' \circ E'^-) \setminus C$
    $C \leftarrow C \cup C'$
    $\overline{C}' \leftarrow (E'^- \circ \overline{C}) \setminus \overline{C}$
    $\overline{C}' \leftarrow \overline{C}' \cup (\overline{C}' \circ E') \setminus \overline{C}$
    $\overline{C} \leftarrow \overline{C} \cup \overline{C}'$
    $\overline{E}' \leftarrow ((\overline{C}' \circ C) \cup (\overline{C} \circ C')) \setminus \overline{E}$
    $\overline{E} \leftarrow \overline{E} \cup \overline{E}'$
    $\overline{E}' \leftarrow ((E'^- \circ \overline{E}) \cup (E^- \circ \overline{E}')) \setminus \overline{E}$
    $\overline{E} \leftarrow \overline{E} \cup \overline{E}' \cup (\overline{E}' \circ E^-) \cup (\overline{E} \circ E'^-)$
  else
    $\overline{E} \leftarrow \overline{E} \cup (E^- \circ \{(\alpha, \beta)\} \circ E^-)$
  end if
end while
while $C \cup \overline{C} \neq \mathcal{O}^? \times \mathcal{O}^?$ **do**
  pick one $(\alpha, \beta) \in \mathcal{O}^? \times \mathcal{O}^? \setminus (C \cup \overline{C})$
  if $\mathcal{O}^{\models} \cup \{\alpha, \beta\} \models \gamma$ for some $\gamma \in \mathcal{O}^{\not\models}$ **then**
    $C' \leftarrow E \circ \{(\alpha, \beta), (\beta, \alpha)\} \circ E^-$
    $C \leftarrow C \cup C'$
    $\overline{E} \leftarrow \overline{E} \cup (E^- \circ \overline{C} \circ C' \circ E^-)$
  else
    $\overline{C}' \leftarrow (E^- \circ \{(\alpha, \beta), (\beta, \alpha)\} \circ E) \setminus \overline{C}$
    $\overline{C} \leftarrow \overline{C} \cup \overline{C}'$
    $\overline{E} \leftarrow \overline{E} \cup (E^- \circ \overline{C}' \circ C \circ E^-)$
  end if
end while

runs in time bounded by $O(n^5)$ and space bounded by $O(n^2)$ if we assume that entailment checking is a constant time operation.

Proof. We first note that $\mathcal{O}^?$ is bounded by n since $|\mathcal{O}^?| = |\mathcal{O}| - (|\mathcal{O}^\models| + |\mathcal{O}^\not\models|)$. Similarly, the size of each relation $E, \overline{E}, C$, and $\overline{C}$ is bounded by n^2 since the relations are binary relations over axioms in $\mathcal{O}$. We first analyze the individual operations. Computing the transitive reflexive closure of a relation can be done in cubic time, i.e., for E^* with E a relation over at most n axioms, we get a bound of n^3. The computation of `transupdatediff` is in the worst case the same as computing the reflexive transitive closure. For a binary join operation ($\circ$), the output is again a binary relation over $\mathcal{O}$ of size bounded by n^2. Each binary join can be computed in at most n^3 steps. Note that multiple joins can be seen as several binary joins, since, in case of formalisms where entailment checking is harder than PTIME, each intermediate relation is again over axioms from $\mathcal{O}$ and is of size at most n^2. The union operation ($\cup$) corresponds to the addition of axioms. Each of the while loops is executed at most n^2 times and requires a fixed number of join operations and possibly in one case the computation of `transupdatediff`, which gives an upper bound of $O(n^2 \cdot n^3) = O(n^5)$ for the both while loops. Together with the reflexive transitive closure and the fixed number of join operations before the while loops, we have that the time complexity of Algorithm 2 is $O(n^5)$ and its space complexity is $O(n^2)$ assuming that entailment checking is a constant time operation. □

Given the complexity $c(n)$ of deciding entailment in a particular logic, we obtain the overall complexity as follows.

Lemma 6. *Let $(\mathcal{O}, \mathcal{O}^\models, \mathcal{O}^\not\models)$ be a revision state with $\mathcal{O}^\not\models \neq \emptyset$, $|\mathcal{O}| := n$ and the axioms in $\mathcal{O}$ expressed in a logic $\mathcal{L}$ in which taking all consequences is a closure operation and for which there is a decision procedure for logical entailment of complexity $c(n)$ where n is the size of the input to the procedure. Let $E, \overline{E}, C, \overline{C}$ be (possibly empty) subsets of the according entailment and conflicts relations. Then there is a polynomial p such that the runtime of Algorithm 2, given $(\mathcal{O}, \mathcal{O}^\models, \mathcal{O}^\not\models)$ and $E, \overline{E}, C, \overline{C}$ as input, is bounded by $p(n) \cdot c(n)$.*

Proof. The input to the entailment checking algorithm is in all cases of size n. Both while loops perform at most n^2 entailment checks, which together with the analysis from Lemma 5 give the desired result. □

In case the entailment checking problem is not tractable, i.e., harder than PTIME, the improved efficiency is clear from the theoretical point of view. In the following, we discuss cost-saving update of decision spaces during the revision, which further increases the added value of decision spaces and explains an improved performance also in case of tractable logics.

Updating Decision Spaces

Now we discuss the change of the decision space as a consequence of approving or declining one axiom with the objective of again minimizing the required number of entailment checks. We first consider the case that an expert approves an axiom $\alpha \in \mathcal{O}^?$, and hence α is added to the set $\mathcal{O}^\models$ of wanted consequences.

Lemma 7. *Let* $\mathbb{D}_{(\mathcal{O},\mathcal{O}^\models,\mathcal{O}^\not\models)} = (\mathcal{O}^?, E, C)$, $\alpha \in \mathcal{O}^?$, *and* $\mathbb{D}_{\mathrm{clos}(\mathcal{O},\mathcal{O}^\models\cup\{\alpha\},\mathcal{O}^\not\models)} = (\mathcal{O}^?_{\mathrm{new}}, E', C')$. *Then*

- $\mathcal{O}^?_{\mathrm{new}} = \mathcal{O}^? \setminus (\uparrow\alpha \cup \wr\alpha)$,
- $\beta E \gamma$ *implies* $\beta E' \gamma$ *for* $\beta, \gamma \in \mathcal{O}^?_{\mathrm{new}}$, *and*
- $\beta C \gamma$ *implies* $\beta C' \gamma$ *for* $\beta, \gamma \in \mathcal{O}^?_{\mathrm{new}}$.

Essentially, Lemma 7 states that all axioms entailed by α (as witnessed by E) as well as all axioms conflicting with α (indicated by C) will be removed from the decision space if α is approved. This is an immediate consequence of Lemma 4. Moreover due to monotonicity, all positive information about entailments and conflicts remains valid. Algorithm 3 takes advantage of these correspondences when fully determining the updated decision space.

Lemma 8. *Let* $\mathbb{D}_{(\mathcal{O},\mathcal{O}^\models,\mathcal{O}^\not\models)}$ *be a decision space,* $\alpha \in \mathcal{O}^?$ *an axiom. We denote the size* $|\mathcal{O}|$ *of* $\mathcal{O}$ *with* n. *Given* $\mathbb{D}_{(\mathcal{O},\mathcal{O}^\models,\mathcal{O}^\not\models)}$ *and* α *as input, Algorithm 3 runs in time bounded by* $O(n^5)$ *and space bounded by* $O(n^2)$ *if we assume that entailment checking is a constant time operation.*

Algorithm 3: Update of Decision Space $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\not\models})}$ on Approving α

Input: $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\not\models})}$, $\alpha \in \mathcal{O}^?$
Output: $\mathbb{D}_{\mathrm{clos}(\mathcal{O},\mathcal{O}^{\models}\cup\{\alpha\},\mathcal{O}^{\not\models})}$ updated decision space
1: $\mathcal{O}^? \leftarrow \mathcal{O}^? \setminus (\uparrow\alpha \cup \wr\alpha)$
2: $E \leftarrow E \cap (\mathcal{O}^? \times \mathcal{O}^?)$
3: $C \leftarrow C \cap (\mathcal{O}^? \times \mathcal{O}^?)$
4: $\overline{C} \leftarrow E^- \circ E$
5: $\overline{E} \leftarrow E^- \circ \overline{C} \circ C \circ E^-$
6: execute lines 8–38 from Alg. 2

Algorithm 4: Update of Decision Space $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\not\models})}$ on Declining α

Input: $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\not\models})}$, $\alpha \in \mathcal{O}^?$
Output: $\mathbb{D}_{\mathrm{clos}(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\not\models}\cup\{\alpha\})}$ updated decision space
1: $\mathcal{O}^? \leftarrow \mathcal{O}^? \setminus \downarrow\alpha$,
2: $E \leftarrow E \cap (\mathcal{O}^? \times \mathcal{O}^?)$
3: $\overline{E} \leftarrow \overline{E} \cap (\mathcal{O}^? \times \mathcal{O}^?)$
4: $C \leftarrow C \cap (\mathcal{O}^? \times \mathcal{O}^?)$
5: $\overline{C} \leftarrow E^- \circ E$
6: **while** $C \cup \overline{C} \neq \mathcal{O}^? \times \mathcal{O}^?$ **do**
7: pick one $(\beta, \gamma) \in \mathcal{O}^? \times \mathcal{O}^? \setminus (C \cup \overline{C})$
8: **if** $\mathcal{O}^{\models} \cup \{\beta, \gamma\} \models \alpha$ **then**
9: $C \leftarrow C \cup (E \circ \{(\beta, \gamma), (\gamma, \beta)\} \circ E^-)$
10: **else**
11: $\overline{C} \leftarrow \overline{C} \cup (E^- \circ \{(\beta, \gamma), (\gamma, \beta)\} \circ E)$
12: **end if**
13: **end while**

Proof. Lines 1–5 of Algorithm 3 can be executed in cubic time and quadratic space due to the same arguments as in Lemma 5. By Lemma 5, executing lines 8–38 from Algorithm 2 under the assumption that entailment checking is a constant time operation can be done in time $O(n^5)$, which proves the claim. □

The next lemma considers changes to be made to the decision space on decline of an axiom α by characterizing it as unwanted consequence.

Lemma 9. *Let* $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\not\models})} = (\mathcal{O}^?, E, C)$*,* $\alpha \in \mathcal{O}^?$*, and* $\mathbb{D}_{\mathrm{clos}(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\not\models}\cup\{\alpha\})} = (\mathcal{O}^?_{\mathrm{new}}, E', C')$*. Then*

- $\mathcal{O}^?_{\mathrm{new}} = \mathcal{O}^? \setminus {\downarrow}\alpha$*,*
- $\beta E \gamma$ *exactly if* $\beta E' \gamma$ *for* $\beta, \gamma \in \mathcal{O}^?_{\mathrm{new}}$*, and*
- $\beta C \gamma$ *implies* $\beta C' \gamma$ *for* $\beta, \gamma \in \mathcal{O}^?_{\mathrm{new}}$*.*

The lemma states that the updated decision space can be obtained by removing all axioms that entail α. This is an immediate consequence of Lemma 4. Furthermore, entailments and non-entailments between remaining axioms remain valid due to the unchanged $\mathcal{O}^{\models}$ whereas the set of conflicts may increase. Algorithm 4 implements the respective decision space update, additionally exploiting that new conflicts can only arise from derivability of the newly declined axiom α. Algorithms 3 and 4 have to be called in Alg. 1 after the accept (line 5) or decline revision step (line 7), respectively.

For n the number of involved axioms, Algorithms 2, 3, and 4 run in time bounded by $O(n^5)$ and space bounded by $O(n^2)$ if we treat entailment checking as a constant time operation. Without the latter assumption, the complexity of reasoning usually dominates. For example, if the axioms use all features of OWL 2 DL, entailment checking is N2EXPTIME-complete [KAZAKOV 2008], which then also applies to our algorithm.

Lemma 10. *Let* $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\not\models})}$ *be a decision space,* $\alpha \in \mathcal{O}^?$ *an axiom. We denote the size* $|\mathcal{O}|$ *of* $\mathcal{O}$ *with* n*. Given* $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\models},\mathcal{O}^{\not\models})}$ *and* α *as input, Algorithm 4 runs in time bounded by* $O(n^5)$ *and space bounded by* $O(n^2)$ *if we assume that entailment checking is a constant time operation.*

Proof. The execution lines 1–5 of Algorithm 4 can be performed in quadratic space and cubic time due to the same arguments as in Lemma 5. We execute the operations within the while loop at most n^2 times, and under the assumption that entailment checking is a constant time operation, we find that the operations can again be performed in cubic time and quadratic space resulting in an overall bound for the time complexity of $O(n^5)$ and $O(n^2)$ space complexity. □

4.3.2 Partitioning

In order to further reduce the number of reasoner calls, we combine the optimization using decision spaces with a straight-forward partitioning approach that is applicable for OWL ontologies and splits ABox axioms (i.e., class and property assertions) into disjoint subsets. Thus, the subsequent discussion is specific to OWL reasoning.

Definition 6. *Let $\mathcal{A}$ be a set of ABox axioms, ind($\mathcal{A}$) the set of individual names used in $\mathcal{A}$, then $\mathcal{A}$ is* connected *if, for all pairs of individuals $a, a' \in$ ind($\mathcal{A}$), there exists a sequence $a_1, \ldots, a_n$ such that $a = a_1$, $a' = a_n$, and, for all $1 \leq i < n$, there exists a property assertion in A containing a_i and a_{i+1}. A collection of ABoxes $\mathcal{A}_1, \ldots, \mathcal{A}_k$ is a* partitioning of $\mathcal{A}$ *if $\mathcal{A} = \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_k$, ind($\mathcal{A}_i$) $\cap$ ind($\mathcal{A}_j$) $= \emptyset$ for $1 \leq i < j \leq k$, and each $\mathcal{A}_i$ is connected.*

The proposed partitioning process can be done in linear time, since it is more or less a straightforward product of the computation of the connected components of the ABox graph. In the absence of nominals (OWL's oneOf constructor), the above described partitions or clusters of an ABox are indeed independent, i.e., the above partitioning does not split any connected decision space components at any stage of the revision. This follows from the results obtained by Cuenca Grau et al. [GRAU et al. 2007c] on locality-based modules. Therefore, the revision of each partition can be carried out independently from others. We apply partitioning once at the beginning of the revision to the whole set of unevaluated axioms and then perform the revision for each partition separately by joining the partition with the remaining terminological axioms. So far, we abstract from the possibility to update the partitioning to a more fine-grained one over the course of revision, since it is unclear whether the additional computational overhead pays off.
In order to also partition non-Abox axioms or to take axioms with nominals into account, other partitioning techniques can be applied, e.g., the signature decomposition approach by Konev et al. [KONEV et al. 2010] that partitions the vocabulary of an ontology into subsets that are independent regarding their meaning. The resulting independent subsets of the ontology can then be reviewed independently from each other analogously to the clusters of ABox axioms used in our evaluation. In the next section, we will present empirical results concerning the added

value achieved when using partitioning within the accuracy-based revision. We will show that

- in particular in case of large datasets containing several partitions, the additional partitioning-based optimization significantly reduces the computational effort;
- partitioning intensifies the effectiveness of decision spaces, since the density of entailment and contradiction relations is always higher within each partition than the density within a set of independent partitions.

4.4 Experimental Results

We evaluate the discussed methodology for accuracy-based revision within the project *NanOn* aiming at ontology-supported literature search. During this project, a hand-crafted ontology modeling the scientific domain of nanotechnology has been developed, capturing substances, structures, and procedures used in that domain. The ontology, denoted here with $\mathcal{O}$, is specified in the Web Ontology Language OWL 2 DL [OWL WORKING GROUP 27 October 2009] and comprises 2,289 logical axioms. This ontology is used as the core resource to automatically analyze scientific documents for the occurrence of NanOn classes and properties by the means of lexical patterns. When such classes and properties are found, the document is automatically annotated with them to facilitate topic-specific information retrieval on a fine-grained level. In this way, one of the project outputs is a large amount of class and property assertions associated with the *NanOn* ontology. In order to estimate the accuracy of such automatically added annotations, they need to be inspected by human experts. This provides a natural application scenario for our approach. The manual inspection of annotations yielded sets of valid and invalid annotation assertions (denoted by $\mathcal{A}^+$ and $\mathcal{A}^-$, respectively). To investigate how the validity ratio $|\mathcal{A}^+|/(|\mathcal{A}^+| + |\mathcal{A}^-|)$ and the size of each axiom set influences the results, we created several distinct annotation sets with different validity ratios.

For each set, we applied our methodology starting from the revision state $(\mathcal{O} \cup \mathcal{O}^- \cup \mathcal{A}^+ \cup \mathcal{A}^-, \mathcal{O}, \mathcal{O}^-)$ with $\mathcal{O}$ containing the axioms of the NanOn ontology and

	validity ratio	*optimal*	$norm$	*best unparametrized*	*random*
L_1	90%	65.6%	65.4%	(*impact*$^+$) 65.4%	41.7%
L_2	76%	59.8%	59.8%	(*impact*$^+$) 55.8%	35.8%
L_3	50%	47.8%	47.6%	(*guaranteed*) 36.5%	24.4%
L_4	25%	59.9%	59.8%	(*impact*$^-$) 54.9%	37.6%
L_5	10%	63.9%	63.9%	(*impact*$^-$) 63.9%	40.3%

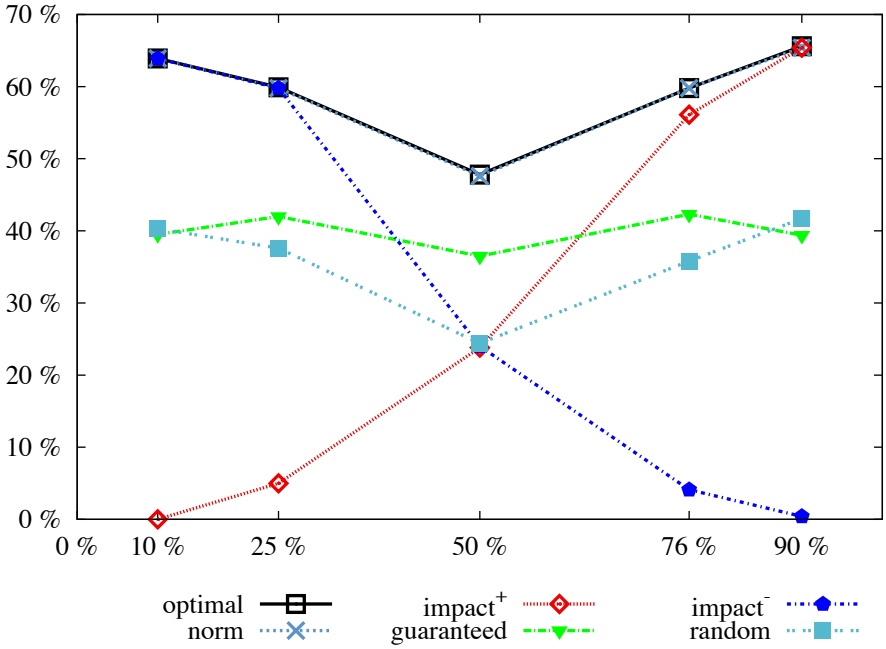


Figure 4.4: *Revision results of* $norm$ *in comparison with other ranking functions for the sets* L_1-L_5

with $\mathcal{O}^-$ containing axioms expressing inconsistency and class unsatisfiability. We obtained a complete revision state $(\mathcal{O} \cup \mathcal{O}^- \cup \mathcal{A}^+ \cup \mathcal{A}^-, \mathcal{O} \cup \mathcal{A}^+, \mathcal{O}^- \cup \mathcal{A}^-)$ where on-the-fly expert decisions about approval or decline were simulated according to the membership in $\mathcal{A}^+$ or $\mathcal{A}^-$. For computing the entailments, we used the OWL reasoner HermiT.[2]

[2] `http://www.hermit-reasoner.com`

For each set, our baseline is the reduction of expert decisions when axioms are evaluated in random order, i.e., no ranking is applied and only the revision closure is used to automatically evaluate axioms. The upper bound for the in principle possible reduction of expert decisions is called the *optimal* ranking, obtained by applying the "impact oracle" for each axiom α that is to be evaluated:

$$\mathit{KnownImpact}(\alpha) = \begin{cases} \mathit{impact}^{+}(\alpha) & \text{if } \alpha \in \mathcal{A}^{+}, \\ \mathit{impact}^{-}(\alpha) & \text{if } \alpha \in \mathcal{A}^{-}. \end{cases}$$

4.4.1 Axiom Impacts versus Parametrized Ranking

To compare the effectiveness of *impact*$^{+}$, *impact*$^{-}$, and *guaranteed* with the parametrized ranking $norm$, we created five sets of annotations L_1 to L_5, each comprising 5,000 axioms with validity ratios varying from 90% to 10%.

The table in Fig. 4.4 shows the results for the different ranking techniques: the column *optimal* shows the upper bound achieved by using the impact oracle, $norm$ shows the results for $norm$ parametrized with the actual validity ratio, *best unparametrized* shows the best possible value achievable with the unparametrized functions, and, finally, the column *random* states the effort reduction already achieved by presenting the axioms in random order. The results show that $norm$ consistently achieves almost the maximum effort reduction with an average difference of 0.1%. The unparametrized functions only work well for the high and low quality datasets, as expected, where *impact*$^{+}$ works well for the former case, while *impact*$^{-}$ works well for the latter. For the dataset with validity ratio 50%, $norm$ achieves an additional 11.1% of automation by using the parametrized ranking. Note that $norm$ does not necessarily achieve the optimum obtained when using the oracle, since the validity ratio within connected decision space components does not necessarily correspond to the average validity ratio.

4.4.2 Effects of Learned Validity Ratio

In order to evaluate our solution for situations where the validity ratio is unknown or only very rough estimates can be given upfront, we further analyze the effec-

	validity ratio	*optimal*	$norm$	$dynnorm_{0.50}$	$dynnorm_{1.00}$	$dynnorm_{0.00}$	*random*	*deviation from* $norm$
S_1	90%	72.4%	72.4%	58.6%	72.4%	65.5%	40.8%	-6.9%
S_2	77%	68.6%	65.7%	57.1%	62.9%	48.6%	38.2%	-10.5%
S_3	48%	65.1%	65.1%	65.1%	60.3%	61.9%	22.0%	-2.7%
S_4	25%	68.3%	68.3%	64.6%	63.4%	67.1%	37.6%	-3.3%
S_5	10%	72.5%	72.5%	71.6%	67.6%	72.5%	29.2%	-1.9%
M_1	91%	66.4%	66.0%	66.2%	66.4%	65.6%	40.8%	+0.1%
M_2	77%	60.0%	60.0%	59.6%	59.8%	59.2%	38.2%	-0.5%
M_3	44%	40.8%	40.6%	40.4%	40.6%	40.4%	22.0%	-0.1%
M_4	25%	60.0%	60.0%	59.6%	59.2%	59.8%	37.6%	-0.5%
M_5	10%	53.2%	53.0%	52.8%	52.8%	53.2%	29.2%	-0.1%
L_1	90%	65.6%	65.4%	65.4%	65.4%	65.3%	41.7%	0.0%
L_2	76%	59.8%	59.8%	59.8%	59.8%	59.9%	35.8%	0.0%
L_3	50%	47.8%	47.6%	47.4%	47.2%	47.3%	24.4%	-0.3%
L_4	25%	59.9%	59.8%	59.8%	59.8%	59.8%	37.6%	0.0%
L_5	10%	63.9%	63.9%	63.9%	63.8%	63.9%	40.3%	0.0%

Table 4.4: *Revision results for datasets S_1 to S_5, M_1 to M_5, and L_1 to L_5*

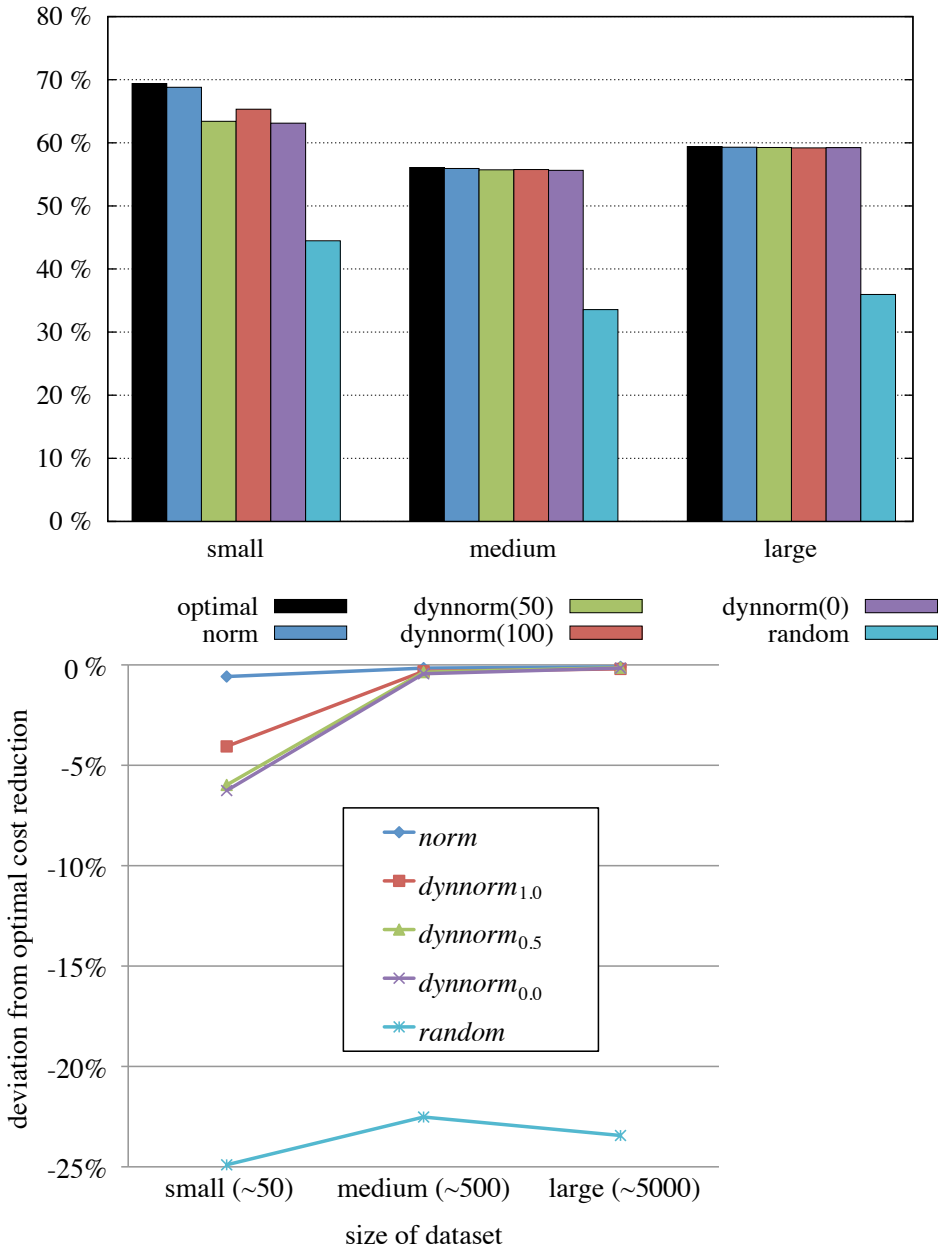


Figure 4.5: *Effect of learning validity ratio for different data set sizes.*

tiveness of the dynamically learning ranking function $dynnorm$. To this end, we created the following annotation sets in addition to the datasets $L_1 - L_5$:

- small datasets S_1 to S_5 with the size constantly growing from 29 to 102 axioms and validity ratios varying from 90% to 10%.
- medium-sized datasets M_1 to M_5 with 500 axioms each and validity ratios varying from 91% to 10%.

Table 4.4 shows the results of the revision: the columns *optimal* and *random* are as described above, the column $norm$ shows the results that we would obtain if we were to assume that the validity ratio is known and given as parameter to the $norm$ ranking function, the columns $dynnorm_{0.50}$, $dynnorm_{1.00}$ and $dynnorm_{0.00}$ show the results for starting the revision with a validity ratio of 50%, 100%, and 0%, respectively, where over the course of the revision, we update the validity ratio estimate. The last column shows the average deviation of the manual effort reduction achieved using $dynnorm$ from those achieved using $norm$. Fig. 4.5 shows the average values for small, medium size and large datasets.
We observe, that, in case of small datasets (S_i), the deviation from $norm$ (on average 5.1%, computed from Table 4.4) as well as the dependency of the results on the initial value of the validity ratio are clearly visible. However, the results of $dynnorm$ are notably better (by around 20.0%, Fig. 4.5) than those of a revision in random order. It is also interesting to observe that the average deviation from $norm$ decreases with the size of a dataset and that the deviation is lower for datasets with an extreme validity ratio (close to 100% or 0%).
For medium-sized and large datasets (M_i and L_i), the deviation from $norm$ (on average 0.3% for both) as well as the dependency on the initial value of the validity ratio are notably lower.
We conclude that

- ranking based on learning validity ratio is already useful for small datasets (30-100 axioms), and improves notably with the growing size of the dataset under revision;
- in case of large datasets, the performance difference between the results with a validity ratio known in advance and a learned validity ratio almost disap-

pears, thereby making the assumption of known average validity ratio obsolete for axiom ranking.

4.4.3 Computational Effort

During our experiments, we measured the average number of seconds after each expert decision required for the automatic evaluation and ranking as well as the average number of reasoning calls. If we compute the average values for the revision based on $dynnorm$ ranking for all 15 datasets, the revision took 0.84 seconds (7.4 reasoning calls) after each expert decision. In the case of small datasets, partitioning yields an additional improvement by an order of magnitude in terms of reasoning calls. For medium-sized datasets without partitioning, the first step out of on average 153 evaluation steps took already 101,101 reasoning calls (ca. 3 hours) even when using decision spaces. Without decision spaces and partitioning, the required number of reasoning calls for the revision of the sets M_1 to M_5 would be more than 500,000, judging by the required reasoning calls to build the corresponding decision space in the worst case. For large datasets, the revision without the two optimizations would even require more than 50 million reasoning calls in the worst case. For this reason, we did not try to run the experiment without partitioning for large datasets. In contrast to that, the average number of required reasoning calls for a complete revision of the sets M_1 to M_5 with partitioning amounts to 3,380. The revision of datasets L_1 to L_5 with partitioning required overall on average 16,175 reasoning calls, which corresponds to between 6 and 7 reasoning calls per evaluation decision. We can summarize the evaluation results as follows:

- The proposed reasoning-based support performs well in an interactive revision process with on average 0.84 seconds per expert decision.
- In particular in case of large datasets containing several partitions, partitioning notably reduces the computational effort.
- Partitioning reduces the number of reasoner call in case of small datasets by an order of magnitude.
- The employment of decision spaces saves in our experiments on average 75% of reasoner calls. As measured in the case of small datasets, partitioning

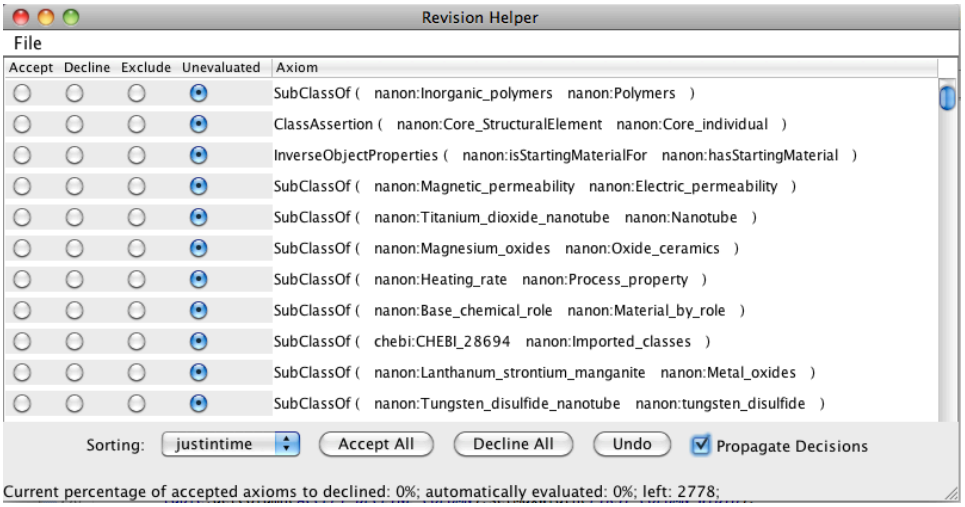


Figure 4.6: *Revision Helper GUI*

further intensifies the effect of decision spaces and we save even 80% of reasoner calls.

4.5 User Front-End

Figure 4.6 shows the user front-end of the *revision helper* tool. It allows the user to load the set $\mathcal{O}$ of axioms under revision and save or load an evaluation state for the currently loaded set $\mathcal{O}$. Thereby, the user can interrupt the revision at any time and proceed later on. If partitioning is activated, revision helper shows the partitions one after another and the revision of each partition is independent from the revision of all other partitions.

By default, revision helper initializes the set $\mathcal{O}^{\models}$ of undesired statements with the minimal set of statements expressing the inconsistency of the ontology and unsatisfiability of its classes. The set of desired statements $\mathcal{O}^{\models}$ can be initialized by loading an arbitrary ontology. A statement can be evaluated by choosing one of the values *Accept* and *Decline*, and it can be excluded from the revision process by choosing *Exclude*. The latter option should be used, if the meaning of a statement is not clear and the user cannot decide whether to accept or to decline it. After

the statement has been evaluated, it is removed from the revision list as well as all statements that could be evaluated automatically, unless the checkbox *Propagate Decisions* is deactivated. The ranking strategy used for sorting the statements can be selected or deactivated at any time and is taken into account after the next evaluation decision. At any stage of the revision, it is possible to export the current set $\mathcal{O}^{\models}$ of accepted statements as an ontology. For the export, we exclude, however, axioms with which $\mathcal{O}^{\models}$ has been initialized at the beginning of the revision.

4.6 Related Work

As discussed in Section 3.1.4, we are aware of three approaches [MEILICKE et al. 2008, JIMÉNEZ-RUIZ et al. 2009a, JIMÉNEZ-RUIZ et al. 2009b] that aim at supporting manual inspection of ontologies. The approach by Meilicke et al. is closely related to this thesis, since it pursues the same goal: a reduction of manual effort required for the accuracy-based revision of ontological data. In the following, we discuss the above approach in detail. For the discussion of the other two, we refer the reader to Section 3.1.4.

Meilicke et al. support a revision of ontology mappings, the expressivity of which is limited to subsumption and equivalence between atomic concepts. Implications of expert decisions about the accuracy of mapping axioms represented as bridge rules [SERAFINI and TAMILIN 2005] are propagated based on incoherence and entailment: any bridge rule entailed by the set of approved bridge rules together with the two ontologies is automatically approved and each bridge rule that would make the set of approved bridge rules together with the two ontologies incoherent is automatically marked as incorrect. In contrast to that, our approach is generic, on the one hand, in the sense that it is applicable to ontologies expressed using any formalism satisfying the properties presented in Section 2.5. On the other hand, the set of unwanted consequences in our approach can be initialized with arbitrary restrictions, including inconsistency, incoherency or any user-defined axioms not causing the initial revision state to become inconsistent. In this way, given the high-level assumption that axioms marked as incorrect should not be logical consequences of the approved axioms, the approach achieves maximum automation possible using

reasoning. Thus, the approach by Meilicke et al. is a special case compared to the one presented in this thesis in terms of both, expressivity of axioms under revision and cases in which consequences of expert decisions are propagated.

4.7 Summary

In this chapter, we proposed a methodology for supporting ontology revision with respect to semantic accuracy. We introduced the notions of revision states, revision state consistency and revision closure, based on which the revision of ontologies can be partially automated.

Further, we showed that, even though a decent effort reduction can already be achieved when axioms are chosen randomly for each expert decision, an evaluation of the axioms in an appropriate order usually yields a higher effort reduction. In order to ensure a decent effectiveness of the proposed reasoning-based support, we investigated different ways of determining a beneficial evaluation order for axioms. To this end, we introduced the notion of axiom impact, which can be used to define simple axiom ranking functions performing well for data with either a very high or a very low average accuracy. In order to achieve higher effectiveness for data with an arbitrary average accuracy, we further refined the ranking functions to take into account the estimated average accuracy of the ontology under revision. We then showed how the average accuracy can be learned on-the-fly over the course of the revision, which deliberates the user from having to provide such an estimate.

Moreover, to account for computational efficiency, we provided an efficient and elegant way of determining the revision closure and axiom impacts by computing and updating structures called *decision spaces* which saved 75% of reasoner calls during our evaluation. Moreover, we introduced a simple partitioning approach, which reduced the number of reasoning call in our evaluation by an order of magnitude.

We evaluated an implementation of the discussed approach in a revision of ontology-based annotations of scientific publications comprising over 25,000 statements with the following results:

- On average, we were able to reduce the number of required evaluation decisions by 36% when the statements were reviewed in an arbitrary order, and by 55.4% when the unparametrized ranking techniques were used. The parametrized ranking technique almost achieved the maximum possible automation (59.4% of evaluation decisions) thereby reducing the manual effort of revision by 59.3%[3]. The gain of the parametrized compared to the unparametrized ranking functions is particularly important for datasets with a validity ratio close to 50% (we observed an improvement of 11.1% in case of L_3, see the table in Fig. 4.4), since for those datasets the potential of automation cannot be fully exploited by the means of simple ranking functions.

- In case of large datasets with an unknown validity ratio, learning the validity ratio is particularly effective due to the law of large numbers. In our experiments, the proportion of automatically evaluated statements is nearly the same as in case where the validity ratio is known *a priori* and is used as a fixed parameter of $norm$, thereby making the assumption of known average validity ratio not necessary for axiom ranking.

- If a dataset allows for an efficient partitioning and an effective application of decision spaces, the proposed reasoning-based support is feasible for an interactive revision process even in case of a large (5,000) number of unevaluated axioms. In our evaluation, reasoning-based support took on average less than one second after each expert decision.

[3]Both values have been computed from the results in Table 4.4 by taking the average for all 15 datasets.

CHAPTER 5

Relevance-Based Revision

Since the size of a terminology has a crucial impact on the maintenance cost and often on the performance of reasoning, it is important to keep it as compact as possible. The aim of a relevance-based revision is to reduce the amount of irrelevant information imported from external sources, and, at the same time, preserve all relevant consequences. In many cases, it is not feasible to decide about the relevance of information on the level of axioms. On the one hand, a single axiom can be partially relevant due to an occurrence of relevant and irrelevant entities within a single axiom, and, on the other hand, an elimination of axioms containing only irrelevant entities can change the meaning of the relevant entities. Example 5 demonstrates the effect of such an elimination.

Example 5. *Consider the following terminology $\mathcal{T}$:*

$$A_2 \sqsubseteq A_1$$
$$A_3 \sqsubseteq A_2$$
$$A_4 \sqsubseteq A_3$$
$$A_5 \sqsubseteq A_4$$
$$A_6 \sqsubseteq A_5$$
$$A_7 \sqsubseteq A_6$$
$$A_8 \sqsubseteq A_7$$
$$A_9 \sqsubseteq A_8$$
$$A_9 \sqsubseteq \exists r.A_9$$
$$A_{10} \sqsubseteq A_9 \qquad A_{13} \sqsubseteq A_9$$
$$A_{11} \sqsubseteq A_{10} \qquad A_{14} \sqsubseteq A_{13}$$
$$A_{12} \sqsubseteq A_{11} \qquad A_{15} \sqsubseteq A_{14}$$
$$A_{10} \sqcap A_{13} \sqsubseteq A_{16}$$

If we are only interested in entities A_1, A_{11}, r, then we might consider to eliminate all axioms except for those that contain at least one relevant entity, namely $A_2 \sqsubseteq A_1, A_{11} \sqsubseteq A_{10}, A_{12} \sqsubseteq A_{11}$ and $A_9 \sqsubseteq \exists r.A_9$. However, in this way we would lose the information about the connection between the relevant entities, for instance $A_{11} \sqsubseteq A_1, A_{11} \sqsubseteq \exists r.A_1, A_{11} \sqsubseteq \exists r.\exists r.A_1, ...$ Indeed, the above reduced ontology does not imply any of these statements. Thus, by omitting axioms based only on the absence of relevant entities can lead to a loss of relevant information. For the above given reasons, we assume that, within the relevance-based revision of an ontology, an expert first decides, which ontology entities are relevant within the particular application context. This yields us a set Σ of relevant entities. The subsequent task is to compute a corresponding terminology that contains as little irrelevant information as possible, and, at the same time, contains all information about the relevant entities. We refer to the task of computing such a terminology as *general module extraction*. Since our requirement is to preserve the meaning of the

relevant entities, in the following we investigate logic-based approaches to general module extraction, i.e., approaches that guarantee a preservation of the semantics for the set of relevant entities.

We say that the semantics is preserved, if all logical consequences concerning only the relevant entities are preserved. The logical foundation for such a preservation of relevant consequences is given by the established notion of *inseparability*. Two terminologies, $\mathcal{T}_1$ and $\mathcal{T}_2$, are inseparable with respect to a signature Σ if they have the same Σ-consequences, i.e., consequences whose signature is a subset of Σ. Depending on the particular application requirements, the expressivity of those Σ consequences can vary from subsumption queries and instance queries to conjunctive queries or even second-order logic queries. Since the above problem has not been solved yet for general terminologies in the lightweight logic $\mathcal{EL}$ and Σ-consequences being subsumption queries, in this chapter we consider the following concrete notion of inseparability, called *concept-inseparability* [KONTCHAKOV et al. 2010, KONEV et al. 2009b, LUTZ et al. 2012]:

Definition 7. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two general $\mathcal{EL}$ terminologies and Σ a signature. $\mathcal{T}_1$ and $\mathcal{T}_2$ are concept-inseparable with respect to Σ, in symbols $\mathcal{T}_1 \equiv^c_\Sigma \mathcal{T}_2$, if for all $\mathcal{EL}$ concepts C, D with $sig(C) \cup sig(D) \subseteq \Sigma$ holds $\mathcal{T}_1 \models C \sqsubseteq D$, iff $\mathcal{T}_2 \models C \sqsubseteq D$.*

Given a signature Σ and a terminology $\mathcal{T}$, the task of terminology extraction in general is to compute a terminology $\mathcal{T}'$, which is entailed by $\mathcal{T}$ and is concept-inseparable from it. We call the result $\mathcal{T}'$ a *general module* of $\mathcal{T}$.

Definition 8. *Let $\mathcal{T}$ be an $\mathcal{EL}$ terminology and Σ a signature. An $\mathcal{EL}$ terminology $\mathcal{T}'$ is a* general module *of $\mathcal{T}$ with respect to Σ, written $\mathcal{T}' \in \texttt{MOD}(\mathcal{T}, \Sigma)$, iff (1) $\mathcal{T} \equiv^c_\Sigma \mathcal{T}'$ and (2) $\mathcal{T} \models \mathcal{T}'$.*

Due to its usefulness for different ontology engineering tasks, the task of general module extraction has been investigated by different authors in the last decade. Among others, approaches arose that compute a subset of the original ontology entailing all relevant consequences (classical module extraction), e.g., [KONTCHAKOV et al. 2010, GRAU et al. 2007b]. While minimal module extraction computes modules by gradually removing axioms from the ontology and checking concept-inseparability with respect to the given signature Σ and is inherently EXPTIME-hard for $\mathcal{EL}$, the so-called *locality-based extractor* is a tractable

alternative, which guarantees a preservation of all relevant consequences, but does not aim at computing a minimal solution. For the signature $\Sigma = \{A_1, A_{11}, r\}$ given in Example 5, minimal module extraction would return the first nine axioms and $A_{10} \sqsubseteq A_9, A_{11} \sqsubseteq A_{10}$ entailing all Σ-consequences of $\mathcal{T}$.

While module extraction guarantees the preservation of all relevant consequences, it is not difficult to see that, by introducing a shortcut axiom $A_9 \sqsubseteq A_1$, we would obtain a much smaller representation of Σ consequences not referring to $A_2 - A_8$. In the same way, we can also introduce a shortcut $A_{11} \sqsubseteq A_9$ for $A_{10} \sqsubseteq A_9, A_{11} \sqsubseteq A_{10}$ and obtain the small general module $\{A_{11} \sqsubseteq A_9, A_9 \sqsubseteq \exists r.A_9, A_9 \sqsubseteq A_1\}$, which only uses A_9 in addition to entities from Σ. Thus, by rewriting a terminology, i.e., exchanging explicitly given axioms by other axioms from the deductive closure, we can significantly improve general modules with respect to the objective of reducing the size of the ontology and the proportion of irrelevant information.

In the next section, we discuss a rewriting technique for general $\mathcal{EL}$ terminologies based on the above notion of concept-inseparability. Based on this rewriting technique, in the subsequent two sections, we propose two strategies for general module extraction. In Section 5.2, we solve the well-known problem of uniform interpolation (and forgetting) and show the corresponding, triple-exponential bound on the size of uniform interpolants. In Section 5.3, we show that neither classical module extraction, nor uniform interpolation yield optimal results for the task of general module extraction in a common application scenario. We revise the requirements for general module extraction and propose a new problem specification that, in our view, is more suitable for the average case. We then show how the problem can be solved by combining rewriting and classical module extraction in ExpTime, and propose a tractable approach, which does not guarantee the minimality of the extracted general modules, but yields on average modules with half as many axioms as the corresponding classical modules.

5.1 Rewriting based on Primitivization

As demonstrated in the last section, exchanging explicitly given axioms by other axioms from the deductive closure allows for more flexibility when choosing a

general module with the objective of reducing the size of the ontology and the proportion of irrelevant information. In this section, we consider the task of rewriting based on concept-inseparability for the lightweight description logic $\mathcal{EL}$.
Since rewriting works on the syntactic structure of terminologies, the task can be significantly simplified by simplifying the syntactic structure of the TBox before the rewriting. To this end, we make use of *primitivization*. Primitivization introduces fresh atomic concepts representing sub-expressions occurring in the TBox. As a result, we obtain a syntactically simple TBox giving for each atomic concept a set of its unnested subsumees and subsumers explicitly given in $\mathcal{T}$. In the following, we show that, by applying primitivization, we can simplify the tracking of subsumption dependencies between general concepts over the course of rewriting. For this purpose, we first give a deduction calculus that is sound and complete for general subsumption in $\mathcal{EL}$. Subsequently, we prove particular properties of "primitivized" TBoxes that allow for a simplified consequence-preserving rewriting.

5.1.1 Gentzen-Style Proof System for $\mathcal{EL}$

In the following, we will use the Gentzen-style calculus for $\mathcal{EL}$ shown in Fig. 5.1.

$$\frac{}{C \sqsubseteq C}(\textsc{Ax}) \quad \frac{}{C \sqsubseteq \top}(\textsc{AxTop})$$

$$\frac{D \sqsubseteq E}{C \sqcap D \sqsubseteq E}(\textsc{AndL})$$

$$\frac{C \sqsubseteq E \quad C \sqsubseteq D}{C \sqsubseteq D \sqcap E}(\textsc{AndR})$$

$$\frac{C \sqsubseteq D}{\exists r.C \sqsubseteq \exists r.D}(\textsc{Ex})$$

$$\frac{C \sqsubseteq E \quad E \sqsubseteq D}{C \sqsubseteq D}(\textsc{Cut})$$

Figure 5.1: *Gentzen-style proof system for general $\mathcal{EL}$ terminologies with C, D, E arbitrary concept expressions.*

We show that the above calculus is sound and complete for subsumptions between arbitrary $\mathcal{EL}$ concepts.

Lemma 11 (Soundness and Completeness)**.** *Let $\mathcal{T}$ be an arbitrary $\mathcal{EL}$ TBox, C, D $\mathcal{EL}$ concepts. Then $\mathcal{T} \models C \sqsubseteq D$, iff $\mathcal{T} \vdash C \sqsubseteq D$.*

Proof. While the soundness of the proof system (if-direction) can be easily checked for each rule separately, the proof of completeness is more sophisticated. In order to show the only-if-direction of the lemma, we construct a model $\mathcal{I}$ for $\mathcal{T}$ wherein *only* the GCIs derivable from $\mathcal{T}$ are valid. This model is constructed as follows:

- $\Delta^{\mathcal{I}}$ contains an element δ_C for every $\mathcal{EL}$ concept expression C
- $A^{\mathcal{I}} := \{\delta_C \in \Delta^{\mathcal{I}} | \mathcal{T} \vdash C \sqsubseteq A, \}$
- $r^{\mathcal{I}} := \{(\delta_C, \delta_D) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} | \mathcal{T} \vdash C \sqsubseteq \exists r.D, r \in \text{sig}_R(\mathcal{T})\}$

We will show that the following claim holds for $\mathcal{I}$:
For all $\delta_E \in \Delta^{\mathcal{I}}$ and $\mathcal{EL}$ concepts F holds $\delta_E \in F^{\mathcal{I}}$ iff $\mathcal{T} \vdash E \sqsubseteq F$. *(*)*

This claim can be exploited in two ways: First, we use it to show that $\mathcal{I}$ is indeed a model of $\mathcal{T}$. Let $C \sqsubseteq D \in \mathcal{T}$ and consider an arbitrary concept expression G with $\delta_G \in C^{\mathcal{I}}$. Via (*) we obtain $\mathcal{T} \vdash G \sqsubseteq C$. Further, $\mathcal{T} \vdash C \sqsubseteq D$ due to $C \sqsubseteq D \in \mathcal{T}$. Thus we can derive $\mathcal{T} \vdash G \sqsubseteq D$ via (CUT) and consequently, applying (*) again, we obtain $\delta_G \in D^{\mathcal{I}}$. Thereby modelhood of $\mathcal{I}$ with respect to $\mathcal{T}$ has been proved.
Second, we use (*) to show that $\mathcal{I}$ is a counter-model for all GCIs not derivable from $\mathcal{T}$ as follows: Assume $\mathcal{I} \models C \sqsubseteq D$ but $\mathcal{T} \nvdash C \sqsubseteq D$. From $\mathcal{T} \vdash C \sqsubseteq C$ and (*) we derive $\delta_C \in C^{\mathcal{I}}$. From $\mathcal{T} \nvdash C \sqsubseteq D$ and (*) we obtain $\delta_C \notin D^{\mathcal{I}}$. Hence we get $C^{\mathcal{I}} \not\subseteq D^{\mathcal{I}}$ and therefore $\mathcal{I} \not\models C \sqsubseteq D$, a contradiction.
It remains to prove (*). This is done by an induction on the maximal nesting depth of the operators $\sqcap$ and $\exists$. There are two base cases:

- for $F = \top$, the claim trivially follows from (AXTOP),
- for $F \in \text{sig}_C(\mathcal{T})$, it is a direct consequence of the definition.

we now consider the cases where F is a complex concept expression

- for $F = C_1 \sqcap \ldots \sqcap C_n$, we note that $\delta_E \in F^{\mathcal{I}}$ exactly if $\delta_E \in C_i^{\mathcal{I}}$ for all $i \in \{1 \ldots n\}$. By induction hypothesis, this means $\mathcal{T} \vdash E \sqsubseteq C_i$ for all $i \in \{1 \ldots n\}$. Finally, observe that $\{E \sqsubseteq C_i \mid 1 \leq i \leq n\}$ and $E \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$ can be mutually derived from each other: (for "$\vdash$" this is a straightforward consequence of (ANDR), for "$\dashv$" note that we can derive $\emptyset \overset{\text{AX}}{\vdash} C_i \sqsubseteq C_i \overset{\text{ANDL}^*}{\vdash} C_1 \sqcap \ldots \sqcap C_n \sqsubseteq C_i$ whence together with $E \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$ follows $E \sqsubseteq C_i$ by (CUT).

- for $F = \exists r.G$, we prove the two directions separately. First assuming $\delta_E \in F^{\mathcal{I}}$ we must find $(\delta_E, \delta_H) \in r^{\mathcal{I}}$ for some H with $\delta_H \in G^{\mathcal{I}}$. This implies both $\mathcal{T} \vdash E \sqsubseteq \exists r.H$ (by definition) and $\mathcal{T} \vdash H \sqsubseteq G$ (via the induction hypothesis). From the latter, we can deduce $\mathcal{T} \vdash \exists r.H \sqsubseteq \exists r.G$ by (EX) and consequently $\mathcal{T} \vdash E \sqsubseteq \exists r.G$. For the other direction, note that by definition, $\mathcal{T} \vdash E \sqsubseteq \exists r.G$ implies $(\delta_E, \delta_G) \in r^{\mathcal{I}}$. On the other hand, we get $\mathcal{T} \vdash G \sqsubseteq G$ by (AX) and therefore $\delta_G \in G^{\mathcal{I}}$ by the induction hypothesis which yields us $\delta_E \in F^{\mathcal{I}}$. □

5.1.2 Subsumee/Subsumer Relation Pairs

In order to obtain a general module of a TBox $\mathcal{T}$, during the rewriting we aim at preserving the part of the deductive closure of $\mathcal{T}$ consisting of Σ-consequences. Since rewriting operates on the syntactic structure of $\mathcal{T}$, it is desirable that the syntactic structure has a close relation to the deductive closure of $\mathcal{T}$ thereby enabling targeted changes of the closure via changes of the syntactic structure. Interestingly, we can transform each $\mathcal{EL}$ TBox into sets of subsumees and sets of subsumers of atomic concepts in such a way that the deductive closure of the TBox is determined by the deductive closures of these subsumees and subsumers of atomic concepts. In this way, we can reduce preservation of the Σ-closure of $\mathcal{T}$ to preservation of the corresponding Σ-closures of subsumees and subsumers. The latter problem is simpler, since the derivation of subsumees and subsumers can in turn be reduced to substitution of atomic concepts by a subset of their subsumees and subsumers, respectively, with the maximal role depth 1. Given the above reduction, we can eliminate exactly the axioms referencing a particular concept from the closure of $\mathcal{T}$ by substituting it within all explicitly given subsumees and subsumers. Thus,

given such a decomposition into subsumees and subsumers of atomic concepts allows for controlled changes of the closure by the means of substitutions.

In what follows, we call any pair of binary relations $\langle R_{\sqsupseteq}^{\mathcal{T}}, R_{\sqsubseteq}^{\mathcal{T}} \rangle$ on concept expressions that respectively relate each atomic concept $B \in \text{sig}_C(\mathcal{T})$ to a subset of subsumees and a subset of subsumers of B entailed by $\mathcal{T}$ a *subsumee/subsumer relation pair for* $\mathcal{T}$. If $\mathcal{T}$ is clear from the context, we simply write $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$. In order to define a deductive closure for a relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$, we construct a TBox $UI(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma)$ that represents the relations between subsumees and subsumers and the corresponding atomic concepts given in $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$ as axioms, e.g., $A \sqsubseteq D$ for $(A, D) \in R_{\sqsubseteq}$. The construction already takes into account that we are only interested in Σ-subsumptions and excludes atomic concepts not from Σ, forming axioms from their subsumees and subsumers directly, e.g., $C \sqsubseteq D$ for $(A, C) \in R_{\sqsupseteq}, (A, D) \in R_{\sqsubseteq}$ instead of $C \sqsubseteq A$ and $A \sqsubseteq D$. This is in particular useful, if the excluded atomic concept A is not referenced by any of the subsumees and subsumers in $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$, since, in this case, the resulting TBox does not reference A. In this way, irrelevant concepts are eliminated from the terminology. As we will show later on, rewriting ideally eliminates references to atomic concepts not from Σ from a relation pair, thereby providing the necessary input for $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma)$.

Definition 9. *Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox and Σ a signature. Further, let $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$ be a subsumee/subsumer relation pair for $\mathcal{T}$. Then,*

$$\begin{aligned}\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma) = &\{C \sqsubseteq A \mid A \in \Sigma, (A, C) \in R_{\sqsupseteq}\} \cup \\ &\{A \sqsubseteq D \mid A \in \Sigma, (A, D) \in R_{\sqsubseteq}\} \cup \\ &\{C \sqsubseteq D \mid \textit{there exists } A \notin \Sigma, (A, C) \in R_{\sqsupseteq}, (A, D) \in R_{\sqsubseteq}\}.\end{aligned}$$

The deductive closure of a set of subsumees (subsumers) of an atomic concept A in a subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$ is the set of all subsumees (subsumers) of A entailed by $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma)$. In the following, we will be in particular interested in the Σ-subset of the corresponding closure.

In the context of general module extraction, subsumee/subsumer relation pairs for $\mathcal{T}$ are required to preserve the Σ-closure of $\mathcal{T}$. To distinguish such relation pairs, we define the property of completeness as follows.

Definition 10. *Let $\mathcal{T}$ be a TBox, $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ a subsumee/subsumer relation pair and Σ a signature. We denote by $\Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$ the extension of Σ with atomic concepts occurring in the range of $R_{\sqsupseteq}$ and $R_{\sqsubseteq}$. $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ is* complete *with respect to Σ, if $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})) \in \mathtt{MOD}(\mathcal{T}, \Sigma)$.*

5.1.3 Primitivization

In order to decompose a TBox into corresponding sets of subsumees and subsumers meeting the above requirement of completeness, we use the so-called *primitivization*: we assign a temporary concept name to each non-atomic sub-expression occurring in $\mathcal{T}$, such that the terminology can be represented without nested expressions, i.e., using only axioms of the form $A \sqsubseteq B$, $A \equiv B_1 \sqcap \ldots \sqcap B_n$, and $A \equiv \exists r.B$, where A and $B_{(i)}$ are atomic concepts or $\top$ and $r \in \mathrm{sig}_R(\mathcal{T})$. For this purpose, we introduce a minimal required set of fresh concept symbols N_D and the corresponding definition axioms $\{A' \equiv C' \mid A' \in N_D\}$ for each $A' \in N_D$ and the corresponding concept C' replaced by A'. This can be realized in time linear in the size of $\mathcal{T}$ by recursively replacing complex concepts $C_{(i)}$ in expressions $C_1 \sqcap \ldots \sqcap C_n$ and $\exists r.C$ by fresh concept symbols with the corresponding equivalence axioms. Note that the original form of the terminology can easily be obtained by replacing the temporary concept names by their definitions. For instance, in case of the terminology $\mathcal{T}$ from Example 5, primitivization is done by introducing two temporary concepts, $B_1 \equiv A_{10} \sqcap A_{13}$ and $B_2 \equiv \exists r.A_9$. By classifying the obtained terminology $\mathcal{T}'$, which we call *normalized*, we can identify all subsumptions between sub-expressions occurring in the original TBox $\mathcal{T}$.

In what follows, we assume that terminologies are normalized and refer to $\mathrm{sig}_C(\mathcal{T}) \cup N_D$ as $\mathrm{sig}_C(\mathcal{T})$. Since concept symbols in N_D are fresh, they do not appear in Σ. W.l.o.g., in what follows we assume that $\mathcal{EL}$ concepts do not contain any equivalent concepts in conjunctions and that equivalent concept symbols have been replaced by a single representative of the corresponding equivalence class.

The following lemma postulates the close semantic relation between a TBox and its normalization.

Lemma 12. *Any $\mathcal{EL}$ TBox $\mathcal{T}$ can be extended into a normalized TBox $\mathcal{T}'$ such that each model of $\mathcal{T}'$ is a model of $\mathcal{T}$ and each model of $\mathcal{T}$ can be extended into a model of $\mathcal{T}'$.*

Proof. All concepts in N_D are defined, i.e., their meaning is uniquely determined by the meaning of subconcepts (concepts that occur in $\mathcal{T}$) of the original TBox $\mathcal{T}$. □

5.1.4 Basic Transformations on Subsumee/Subsumer Relation Pairs

Since a normalized TBox consists only of axioms stating for each atomic concept a set of its subsumees and subsumers of the simple form $B, \exists r.B$ and $B_1 \sqcap ... \sqcap B_n$, it can be easily transformed into a subsumee/subsumer relation pair for $\mathcal{T}$. If we additionally classify $\mathcal{T}$, we obtain a complete subsumee/subsumer relation pair for $\mathcal{T}$ as follows:

Definition 11. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ terminology extended with all implicit subsumptions between atomic concepts. The* initial subsumee/subsumer relation pair for $\mathcal{T}$ $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ *is defined as follows:*

1. $R^{\mathcal{T}}_{\sqsupseteq} = \{(B, C) \mid B \in sig_C(\mathcal{T}), C \sqsubseteq B \in \mathcal{T}$ *or* $B \equiv C \in \mathcal{T}\}$,
2. $R^{\mathcal{T}}_{\sqsubseteq} = \{(B, C) \mid B \in sig_C(\mathcal{T}), B \sqsubseteq C \in \mathcal{T}$ *or* $B \equiv C \in \mathcal{T}\}$.

Before showing the completeness of the initial subsumee/subsumer relation pair, we demonstrate that already the initial subsumee/subsumer relation pair allows us to obtain general modules that are notably smaller than classical minimal modules. For instance, given the signature $\Sigma = \{A_1, A_8, A_{12}, A_{15}, A_{16}, r\}$ in Example 5, classical module extraction applied to $\mathcal{T}$ directly would return $\mathcal{T}$ itself. Clearly, this result is not optimal in terms of size. In contrast to that, we can obtain a small general module consisting of sub-expressions of $\mathcal{T}$, namely $\{A_{12} \sqsubseteq A_{10}$, $A_{15} \sqsubseteq A_{13}$, $A_{10} \sqcap A_{13} \sqsubseteq A_{16}$, $A_{10} \sqsubseteq A_9$, $A_{13} \sqsubseteq A_9$, $A_9 \sqsubseteq \exists r.A_9$, $A_9 \sqsubseteq A_8$, $A_8 \sqsubseteq A_1\}$, from an initial subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ for the terminology $\mathcal{T}$ from Example 5 as follows.

	$R_{\sqsupseteq}$	$R_{\sqsubseteq}$
A_{16}	B_1	$\emptyset$
A_{15}	$\emptyset$	$A_1, \ldots, A_9, B_2, A_{13}, A_{14}$
A_{14}	A_{15}	$A_1, \ldots, A_9, B_2, A_{13}$
A_{13}	B_1, A_{14}, A_{15}	$A_1, \ldots, A_9, B_2$
A_{12}	$\emptyset$	$A_1, \ldots, A_{11}, B_2$
A_{11}	A_{12}	$A_1, \ldots, A_{10}, B_2$
A_{10}	$B_1, A_{11}, A_{12}, A_{15}$	$A_1, \ldots, A_9, B_2$
A_9	$B_1, A_{10}, \ldots, A_{15}$	$A_1, \ldots, A_8, B_2$
A_8	$B_1, A_9, \ldots, A_{15}$	$A_1, \ldots, A_7$
A_7	$B_1, A_8, \ldots, A_{15}$	$A_1, \ldots, A_6$
A_6	$B_1, A_7, \ldots, A_{15}$	$A_1, \ldots, A_5$
A_5	$B_1, A_6, \ldots, A_{15}$	$A_1, \ldots, A_4$
A_4	$B_1, A_5, \ldots, A_{15}$	$A_1, \ldots, A_3$
A_3	$B_1, A_4, \ldots, A_{15}$	A_1, A_2
A_2	$B_1, A_3, \ldots, A_{15}$	A_1
A_1	$B_1, A_2, \ldots, A_{15}$	$\emptyset$
B_2	$\exists r.A_9, A_9, \ldots, A_{15}, B_1$	$\exists r.A_9$
B_1	$A_{10} \sqcap A_{13}$	$A_{10} \sqcap A_{13}, A_1, \ldots, A_{10}, A_{13}, B_2, A_{16}$

Figure 5.2: *The initial subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ for Example 5.*

In order to compute the initial subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$, we first normalize $\mathcal{T}$ by introducing two temporary concepts, $B_1 \equiv A_{10} \sqcap A_{13}$ and $B_2 \equiv \exists r.A_9$. Subsequently, we classify the normalized terminology and obtain the relation pair shown in Fig. 5.2.

Now, we consider the according general module given by $\mathcal{T}_{\mathtt{M}} = \mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq}))$. The terminology $\mathcal{T}_{\mathtt{M}}$ contains, for each A_i with $i \in \{1, \ldots, 16\}$, all axioms of the form $C \sqsubseteq A_i$ with $(A_i, C) \in R_{\sqsupseteq}$ and $A_i \sqsubseteq C$ with $(A_i, C) \in R_{\sqsubseteq}$. This also holds for B_1 and B_2. It is not difficult to check that, after replacing B_1 by $A_{10} \sqcap A_{13}$ and B_2 by $\exists r.A_9$ in $\mathcal{T}_{\mathtt{M}}$, each axiom of the above given module is contained in it. In fact, it can be shown that, after replacing fresh concepts by their definitions, the general module constructed from the initial subsumee/subsumer relation pair contains all general modules consisting of sub-expressions of $\mathcal{T}$. Thus, by applying minimal module extraction to the obtained general module, we would identify a minimal general module consisting of sub-expressions of $\mathcal{T}$.

5.1.5 Showing Completeness of Relation Pairs

In order to show the completeness of the initial or some other subsumee/subsumer relation pair, we make use of the deduction calculus introduced in Section 5.1.1. In the following, we give two lemmas that show how, in normalized TBoxes, any subsumption between complex concepts can be traced back to axioms connecting an atomic concept with its subsumees and subsumers. We introduce the following auxiliary function $\texttt{Pre} : \text{sig}_C(\mathcal{T}) \rightarrow 2^{2^{\text{sig}_C(\mathcal{T})}}$, which allows us for any atomic concept A to refer to its subconcepts of the form $B_1 \sqcap ... \sqcap B_n$. For each such conjunction, the set of its conjuncts is an element of $\texttt{Pre}$.

Definition 12. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ TBox and $A \in sig_C(\mathcal{T})$. $\texttt{Pre}(A)$ is the smallest set with the following properties:*

- $\{A\} \in \texttt{Pre}(A)$.
- *For each $K \in \texttt{Pre}(A)$ and each $B \in K$, if there is $B \equiv B_1 \sqcap ... \sqcap B_n \in \mathcal{T}$, then also $(K \setminus \{B\}) \cup \{B_1, ..., B_n\} \in \texttt{Pre}(A)$.*
- *For each $K \in \texttt{Pre}(A)$ and each $B \in K$, if there is $\mathcal{T} \models B' \sqsubseteq B$, then also $(K \setminus \{B\}) \cup \{B'\} \in \texttt{Pre}(A)$.*

Concerning a subsumption between two complex concepts, we can show the following property.

Lemma 13. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ TBox and C, D two $\mathcal{EL}$ concepts with $sig(C) \cup sig(D) \subseteq sig(\mathcal{T})$ such that $\mathcal{T} \models C \sqsubseteq D$. For any $A \in sig_C(\mathcal{T})$. W.l.o.g., assume that*

$$C = \prod_{1 \leq j \leq n} A_j \sqcap \prod_{1 \leq k \leq m} \exists r_k.E_k$$

for $A_j \in sig_C(\mathcal{T})$ and $r_k \in sig_R(\mathcal{T})$, E_k $\mathcal{EL}$ concepts with $sig(E_k) \subseteq sig(\mathcal{T})$ for $1 \leq k \leq m$. Then, for all conjuncts D_i of D, the following is true: If $D_i \in sig_C(\mathcal{T})$, there is a set $M \in \texttt{Pre}(D_i)$ of $sig_C(\mathcal{T})$ concepts such that for each element B of M holds at least one of the conditions [A1]-[A2]:

(A1) There is an A_j in C such that $A_j = B$.

(A2) There are r_k, E_k and there exists $B' \in sig_C(\mathcal{T})$ such that $\mathcal{T} \models E_k \sqsubseteq B'$ and $B \equiv \exists r_k.B' \in \mathcal{T}$.

If $D_i = \exists r'.D'$ for $r' \in sig_R(\mathcal{T})$ and D' an $\mathcal{EL}$ concept , at least one of the conditions [A3]-[A4] holds:

(A3) There are r_k, E_k such that $r_k = r'$ and $\mathcal{T} \models E_k \sqsubseteq D'$.

(A4) There is $B \in sig_C(\mathcal{T})$ such that $\mathcal{T} \models B \sqsubseteq \exists r'.D'$ and $\mathcal{T} \models C \sqsubseteq B$ and for $C \sqsubseteq B$ at least one of the conditions [A1]-[A2] holds.

Proof. We consider all rules, that could have been the last rule applied in order to obtain the above sequent and show by induction on the length of the proof that, in each case, the lemma holds. Rules AXTOP, AX are the basecase, since each proof begins with one of them.

$(C \bowtie D \in \mathcal{T})$ In the case that $C \sqsubseteq D \in \mathcal{T}$ or $C \equiv D \in \mathcal{T}$, the lemma holds due to the normalization. Axioms within $\mathcal{T}$ can have the following form:

- $C, D \in \text{sig}_C(\mathcal{T})$. In this case, $\{C\} \in \texttt{Pre}(D)$. Therefore, condition [A1] holds.
- $C \in \text{sig}_C(\mathcal{T}), D = D_1 \sqcap ... \sqcap D_m$ with $D_1, ..., D_m \in \text{sig}_C(\mathcal{T})$. In this case, for each D_i with $1 \leq i \leq m$ holds $\{C\} \in \texttt{Pre}(D_i)$. Therefore, condition [A1] holds for each D_i.
- $C \in \text{sig}_C(\mathcal{T}), D = \exists r'.D'$ with $D' \in \text{sig}_C(\mathcal{T})$. This case corresponds to the condition [A4].

(AXTOP) Since the conjunction is empty in case $D = \top$, the lemma holds.

(AX) Since $C = D$, for each D_i there is a conjunct C_i of C with $C_i = D_i$. If $D_i \in \text{sig}_C(\mathcal{T})$, condition [A1] of the lemma holds. Otherwise, [A3].

(EX) If EX was the last applied rule, then $D_i = \exists r_k.D'$ and $\mathcal{T} \vdash D_k \sqsubseteq D'$. Therefore, [A3] of the lemma holds.

(ANDL) Assume that $C' \sqcap C'' = C$ such that $C' \sqsubseteq D$ is the antecedent. By induction hypothesis, the lemma holds for $C' \sqsubseteq D$. Since all conjuncts of C' are also conjuncts of C, the lemma holds also for $C \sqsubseteq D$.

(ANDR) Assume that $D = D_1 \sqcap D_2$, therefore, $C \sqsubseteq D_1$ and $C \sqsubseteq D_2$ is the antecedent. By induction hypothesis, the lemma holds for both, $C \sqsubseteq D_1$ and $C \sqsubseteq D_2$. Since all conjuncts of D are from either D_1 or D_2, the lemma also holds for $C \sqsubseteq D$.

(CUT) By induction hypothesis, the lemma holds for both elements of the antecedent, $C \sqsubseteq C_1$ and $C_1 \sqsubseteq D$. W.l.o.g., assume that $C_1 = \bigsqcap_{1 \leq p \leq r} A_p \sqcap \bigsqcap_{1 \leq s \leq t} \exists r'_s.E'_s$.

1. Assume that $D_i \in \text{sig}_C(\mathcal{T})$. Then, there is $M_1 \in \texttt{Pre}(D_i)$ such that [A1] or [A2] holds for each $B_1 \in M_1$.

   A1 Assume that there is A_p with $A_p = B_1$. Then, by induction hypothesis, for $C \sqsubseteq A_p$, there is $M_p \in \texttt{Pre}(A_p)$ such that [A1] or [A2] holds for each $B'_1 \in M_p$. Let $M_{\text{part}}(B_1) = M_p$ and $M_{1,A1} \subseteq M_1$ be the set of all such B_1. Then, let $M_{\text{new}} = M_1 \setminus M_{1,A1} \cup \bigcup\{M_{\text{part}}(B_1) \mid B_1 \in M_{1,A1}\}$.

   A2 Assume that for B_1 there are r'_s, E'_s and there exists $B' \in \text{sig}_C(\mathcal{T})$ such that $\mathcal{T} \models E'_s \sqsubseteq B'$ and $B \equiv \exists r'_s.B' \in \mathcal{T}$. Then, for $C \sqsubseteq \exists r'_s.E'_s$ can hold [A3] or [A4].

   -(A3) There are r_k, E_k such that $r_k = r'_s$ and $\mathcal{T} \models E_k \sqsubseteq E'_s$. Then [A2] holds for $C \sqsubseteq B_1$, since $\mathcal{T} \models E_k \sqsubseteq B'$ and $B \equiv \exists r_k.B' \in \mathcal{T}$.

   -(A4) There is $B'' \in nct$ such that $\mathcal{T} \models B'' \sqsubseteq \exists r'_s.E'_s$, $\mathcal{T} \models C \sqsubseteq B''$ and there is a set $M'' \in \texttt{Pre}(B'')$ such that for each element B' of M'' holds at least one of the conditions [A1]-[A2] with respect to $C \sqsubseteq B'$. Let $M_{\text{part}}(B_1) = M''$ and $M_{1,A4} \subseteq M_1$ be the set of all such B_1. Then, let $M'_{\text{new}} = M_{\text{new}} \setminus M_{1,A4} \cup \bigcup\{M_{\text{part}}(B_1) \mid B_1 \in (M_{1,A4} \setminus M_{1,A1})\}$.

   Clearly, $M'_{\text{new}} \in \texttt{Pre}(D_i)$ and [A1] or [A2] holds for each $B_1 \in M'_{\text{new}}$ with respect to $C \sqsubseteq B_1$, i.e., the lemma holds for $C \sqsubseteq D_i$.

2. Assume that $D_i = \exists r'.D'$. Then, [A3] or [A4] hold.

A3 There are r'_s, E'_s such that $r' = r'_s$ and $\mathcal{T} \models E'_s \sqsubseteq D'$. Then, for $C \sqsubseteq \exists r'_s.E'_s$ one of [A3], [A4] holds:

-(A3) There are r_k, E_k such that $r_k = r'_s$ and $\mathcal{T} \models E_k \sqsubseteq E'_s$. Then [A3] holds for $C \sqsubseteq D_i$, since $\mathcal{T} \models E_k \sqsubseteq D'$ and $r_k = r'$.

-(A4) There is $B'' \in$ nct such that $\mathcal{T} \models B'' \sqsubseteq \exists r'_s.E'_s$, $\mathcal{T} \models C \sqsubseteq B''$ and there is a set $M'' \in \mathtt{Pre}(B'')$ of $\text{sig}_C(\mathcal{T})$ concepts such that for each element B' of M'' holds at least one of the conditions [A1]-[A2] with respect to $C \sqsubseteq B'$. Since $\mathcal{T} \models B'' \sqsubseteq D_i$, [A4] holds for $\mathcal{T} \models C \sqsubseteq D_i$.

A4 There is $B \in$ nct such that $\mathcal{T} \models B \sqsubseteq \exists r'.D'$, $\mathcal{T} \models C_1 \sqsubseteq B$ and there is a set $M' \in \mathtt{Pre}(B)$ such that for each element B' of M holds at least one of the conditions [A1]-[A2] with respect to $C_1 \sqsubseteq B'$. Then, we have the same situation as above with two subsumptions $C \sqsubseteq C_1$ and $C_1 \sqsubseteq B$, where $B \in \text{sig}_C(\mathcal{T})$. Therefore, the argumentation is the same as above implying that the claim of the lemma holds for $C \sqsubseteq B$, i.e., there is $M_1 \in \mathtt{Pre}(B)$ such that [A1] or [A2] holds for each $B_1 \in M_1$. Then, [A4] holds for $C \sqsubseteq D_i$. □

Concerning subsumers of atomic concepts being existential restrictions, we can show the following property.

Lemma 14. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ TBox, $A \in sig_C(\mathcal{T})$ and $r \in sig_R(\mathcal{T})$. Let C an $\mathcal{EL}$ concept such that $\mathcal{T} \models A \sqsubseteq \exists r.C$. Then, there are $B_1, B_2 \in sig_C(\mathcal{T})$ with $B_1 \equiv \exists r.B_2 \in \mathcal{T}$ such that $\mathcal{T} \models A \sqsubseteq B_1$, $\mathcal{T} \models B_2 \sqsubseteq C$.*

Proof. Lemma 16 in [LUTZ and WOLTER 2010] states that for a general $\mathcal{EL}$ TBox $\mathcal{T}$ with $\mathcal{T} \models C_1 \sqsubseteq \exists r.C_2$, where C_1, C_2 are $\mathcal{EL}$-concepts one of the following holds:

- there is a conjunct $\exists r.C'$ of C_1 such that $\mathcal{T} \models C' \sqsubseteq C_2$;

- there is a subconcept $\exists r.C'$ of $\mathcal{T}$ such that $\mathcal{T} \models C_1 \sqsubseteq \exists r.C'$ and $\mathcal{T} \models C' \sqsubseteq C_2$;

The first condition does not hold in this lemma, since $A \in \text{sig}_C(\mathcal{T})$. Moreover, since in our case $\mathcal{T}$ is normalized, for each subconcept $\exists r.C'$ of $\mathcal{T}$ containing an existential restriction holds: there is an atomic concept $B_2 \in \text{sig}_C(\mathcal{T})$ such that $B_2 = C'$ and there is an axiom of the form $B_1 \equiv \exists r.B_2 \in \mathcal{T}$ with $B_1 \in \text{sig}_C(\mathcal{T})$. Additionally, from the above Lemma 16 follows $\mathcal{T} \models A \sqsubseteq \exists r.B_2$ and $\mathcal{T} \models B_2 \sqsubseteq C$. Since $\mathcal{T} \models B_1 \equiv \exists r.B_2$, it follows that also $\mathcal{T} \models A \sqsubseteq B_1$. □

The completeness of the initial subsumee/subsumer relation pair can be shown by induction using the following, more narrow notion of closure for subsumee/subsumer relation pairs. We refer to it as *deweakenized closure*, since, in contrast to the full closure as introduced in Section 5.1.2 after Definitoin 9, it does not contain all *weak* subsumees and subsumers. These are subsumees obtained by adding arbitrary conjuncts to arbitrary sub-expressions of other subsumees of the same concept and subsumers obtained by omitting arbitrary conjuncts from arbitrary sub-expressions of other subsumers.

Definition 13. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ TBox and $\langle R_\sqsupseteq, R_\sqsubseteq \rangle$ a subsumee/subsumer relation pair for $\mathcal{T}$.*

1. *The* deweakenized closure $R_\sqsupseteq^+$ *of $R_\sqsupseteq$ is the smallest set such that $R_\sqsupseteq \subseteq R_\sqsupseteq^+$ and for all $B \in sig_C(\mathcal{T})$ and C with $(B, C) \in R_\sqsupseteq^+$ holds: if some $B' \in sig_C(\mathcal{T})$ occurs in C, then also $(B, C') \in R_\sqsupseteq^+$ for all C' obtained by replacing any occurrence of B' by any element C with $(B', C) \in R_\sqsupseteq$.*

2. *The* deweakenized closure $R_\sqsubseteq^+$ *of $R_\sqsubseteq$ is the smallest set such that $R_\sqsubseteq \subseteq R_\sqsubseteq^+$ and the following condition hold for all $B \in sig_C(\mathcal{T})$: if $(B, C) \in R_\sqsubseteq^+$ and some $B' \in sig_C(\mathcal{T})$ occurs in C, then also $(B, C') \in R_\sqsubseteq^+$ for all C' obtained by replacing any occurrence of B' either by $\bigsqcap_{(B',D)\in R_\sqsubseteq} D$ or by any element D such that $(B', D) \in R_\sqsubseteq$.*

In what follows, we write $\mathtt{M}(R_\sqsupseteq, R_\sqsubseteq)$ instead of $\mathtt{M}(R_\sqsupseteq, R_\sqsubseteq, \Sigma^{\mathtt{ext}})$, if $\Sigma^{\mathtt{ext}} = \Sigma^{\mathtt{ext}}(R_\sqsupseteq, R_\sqsubseteq)$. Based on the above definition, we can prove that $\mathtt{M}(R_\sqsupseteq, R_\sqsubseteq) \equiv \mathtt{M}(R_\sqsupseteq^+, R_\sqsubseteq^+)$.

Lemma 15. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ TBox and $\langle R_\sqsupseteq, R_\sqsubseteq\rangle$ a subsumee/subsumer relation pair for $\mathcal{T}$. Then, $\langle R_\sqsupseteq^+, R_\sqsubseteq\rangle$ and $\langle R_\sqsupseteq, R_\sqsubseteq^+\rangle$ are subsumee/subsumer relation pairs for $\mathcal{T}$. If $\langle R_\sqsupseteq, R_\sqsubseteq\rangle$ is complete with respect to Σ, then $\mathtt{M}(R_\sqsupseteq, R_\sqsubseteq) \equiv \mathtt{M}(R_\sqsupseteq^+, R_\sqsubseteq^+)$.*

Proof. The property of being a subsumee/subsumer relation pair for $\mathcal{T}$:

- We show that $\langle R_\sqsupseteq^+, R_\sqsubseteq\rangle$ is a subsumee/subsumer relation pair for $\mathcal{T}$. The theorem holds, if for each $B \in \mathrm{sig}_C(\mathcal{T})$ and for each C with $(B, C) \in R_\sqsupseteq^+$ holds $\mathcal{T} \models C \sqsubseteq B$. Since for each $(B, C) \in R_\sqsupseteq^+$, there is a finite sequence of derivations according to Definition 13, we prove by induction on the length of the derivation that $\mathcal{T} \models C \sqsubseteq B$. If the length of the derivation is 0, then $(B, C) \in R_\sqsupseteq$ by Definition 13 and the claim holds.

  Assume that the claim holds for some C'' with $(B, C'') \in R_\sqsupseteq^+$, i.e., $\mathcal{T} \models C'' \sqsubseteq B$, and assume that C has been derived from C'' according to Definition 13 by replacing B' within C'' by some C' with $(B', C') \in R_\sqsupseteq$. Then, $\mathcal{T} \models C' \sqsubseteq B'$ and, therefore, $\mathcal{T} \models C \sqsubseteq B$.

- The proof for $\langle R_\sqsupseteq, R_\sqsubseteq^+\rangle$ being a subsumee/subsumer relation pair for $\mathcal{T}$ is done in the same way by induction on the length of the derivation, since $\mathcal{T} \models \bigsqcap_{(B', D) \in R_\sqsubseteq} D \sqsubseteq B'$.

The claim $\mathtt{M}(R_\sqsupseteq, R_\sqsubseteq) \equiv \mathtt{M}(R_\sqsupseteq^+, R_\sqsubseteq^+)$ can be shown as follows. We start with $\mathtt{M}(R_\sqsupseteq, R_\sqsubseteq) \equiv \mathtt{M}(R_\sqsupseteq^+, R_\sqsubseteq)$.

- First, note that replacements in Definition 13 do not add any new elements to $\Sigma^{\mathtt{ext}}(R_\sqsupseteq, R_\sqsubseteq)$: if a concept is not referenced, then it will not be referenced after a replacement. Due to an inclusion of the relation by its closure, $\Sigma^{\mathtt{ext}}(R_\sqsupseteq^+, R_\sqsubseteq^+)$=$\Sigma^{\mathtt{ext}}(R_\sqsupseteq, R_\sqsubseteq)$. The direction $\mathtt{M}(R_\sqsupseteq^+, R_\sqsubseteq) \models \mathtt{M}(R_\sqsupseteq, R_\sqsubseteq)$ follows from $R_\sqsupseteq \subseteq R_\sqsupseteq^+$ and the monotonicity of reasoning in $\mathcal{EL}$. In order to show the direction $\mathtt{M}(R_\sqsupseteq, R_\sqsubseteq) \models \mathtt{M}(R_\sqsupseteq^+, R_\sqsubseteq)$, we show by induction on the length of the derivation that, for any $B \in \mathrm{sig}_C(\mathcal{T})$ and for any C with $(B, C) \in R_\sqsupseteq^+$ holds $\mathtt{M}(R_\sqsupseteq, R_\sqsubseteq) \models \mathtt{M}(R_\sqsupseteq \cup \{(B, C)\}, R_\sqsubseteq)$. Let us denote the first TBox by $\mathtt{M}_1$ and the second one by $\mathtt{M}_2$. Assume that the length of the derivation is 0. Then, $(B, C) \in R_\sqsupseteq$ by Definition 13 and we obtain $\mathtt{M}_1 = \mathtt{M}_2$.

Assume that the claim holds for some $R'_{\sqsupseteq}$ with $R_{\sqsupseteq} \subseteq R'_{\sqsupseteq} \subseteq R^+_{\sqsupseteq}$, i.e., $\mathtt{M}_1 \models \mathtt{M}(R'_{\sqsupseteq}, R_{\sqsubseteq})$. Then, for any B and any C'' with $(B, C'') \in R'_{\sqsupseteq}$ holds $\mathtt{M}_1 \models \mathtt{M}(R_{\sqsupseteq} \cup \{(B, C'')\}, R_{\sqsubseteq})$ (last TBox denoted as $\mathtt{M}'_2$). Assume that some C has been derived from C'' according to Definition 13 by replacing B' within C'' by some C' with $(B', C') \in R_{\sqsupseteq}$. Note that $B' \in \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$. Now we distinguish two cases:

1. $B \in \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$: By Definition 9, $\mathtt{M}(R'_{\sqsupseteq} \cup \{(B, C)\}, R_{\sqsubseteq}) = \mathtt{M}(R'_{\sqsupseteq}, R_{\sqsubseteq}) \cup \{C \sqsubseteq B\}$. To show that the claim holds for the latter extension of the subsumee relation with (B, C), we need to show that $\mathtt{M}_1 \models C \sqsubseteq B$. Since $C' \sqsubseteq B' \in \mathtt{M}_1$ and $C'' \sqsubseteq B \in \mathtt{M}'_2$ by Definition 9, we obtain $\mathtt{M}_1 \models C'' \sqsubseteq B$, and, therefore, $\mathtt{M}_1 \models C \sqsubseteq B$.
2. $B \notin \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$: By Definition 9, $\mathtt{M}(R'_{\sqsupseteq} \cup \{(B, C)\}, R_{\sqsubseteq}) = \mathtt{M}(R'_{\sqsupseteq}, R_{\sqsubseteq}) \cup \{C \sqsubseteq D \mid (B, D) \in R_{\sqsubseteq}\}$. To show that the claim holds for the latter extension of the subsumee relation with (B, C), we need to show that $\mathtt{M}_1 \models \{C \sqsubseteq D \mid (B, D) \in R_{\sqsubseteq}\}$. Since $C' \sqsubseteq B' \in \mathtt{M}_1$ and $C'' \sqsubseteq D \in \mathtt{M}'_2$ for any D with $(B, D) \in R_{\sqsubseteq}$ by Definition 9, we obtain $\mathtt{M}_1 \models C'' \sqsubseteq D$ for any D with $(B, D) \in R_{\sqsubseteq}$. Therefore, $\mathtt{M}_1 \models \{C \sqsubseteq D \mid (B, D) \in R_{\sqsubseteq}\}$.

It follows that $\mathtt{M}_1 \models \mathtt{M}(R'_{\sqsupseteq} \cup \{(B, C)\}, R_{\sqsubseteq})$.

- The proof for $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}) \equiv \mathtt{M}(R_{\sqsupseteq}, R^+_{\sqsubseteq})$ is done in the same way by induction on the length of the derivation with the additional implication of $\mathtt{M}_1 \models B' \sqsubseteq \bigsqcap_{(B', C'_i) \in R_{\sqsubseteq}} C'_i$ from $B' \sqsubseteq C'_i \in \mathtt{M}_1$.

From the above claims follows that $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}) \equiv \mathtt{M}(R^+_{\sqsupseteq}, R^+_{\sqsubseteq})$. □

To be able to use the Lemma 13 directly, we additionally prove that elements of $\mathtt{Pre}$ are contained in the deweakenized closure $R^+_{\sqsupseteq}$.

Lemma 16. *Let $\mathcal{T}$ be normalized $\mathcal{EL}$ TBox. For $A \in sig_C(\mathcal{T})$, let $K \in \mathtt{Pre}(A)$ such that $K \neq \{A\}$. Then, $K \subseteq \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$ and $(A, \bigsqcap_{B_i \in K} B_i) \in R^+_{\sqsupseteq}$.*

Proof. First, note that, since $\mathrm{sig}_C(\mathcal{T})$ is finite, also $\mathtt{Pre}(B)$ for each $B \in \mathrm{sig}_C(\mathcal{T})$ is finite and has a finite derivation using the two derivation rules in Definition 12. We show the lemma by induction on the length of derivation for K starting with

1. The derivation starts with $\{A\}$ and consists of a second (last) derivation rule, which can be one of the following:

- $A \equiv B_1 \sqcap ... \sqcap B_n \in \mathcal{T}$, and, therefore, $(K/\{A\}) \cup \{B_1, ..., B_n\} \in \mathtt{Pre}(A)$. By Definition 11, $(A, B_1 \sqcap ... \sqcap B_n) \in R_{\sqsupseteq}$. Therefore, $(A, B_1 \sqcap ... \sqcap B_n) \in R^+_{\sqsupseteq}$. Note that $\{B_1, ..., B_n\} \subseteq \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$ by definition.

- $\mathcal{T} \models B' \sqsubseteq A$, and, therefore, $(K/\{A\}) \cup \{B'\} \in \mathtt{Pre}(A)$. By Definition 11, $(A, B') \in R_{\sqsupseteq}$. Therefore, $(A, B') \in R^+_{\sqsupseteq}$. Note that $\{B'\} \subseteq \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$ by definition.

Assume that for some $K' \in \mathtt{Pre}(A)$ such that $K' = \{B_1, ..., B_n\}$ the lemma holds. There are the following possibilities to derive K from K' by a single additional derivation:

- $B \equiv B'_1 \sqcap ... \sqcap B'_n \in \mathcal{T}$ for some $B \in K'$ and $(K'/\{B\}) \cup \{B'_1, ..., B'_n\} = K$. By Definition 13, replacing the corresponding B within $B_1 \sqcap ... \sqcap B_n$ such that $(A, B_1 \sqcap ... \sqcap B_n) \in R^+_{\sqsupseteq}$ by $B'_1 \sqcap ... \sqcap B'_n$ yields again an element C with $(A, C) \in R^+_{\sqsupseteq}$. Note that $\{B'_1, ..., B'_n\} \subseteq \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$ by definition, therefore $K \subseteq \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$.

- $\mathcal{T} \models B' \sqsubseteq B$ for some $B \in K'$ and $(K'/\{B\}) \cup \{B'\} = K$. By Definition 13, replacing the corresponding B within $B_1 \sqcap ... \sqcap B_n$ such that $(A, B_1 \sqcap ... \sqcap B_n) \in R^+_{\sqsupseteq}$ by B' yields again an element C with $(A, C) \in R^+_{\sqsupseteq}$. Note that $\{B'\} \subseteq \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$ by definition, therefore $K \subseteq \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$.

□

Now we can show that the initial subsumee/subsumer relation pair meets the completeness criterion:

Theorem 1. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ ontology, $\Sigma \subseteq sig(\mathcal{T})$ a signature, and $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$ the initial subsumee/subsumer relation pair for $\mathcal{T}$, then $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$ is complete with respect to Σ.*

Proof. By Definition of completeness, we need to show that $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}) \in \mathtt{MOD}(\mathcal{T}, \Sigma)$ (to simplify the notations, we denote the latter TBox by $\mathtt{M}$). This is the case, if, by Definition 8, $\mathcal{T} \equiv^{c}_{\Sigma} \mathtt{M}$ and $\mathcal{T} \models \mathtt{M}$. The second of the two

statements follows from Definition 9 and the property of being a subsumee/subsumer relation pair for $\mathcal{T}$. By Definition 7, the statement $\mathcal{T} \equiv^{c}_{\Sigma} \mathtt{M}$ consists of two directions: (1) for all $\mathcal{EL}$ concepts C, D with $\text{sig}(C) \cup \text{sig}(D) \subseteq \Sigma$ holds $\mathtt{M} \models C \sqsubseteq D \Rightarrow \mathcal{T} \models C \sqsubseteq D$ and (2) for all $\mathcal{EL}$ concepts C, D with $\text{sig}(C) \cup \text{sig}(D) \subseteq \Sigma$ holds $\mathtt{M} \models C \sqsubseteq D \Leftarrow \mathcal{T} \models C \sqsubseteq D$. The first direction follows from $\mathcal{T} \models \mathtt{M}$. For the second direction, assume that there are such C, D with $\text{sig}(C) \cup \text{sig}(D) \subseteq \Sigma$ and it holds that $\mathcal{T} \models C \sqsubseteq D$. We prove by induction on maximal role depth of C, D that also $\mathtt{M} \models C \sqsubseteq D$. W.l.o.g., let $D = \bigsqcap_{1 \leq i \leq l} D_i$ and

$$C = \bigsqcap_{1 \leq j \leq n} A_j \sqcap \bigsqcap_{1 \leq k \leq m} \exists r_k.E_k$$

with $A_j \in \Sigma \cap \text{sig}_C(\mathcal{T})$ for $1 \leq j \leq n$, $r_k \in \Sigma \cap \text{sig}_R(\mathcal{T})$ for $1 \leq k \leq m$ and E_k with $1 \leq k \leq m$ a set of $\mathcal{EL}$ concepts such that $\text{sig}(E_k) \subseteq \Sigma$. Clearly, $\mathcal{T} \models C \sqsubseteq D$, iff $\mathcal{T} \models C \sqsubseteq D_i$ for all i with $1 \leq i \leq l$. The proof of this theorem uses the following two claims concerning the deweakenized relation closures:

- Claim 1: We show that, for each such general C with $\text{sig}(C) \subseteq \Sigma$ and each $A \in \text{sig}_C(\mathcal{T})$ with $\mathcal{T} \models C \sqsubseteq A$ there is C' with $\text{sig}(C') \subseteq \Sigma, \{\} \models C \sqsubseteq C'$ such that holds $(A, C') \in R^{+}_{\sqsupseteq}$. We prove the claim by induction on the role depth of C:
  - Assume role depth = 0. Then C is a conjunction of atomic concepts, i.e., $m = 0$ and $C = \bigsqcap_{1 \leq j \leq n} A_j$. Then, by Lemma 13, there is a set $K' \in \mathtt{Pre}(A)$ of atomic concepts such that, for each $B \in K'$, there is an A_j with $A_j = B$. Therefore, each $B \in K'$ is in Σ. By Lemma 16, $(A, C) \in R^{+}_{\sqsupseteq}$.
  - Assume that the role depth is greater than 0. As in the case above, by Lemma 13 there is a set $K' \in \mathtt{Pre}(A)$ of atomic concepts such that, for each $B \in K'$, [A1] or [A2] holds. Let $K'_1 = K' \cap \{A_1, ...A_n\}$ and $K'_2 = K' \setminus K'_1$. Let $C'_1 = \bigsqcap_{B \in K'_1} B$, and $C'_2 = \bigsqcap_{1 \leq f \leq p} \exists r'_f.E'_f$ with $\{\exists r'_1.E'_1, ..., \exists r'_p.E'_p\} = \{\exists r'_f.E'_f \mid \exists r'_f.E'_f = \exists r_k.E_k$ for some k and for one of $B_f \in K'_2$ holds [A2] such that there exists $B'_f \in \text{sig}_C(\mathcal{T})$ with $\mathcal{T} \models E'_f \sqsubseteq B'_f$ and $B_f \equiv \exists r'_f.B'_f \in \mathcal{T}\}$. Since $B'_f \in \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$, by induction hypothesis, there is such E''_f

with $(B'_f, E''_f) \in R^+_\sqsupseteq$ that $\text{sig}(E''_f) \subseteq \Sigma, \{\} \models E'_f \sqsubseteq E''_f$. Then, $(B_f, \exists r'_f.E''_f) \in R^+_\sqsupseteq$. Let $C''_2 = \exists r'_1.E''_1 \sqcap ... \sqcap \exists r'_p.E''_p$. Note that $\{\} \models C'_2 \sqsubseteq C''_2$. Since, by Lemma 16, we can assume that $K' \subseteq \Sigma^{\texttt{ext}}(R_\sqsupseteq, R_\sqsubseteq)$, by Lemma 16, $(A, \bigsqcap_{B' \in K'} B') \in R^+_\sqsupseteq$. By applying Definition 13, we can obtain $C'_1 \sqcap C''_2$ with $(A, C'_1 \sqcap C''_2) \in R^+_\sqsupseteq$ by replacing each B' by the corresponding conjunct. Additionally, $\{\} \models C \sqsubseteq C'_1 \sqcap C''_2$ and $\text{sig}(C'_1 \sqcap C''_2) \subseteq \Sigma$. Therefore, the claim holds.

- Claim 2: We show that, for each such general C with $\text{sig}(C) \subseteq \Sigma$ and each $A \in \text{sig}_C(\mathcal{T})$ with $\mathcal{T} \models A \sqsubseteq C$ there is C' with $\{\} \models C' \sqsubseteq C$ such that holds $(A, C') \in R^+_\sqsubseteq$. We prove the claim by induction of the role depth of C.

  - For each A_j, we know that $\mathcal{T} \models A \sqsubseteq A_j$, i.e., $(A, A_j) \in R^+_\sqsubseteq$, and $A_j \in \Sigma$. Therefore, by Definition 13, there is D' such that $(A, \bigsqcap_{1 \leq j \leq n} A_j \sqcap D') \in R^+_\sqsubseteq$, i.e., $(A, C \sqcap D') \in R^+_\sqsubseteq$.
  - Assume that the role depth is greater than 0. For each $\exists r_k.E_k$, it follows from Lemma 14 that there are $B_k, B'_k \in \text{sig}_C(\mathcal{T})$ with $B_k \equiv \exists r_k.B'_k \in \mathcal{T}$ such that $\mathcal{T} \models A \sqsubseteq B_k, \mathcal{T} \models B'_k \sqsubseteq E_k$. Therefore, for each k holds $(A, B_k) \in R_\sqsubseteq$ and $(B_k, \exists r_k.B'_k) \in R_\sqsubseteq$. By Definition 13 follows that $(A, \bigsqcap_{1 \leq j \leq n} A_j \sqcap \bigsqcap_{1 \leq k \leq m} B_k \sqcap D') \in R^+_\sqsubseteq$ for some D'. Further, By Definition 13 follows that for some D'' holds $(A, \bigsqcap_{1 \leq j \leq n} A_j \sqcap \bigsqcap_{1 \leq k \leq m} \exists r_k.B'_k \sqcap D'') \in R^+_\sqsubseteq$. Moreover, by induction hypothesis follows that for each k there is such E'_k that $\{\} \models E'_k \sqsubseteq E_k$ and $(B'_k, E'_k) \in R^+_\sqsubseteq$. Therefore, after several replacement steps starting with $(A, \bigsqcap_{1 \leq j \leq n} A_j \sqcap \bigsqcap_{1 \leq k \leq m} \exists r_k.B'_k \sqcap D'') \in R^+_\sqsubseteq$ according to Definition 13, we obtain $C' = \bigsqcap_{1 \leq j \leq n} A_j \sqcap \bigsqcap_{1 \leq k \leq m} \exists r_k.E'_k \sqcap D''$ with the corresponding inclusion $(A, C') \in R^+_\sqsubseteq$. Since $\{\} \models C' \sqsubseteq \bigsqcap_{1 \leq j \leq n} A_j \sqcap \bigsqcap_{1 \leq k \leq m} \exists r_k.E_k$, the claim holds.

Now we prove the theorem by distinguishing the following two cases according to the cases in Lemma 13:

- If $D_i = A \in \Sigma$, then the theorem follows from the Claim 1, stating that there is C' with $\text{sig}(C') \subseteq \Sigma, \{\} \models C \sqsubseteq C'$ and $(A, C') \in R^+_{\sqsupseteq}$, and Lemma 18, stating that $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}) \equiv \mathtt{M}(R^+_{\sqsupseteq}, R^+_{\sqsubseteq})$.

- If $D_i = \exists r.D'$ for some r, D', then, by Lemma 13, one of the following is true:

  (A3) There are r_k, E_k in C such that $r_k = r$ and $\mathcal{T} \models E_k \sqsubseteq D'$. By induction hypothesis holds $\mathtt{M} \models E_k \sqsubseteq D'$. It follows that $\mathtt{M} \models \exists r_k.E_k \sqsubseteq D_i$ and $\mathtt{M} \models C \sqsubseteq D_i$.

  (A4) There is $B \in \text{sig}_C(\mathcal{T})$ of $\mathcal{T}$ such that $\mathcal{T} \models B \sqsubseteq \exists r.D'$ and $\mathcal{T} \models C \sqsubseteq B$. Then, it follows from Claim 1 that there is C' with $\text{sig}(C') \subseteq \Sigma, \{\} \models C \sqsubseteq C'$ with $(B, C') \in R^+_{\sqsupseteq}$ and it follows from Claim 2 that there is D'' with $\{\} \models D'' \sqsubseteq \exists r.D'$ and $(B, D'') \in R^+_{\sqsubseteq}$. Therefore, by Definition 9, $\mathtt{M} \models C' \sqsubseteq \exists r.D'$, i.e., $\mathtt{M} \models C \sqsubseteq D_i$.

□

Up to now, we discussed the usefulness of primitivization and the decomposition of a TBox into subsumee/subsumer relation pairs for rewriting-based general module extraction. In the following, we will discuss the properties of two different rewriting approaches based on primitivization.

5.2 Uniform Interpolation

In this section, we consider a particular type of rewriting, namely uniform interpolation. Given a relevant signature Σ and a TBox $\mathcal{T}$, the task of uniform interpolation is to determine a TBox $\mathcal{T}'$ with $\text{sig}(\mathcal{T}') \subseteq \Sigma$ such that $\mathcal{T} \equiv^{\text{c}}_{\Sigma} \mathcal{T}'$. $\mathcal{T}'$ is also called a *uniform $\mathcal{EL}$ Σ-interpolant* of $\mathcal{T}$.
Here, we consider the task of computing uniform interpolants in general $\mathcal{EL}$ terminologies, which turns out to be a difficult problem that has not been solved before this thesis. An existing approach [KONEV et al. 2009b] to uniform interpolation in $\mathcal{EL}$ is restricted to terminologies containing each atomic concept at most once on the left-hand side of concept inclusions and additionally satisfying sufficient, but not necessary acyclicity conditions. Lutz and Wolter [LUTZ and WOLTER 2011]

propose an approach to uniform interpolation in expressive description logics based on $\mathcal{ALC}$ featuring general terminologies, which, however does not solve the problem of uniform interpolation in $\mathcal{EL}$. Recently, Lutz, Seylan and Wolter [LUTZ et al. 2012] proposed an EXPTIME procedure for deciding, whether a finite uniform $\mathcal{EL}$ interpolant exists for a particular general terminology and a particular set of relevant terms. However, the authors do not address the actual computation of such a uniform interpolant. Also the bound on the size of uniform $\mathcal{EL}$ interpolants remained unknown.
In the following, we discuss a worst-case-optimal, proof-theoretic approach to computing a finite uniform $\mathcal{EL}$ interpolant for a general terminology. For a successful computation, it is required that such a finite uniform interpolant exists. As demonstrated by the following example, in the presence of cyclic concept inclusions, a finite uniform $\mathcal{EL}$ Σ-interpolant might not exist for a particular TBox $\mathcal{T}$ and a particular Σ.

Example 6. *Consider uniform interpolants of the TBox* $\mathcal{T} = \{A' \sqsubseteq A, A \sqsubseteq A'', A \sqsubseteq \exists r.A, \exists s.A \sqsubseteq A\}$. *with respect to* $\Sigma = \{s, r, A', A''\}$. *We obtain an infinite chain of consequences* $A' \sqsubseteq \exists r.\exists r.\exists r....A''$ *and* $\exists s.\exists s.\exists s....A' \sqsubseteq A''$ *containing nested existential quantifiers of unbounded depth.*

We show that, if such a finite uniform $\mathcal{EL}$ interpolant exists for the given terminology and signature, then there exists a uniform $\mathcal{EL}$ interpolant of at most triple exponential size. Further, we show that, in the worst-case, no shorter interpolants exist, thereby establishing the triple exponential tight bounds on the size of uniform interpolants in $\mathcal{EL}$.

5.2.1 Upper Bound

Now we discuss the upper bound on the size of uniform $\mathcal{EL}$ interpolants as well as their computation. Since, for a TBox $\mathcal{T}$ and a signature Σ, there are in general infinitely many Σ-consequences, in the following, we aim at identifying a subset of such consequences, the deductive closure of which contains the whole set. Interestingly, we can give a bound on the role depth of Σ-consequences such that, for the set $\mathcal{T}_{\Sigma,N}$ of all Σ-consequences of $\mathcal{T}$ with the maximal role depth N holds:

either $\mathcal{T}_{\Sigma,N}$ is a uniform $\mathcal{EL}$ interpolant of $\mathcal{T}$ with respect to Σ or such a finite uniform $\mathcal{EL}$ interpolant of $\mathcal{T}$ does not exist. This can easily be shown given the results obtained by Lutz and Wolter [LUTZ et al. 2012] while investigating the problem of existence of uniform $\mathcal{EL}$ interpolants. For a concept C, let $d(C)$ denote the maximal role depth within C. For a TBox $\mathcal{T}$, $d(\mathcal{T}) = \mathtt{max}\{d(C) \mid C \text{ is a sub-expression of } \mathcal{T}\}$.

Lemma 17. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ TBox, Σ a signature. Let def($\mathcal{T}$) be the number of definitions in $\mathcal{T}$. The following statements are equivalent:*

1. *There exists a uniform $\mathcal{EL}$ Σ-interpolant of $\mathcal{T}$.*
2. *There exists a uniform $\mathcal{EL}$ Σ-interpolant $\mathcal{T}'$ of $\mathcal{T}$ such that $d(\mathcal{T}') \leq 2^{4\cdot(|sig_C(\mathcal{T})|+def(\mathcal{T}))} + 1$.*

Proof. In a normalized TBox $\mathcal{T}$, the number of sub-expressions[1] is $|\text{sig}_C(\mathcal{T})| + \text{def}(\mathcal{T})$. Therefore, we can replace the last statement of Condition 2 by $d(\mathcal{T}') \leq 2^{2\cdot n} + 1$, where n is twice the number of sub-expressions within $\mathcal{T}$. Then, the lemma follows from Conditions (1) and (4) of Lemma 55 in [LUTZ et al. 2012]. □

However, knowing the above bound on the role depth is only sufficient to show the non-elementary upper bound on the size of uniform interpolants for the following reasons. There are 2^n many different conjunctions of n different conjuncts, and, accordingly, for each role, 2^n many different existential restrictions of depth $i+1$ if n is the number of existential restrictions of depth i. Since, for any role depth n, we can find a TBox such that n is the corresponding maximal role depth, for each number n of exponents, we can find a TBox with the corresponding size of the uniform interpolant.

In order to obtain a tight upper bound, we need to further reduce the subset of Σ-consequences required to obtain a uniform interpolant. On the one hand, we show that, in case of normalized terminologies, Σ-consequences consisting of subsumees and subsumers of atomic concepts in $\mathcal{T}$ of depth $2^{4\cdot(|\text{sig}_C(\mathcal{T})|+\text{def}(\mathcal{T}))}+1$ are sufficient. On the other hand, we show that subsumees of a particular type do not

[1] In a conjunction, only the concepts not being a conjunction itself are considered as proper sub-expressions. Therefore, a conjunction with n elements has n proper sub-expressions.

add any consequences to the deductive closure. These are subsumees obtained by adding arbitrary conjuncts to arbitrary sub-expressions of other subsumees of the same concept. In the following, we refer to this type of subsumees as *weak* subsumees. In the following, we show that, in case a finite uniform $\mathcal{EL}$ interpolant of $\mathcal{T}$ with respect to Σ exists, there are at most triple-exponentially many non-weak subsumees and subsumers. Moreover, we show that each of them is of at most double-exponential size.

For this purpose, we represent the above specified required subsets of subsumees and subsumers as languages of regular tree grammars on ranked unordered trees, in which transition rules are given by subsumees and subsumers with a maximal role depth 1 (with atomic concepts interpreted as non-terminals). The latter representation is convenient for us, since the generation of trees exactly reflects the deduction of subsumees and subsumers of a higher role depth and allows us to derive the corresponding bound on the size of the above sets.

Grammar Representation of Subsumees and Subsumers

First, we briefly recall the basics on tree languages and regular tree grammars. A *ranked alphabet* is a pair ($\mathcal{F}$, Arity) where $\mathcal{F}$ is a finite set and Arity is a mapping from $\mathcal{F}$ into $\mathbb{N}$. $T(\mathcal{F})$ denotes the set of ground terms over the alphabet $\mathcal{F}$. Let $\mathcal{X}_n$ be a set of n variables. A term $C \in T(\mathcal{F}, \mathcal{X}_n)$ containing each variable from $\mathcal{X}_n$ at most once is called a *context*. We denote by $C(\mathcal{F})$ the set of contexts containing a single variable.

Example 7. *Let $\mathcal{F} = \{f_2, g_1, a\}$ with the arity of symbols denoted by the subscript and X, Y two variables. Expressions $f_2(g_1(a), X)$, $f_2(g_1(Y), X)$ and $f_2(Y, X)$ are contexts derived from the tree $f_2(g_1(a), a)$, while $f_2(g_1(X), X)$ is not.*

A *regular tree grammar* $G = (S, \mathcal{N}, \mathcal{F}, R)$ is composed of a *start symbol* S, a set $\mathcal{N}$ of *non-terminal symbols* (non-terminal symbols have arity 0) with $S \in \mathcal{N}$, a ranked alphabet $\mathcal{F}$ of *terminal symbols* with a fixed arity such that $\mathcal{F} \cap \mathcal{N} = \emptyset$, and a set R of derivation rules of the form $X \to \beta$ where β is a tree from $T(\mathcal{F} \cup \mathcal{N})$ and $X \in \mathcal{N}$. Given a regular tree grammar $G = (S, \mathcal{N}, \mathcal{F}, R)$, the derivation relation $\to_G$ associated to G is a relation on pairs of terms from $T(\mathcal{F} \cup \mathcal{N})$ such

that $s \rightarrow_G t$ if and only if there is a rule $X \rightarrow \alpha \in R$ and there is a context C such that $s = C$ and $t = C[\alpha/X]$. The language generated by G, denoted by $L(G)$ is a subset of $T(\mathcal{F})$ which can be reached by successive derivations starting with the start symbol, i.e. $L(G) = \{s \in T \mid S \rightarrow_G^+ s\}$ with $\rightarrow_G^+$ the transitive closure of $\rightarrow_G$. We omit the substript G when the grammar G is clear from the context.

Example 8. *Let $G = (A, \{A, B\}, \{f_2, g_1, a, b\}, R)$ with R given by the following derivation rules:*

- $A \rightarrow f_2(B, A) \mid a$
- $B \rightarrow g_1(A) \mid b$

Then, $f_2(g_1(a), a) \in L(G)$, since $A \rightarrow f_2(B, A) \rightarrow f_2(B, a) \rightarrow f_2(g_1(A), a) \rightarrow f_2(g_1(a), a)$.

For further details on regular tree grammars, we refer the reader, for instance, to [COMON et al. 2008].

In our definition of grammars, we uniquely represent each atomic concept $A \in \text{sig}_C(\mathcal{T})$ by a non-terminal $\mathfrak{n}_A$ (and denote the set of all non-terminals by $\mathcal{N}^{\mathcal{T}} = \{\mathfrak{n}_x | x \in \text{sig}_C(\mathcal{T}) \cup \{\top\}\}$). In what follows, we use the ranked alphabet $\mathcal{F} = (\text{sig}_C(\mathcal{T}) \cap \Sigma) \cup \{\top\} \cup \{\exists r_1 \mid r \in \text{sig}_R(\mathcal{T}) \cap \Sigma\} \cup \{\sqcap_i \mid i \leq n\}$, where atomic concepts in $\text{sig}_C(\mathcal{T}) \cap \Sigma$ are constants, $\exists r$ for $r \in \text{sig}_R(\mathcal{T}) \cap \Sigma$ are unary functions and $\sqcap_i$ are functions of the arity i bounded by $n = |\text{sig}_C(\mathcal{T})| \cdot (|\text{sig}_R(\mathcal{T})| + 1)$, i.e., the number of all possible simple concepts in $\mathcal{T}$ (atomic concepts and all existential restrictions on atomic concepts). The restriction to the maximum arity of n is w.l.o.g., since we can always split longer conjunctions into a nested conjunction with at most n elements in each sub-expression. In the following, it will be convenient to simply write $\sqcap$ and $\exists r$ if the arity of the corresponding function is clear from the context. Clearly, every $\mathcal{EL}$ concept C with $\text{sig}(C) \subseteq \Sigma$ and at most n conjuncts in each sub-expression has a unique representation by the means of the above functions. We denote such a term representation of C using $\mathcal{F}$ by t_C.

In what follows, we use a substituting function $\sigma_{\mathcal{T},\mathcal{F}} : \{C \mid \text{sig}(C) \subseteq \text{sig}(\mathcal{T})\} \rightarrow T(\mathcal{F}, \mathcal{N}^{\mathcal{T}})$ with $\sigma_{\mathcal{T},\mathcal{F}}(C) = t_C\{\mathfrak{n}_\top/\top, \mathfrak{n}_{B_1}/B_1, ..., \mathfrak{n}_{B_n}/B_n\}$, where $B_1, ..., B_n$ are all atomic sub-expressions of C. If the TBox and the set of non-terminals are

clear from the context, we will denote such a representation of a concept C simply by $\sigma(C)$.

As mentioned above, weak subsumees are not required in order to obtain a uniform $\mathcal{EL}$ interpolant. In fact, including weak subsumees into our definition of the grammars would be inconvenient, since, in general, adding arbitrary conjuncts to arbitrary sub-expressions allows us to obtain subsumees being conjunctions of unbounded size. This would cause the corresponding language to contain terms with $\sqcap$-functions of unbounded arity and lead to several disadvantages. On the one hand, it would make the definition of the gammars deriving subsumees and subsumers more complex. On the other hand, it is not convinient for the estimation of the grammar size, which is required for a derivation of an upper bound on the size of uniform interpolants. Thus, the design decision to exclude weak subsumees from the corresponding languages is straightforward.

Bounded arity of the $\sqcap$-function is not the only advantage of the exclusion of weak subsumees from the corresponding languages. Interestingly, instead of introducing transition rules for all subsumees containing existential restrictions with a role depth 1 as has been described at the beginning of this section, in the above case it is sufficient to introduce transition rules only for such subsumees explicitly given in the normalized TBox (see proof of Theorem 4). This simplifies the grammar construction, since the only implicit subsumees required within the transition rules are conjunctions of atomic concepts, i.e., subsumees given by $\mathtt{Pre}(A) = \{M \subseteq \text{sig}_C(\mathcal{T}) \mid \mathcal{T} \models \bigsqcap_{B_i \in M} B_i \sqsubseteq A\}$ for each $A \in \text{sig}_C(\mathcal{T})$.

In case of subsumers, the situation is different: In order to show the corresponding upper bound, we require all subsumers of atomic concepts (see proof of Theorem 5). Thus, the corresponding transition rules of subsumer grammars are required to contain all subsumers with a maximal role depth 1, which we denote by $\mathtt{Post}_{\text{Base}}(A) = \{A' \in \text{sig}_C(\mathcal{T}) \cup \{\top\} \mid \mathcal{T} \models A \sqsubseteq A'\} \cup \{\exists r.A' \mid A' \in \text{sig}_C(\mathcal{T}) \cup \{\top\}, \mathcal{T} \models A \sqsubseteq \exists r.A', r \in \Sigma\}$. Further, we use the notation $\mathtt{Post}(A) = 2^{\mathtt{Post}_{\text{Base}}(A)}$ for the power set of $\mathtt{Post}_{\text{Base}}(A)$ and obtain the following definition.

Definition 14. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ TBox, Σ a signature. Further, let $\mathtt{Pre}(A) = \{M \subseteq sig_C(\mathcal{T}) \mid \mathcal{T} \models \bigsqcap_{B_i \in M} B_i \sqsubseteq A\}$, $\mathtt{Post}_{Base}(A) = \{A' \in$*

$sig_C(\mathcal{T}) \cup \{\top\} \mid \mathcal{T} \models A \sqsubseteq A'\} \cup \{\exists r.A' \mid A' \in sig_C(\mathcal{T}) \cup \{\top\}, \mathcal{T} \models A \sqsubseteq \exists r.A', r \in \Sigma\}$ *and* $\texttt{Post}(A) = 2^{\texttt{Post}_{Base}(A)}$. *Further, for each* $B \in sig_C(\mathcal{T})$, *let* $R^{\sqsupseteq}$ *be given by*

(GL1) $\mathfrak{n}_B \to B$ *if* $B \in \Sigma$,

(GL2) $\mathfrak{n}_B \to \mathfrak{n}_{B'}$ *for all* $\{B'\} \in \texttt{Pre}(B)$,

(GL3) $\mathfrak{n}_B \to \sqcap(\mathfrak{n}_{B'_1}, ..., \mathfrak{n}_{B'_n})$ *for all* $\{B'_1, ..., B'_n\} \in \texttt{Pre}(B)$ *with* $n \geq 1$,

(GL4) $\mathfrak{n}_B \to \exists r(\mathfrak{n}_{B'})$ *for all* B' *with* $B \equiv \exists r.B' \in \mathcal{T}$ *and* $r \in sig_R(\mathcal{T}) \cap \Sigma$.

Let $R^{\sqsubseteq}$ *be given for all* $B \in sig_C(\mathcal{T}) \cup \{\top\}$ *by*

(GR1) $\mathfrak{n}_B \to B$ *if* $B \in \Sigma \cup \{\top\}$,

(GR2) $\mathfrak{n}_B \to \sigma(C)$ *for all* $\{C\} \in \texttt{Post}(B)$,

(GL3) $\mathfrak{n}_B \to \sqcap(\sigma(C'_1), ..., \sigma(C'_n))$ *for all* $\{C'_1, ..., C'_n\} \in \texttt{Post}(B)$ *with* $n \geq 1$.

For each $A \in sig_C(\mathcal{T})$, *the regular tree grammar* $G^{\sqsupseteq}(\mathcal{T}, \Sigma, A)$ *is then given by* $(\mathfrak{n}_A, \mathcal{N}^{\mathcal{T}}, \mathcal{F}, R^{\sqsupseteq})$, *and the regular tree grammar* $G^{\sqsubseteq}(\mathcal{T}, \Sigma, A)$ *is given by* $(\mathfrak{n}_A, \mathcal{N}^{\mathcal{T}}, \mathcal{F}, R^{\sqsubseteq})$.

We denote the set of tree grammars $\{G^{\sqsupseteq}(\mathcal{T}, \Sigma, A) \mid A \in \text{sig}_C(\mathcal{T})\}$ by $\mathbb{G}^{\sqsupseteq}(\mathcal{T}, \Sigma)$ and the set $\{G^{\sqsubseteq}(\mathcal{T}, \Sigma, A) \mid A \in \text{sig}_C(\mathcal{T})\}$ by $\mathbb{G}^{\sqsubseteq}(\mathcal{T}, \Sigma)$.
Since $\text{sig}(\mathcal{T})$ is finite, all elements of `Pre` and `Post` can be effectively computed. For the construction of grammars the following result holds.

Theorem 2. *Let* $\mathcal{T}$ *be a normalized* $\mathcal{EL}$ *TBox,* Σ *a signature.* $\mathbb{G}^{\sqsupseteq}(\mathcal{T}, \Sigma)$ *and* $\mathbb{G}^{\sqsubseteq}(\mathcal{T}, \Sigma)$ *can be computed from* $\mathcal{T}$ *in exponential time and are exponentially bounded in the size of* $\mathcal{T}$.

Proof. The exponentially bounded size and time hold due to the exponential number of elements in `Pre` and `Post` and tractable reasoning in $\mathcal{EL}$ [BAADER et al. 2005]. □

The following example demonstrates the grammar construction.

Example 9. *For $\mathcal{T}$ and Σ from Example 6, we obtain a normalized TBox $\mathcal{T}' = \{A' \sqsubseteq A, A \sqsubseteq A'', A \sqsubseteq B, B \equiv \exists r.A, B' \equiv \exists s.A, B' \sqsubseteq A\}$, which yields $\texttt{Pre} = \{(A, 2^{\{A',B'\}}), (A'', 2^{\{A',B',A\}}), (A', \{\}), (B, 2^{\{A',A\}}), (B', \{\})\}$, $\texttt{Post}_{Base} = \{(A, \{A'', B, \top, \exists r(\mathfrak{n}_A), \exists r(\mathfrak{n}_\top)\}), (A', \{A, A'', B, \top, \exists r(\mathfrak{n}_A), \exists r(\mathfrak{n}_\top)\}), (B, \{\top, \exists r(\mathfrak{n}_A), \exists r(\mathfrak{n}_\top)\}), (A'', \{\top\}), (B', \{A'', A, \top, \exists s(\mathfrak{n}_A), \exists s(\mathfrak{n}_\top)\})\}$ and the following set of transitions for $R^{\sqsupseteq}$:*

$$\mathfrak{n}_B \to \mathfrak{n}_A$$

$$\begin{array}{rr}
\mathfrak{n}_{A''} \to \mathfrak{n}_{A'} & \mathfrak{n}_A \to \mathfrak{n}_{B'} \\
\mathfrak{n}_{A''} \to \mathfrak{n}_A & \mathfrak{n}_A \to \mathfrak{n}_{A'} \\
\mathfrak{n}_{A''} \to \mathfrak{n}_{B'} & \mathfrak{n}_B \to \mathfrak{n}_{A'} \\
\mathfrak{n}_{A''} \to A'' & \mathfrak{n}_{A'} \to A' \\
\mathfrak{n}_{B'} \to \exists s(\mathfrak{n}_A) & \mathfrak{n}_B \to \exists r(\mathfrak{n}_A) \\
\mathfrak{n}_A \to \sqcap(\mathfrak{n}_{A'}, \mathfrak{n}_{B'}) & \mathfrak{n}_B \to \sqcap(\mathfrak{n}_{A'}, \mathfrak{n}_A) \\
\mathfrak{n}_{A''} \to \sqcap(\mathfrak{n}_{A'}, \mathfrak{n}_A) & \mathfrak{n}_{A''} \to \sqcap(\mathfrak{n}_A, \mathfrak{n}_{B'}) \\
\mathfrak{n}_{A''} \to \sqcap(\mathfrak{n}_{A'}, \mathfrak{n}_{B'}) & \mathfrak{n}_{A''} \to \sqcap(\mathfrak{n}_A, \mathfrak{n}_{A'}, \mathfrak{n}_{B'})
\end{array}$$

For $R^{\sqsubseteq}$, we obtain $\mathfrak{n} \to \mathfrak{n}_\top$ for all $\mathfrak{n} \in \mathcal{N}$ and

$$\begin{array}{rr}
\mathfrak{n}_{A''} \to A'' & \mathfrak{n}_\top \to \top \\
\mathfrak{n}_{A'} \to A' & \mathfrak{n}_{A'} \to \mathfrak{n}_B \\
\mathfrak{n}_A \to \mathfrak{n}_{A''} & \mathfrak{n}_{A'} \to \mathfrak{n}_A \\
\mathfrak{n}_A \to \mathfrak{n}_B & \mathfrak{n}_{A'} \to \mathfrak{n}_{A''} \\
\mathfrak{n}_{B'} \to \mathfrak{n}_A & \mathfrak{n}_{B'} \to \mathfrak{n}_{A''} \\
\mathfrak{n}_{B'} \to \exists s(\mathfrak{n}_A) & \mathfrak{n}_B \to \exists r(\mathfrak{n}_A) \\
\mathfrak{n}_A \to \exists r(\mathfrak{n}_A) & \mathfrak{n}_{A'} \to \exists r(\mathfrak{n}_A) \\
\mathfrak{n}_{B'} \to \exists s(\mathfrak{n}_\top) & \mathfrak{n}_B \to \exists r(\mathfrak{n}_\top) \\
\mathfrak{n}_A \to \exists r(\mathfrak{n}_\top) & \mathfrak{n}_{A'} \to \exists r(\mathfrak{n}_\top)
\end{array}$$

Additionally, $R^{\sqsubseteq}$ contains rules for conjunctions of all elements of $\texttt{Post}_{Base}$ corresponding to (**GR3**)*, which we do not give for space reasons.*

By applying the rules $\mathfrak{n}_A \rightarrow \mathfrak{n}_{B'}, \mathfrak{n}_{B'} \rightarrow \exists s(\mathfrak{n}_A)$ contained in $R^{\sqsupseteq}$ n times, we obtain a term $\exists s(\exists s(...\exists s(A)))$ of depth n, which represents the corresponding subsumee of A of the same depth.

Grammar Properties

The following theorem states that the grammars derive only terms representing Σ-subsumees and Σ-subsumers of the corresponding atomic concept.

Theorem 3. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ TBox, Σ a signature and $A \in sig_C(\mathcal{T})$.*

1. *For each $t \in L(G^{\sqsupseteq}(\mathcal{T}, \Sigma, A))$, there is a concept C with $t_C = t$ and $sig(C) \subseteq \Sigma$ such that $\mathcal{T} \models C \sqsubseteq A$.*
2. *For each $t \in L(G^{\sqsubseteq}(\mathcal{T}, \Sigma, A))$, there is a concept C with $t_C = t$ and $sig(C) \subseteq \Sigma$ such that $\mathcal{T} \models A \sqsubseteq C$.*

Proof. The theorem is proved by an easy induction on the maximal nesting depth of functions in t using the rules given in Definition 14. It is easy to check in Definition 14 that the grammars derive only terms containing atomic concepts and roles from Σ, since $\mathfrak{n}_B \rightarrow B$ only if $B \in \Sigma$ and $\mathfrak{n}_B \rightarrow \exists r(t)$ only if $r \in \Sigma$. Therefore, for any $A \in \text{sig}_C(\mathcal{T})$ and any $t_C \in L(G^{\sqsubseteq}(\mathcal{T}, \Sigma, A)) \cup L(G^{\sqsupseteq}(\mathcal{T}, \Sigma, A))$ holds $\text{sig}(C) \subseteq \Sigma$.

1. Let t be a term such that $t \in L(G^{\sqsupseteq}(\mathcal{T}, \Sigma, A))$. We prove the theorem by induction on the maximal nesting depth of functions in t.
   - Assume that t is an atomic concept B. B can only be derived from $\mathfrak{n}_A$ by n empty transitions (GL2), and, once $\mathfrak{n}_B$ is reached, the rule (GL1). Let $B_1, ..., B_n$ be such that $\mathfrak{n}_A \rightarrow \mathfrak{n}_{B_1} \rightarrow ... \rightarrow \mathfrak{n}_{B_n} \rightarrow \mathfrak{n}_B$. Then, by Definition 14, for each pair B_i, B_{i+1} holds $\mathcal{T} \models B_i \sqsupseteq B_{i+1}$, for B_n, B holds $\mathcal{T} \models B_n \sqsupseteq B$ and for A, B_1 holds $\mathcal{T} \models A \sqsupseteq B_1$. It follows that also $\mathcal{T} \models A \sqsupseteq B$, while $t = t_B$.
   - Assume that $t = \exists r(t')$ for some term t'. Then, the derivation of t from $\mathfrak{n}_A$ starts with n empty transitions (GL2) such that $\mathfrak{n}_{B'}$ for some $B' \in \text{sig}_C(\mathcal{T})$ is reached, and a subsequent application of (GL4) such

that $\mathfrak{n}_B$ for some $B \in \text{sig}_C(\mathcal{T})$ is reached. As argued above about the applications of empty transitions, $\mathcal{T} \models A \sqsupseteq B'$ holds. Moreover, By Definition 14 (GL4) holds $B' \equiv \exists r.B \in \mathcal{T}$, and, therefore, $\mathcal{T} \models A \sqsupseteq \exists r.B$. Let C' be a concept with $t' = t_{C'}$. Then, the theorem holds for C' and $\mathfrak{n}_B$ by induction hypothesis, i.e., $\mathcal{T} \models B \sqsupseteq C'$. Therefore, $\mathcal{T} \models A \sqsupseteq \exists r.C'$, while $t = t_{\exists r.C'}$.

- Assume that $t = \sqcap(t_1, ..., t_n)$ for a set of terms $t_1, ..., t_n$. Then, the derivation of t from $\mathfrak{n}_A$ starts with n empty transitions (GL2) such that $\mathfrak{n}_{B'}$ for some $B' \in \text{sig}_C(\mathcal{T})$ is reached, and a subsequent application of (GL3) such that, for a set of concepts $B_i \in \text{sig}_C(\mathcal{T})$ with $1 \leq i \leq n$ and $t_i \in L(G^{\sqsupseteq}(\mathcal{T}, \Sigma, \mathfrak{n}_{B_i}))$, $\mathfrak{n}_{B_i}$ is reached. As argued above about the applications of empty transitions, $\mathcal{T} \models A \sqsupseteq B'$ holds. Let C_i be a concept with $t_i = t_{C_i}$. By induction hypothesis, $\mathcal{T} \models B_i \sqsupseteq C_i$. By Definition 14, $\mathcal{T} \models B' \sqsupseteq B_1 \sqcap ... \sqcap B_n$. Therefore, $\mathcal{T} \models B' \sqsupseteq C_1 \sqcap ... \sqcap C_n$ and $\mathcal{T} \models A \sqsupseteq C_1 \sqcap ... \sqcap C_n$ with $t = t_{C_1 \sqcap ... \sqcap C_n}$.

2. The proof of soundness of $\mathbb{G}^{\sqsubseteq}(\mathcal{T}, \Sigma))$ can be done in the same manner. Let t be a term such that $t \in L(G^{\sqsubseteq}(\mathcal{T}, \Sigma, A))$. We prove the theorem by induction on the maximal nesting depth of functions in t.

   - Assume that t is an atomic concept B. B can only be derived from $\mathfrak{n}_A$ by n empty transitions (GR2), and, once $\mathfrak{n}_B$ is reached, the rule (GR1). Let $B_1, ..., B_n$ be such that $\mathfrak{n}_A \rightarrow \mathfrak{n}_{B_1} \rightarrow ... \rightarrow \mathfrak{n}_{B_n} \rightarrow \mathfrak{n}_B$. Then, by Definition 14, for each pair B_i, B_{i+1} holds $\mathcal{T} \models B_i \sqsubseteq B_{i+1}$, for B_n, B holds $\mathcal{T} \models B_n \sqsubseteq B$ and for A, B_1 holds $\mathcal{T} \models A \sqsubseteq B_1$. It follows that also $\mathcal{T} \models A \sqsubseteq B$ with $t = t_B$.

   - Assume that $t = \exists r(t')$ for some term t'. Then, the derivation of t from $\mathfrak{n}_A$ starts with n empty transitions (GR2) such that $\mathfrak{n}_{B'}$ for some $B' \in \text{sig}_C(\mathcal{T})$ is reached, and a subsequent application of a non-empty transition (GR2) such that $\exists r.\mathfrak{n}_B$ for some $B \in \text{sig}_C(\mathcal{T})$ is reached. As argued above about the applications of empty transitions, $\mathcal{T} \models A \sqsubseteq B'$ holds. Moreover, By Definition 14 holds $\mathcal{T} \models B' \sqsubseteq \exists r.B$, and, therefore, $\mathcal{T} \models A \sqsubseteq \exists r.B$. Let C' be a concept with $t' = t_{C'}$. By

induction hypothesis, $\mathcal{T} \models B \sqsubseteq C'$. Therefore, $\mathcal{T} \models A \sqsubseteq \exists r.C'$ with $t = t_{\exists r.C'}$.

- Assume that $t = \sqcap(t_1, ..., t_n)$ for a set of terms $t_1, ..., t_n$. Then, the derivation of t from $\mathfrak{n}_A$ starts with n empty transitions (GR2) such that $\mathfrak{n}_{B'}$ for some $B' \in \text{sig}_C(\mathcal{T})$ is reached, and a subsequent application of (GR2) such that, for a set of concepts $B_i \in \text{sig}_C(\mathcal{T})$ with $1 \leq i \leq n$, we reach $\sqcap(\sigma(C_1), ..., \sigma(C_n))$ where for each i holds either $C_i = B_i$ and $t_i \in L(G^{\sqsubseteq}(\mathcal{T}, \Sigma, \mathfrak{n}_{B_i}))$ or $C_i = \exists r.B_i$ and $t'_i \in L(G^{\sqsubseteq}(\mathcal{T}, \Sigma, \mathfrak{n}_{B_i}))$ for $t_i = \exists r.t'_i$. By induction hypothesis, for each B_i there is a concept C'_i with $t_{C'_i} = t_i$ in case $C_i = B_i$ and $t_{C'_i} = t'_i$, otherwise, such that $\mathcal{T} \models B_i \sqsubseteq C'_i$. Since, for each C_i, by Definition 14 holds $\mathcal{T} \models B' \sqsubseteq C_i$, we obtain a concept C' by replacing each B_i with C'_i such that $\mathcal{T} \models B' \sqsubseteq C'$, and $t_{C'} \in L(G^{\sqsubseteq}(\mathcal{T}, \Sigma, \mathfrak{n}_{B'}))$. Therefore, also $\mathcal{T} \models A \sqsubseteq C'$, and $t_{C'} \in L(G^{\sqsubseteq}(\mathcal{T}, \Sigma, \mathfrak{n}_A))$. □

□

As discussed above, subsumee grammars do not guarantee to capture weak subsumees. Therefore, we obtain the following result for the completeness of the grammars.

Theorem 4. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ TBox, Σ a signature and $A \in sig_C(\mathcal{T})$.*

1. *For each C with $sig(C) \subseteq \Sigma$ such that $\mathcal{T} \models C \sqsubseteq A$ there is a concept C' such that C can be obtained from C' by adding arbitrary conjuncts to arbitrary sub-expressions and $t_{C'} \in L(G^{\sqsupseteq}(\mathcal{T}, \Sigma, A))$.*

2. *For each D with $sig(D) \subseteq \Sigma$ such that $\mathcal{T} \models A \sqsubseteq D$ holds: $t_D \in L(G^{\sqsubseteq}(\mathcal{T}, \Sigma, A))$.*

Proof. The theorem is proved by induction on the role depth of C using the properties of the normalization, for instance, stated in Lemmas 13, in addition to Definition 14. Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ TBox, Σ a signature and $A \in \text{sig}_C(\mathcal{T})$.

W.l.o.g., we can assume that there is a concept C with

$$C = \bigsqcap_{1 \leq j \leq n} A_j \sqcap \bigsqcap_{1 \leq k \leq m} \exists r_k.E_k$$

with $A_j \in \Sigma$ for $1 \leq j \leq n$, $r_k \in \Sigma$ for $1 \leq k \leq m$ and E_k with $1 \leq k \leq m$ a set of $\mathcal{EL}$ concepts such that $\text{sig}(E_k) \subseteq \Sigma$. Further, w.l.o.g., we can assume that all A_j are pairwise different.

1. We show that, for each such general C with $\text{sig}(C) \subseteq \Sigma$ and $\mathcal{T} \models C \sqsubseteq A$, there is a concept C' such that C can be obtained from C' by weakening and $t_{C'} \in L(G^{\sqsupseteq}(\mathcal{T}, \Sigma, A))$. We prove the claim by induction of the role depth of C.

   - Assume role depth = 0. Then C is a conjunction of atomic concepts, i.e., $m = 0$ and $C = \bigsqcap_{1 \leq j \leq n} A_j$. Then, by Lemma 13, there is a set $M' \in \texttt{Pre}(A)$ of atomic concepts such that, for each $B \in M'$, there is an A_j with $A_j = B$. Therefore, each $B \in M'$ is in Σ. Let $C'_1 = \bigsqcap_{B \in M'} B$. Since $M' \subseteq \{A_1, ...A_n\}$, C can be obtained from C'_1 by weakening. By Definition 14 (GL3), there is a rule $\mathfrak{n}_A \to \sqcap(\mathfrak{n}_{B_1}, ..., \mathfrak{n}_{B_o})$ with $\{B_1, ..., B_o\} = M'$. Since each $B \in M'$ is in Σ, we obtain by (GL1) $\mathfrak{n}_B \to B$. Since our grammars operate on unordered trees, it follows that $\mathfrak{n}_A \to^+_{G^{\sqsupseteq}(\mathcal{T},\Sigma,A))} t_{C'_1}$, i.e., $t_{C'_1} \in L(G^{\sqsupseteq}(\mathcal{T}, \Sigma, A))$ for any order of conjuncts in C'_1. Therefore, the theorem holds with $C' = C'_1$.
   - Assume that the role depth is greater than 0. As in the case above, there is a set $M' \in \texttt{Pre}(A)$ of atomic concepts such that, for each $B \in M'$, [A1] or [A2] holds. Let $M'_1 = M' \cap \{A_1, ...A_n\}$ and $M'_2 = M' \setminus M'_1$. Let $C'_1 = \bigsqcap_{B \in M'_1} B$, and $C'_2 = \bigsqcap_{1 \leq f \leq p} \exists r'_f.E'_f$ with $\{\exists r'_1.E'_1, ..., \exists r'_p.E'_p\} = \{\exists r.E \mid$ for one of $B \in M'_2$ holds [A2] such that there exists $B' \in \text{sig}_C(\mathcal{T})$ with $\mathcal{T} \models E \sqsubseteq B'$ and $B \equiv \exists r.B' \in \mathcal{T}\}$. Clearly,$C$ can be obtained from $C'_1 \sqcap C'_2$ by weakening. By Definition 14 (GL3), there is a rule $\mathfrak{n}_A \to \sqcap(\mathfrak{n}_{B_1}, ..., \mathfrak{n}_{B_o})$ with $\{B_1, ..., B_o\} = M'$. Moreover, for all $B \in M'_1$ holds $\mathfrak{n}_B \to B$ and for all $B_f \in M'_2$, there is $\exists r'_f.E'_f$ such that there exists $B'_f \in \text{sig}_C(\mathcal{T})$

with $\mathcal{T} \models E'_f \sqsubseteq B'_f$ and $B_f \equiv \exists r'_f.B'_f \in \mathcal{T}$. By Definition 14 (GL4), $\mathfrak{n}_{B_f} \to \exists r'_f(\mathfrak{n}_{B'_f})$. By induction hypothesis, there is a concept E''_f such that $\mathfrak{n}_{B'_f} \to^+_{G^{\sqsupseteq}(\mathcal{T},\Sigma,A))} t_{E''_f}$ and E'_f can be obtained from E''_f by weakening. Therefore, $\mathfrak{n}_{B_f} \to^+_{G^{\sqsupseteq}(\mathcal{T},\Sigma,A))} \exists r'_f(t_{E''_f})$ and $\exists r'_f.E'_f$ can be obtained from $\exists r'_f.E''_f$ by weakening. Let $C''' = C'_1 \sqcap \bigsqcap_{B_f \in M'_2} \exists r'_f.E''_f$. Then, C can be obtained from C''' by weakening. Since our grammars operate on unordered trees, we obtain $\mathfrak{n}_A \to^+_{G^{\sqsupseteq}(\mathcal{T},\Sigma,A))} t_{C'''}$, i.e., $t_{C'''} \in L(G^{\sqsupseteq}(\mathcal{T},\Sigma,A))$ for any order of conjuncts. Therefore, the theorem holds with $C' = C'''$.

2. We proceed with showing that for each such general C with $\text{sig}(C) \subseteq \Sigma$ and $\mathcal{T} \models A \sqsubseteq C$ holds: $t_C \in L(G^{\sqsubseteq}(\mathcal{T},\Sigma,A))$. We prove the claim by induction of the role depth of C. For each A_j, we know that $\mathcal{T} \models A \sqsubseteq A_j$ and $A_j \in \Sigma$, i.e., $A_j \in \mathtt{Post}_{\text{Base}}(A)$. By Definition 14 (GR2) or (GR3), $\mathfrak{n}_{A_j} \to A_j$ for all A_j and $\mathfrak{n}_A \to \sqcap(\mathfrak{n}_{A_1},...,\mathfrak{n}_{A_n})$, and, therefore, $t_C \in L(G^{\sqsubseteq}(\mathcal{T},\Sigma,A))$. Assume a role depth > 0. For each $\exists r_k.E_k$, it follows from Lemma 14 that there are $B_1, B_2 \in \text{sig}_C(\mathcal{T})$ with $B_1 \equiv \exists r_k.B_2 \in \mathcal{T}$ such that $\mathcal{T} \models A \sqsubseteq B_1$, $\mathcal{T} \models B_2 \sqsubseteq E_k$. Since $r_k \in \Sigma$, follows that $\exists r_k.B_2 \in \mathtt{Post}_{\text{Base}}(A)$. Moreover, by induction hypothesis follows that $t_{E_k} \in L(G^{\sqsubseteq}(\mathcal{T},\Sigma,B_2))$. An application of (GR2) or (GR3) in Definition 14 yields $t_C \in L(G^{\sqsubseteq}(\mathcal{T},\Sigma,A))$. □

From Grammars to Uniform Interpolants

Now we show that, as a consequence of Lemma 17 and Theorem 4, in case a finite uniform interpolant exists, we can construct it from the subsumees and subsumers of maximal depth $N = 2^{4\cdot(|\text{sig}_C(\mathcal{T})|+\text{def}(\mathcal{T}))} + 1$ generated by the grammars $\mathbb{G}^{\sqsupseteq}(\mathcal{T},\Sigma), \mathbb{G}^{\sqsubseteq}(\mathcal{T},\Sigma)$. Note that, if all subsumees and subsumers are using only concepts and roles from Σ (follows from Theorem 3), $\text{sig}(\mathtt{M}(L_{\sqsupseteq}, L_{\sqsubseteq}, \Sigma)) \subseteq \Sigma$. We obtain the following result concerning the size of uniform $\mathcal{EL}$ Σ-interpolants of $\mathcal{T}$.

Theorem 5. *Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox and Σ a signature. For $N = 2^{4\cdot(|sig_C(\mathcal{T})|+def(\mathcal{T}))} + 1$, $\bowtie \in \{\sqsupseteq, \sqsubseteq\}$ and $A \in sig_C(\mathcal{T})$, let $L_{\bowtie}(A) = \{C \mid t_C \in L(G^{\bowtie}(\mathcal{T},\Sigma,A)), d(C) \leq N\}$. The following statements are equivalent:*

1. *There exists a uniform $\mathcal{EL}$ Σ-interpolant of $\mathcal{T}$.*
2. $\mathtt{M}(L_{\sqsupseteq}, L_{\sqsubseteq}, \Sigma) \equiv^{c}_{\Sigma} \mathcal{T}$
3. *There exists a uniform $\mathcal{EL}$ Σ-interpolant $\mathcal{T}'$ with $|\mathcal{T}'| \in O(2^{2^{2^{|\mathcal{T}|}}})$.*

Proof. The non-trivial parts of the proof are implications $1 \Rightarrow 2$ and $2 \Rightarrow 3$. For convenience, let $\mathcal{T}_\Sigma$ denote the TBox $\mathtt{M}(L_{\sqsupseteq}, L_{\sqsubseteq}, \Sigma)$.

$1 \Rightarrow 2$**:** By Definition 7, the statement $\mathcal{T}_\Sigma \equiv^{c}_{\Sigma} \mathcal{T}$ consists of two directions: (1) for all $\mathcal{EL}$ concepts C, D with $\text{sig}(C) \cup \text{sig}(D) \subseteq \Sigma$ holds $\mathcal{T}_\Sigma \models C \sqsubseteq D \Rightarrow \mathcal{T} \models C \sqsubseteq D$ and (2) for all $\mathcal{EL}$ concepts C, D with $\text{sig}(C) \cup \text{sig}(D) \subseteq \Sigma$ holds $\mathcal{T}_\Sigma \models C \sqsubseteq D \Leftarrow \mathcal{T} \models C \sqsubseteq D$.

(1) The first direction follows from Theorem 3 and Definition 9, which does not introduce any consequences not being consequences of $\mathcal{T}$.

(2) For the second direction, assume that there exists a uniform $\mathcal{EL}$ Σ-interpolant of $\mathcal{T}$. Then, by Lemma 17, there exists a uniform $\mathcal{EL}$ Σ-interpolant $\mathcal{T}'$ of $\mathcal{T}$ with $d(\mathcal{T}') \leq N$. It is sufficient to show that for each $C \sqsubseteq D \in \mathcal{T}'$ holds $\mathcal{T}_\Sigma \models C \sqsubseteq D$. Assume that $C \sqsubseteq D \in \mathcal{T}'$. Then, $\mathcal{T} \models C \sqsubseteq D$ and we prove by induction on maximal role depth of C, D that also $\mathcal{T}_\Sigma \models C \sqsubseteq D$. W.l.o.g., let $D = \bigsqcap_{1 \leq i \leq l} D_i$ and

$$C = \bigsqcap_{1 \leq j \leq n} A_j \sqcap \bigsqcap_{1 \leq k \leq m} \exists r_k.E_k$$

with $A_j \in \Sigma \cap \text{sig}_C(\mathcal{T})$ for $1 \leq j \leq n$, $r_k \in \Sigma \cap \text{sig}_R(\mathcal{T})$ for $1 \leq k \leq m$ and E_k with $1 \leq k \leq m$ a set of $\mathcal{EL}$ concepts such that $\text{sig}(E_k) \subseteq \Sigma$. Clearly, $\mathcal{T} \models C \sqsubseteq D$, iff $\mathcal{T} \models C \sqsubseteq D_i$ for all i with $1 \leq i \leq l$.

- If $D_i = A \in \Sigma$, then, it follows from Theorem 4 that there is a concept C' such that C can be obtained from C' by adding arbitrary conjuncts to arbitrary sub-expressions with $t_{C'} \in L(G^{\sqsupseteq}(\mathcal{T}, \Sigma, A))$. Since $d(C) \leq N$ and C has been obtained from C' by weakening, also $d(C') \leq N$. Therefore, $t_{C'} \in L_{\sqsupseteq}(A)$, and $\mathcal{T}_\Sigma \models C \sqsubseteq D_i$.

- If $D_i = \exists r.D'$ for some r, D', then, by Lemma 13, one of the following is true:

(A3) There are r_k, E_k in C such that $r_k = r$ and $\mathcal{T} \models E_k \sqsubseteq D'$. Since $d(E_k) < N$ and $d(D') < N$, by induction hypothesis holds $\mathcal{T}_\Sigma \models E_k \sqsubseteq D'$. It follows that $\mathcal{T}_\Sigma \models \exists r_k.E_k \sqsubseteq D_i$ and $\mathcal{T}_\Sigma \models C \sqsubseteq D_i$.

(A4) There is $B \in \text{sig}_C(\mathcal{T})$ of $\mathcal{T}$ such that $\mathcal{T} \models B \sqsubseteq \exists r.D'$ and $\mathcal{T} \models C \sqsubseteq B$. Then,

- it follows from Theorem 4 that there is a concept C' such that C can be obtained from C' by adding arbitrary conjuncts to arbitrary sub-expressions with $t_{C'} \in L(G^{\sqsupseteq}(\mathcal{T}, \Sigma, B))$. Since $d(C) \leq N$ and C has been obtained from C' by weakening, also $d(C') \leq N$. Therefore, $t_{C'} \in L_{\sqsupseteq}(B)$
- it follows from Theorem 4 that $t_{\exists r.D'} \in L(G^{\sqsubseteq}(\mathcal{T}, \Sigma, B))$. Since $d(\exists r.D') \leq N$, it follows that $t_{\exists r.D'} \in L_{\sqsubseteq}(B)$.

Therefore, by Definition 9, $\mathcal{T}_\Sigma \models C' \sqsubseteq \exists r.D'$, and $\mathcal{T}_\Sigma \models C \sqsubseteq D_i$.

$2 \Rightarrow 3$**:** Observe that $\mathbb{G}_1, \mathbb{G}_2$ have $|\text{sig}_C(\mathcal{T})|$ non-terminals and at most $2^{2 \cdot n} + |\text{sig}_C(\mathcal{T})|$ outgoing transitions for each non-terminal, n the maximal arity of $\sqcap$, each of which has at most n occurring non-terminals. Now we consider the stepwise generation of the languages $L_{\sqsupseteq}(A)$ and $L_{\sqsubseteq}(A)$ for an arbitrary A by the grammars $\mathbb{G}_1, \mathbb{G}_2$ in order to find an upper bound for the size of the two languages. In order to do so, we identify the upper bound for the set of all generated (ground and unground) terms of depth N. In each step i, the set of generated terms is extended with the set R_i of all terms obtained by replacing non-terminals on the right-hand side of rules by the corresponding right-hand sides from $\mathbb{G}_1, \mathbb{G}_2$. Note that, after N steps, the corresponding sets of generated terms contain $L_{\sqsupseteq}(A)$ and $L_{\sqsubseteq}(A)$. Let $\texttt{leaves}_i$ be the maximal number of non-terminals occurring in a transition after step i and $\texttt{tran}_i$ the maximal number of outgoing transitions for a non-terminal after step i. Then, $\texttt{tran}_0 = 2^{2 \cdot n} + |\text{sig}_C(\mathcal{T})|$ and $\texttt{leaves}_0 = n$.

Further, $\texttt{leaves}_{i+1} = n \cdot \texttt{leaves}_i$, i.e., $\texttt{leaves}_i = n^{i+1}$. For each non-terminal, there are at most $2^{2 \cdot n} + |\text{sig}_C(\mathcal{T})|$ possible replacing transitions, therefore, for each $t \in R_i$, there are $(2^{2 \cdot n} + |\text{sig}_C(\mathcal{T})|)^{\texttt{leaves}_{i+1}}$ possibilities to replace all non-terminals $\mathfrak{n} \in \mathcal{N}$ by the corresponding transitions from R_0. We obtain $\texttt{tran}_{i+1} = \texttt{tran}_i \cdot (2^{2 \cdot n} + |\text{sig}_C(\mathcal{T})|)^{\texttt{leaves}_{i+1}}$, i.e., $\texttt{tran}_i \leq (2^{2 \cdot n} + |\text{sig}_C(\mathcal{T})|)^{i \cdot n^{i+2}}$. For $i = N$, we obtain $\texttt{leaves}_i = n^N \in O(2^{2^{|\mathcal{T}|}})$ and $\texttt{tran}_i \leq (2^{2 \cdot n} + |\text{sig}_C(\mathcal{T})|)^{(N) \cdot n^{N+2}} \in O(2^{2^{2^{|\mathcal{T}|}}})$.

These complexity results correspond to the size and number of axioms in Example 10 used to demonstrate the triple-exponential lower bound. □

5.2.2 Lower Bound

It is interesting that, while deciding the existence of uniform interpolants in $\mathcal{EL}$ [LUTZ et al. 2012] is one exponential less complex than the same decision problem for the more complex logic $\mathcal{ALC}$ [LUTZ and WOLTER 2011], the size of uniform interpolants remains triple-exponential due to the unavailability of disjunction. We demonstrate that this is in fact the lower bound by the means of the following example (obtained by a slight modification of the corresponding example given in [LUTZ and WOLTER 2010] originally demonstrating a double exponential lower bound in the context of conservative extensions).

Example 10. *The $\mathcal{EL}$ TBox $\mathcal{T}_n$ for a natural number n is given by*

$$A_1 \sqsubseteq \overline{X_0} \sqcap \ldots \sqcap \overline{X_{n-1}} \tag{5.1}$$

$$A_2 \sqsubseteq \overline{X_0} \sqcap \ldots \sqcap \overline{X_{n-1}} \tag{5.2}$$

$$\sqcap_{\sigma \in \{r,s\}} \exists \sigma.(\overline{X_i} \sqcap X_0 \sqcap \ldots \sqcap X_{i-1}) \sqsubseteq X_i \qquad i < n \tag{5.3}$$

$$\sqcap_{\sigma \in \{r,s\}} \exists \sigma.(X_i \sqcap X_0 \sqcap \ldots \sqcap X_{i-1}) \sqsubseteq \overline{X_i} \qquad i < n \tag{5.4}$$

$$\sqcap_{\sigma \in \{r,s\}} \exists \sigma.(\overline{X_i} \sqcap \overline{X_j}) \sqsubseteq \overline{X_i} \quad j < i < n \tag{5.5}$$

$$\sqcap_{\sigma \in \{r,s\}} \exists \sigma.(X_i \sqcap \overline{X_j}) \sqsubseteq X_i \quad j < i < n \tag{5.6}$$

$$X_0 \sqcap \ldots \sqcap X_{n-1} \sqsubseteq B \tag{5.7}$$

In the above TBox, Axiom 5.3 ensures that an unset bit will be set in the successor number, if all bits before it are already set. The subsequent axiom 5.4 ensures that

a set bit will be unset in the successor number, if all bits before it are also set. Axioms 5.5 and 5.6 ensure that in all other cases, bits do not switch. For instance, Axioms 5.5 states that, if any bit before bit i is unset yet, then bit i will remain unset also in the successor number.
If we now consider sets $\mathcal{C}_i$ of concept descriptions inductively defined by $\mathcal{C}_0 = \{A_1, A_2\}$,$\mathcal{C}_{i+1} = \{\exists r.C_1 \sqcap \exists s.C_2 \mid C_1, C_2 \in \mathcal{C}_i\}$, then we find that $|\mathcal{C}_{i+1}| = |\mathcal{C}_i|^2$ and consequently $|\mathcal{C}_i| = 2^{(2^i)}$. Thus, the set $\mathcal{C}_{2^n-1}$ contains triply exponentially many different concepts, each of which is doubly exponential in the size of $\mathcal{T}_n$ (intuitively, we obtain concepts having the shape of binary trees of exponential depth, thus having doubly exponentially many leaves, each of which can be endowed with A_1 or A_2, which gives rise to triply exponentially many different such trees). Then it can be shown that for each concept $C \in \mathcal{C}_{2^n-1}$ holds $\mathcal{T}_n \models C \sqsubseteq B$ and that there cannot be a smaller uniform interpolant with respect to the signature $\Sigma = \{A_1, A_2, B, r, s\}$ than the one containing all these GCIs.

Based on the above example, we now prove the following result.

Theorem 6. *There exists a sequence of $(\mathcal{T}_n)$ of $\mathcal{EL}$ TBoxes and a fixed signature Σ such that*

- *the size of $\mathcal{T}_n$ is upper-bounded by a polynomial in n and*
- *the size of the smallest uniform interpolant of $\mathcal{T}_n$ with respect to Σ is lower-bounded by $2^{(2^{(2^n-1)})}$.*

Proof. Obviously, the size of $\mathcal{T}_n$ is polynomially bounded by n. We now consider sets $\mathcal{C}_k$ of concept descriptions inductively defined by $\mathcal{C}_0 = \{A_1, A_2\}$ and $\mathcal{C}_{k+1} = \{\exists r.C_1 \sqcap \exists s.C_2 \mid C_1, C_2 \in \mathcal{C}_k\}$. We find that $|\mathcal{C}_{k+1}| = |\mathcal{C}_k|^2$ and consequently $|\mathcal{C}_k| = 2^{(2^k)}$. Thus, the set $\mathcal{C}_{2^n-1}$ contains triply exponentially many different concepts, each of which is doubly exponential in the size of $\mathcal{T}_n$.
Obviously, for any k, every concept description from $\mathcal{C}_k$ contains only signature elements from A_1, A_2, r, s.
It is rather straightforward to check that $\mathcal{T}_n \models C \sqsubseteq B$ holds for each concept $C \in \mathcal{C}_{2^n-1}$: by induction on k, we can show that for any $C \in \mathcal{C}_k$ with $k < 2^n$ holds $\mathcal{T}_n \models C \sqsubseteq Y_0^k \sqcap \ldots \sqcap Y_{n-1}^k$ with

$$Y_i^k = \begin{cases} X_i \text{ if } \lfloor \frac{k}{2^i} \rfloor \mathsf{mod}\ 2 = 1 \\ \overline{X_i} \text{ if } \lfloor \frac{k}{2^i} \rfloor \mathsf{mod}\ 2 = 0 \end{cases},$$

i.e., Y_i^k indicates the ith bit of the number k in binary encoding. Then, $C \sqsubseteq B$ follows via the last axiom of $\mathcal{T}_n$.

Toward the claimed triple-exponential lower bound, we now show that every uniform interpolant of $\mathcal{T}_n$ for $\Sigma = \{A_1, A_2, B, r, s\}$ must contain for each $C \in \mathcal{C}_{2^n-1}$ a GCI of the form $C \sqsubseteq B'$ with $B' = B$ or $B' = B \sqcap F$ for some F (where we consider structural variants – i.e., concept expressions which are equivalent with respect to the empty knowledge base – as syntactically equal). Toward a contradiction, we assume that this is not the case, i.e., there is a uniform interpolant $\mathcal{T}'$ and a $C \in \mathcal{C}_{2^n-1}$ where $C \sqsubseteq B' \notin \mathcal{T}'$ for any B' containing B as a conjunct.

Yet, since $C \sqsubseteq B$ must be a consequence of $\mathcal{T}'$, there must be a derivation of it. Looking at the derivation calculus from the last section, the last derivation step must be (ANDL) or (CUT). We can exclude (ANDL) since neither $\exists r.C' \sqsubseteq B$ nor $\exists s.C' \sqsubseteq B$ is the consequence of $\mathcal{T}'$ for any $C' \in \mathcal{C}_{2^n-2}$ (which can be easily shown by providing appropriate witness models of $\mathcal{T}'$). Consequently, the last derivation step must be an application of (CUT), i.e., there must be a concept $E \neq C$ such that $\mathcal{T}' \models C \sqsubseteq E$ and $\mathcal{T}' \models E \sqsubseteq B$. Without loss of generality, we assume that we consider a derivation where the branch of the derivation branch for $C \sqsubseteq E$ has minimal depth.

We now distinguish two cases: either E contains B as a conjunct or not.

- First we assume $E = E' \sqcap B$, i.e. the CUT rule was used to derive $C \sqsubseteq B$ from $C \sqsubseteq E' \sqcap B$ and $E' \sqcap B \sqsubseteq B$. The former cannot be contained in $\mathcal{T}'$ by assumption, hence it must have been derived itself. Again, it cannot have been derived via (ANDL) for the same reasons as given above, which again leaves (CUT) as the only possible derivation rule for obtaining $C \sqsubseteq E' \sqcap B$. Thus, there must be some concept G with $\mathcal{T}' \models C \sqsubseteq G$ and $\mathcal{T}' \models G \sqsubseteq E' \sqcap B$. Once more, we distinguish two cases: either G contains B as a conjunct or not.
  - If G contains B as a conjunct, i.e., $G = G' \sqcap B$, the derivation of $C \sqsubseteq E$ was not depth-minimal since there is a better proof where

$C \sqsubseteq B$ is derived from $C \sqsubseteq G' \sqcap B$ and $G' \sqcap B \sqsubseteq B$ via (CUT). Hence we have a contradiction.

- If G does not contain B as a conjunct, the original derivation of $C \sqsubseteq E$ was not depth-minimal since we can construct a better one that derives $C \sqsubseteq B$ directly from $C \sqsubseteq G$ and $G \sqsubseteq B$ (the latter being derived from $G \sqsubseteq E' \sqcap B$ via (ANDR)).

- Now assume E does not contain B as a conjunct.

  We construct $(\Delta, \cdot^{\mathcal{I}})$, the "characteristic interpretation" of C as follows (ϵ denoting the empty word):

  - $\Delta = \{w \mid w \in \{r, s\}^*,\ \text{length}(w) < 2^n\}$
  - We define an auxiliary function χ associating a concept expression to each domain element: we let $\chi(\epsilon) = C$ and for every $wr, ws \in \Delta$ with $\chi(w) = \exists r.C_1 \sqcap \exists s.C_2$, we let $\chi(wr) = C_1$ and $\chi(ws) = C_2$.
  - the concepts and roles are interpreted as follows:
    - $A_\iota^{\mathcal{I}} = \{w \mid \chi(w) = A_\iota\}$ for $\iota \in \{1, 2\}$
    - $B^{\mathcal{I}} = \{\epsilon\}$
    - $X_i^{\mathcal{I}} = \{w \mid \lfloor \frac{\text{length}(w)}{2^i} \rfloor \mathsf{mod}\ 2 = 0\}$ for $i < n$
    - $\overline{X_i}^{\mathcal{I}} = \{w \mid \lfloor \frac{\text{length}(w)}{2^i} \rfloor \mathsf{mod}\ 2 = 1\}$ for $i < n$
    - $r^{\mathcal{I}} = \{\langle w, wr\rangle \mid wr \in \Delta\}$
    - $s^{\mathcal{I}} = \{\langle w, ws\rangle \mid ws \in \Delta\}$

  It is straightforward to check that $\mathcal{I}$ is a model of $\mathcal{T}_n$ and that $\epsilon \in C^{\mathcal{I}}$. Consequently, due to our assumption, $\epsilon \in E^{\mathcal{I}}$ must hold. Yet then, by construction, E can only be a proper "structural superconcept" of C, i.e., $\emptyset \models C \sqsubseteq E$ and $\emptyset \not\models E \sqsubseteq C$ must hold.

  We now obtain $\widetilde{E}$ by enriching E as follows: recursively, for every subexpression G of E satisfying $\emptyset \models G \sqsubseteq C'$ for some $C' \in \mathcal{C}_k$ for some $k < 2^n$, we substitute G by $G \sqcap Y_0^k \sqcap \ldots \sqcap Y_{n-1}^k$. Then, $\widetilde{E}$ directly corresponds to a finite tree interpretation $\mathcal{I}'$ which is a model of $\mathcal{T}_n$ (following from structural induction on subexpressions of $\widetilde{E}$) and the root individual of which

satisfies $\widetilde{E}$ but not C (by assumption). Yet, the root individual cannot satisfy any other concept expression C'' from $\mathcal{C}_{2^n-1} \setminus \{C\}$ either, since this, via $\emptyset \models E \sqsubseteq C''$, would imply $\emptyset \models C \sqsubseteq C''$ which is not the case (by induction on k one can show that there cannot be a homomorphism between the associated tree interpretations of any two distinct concepts from any $\mathcal{C}_k$). In particular, we note that the root individual of $\mathcal{I}'$ also does not satisfy B. Thus, we have found a model of $\mathcal{T}_n$ witnessing $\mathcal{T}_n \not\models E \sqsubseteq B$, contradicting our assumption that $\mathcal{T}' \models E \sqsubseteq B$.

□

Hence we have found a class $\mathcal{T}_n$ of TBoxes giving rise to uniform interpolants of triple-exponential size in terms of the original TBox.

Both, the possibility of the non-existence as well as the triple-exponential lower bound, are negative results from practical point of view. There are various ways how to deal with these results in practical applications. For instance, expressing uniform interpolants in $\mathcal{EL}$ extended with fixpoint constructs [NIKITINA 2011] allows us to avoid both problems, the non-existence and the triple exponential blowup. This option is, however, not optimal in terms of usability, since fixpoint constructs are known to be found unintuitive by users. Further, there are some approaches to approximating interpolants, which, however, leads to a loss of relevant information.

In the next section, we propose a different solution. We consider three conflicting objectives for general module extraction: reducing the size of the extracted knowledge base, reducing the size of its signature and preserving the syntactic similarity of the extracted knowledge base with the originally given one. We demonstrate that, both, classical module extraction and uniform interpolation, assign an absolute priority to one of these objectives, thereby limiting the possibilities to achieve an improvement with respect to the other two. We show that general module extraction gains in effectiveness in terms of knowledge base size, when modules are neither required to use only the relevant set of entities nor required to be subsets of the original knowledge base. We present an alternative, tractable approach to general module extraction preserving all relevant consequences based on a different prioritization of objectives.

5.3 Hybrid Module Extraction

In the last sections, we discussed classical module extraction and uniform interpolation as two possible approaches to general module extraction. However, there are many practical scenarios, in which both approaches are of a limited use. First, the complexity results for both approaches are not very promising: even for the lightweight logic $\mathcal{EL}$, the task of minimal module extraction is EXPTIME-hard and the task of uniform interpolation is even 3EXPTIME-hard with a tight triple-exponential bound on the size of uniform interpolants in case a finite result exists [NIKITINA and RUDOLPH 2012]. Given that most applications of this non-standard reasoning task in ontology engineering, including relevance-based revision, are of particular interest for large ontologies and that there are scenarios, in which long computation times are not feasible due to user interaction, tractable approaches computing a small but not necessarily minimal solution would often be a reasonable alternative. Moreover, both types of approaches are based on a specific prioritization of objectives that might be necessary in particular scenarios, but is disadvantageous in many others due to its negative impact on the size of the extracted general modules. In this section, we consider the following requirements for the task of general module extraction:

1. **Syntactic Similarity**: In scenarios, where the ontology is intended to be used by human experts, the syntactic structure of the module determining its comprehensiveness or cognitive complexity has to be taken into account. The extent, to which a general module has to be syntactically similar to the original ontology $\mathcal{T}$ depends on the particular application requirements. For instance, modules can be required to be a subset of $\mathcal{T}$, to consist only of sub-expressions occurring in $\mathcal{T}$ or to consist only of concepts structurally equivalent to sub-expressions occurring in $\mathcal{T}$, but possibly referencing different atomic concepts.

2. **Small Knowledge Base Size**: Reducing the size of the ontology is a core objective for the task of general module extraction, since smaller ontologies (assuming that the particular syntactic similarity requirement is fulfilled in

both cases) require less computational and manual effort in many different ontology management activities.

3. **Small Signature Size**: Decreasing the size of the signature results in a decrease of irrelevant entities occurring in the ontology, which is also one of the core objectives of general module extraction.

While uniform interpolation clearly prioritizes small signature size making no compromises with respect to the other two requirements, minimal module extraction requires a very strong notion of syntactic similarity by not allowing for rewriting and, therefore, limiting the possibilities to reduce the size of both, the signature and the module. While such uncompromising prioritization can be required in some particular scenarios, in other scenarios it leads to a disadvantage. General module extraction for the TBox given in Example 5 and the signature $\Sigma = \{A_1, A_8, A_{12}, A_{15}, A_{16}, r\}$ demonstrates the drawbacks of minimal module extraction and uniform interpolation in terms of ontology size caused by the extreme choice of priorities. While minimal module extraction would return the whole ontology, uniform interpolation fails to extract a finite ontology due to the cyclic dependency given by $A_9 \sqsubseteq \exists r.A_9$. However, if we are not restricted to subsets of $\mathcal{T}$, but are also interested in modules consisting of sub-expressions occurring in $\mathcal{T}$, then there is a representation of the relevant information about Σ, which uses half as many axioms as the original TBox: $\{A_{12} \sqsubseteq A_{10}$, $A_{15} \sqsubseteq A_{13}$, $A_{10} \sqcap A_{13} \sqsubseteq A_{16}$, $A_{10} \sqsubseteq A_9$, $A_{13} \sqsubseteq A_9$, $A_9 \sqsubseteq \exists r.A_9$, $A_9 \sqsubseteq A_8$, $A_8 \sqsubseteq A_1\}$. If we are, additionally, allowed to exchange atomic concepts within sub-expressions while leaving the structure of expressions unchanged, then there is an even smaller representation consisting of 6 axioms: $\{A_{12} \sqcap A_{15} \sqsubseteq A_{16}$, $A_{12} \sqsubseteq A_9$, $A_{15} \sqsubseteq A_9$, $A_8 \sqsubseteq A_1$, $A_9 \sqsubseteq A_8$, $A_9 \sqsubseteq \exists r.A_9\}$.
The following example completes the picture that has been roughly sketched above and demonstrates the effect of unrestricted rewriting aiming at signature reduction on the module size.

Example 11. *The following TBox $\mathcal{T}$ models a "counter" with numbers $X_0, \ldots, X_{10}$, where the lowest number X_0 has two subsumees:*

$$A_1 \sqsubseteq X_0 \qquad A_2 \sqsubseteq X_0 \qquad \exists r.X_i \sqcap \exists s.X_i \sqsubseteq X_{i+1} \quad 0 \leq i \leq 9$$

Given this TBox, we could extract a ontology not referencing a particular atomic concept by replacing its occurrence by its direct subsumees. For instance, if we want to represent the information without using X_1, we can omit $\exists r.X_0 \sqcap \exists s.X_0 \sqsubseteq X_1$ and replace X_1 on the left-hand side of the remaining axioms by its direct subsumee $\exists r.X_0 \sqcap \exists s.X_0$, leading to $\exists r.(\exists r.X_0 \sqcap \exists s.X_0) \sqcap \exists s.(\exists r.X_0 \sqcap \exists s.X_0) \sqsubseteq X_2$. Concerning the extraction of ontologies from $\mathcal{T}$, we can more generally observe the following:

- *Assume that we are interested in the dependencies between X_0, and X_{10} including those using roles r, s. By replacing any of the concepts $X_1, ..., X_9$ by their direct subsumees, we reduce both, the number of axioms and the number of referenced concept names, but we increase the nesting depth of the resulting TBox. A complete replacement of $X_1, ..., X_9$ would yield a subsumee of X_{10} with a nesting depth of 10 and exponentially many occurrences of X_0. Even though the TBox contains only three axioms and no irrelevant concept names, it is less comprehensive than the original ontology.*

- *Assume that we are interested in A_1, A_2 instead of X_0. Eliminating X_0 from $\mathcal{T}$ would yield four different subsumees of X_1, namely $\exists r.A_1 \sqcap \exists s.A_1$, $\exists r.A_1 \sqcap \exists s.A_2$, $\exists r.A_2 \sqcap \exists s.A_1$ and $\exists r.A_2 \sqcap \exists s.A_2$. Each of these subsumees is required in order to preserve the relevant consequences, since none of the four concepts subsumes one of the other. Replacing X_0 in the general module using only A_1, A_2, X_0, X_{10} and r, s by its two subsumees, A_1 and A_2, would result in double exponentially many ($2^{2^{10}}$) different subsumees of X_{10}. Therefore, the elimination of a single concept name is, in most cases, not justified from the practical point of view.*

To address scenarios, where the above uncompromising prioritization is not required, in this section we investigate an alternative prioritization, allowing for a more balanced relationship between the extents to which the objectives are achieved. Similarly to minimal module extraction, we aim at preserving syntactic similarity between the general module and the original ontology.

However, we consider the extraction of general modules that consist of concepts structurally equivalent to sub-expressions occurring in the original ontology, i.e.,

concepts with the same structure but possibly a different set of atomic concepts. For instance, $A \sqcap \exists r.A$ is structurally equivalent to $B_1 \sqcap \exists r.B_2$.

Adding the computational complexity as a forth dimension, we investigate how we can obtain a tractable alternative to minimal module extraction and uniform interpolation by sacrificing the minimality guarantee, while fulfilling the requirement of syntactic similarity and reaching a decent effectiveness in terms of module size. As we show in the next section, ontologies obtained from the Gene Ontology by our approach on average contain half as many axioms as their minimal justifications within the original ontology. A comparison with the existing implementations also yields promising results. In case of the minimal module extractor for DL-Lite$_{\texttt{bool}}$, the extracted modules are 2 to 2.2 times larger than the ontologies obtained by our approach. In case of the locality-based module extractor, which is a tractable approach for extracting small but not necessarily minimal subsets of the original ontology, the extracted modules are on average 12 times larger than the ontologies obtained by our approach.

Similarly to the uniform interpolation approach presented in the last section, the discussed tractable rewriting approach uses normalization and the proof-theoretic results obtained in Section 5.1. We apply primitivization, thereby transforming the originally given ontology into sets of simple subsumees and subsumers of atomic concepts (concepts of the form $B, \exists r.B$ and $B_1 \sqcap ... \sqcap B_n$), i.e., into a subsumee/-subsumer relation pair for $\mathcal{T}$. The latter transformation is advantageous, since the derivation of arbitrary subsumees and subsumers from subsumees and subsumers with the maximal depth 1 becomes very easy and can be done by substituting an atomic concept in a subsumee or subsumer by any of its subsumees and subsumers with the maximal depth 1. This simplification establishes a close connection between the syntactic representation and the deductive closure: we can eliminate exactly the axioms referencing a particular concept from the closure of $\mathcal{T}$ by substituting it in all explicitly given subsumees and subsumers by its subsumees and subsumers, respectively, with the maximal role depth 1. Such a controlled stepwise reduction of the closure by the means of substitutions is also the main idea of the approach discussed in this section. However, in order to obtain a polynomial upper bound and preserve the syntactic similarity of the general module with the original ontology, we impose particular restrictions on the application of substitu-

tion. After each rewriting step, we identify and exclude *invalid* substitutions, i.e., those introducing structurally new subsumees and subsumers and those causing a growth of the corresponding subsumee/subsumer relation pair. In this way, we obtain a polynomial upper bound on the size of the resulting general module and guarantee their syntactic similarity with the originally given ontology. However, in order to ensure a decent effectiveness, it is crucial to exclude as few substitutions as possible. In the following, we discuss the choice of the initial subsumee/subsumer relation pair as well as the choice of substituents given our current objectives.

5.3.1 Choice of Substituents

For the approach to uniform interpolation, it was convenient to represent sets of subsumees and subsumers as languages generated by regular tree grammars, since the above described derivation of arbitrary subsumees and subsumers is naturally represented by the generation of trees in grammars. Here, we cannot benefit from this representation: Since regular tree grammars are not context-sensitive, the grammer representation does not allow for a context-dependent choice of subsumees or subsumers that should be used as substituents. For instance, it is not possible to use only the conjunction of the direct subsumers of B as its substituent within the scope of an existential restriction, and use the set of all its direct subsumers, otherwise. However, the additional flexibility of a context-dependent choice of substituents is important given our current objectives to obtain a polynomial upper bound and preserve the syntactic similarity of the general module with the original ontology. Since a preservation of syntactic similarity requires an exclusion of substitutions introducing structurally new subsumees and subsumers, context-dependent choice of substituents allows us to omit substituents yielding structurally new concepts in cases they are not required for the preservation of all relevant consequences due to the particular context. In this way, the subsitution remains eligible and does not destroy the syntactic similarity when applied. Thereby, we increase the number of eligible substitutions and, therefore, the effectiveness of the approach.

Moreover, context-dependent choice of substituents gives us an additional possibility to reduce the size of the subsumee and subsumer sets obtained after each sub-

stitution. Among other things, it allows us to exclude weak subsumees and weak subsumers as substituents also in cases where we do not require them due to the particular context. Since we want to exclude substitutions that cause a growth of the corresponding relation pair, it is also crucial to reduce the size of the subsumee and subsumer sets kept during the rewriting in order to increase the number of eligible substitutions. Given our objective to obtain general modules of a small size, keeping the corresponding subsumee/subsumer relation pair as small as possible is important in general. In order to minimize the number of introduced subsumees/-subsumers and, at the same time, maximize the number of eligible substitutions, instead of using regular tree languages, we define elementary rewriting operations with a context-dependent choice of substituents.

Starting with an initial subsumee/subsumer relation pair, e.g., the one given in Definition 11, that is complete with respect to a signature Σ, i.e., allows for constructing a general module of $\mathcal{T}$ with respect to Σ, the pair of relations obtained after each rewriting step should ideally still be complete with respect to Σ. Using the standard substitution notation $C[A/B]$ for denoting the concept obtained by replacing all occurrences of B within C by A, we use the following definition of an elementary rewriting preserving the completeness of subsumee/subsumer relation pairs.

Definition 15. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ ontology, $\Sigma \subseteq sig(\mathcal{T})$ a signature, and $\langle R_\sqsupseteq, R_\sqsubseteq \rangle$ a subsumee/subsumer relation pair for $\mathcal{T}$. For atomic concepts $A, B \in sig_C(\mathcal{T})$ and $\bowtie \in \{\sqsupseteq, \sqsubseteq\}$, an* elementary rewriting $\mathtt{Rew}_{R_\bowtie}(B, C, A)$ *of a subsumee/subsumer $C \in R_\bowtie(B)$ with respect to A is given by*

1. $\mathtt{Rew}_{R_\sqsupseteq}(B, C, A) = \{(B, C') \mid A' \in R_\sqsupseteq(A), C' = C[A'/A]\}$.
2. $\mathtt{Rew}_{R_\sqsubseteq}(B, C, A) = \begin{cases} \{(B, C') \mid D' = \bigsqcap_{D \in R_\sqsubseteq(A)} D, C' = C[D'/A]\}, & (a) \\ \{(B, C') \mid A' \in R_\sqsubseteq(A), C' = C[A'/A]\}, & (b) \end{cases}$

where (a) is used when A is within the scope of an existential restriction and (b) is used otherwise. Let $S_A = \{(B, C) \mid C \in R_\bowtie(B)$ and A occurs in $C\}$. A rewriting with respect to A is given by $\mathtt{Rew}_{R_\bowtie}(A) = \bigcup_{(B,C) \in S_A} \mathtt{Rew}_{R_\bowtie}(B, C, A) \cup R_\bowtie \setminus S_A$.

While, according to $\mathtt{Rew}_{R_\sqsupseteq}(B, C, A)$, we always omit the weak subsumees (conjunctions of any $A' \in R_\sqsupseteq(A)$), in case of subsumers, the strongest subsumer is the

conjunction $\bigsqcap_{D \in R_{\sqsubseteq}(A)} D$. However, in order to avoid an unnecessary exclusion of substitutions (those introducing new conjunctions) due to the requirement of syntactic similarity, we use weak subsumers ($A' \in R_{\sqsupseteq}(A)$) instead of $\bigsqcap_{D \in R_{\sqsubseteq}(A)} D$ where possible. This is exactly the case, when the substitution does not take place within the scope of an existential restriction. The latter is the case, since the corresponding deductive closure of $R_{\sqsubseteq}(B)$ contains $C[D'/A]$ with $D' = \bigsqcap_{D \in R_{\sqsubseteq}(A)} D$ if $R_{\sqsubseteq}(B)$ contains all $C[A'/A]$ with $A' \in R_{\sqsubseteq}(A)$ and A is not within the scope of an existential restriction in C. Thus, in the latter case, it is indeed sufficient to use weaker subsumers to preserve the completeness of the corresponding subsumee/-subsumer relation pair. We obtain the following result concerning completeness with respect to Σ:

Theorem 7. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ ontology, $\Sigma \subseteq sig(\mathcal{T})$ a signature, $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$ a subsumee/subsumer relation pair for $\mathcal{T}$ that is complete with respect to Σ. Then, for any $B' \notin \Sigma$ holds $\langle \mathtt{Rew}_{R_{\sqsupseteq}}(B'), R_{\sqsubseteq} \rangle$ and $\langle R_{\sqsupseteq}, \mathtt{Rew}_{R_{\sqsubseteq}}(B') \rangle$ are subsumee/subsumer relation pairs for $\mathcal{T}$, which are complete with respect to Σ.*

Proof. We prove the theorem using the notion of deweakenized closure given in Definition 13. Let $(B, C) \in R^+_{\sqsupseteq}$ for some B such that $\mathrm{sig}(C) \subseteq \Sigma$. It is easy to see that $(B, C) \in (\mathtt{Rew}_{R_{\sqsupseteq}}(B'))^+$, since $C \neq B'$ and all derivations obtained by replacing B' by its direct subsumees are now direct subsumees of the corresponding predecessors. For all D with $(B, D) \in R^+_{\sqsubseteq}$ and $\mathrm{sig}(D) \subseteq \Sigma$ holds that there is a D' such that D can be obtained from D' by omitting arbitrary conjuncts from arbitrary sub-expressions, i.e., $\{\} \models D' \sqsubseteq D$, and $(B, D') \in (\mathtt{Rew}_{R_{\sqsupseteq}}(B'))^+$ for the same reasons as above. It follows that $\mathtt{M}(R^+_{\sqsupseteq}, R^+_{\sqsubseteq}) \equiv^c_\Sigma \mathtt{M}((\mathtt{Rew}_{R_{\sqsupseteq}}(B'))^+, (\mathtt{Rew}_{R_{\sqsubseteq}}(B'))^+)$. Since, for a subsumee/subsumer relation pair $\langle R'_{\sqsupseteq}, R'_{\sqsubseteq} \rangle$ holds $\mathtt{M}(R'_{\sqsupseteq}, R'_{\sqsubseteq}) \equiv \mathtt{M}(R'^+_{\sqsupseteq}, R'^+_{\sqsubseteq})$, we obtain $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}) \equiv^c_\Sigma \mathtt{M}(\mathtt{Rew}_{R_{\sqsupseteq}}(B'), \mathtt{Rew}_{R_{\sqsubseteq}}(B'))$. Moreover, since $\mathtt{Rew}_{R_{\sqsupseteq}}(B'))^+ \subseteq R^+_{\sqsupseteq}$ and $\mathtt{Rew}_{R_{\sqsubseteq}}(B'))^+ \subseteq R^+_{\sqsubseteq}$, we obtain $\mathtt{M}(R^+_{\sqsupseteq}, R^+_{\sqsubseteq}) \models \mathtt{M}((\mathtt{Rew}_{R_{\sqsupseteq}}(B'))^+, (\mathtt{Rew}_{R_{\sqsubseteq}}(B'))^+)$, and, therefore, $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}) \models \mathtt{M}(\mathtt{Rew}_{R_{\sqsupseteq}}(B'), \mathtt{Rew}_{R_{\sqsubseteq}}(B'))$. Since $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$ is complete with respect to Σ, we obtain $\mathcal{T} \equiv^c_\Sigma \mathtt{M}(\mathtt{Rew}_{R_{\sqsupseteq}}(B'), \mathtt{Rew}_{R_{\sqsubseteq}}(B'))$ and $\mathcal{T} \models \mathtt{M}(\mathtt{Rew}_{R_{\sqsupseteq}}(B'), \mathtt{Rew}_{R_{\sqsubseteq}}(B'))$ after replacing temporary concepts by their definitions. □

In order to keep the relations as small as possible, we further remove trivial subsumees and subsumers obtained during the rewriting, namely those entailed by the empty TBox and, therefore, not necessary within the subsumee/subsumer relations to guarantee the completeness. These are atomic concepts themselves and, in case of subsumee relations, conjunctions with the atomic concept itself as one of the conjuncts. The corresponding check is inexpensive from the computational point of view, since such trivial subsumees and subsumers can be identified independently from other subsumees and subsumers. In what follows, we assume that such trivial subsumees and subsumers are removed after each rewriting.

Given a normalized $\mathcal{EL}$ ontology, the elimination of roles can be done by omitting all axioms with subsumees and subsumers containing irrelevant roles without loosing any relevant consequences. Therefore, in the following we focus on the elimination of irrelevant concept names and assume w.l.o.g. that the sets of subsumees and subsumers do not contain any non-Σ roles.

5.3.2 Choice of Initial Subsumees and Subsumers

Since rewritings yield smaller modules for sparse relation pairs, we will only use a subset of the initial subsumee/subsumer relation pair given in Definition 11. We compute a reduced subsumee/subsumer relation pair that only uses the transitive reduction of the classification results, i.e., we include $B_1 \sqsubseteq B_2$ only if there is no B_3 such that $B_1 \sqsubseteq B_3$ and $B_3 \sqsubseteq B_2$. It is easy to check that the completeness of the initial subsumee/subsumer relation pair stated in Theorem 1 still holds (see Lemma 18).

Definition 16. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ ontology. A subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ is called the* reduced initial subsumee/subsumer relation pair for $\mathcal{T}$ *if $R_{\sqsupseteq}$ and $R_{\sqsubseteq}$ are as follows:*

1. $R_{\sqsupseteq}(B) = \{C \mid C \sqsubseteq B \in \mathcal{T}$ *or* $C \equiv B \in \mathcal{T}$ *or* $C \in sig_C(\mathcal{T})$ *and* $\mathcal{T} \models C \sqsubseteq B\} \setminus \{A \in sig_C(\mathcal{T}) \mid$*there is* $A' \neq A$ *such that* $\mathcal{T} \models A \sqsubseteq A'$ *and* $\mathcal{T} \models A' \sqsubseteq B\}$,

2. $R_{\sqsubseteq}(B) = \{C \mid B \sqsubseteq C \in \mathcal{T}$ *or* $B \equiv C \in \mathcal{T}$ *or* $C \in sig_C(\mathcal{T})$ *and* $\mathcal{T} \models B \sqsubseteq C\} \setminus \{A \in sig_C(\mathcal{T}) \mid$ *there is* $A' \neq A$ *such that* $\mathcal{T} \models B \sqsubseteq A'$ *and* $\mathcal{T} \models A' \sqsubseteq A\}$.

We show that, if the subsumee/subsumer relation pair is complete with respect to Σ, then removing a transitive subsumption between two atomic concepts in any of the relations yields again a subsumee/subsumer relation pair complete with respect to Σ.

Lemma 18. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ TBox, $\Sigma \subseteq sig(\mathcal{T})$ a signature, and $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ a subsumee/subsumer relation pair for $\mathcal{T}$ complete with respect to Σ. Assume that, for some $A_1, A_2, A_3 \in sig_C(\mathcal{T})$ holds $\{(A_3, A_1), (A_3, A_2)\} \subseteq R_{\sqsupseteq}$,$\{(A_2, A_1)\} \subseteq R_{\sqsupseteq}$,$\{(A_1, A_2), (A_1, A_3)\} \subseteq R_{\sqsubseteq}$,$\{(A_2, A_3)\} \subseteq R_{\sqsubseteq}$. Then, $\langle R_{\sqsupseteq} \setminus \{(A_1, A_3)\}, R_{\sqsubseteq} \setminus \{(A_3, A_1)\}\rangle$ is complete with respect to Σ.*

Proof. The lemma is an immediate consequence of $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}) \equiv \mathtt{M}(R^+_{\sqsupseteq}, R^+_{\sqsubseteq})$ and $(A_1, A_3) \in R^+_{\sqsupseteq}, (A_3, A_1) \in R^+_{\sqsubseteq}$. □

In the next section, we assume this reduced form of initial subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$. Above, we have shown that, starting with the reduced initial subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$, after each rewriting step we obtain a subsumee/subsumer relation pair over $\mathcal{T}$ that is complete with respect to Σ. However, without further restrictions, the above rewritings would potentially introduce many large nested concept expressions or might not even terminate. In the following, we show how these problems can be avoided by stating the corresponding validity criteria for rewritings on subsumee/subsumer relation pairs.

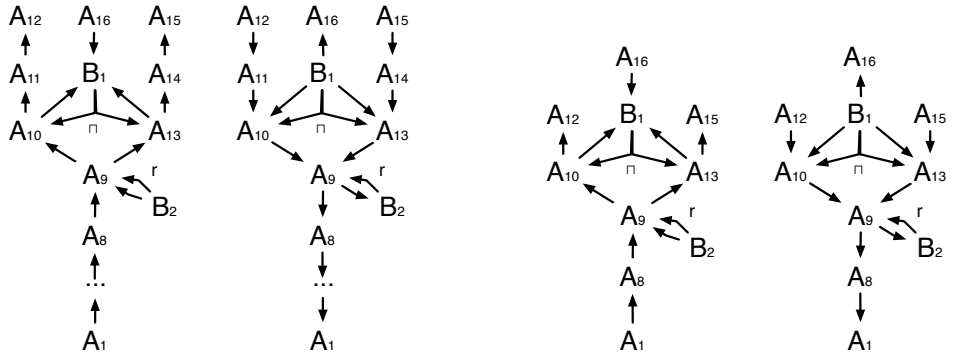
5.3.3 Restricting Rewriting

In this section, we address the problems caused by unrestricted application of rewriting demonstrated in Example 11. On the one hand, the example shows that rewriting can significantly change the syntactic structure of a ontology. On the other hand, it demonstrates that, while in some cases an elimination of a particular concept name can lead to a smaller ontology, it can as well cause the ontology to grow by several factors or, in the worst case, become infinite.

In order to avoid the above negative effects of rewriting, after each rewriting step we identify and exclude *invalid* rewritings, i.e., rewritings having a negative impact on the structure of the resulting module or the size of the relation pair.

Invalidity Conditions Guaranteeing Syntactic Similarity

As already discussed above, we exclude rewritings replacing atomic concepts by the conjunction of their direct subsumers corresponding to case (a) in Definition 15, since such a replacement possibly introduces concept expressions with a new structure not occurring in the original ontology. Thus, the set of valid rewritings is restricted to replacements of atomic concepts by their direct subsumees and subsumers. For the same reason, we additionally exclude rewritings that yield nested concept expressions, i.e., replacements of an atomic concept within a conjunction or existential restriction by one of its non-atomic subsumees or subsumers. Since the initial subsumee/subsumer relation pair contains only concepts of the form $B, \exists r.B$ and $B_1 \sqcap ... \sqcap B_n$, after each such valid rewriting step, all subsumees and subsumers are guaranteed to have this simple form as well. In this way, subsumee/subsumer relations can be represented as hypergraphs with atomic concepts as nodes and three types of edge, namely $A \rightarrow B$ representing atomic subsumees/subsumers, $A \xrightarrow{r} B$ representing existential restrictions, and multi-edges $A \xrightarrow{\sqcap} B_1, ..., B_n$ representing conjunctions.
The corresponding hypergraphs for the initial subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$ for the ontology in Example 5 are shown in Fig. 5.3(a).
The two exclusion cases given above indeed guarantee that no structurally new subsumees or subsumers are introduced during the rewriting. However, it is not yet sufficient to exclude an introduction of structurally new concept expressions in the resulting module. The reason for this are temporary concepts introduced during the primitivization (see Section 5.1.3) to represent non-atomic concept expressions. In the case that an atomic concept occurring in the original TBox has been substituted by a temporary concept B within a subsumee or subsumer and the latter is included into the resulting general module, a replacement of B within the module by the corresponding definition would possibly introduce a structurally new concept expression. Therefore, in order to guarantee the preservation of the syntactic



(a) Hypergraphs of the initial relations $R_{\sqsupseteq}$ (left) and $R_{\sqsubseteq}$ (right)

(b) Rewriting with respect to $A_2 - A_7$, A_{11}, A_{14}

Figure 5.3: *Hypergraphs for the ontology in Example 5*

similarity of the resulting general module after such a replacement of temporary concepts with their original definitions, we additionally exclude substitutions of non-temporary atomic concepts by the temporary ones.

In order to give the excluding conditions for the three above discussed cases, we we distinguish the following three types of successors and predecessors according to the types of edges in the subsumee/subsumer hypergraphs. For an atomic concept A and a relation $R_{\bowtie}$ with $\bowtie \in \{\sqsupseteq, \sqsubseteq\}$, we use

$$
\begin{aligned}
\mathtt{IN}_A(A) &:= \{B \in \mathrm{sig}_C(\mathcal{T}) \mid (B, A) \in R_{\bowtie}\} \\
\mathtt{OUT}_A(A) &:= \{B \in \mathrm{sig}_C(\mathcal{T}) \mid (A, B) \in R_{\bowtie}\} \\
\mathtt{IN}_{\mathtt{Roles}}(A) &:= \{B \mid (B, \exists r.A) \in R_{\bowtie}\} \\
\mathtt{OUT}_{\mathtt{Roles}}(A) &:= \{B \mid (A, \exists r.B) \in R_{\bowtie}\} \\
\mathtt{IN}_{\mathtt{Con}}(A) &:= \{B \mid (B, B'_1 \sqcap ... \sqcap B'_n) \in R_{\bowtie} \text{ with } A = B'_i \text{ for some } i \in \{1, ..., n\}\} \\
\mathtt{OUT}_{\mathtt{Con}}(A) &:= \{B'_1 \sqcap ... \sqcap B'_n \mid (A, B'_1 \sqcap \ldots \sqcap B'_n) \in R_{\bowtie}\}
\end{aligned}
$$

Further, let $\mathtt{IN}(A) = \mathtt{IN}_A(A) \cup \mathtt{IN}_{\mathtt{Roles}}(A) \cup \mathtt{IN}_{\mathtt{Con}}(A)$ and $\mathtt{OUT}(A) = \mathtt{OUT}_A(A) \cup \mathtt{OUT}_{\mathtt{Roles}}(A) \cup \mathtt{OUT}_{\mathtt{Con}}(A)$.

In order to avoid an introduction of structurally new concept expressions during the rewriting ((5.8)-(5.10)) and ensure termination ((5.11)), we exclude a rewriting

with respect to an atomic concept A if one of the following conditions is true:

$(\mathtt{IN}_{\mathtt{Roles}}(A) \cup \mathtt{IN}_{\mathtt{Con}}(A) \neq \emptyset)$ and $\mathtt{OUT}_A(A)$ contains temporary concepts; (5.8)

$(\mathtt{IN}_{\mathtt{Roles}}(A) \cup \mathtt{IN}_{\mathtt{Con}}(A) \neq \emptyset)$ and $(\mathtt{OUT}_{\mathtt{Roles}}(A) \cup \mathtt{OUT}_{\mathtt{Con}}(A) \neq \emptyset)$; (5.9)

$R_{\bowtie}$ is a subsumer relation and $|\mathtt{IN}_{\mathtt{Roles}}(A)| \geq 1$ and $|\mathtt{OUT}(A)| \geq 2$; (5.10)

Some C with $(A, C) \in R_{\bowtie}$ contains A; (5.11)

Going back to Example 5, the rewriting with respect to A_9 in $R_{\sqsubseteq}$ is invalid due to Condition (5.10) and rewriting with respect to A_{10}, A_{13} in $R_{\sqsupseteq}$ are invalid due to Condition (5.9).

Invalidity Conditions Guaranteeing Polynomial Bound

In order to identify rewritings that would increase the size of a relation, we compare the number of edges before and after the rewriting. While the number of edges potentially affected by a rewriting with respect to a concept A can be given by $|\mathtt{IN}(A)| + |\mathtt{OUT}(A)|$, the corresponding number of affected edges after the rewriting is in general bounded by $|\mathtt{OUT}(A)| + |\mathtt{IN}(A)| \cdot |\mathtt{OUT}(A)|$. Interestingly, if a concept B is unreferenced, it is usually possible to remove some elements from the corresponding sets of subsumees and subsumers without losing any Σ-consequences, or even without losing any axioms in $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq})$. We can remove subsumees and subsumers of unreferenced concepts, if none of the corresponding axioms in $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq})$ that contain these subsumees and subsumers, add any new Σ-consequences to $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq})$. Thus, in order to determine if a subsumee C with $(B, C) \in R_{\sqsupseteq}$ of $B \notin \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$ is unnecessary, we check for each element D with $(B, D) \in R_{\sqsubseteq}$, if $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}) \setminus \{C \sqsubseteq D\} \models C \sqsubseteq D$. Unnecessary subsumers can be determined in the same manner. For instance, in case of A_2 in Example 5, after the corresponding rewriting of both relations, we can remove its subsumee A_3 and subsumer A_1, if $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}) \setminus \{A_1 \sqsubseteq A_3\} \models A_1 \sqsubseteq A_3$. It is easy to check given the corresponding hypergraphs that this is indeed the case. In fact, the corresponding sets of necessary subsumees and subsumers after the rewriting are empty for $A_2, \ldots, A_7, A_{10}, A_{11}, A_{13}, A_{14}$ and B_1, B_2.

Algorithm 5: Rewriting of Subsumee/Subsumer Relation Pairs

Data: $\langle R^0_{\sqsupseteq}, R^0_{\sqsubseteq}\rangle$ initial subsumee/subsumer relation pair
Result: $\langle R_{\sqsupseteq \mathtt{r}}, R_{\sqsubseteq \mathtt{r}}\rangle$ rewritten subsumee/subsumer relation pair

$\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle \leftarrow \langle R^0_{\sqsupseteq}, R^0_{\sqsubseteq}\rangle$;
while *fixpoint is not reached* **do**
  for $B \in sig_C(\mathcal{T}) \setminus \Sigma$ **do**
    if *Conditions* (5.8)–(5.11) *are false* **then**
      Compute $n_{R_{\sqsupseteq}}$ and $n_{R_{\sqsubseteq}}$ for B;
      if *Inequation* (5.12) *does not hold* **then**
        $R_{\sqsupseteq} \leftarrow \mathtt{Rew}_{R_{\sqsupseteq}}(B) \setminus \{(B,C) \mid (B,C) \in R_{\sqsupseteq} \setminus R^{\mathtt{red}}_{\sqsupseteq}\}$;
        $R_{\sqsubseteq} \leftarrow \mathtt{Rew}_{R_{\sqsubseteq}}(B) \setminus \{(B,C) \mid (B,C) \in R_{\sqsubseteq} \setminus R^{\mathtt{red}}_{\sqsubseteq}\}$;
$\langle R_{\sqsupseteq \mathtt{r}}, R_{\sqsubseteq \mathtt{r}}\rangle \leftarrow \langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$;
return $\langle R_{\sqsupseteq \mathtt{r}}, R_{\sqsubseteq \mathtt{r}}\rangle$;

Let $\bowtie \in \{\sqsubseteq, \sqsupseteq\}$. Given the relation $R^{\mathtt{red}}_{\bowtie}$ obtained by omitting such unnecessary elements from $R_{\bowtie}$, we can use a tighter bound on the number of edges after rewriting based on $n_{R_{\bowtie}} = |\{C \mid (B,C) \in R^{\mathtt{red}}_{\bowtie}\}|$ instead of $|\{C \mid (B,C) \in R_{\bowtie}\}|$. Thus, we obtain the following inequation that holds for rewritings potentially increasing the size of relations:

$$|\mathtt{IN}(A)| + |\mathtt{OUT}(A)| < n_{R_{\bowtie}} + |\mathtt{IN}(A)| \cdot |\mathtt{OUT}(A)| \tag{5.12}$$

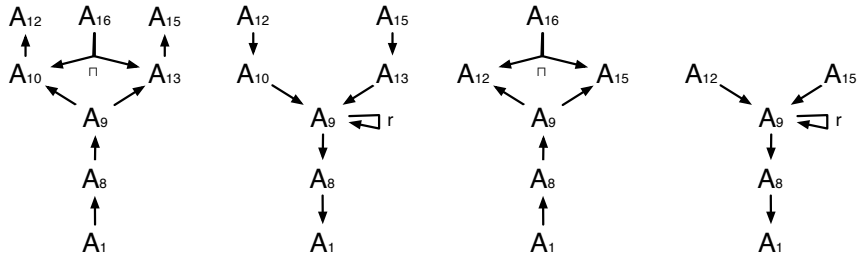
Based on the invalidity conditions (5.8)–(5.12), Algorithm 5 shows the rewriting process starting with the initial subsumee/subsumer relation pair $\langle R^0_{\sqsupseteq}, R^0_{\sqsubseteq}\rangle$. The computation terminates, when no further subsumees/subsumers can be eliminated during a single iteration. We obtain a rewritten subsumee/subsumer relation pair $\langle R_{\sqsupseteq \mathtt{r}}, R_{\sqsubseteq \mathtt{r}}\rangle$ over $\mathcal{T}$ complete with respect to Σ, which is of a polynomial size in the size of the original (not normalized) ontology $\mathcal{T}'$ and does not contain any nested concept expressions. Moreover, after replacing all temporary concept names in $\mathtt{M}(R_{\sqsupseteq \mathtt{r}}, R_{\sqsubseteq \mathtt{r}}, \Sigma^{\mathtt{ext}}(R_{\sqsupseteq \mathtt{r}}, R_{\sqsubseteq \mathtt{r}}))$ by their definitions, we obtain a general module of $\mathcal{T}'$, which does not contain any structurally new concept expressions not occurring in $\mathcal{T}'$. We can summarize the results as follows.

Theorem 8. *Let $\mathcal{T}$ be an $\mathcal{EL}$ ontology and $\Sigma \subseteq sig(\mathcal{T})$ a signature. Let $\mathcal{T}'$ be a normalization of $\mathcal{T}$ and $\mathcal{T}_r$ the ontology obtained by replacing all temporary concept names in $\mathtt{M}(R_{\sqsupseteq \mathtt{r}}, R_{\sqsubseteq \mathtt{r}}, \Sigma^{\mathtt{ext}}(R_{\sqsupseteq \mathtt{r}}, R_{\sqsubseteq \mathtt{r}}))$ by their definitions.*

- *$\mathtt{M}(R^{\mathcal{T}'}_{\sqsupseteq \mathtt{r}}, R^{\mathcal{T}'}_{\sqsubseteq \mathtt{r}}, \Sigma^{\mathtt{ext}}(R^{\mathcal{T}'}_{\sqsupseteq \mathtt{r}}, R^{\mathcal{T}'}_{\sqsubseteq \mathtt{r}}))$ can be computed in polynomial time and is polynomial in the size of $\mathcal{T}$;*
- *for all sub-expressions C' occurring in $\mathcal{T}_r$ there is a sub-expression C of $\mathcal{T}$ such that C' can be obtained from C by exchanging atomic concepts.*

Proof. The first statement is an immediate consequence of the polynomiality of $\mathcal{T}'$ and polynomiality of its computation, since rewriting steps do not increase the size of the relations due to Inequality (5.12), and the number of elementary rewritings is also polynomial (with number of rewriting steps limited by the number of concepts and the number of rewritten edges at each step at most polynomial with the size of $\mathcal{T}$). Note that, due to Condition (5.11), we only perform a rewriting, if the corresponding substituted concept becomes unreferenced after that.
To see the correctness of the second statement, note that, due to Conditions (5.9) and (5.10), for any subsumee or subsumer C in $R^{\mathcal{T}'}_{\sqsupseteq \mathtt{r}}$ and $R^{\mathcal{T}'}_{\sqsubseteq \mathtt{r}}$ holds that it is structurally equivalent to a concept C' being on the left- or the right-hand side of an axiom in $\mathcal{T}'$. In addition, due to Condition (5.8), if a temporary atomic concept occurs in C, then it occurs on exactly the same place also in C', i.e., we do not replace any atomic concepts by temporary concepts. Thus, after replacing temporary atomic concepts in C, we obtain a concept structurally equivalent to a sub-expression of $\mathcal{T}$. □

Now, we demonstrate rewriting based on Conditions (5.8)–(5.12) for Example 5 with $\Sigma = \{A_1, A_8, A_{12}, A_{15}, A_{16}, r\}$. We notice that, for instance, for all $i \in \{2, ..., 7, 11, 14\}$ holds $|\mathtt{IN}_A(A_i)| = 1$ and $|\mathtt{OUT}_A(A_i)| = 1$. Since both, $n_{R_\sqsupseteq}$ and $n_{R_\sqsubseteq}$ are 0 for all A_i, the number of edges decreases by one in case of each rewriting. After each rewriting including the subsequent omitting of unnecessary successors of the substituted concept, the number of edges as well as $n_{R_\sqsupseteq}$ and $n_{R_\sqsubseteq}$ remain the same for all remaining concepts. Thus, the conditions for the remaining concepts A_i with $i \in \{2, ..., 7, 11, 14\}$ do not change during any of the above rewritings. After performing all of the above rewritings, we obtain the subsumee/subsumer relation pair shown in Fig. 5.3(b).



(a) Rewriting with respect to B_1, B_2 (b) Rewriting with respect to A_{10}, A_{13}

Figure 5.4: *Rewriting for the ontology in Example 5*

In case of B_1, in $R_{\sqsubseteq}$ we have only outgoing edges. Since both, $n_{R_{\sqsupseteq}}$ and $n_{R_{\sqsubseteq}}$ are 0, we can eliminate the concept from in $R_{\sqsubseteq}$ by omitting its subsumers. In $R_{\sqsupseteq}$, we have three incoming and one outgoing edge, i.e., Inequation (5.12) does not hold. The number of edges decreases also in this case, since two of the conjunction edges obtained by rewriting are trivial (see 5.3.1) and are removed directly after the rewriting. In case of B_2, we only need to consider $R_{\sqsubseteq}$, since in $R_{\sqsupseteq}$ the concept is already unreferenced. Since we again have one incoming and one outgoing edge and $n_{R_{\sqsubseteq}}$ is 0, we can also perform the corresponding rewriting and eliminate B_2, thereby obtaining the relation pair shown in Fig. 5.4(a).

Now, we can also perform rewriting with respect to A_{10}, A_{13} in $R_{\sqsupseteq}$, since Condition (5.9) does not hold any more. Checking for unnecessary subsumees and subsumers reveals that both, $n_{R_{\sqsupseteq}}$ and $n_{R_{\sqsubseteq}}$ are still 0 for both, A_{10} and A_{13}. Since Inequation (5.12) does not hold in any of the two graphs, we can perform the corresponding rewriting and eliminate both, A_{10} and A_{13}, thereby obtaining the relation pair shown in Fig. 5.4(b).

We recall that $\Sigma = \{A_1, A_8, A_{12}, A_{15}, A_{16}, r\}$. Thus, the only atomic concept not from Σ still referenced within the subsumee/subsumer relations is A_9, which is not eligible for rewriting due to Condition (5.11). Therefore, the rewriting process is finished. After computing $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq})$, we obtain the smaller of the two general modules given for Example 5 earlier in this section, namely $\{A_{12} \sqcap A_{15} \sqsubseteq A_{16}$, $A_{12} \sqsubseteq A_9$, $A_{15} \sqsubseteq A_9$, $A_8 \sqsubseteq A_1$, $A_9 \sqsubseteq A_8$, $A_9 \sqsubseteq \exists r.A_9\}$.

While the advantages of the discussed approach in comparison to uniform interpolation are rather clear based on the obtained theoretical results, in case of minimal

module extraction and other approaches computing modules being a subset of the original ontology, the advantages, in particular in terms of module size, require an additional empirical investigation. In the following section, we will present encouraging empirical results for general module extraction using the presented approach and two existing implementations computing modules being a subset of the original ontology.

5.4 Experimental Results

In this chapter, we have discussed three different strategies to logic-based general module extraction: extraction of modules being a subset of the original ontology, uniform interpolation and a tractable rewriting approach extracting general modules consisting of concepts structurally equivalent to those occurring in the original ontology. The limitations of the uniform interpolation for practical scenarios are clear: the task is 3-$ExpTime$-hard with a tight triple-exponential bound on the size of general modules in case a finite result exists [NIKITINA and RUDOLPH 2012]. Moreover, uniform interpolation can significantly change the syntactic structure of the TBox, yielding in the worst-case double-exponential concept expressions, which makes uniform interpolation feasible only in scenarios, where the comprehensiveness of modules is not required.

In case of minimal module extraction and other approaches computing modules being a subset of the original ontology, a comparison from the theoretical point of view is difficult due to the different notion of syntactic similarity. Even though minimal module extraction guarantees the minimality of modules, it is based on a very strong notion of syntactic similarity not allowing for rewriting and, therefore, is usually less effective in terms of both, signature size and module size. In this section, we compare the tractable rewriting approach discussed in the last section, referred to as *Rewriter*, with existing implementations computing modules being a subset of the original ontology in terms of module size and computation time.

We are aware of two such implementations: an approach to minimal module extraction for DL-Lite$_{\texttt{bool}}$ [KONTCHAKOV et al. 2010] and *Locality-based extractor* [GRAU et al. 2007b] – an existing tractable approach to (not necessarily minimal) module extraction not based on rewriting. To the best of our knowledge, there

Table 5.1: *Evaluation results (module size) on DL-Lite$_{\texttt{bool}}$ fragment of $\mathcal{EL}$*

Signature size	Rewriter	Minimal module extractor	Locality-based extractor
10	4.8	9.7 (2.0)	167 (34.8)
30	10.3	22.2 (2.2)	436 (41.1)
50	28.8	60.4 (2.1)	1245 (43.2)

Table 5.2: *Evaluation results (module size) on $\mathcal{EL}$*

Signature size	Rewriter	Minimal justification extractor	Locality-based extractor
10	21	43 (2.0)	259 (12.3)
30	45	104 (2.3)	659 (14.6)
50	151	306 (2.0)	1787 (11.8)

are currently no existing implementations of minimal module extraction for $\mathcal{EL}$. Therefore, we compare the two implementations on the DL-Lite$_{\texttt{bool}}$ fragment of $\mathcal{EL}$, obtained from an $\mathcal{EL}$ ontology by replacing qualified existential restrictions by the corresponding unqualified restrictions. In order to also estimate the difference in the module size for $\mathcal{EL}$, we implemented a module extractor based on minimal justifications, which, given a general module obtained using our approach, computes a subset of the original ontology entailing the general module.

For our evaluation, we use the $\mathcal{EL}$ fragment of the Gene Ontology[2] describing gene product characteristics in terms of how gene products behave in a cellular context. The OWL version of the ontology (April 2012) comprises 36,251 atomic classes, 8 object properties and 316,580 logical axioms, out of which 66,117 axioms are terminological (the $\mathcal{EL}$ fragment contains 66,101 terminological axioms).

We use signatures with 10, 30 and 50 atomic concepts and 4 roles each. For each signature size, we randomly choose 10 signatures and let the different extractors compute the corresponding general module. Subsequently, we compute the average module size, shown in Tables 5.1 and 5.2 (the number in brackets is the average module size measured in the corresponding average size of the modules computed by Rewriter). The first table shows the results for the DL-Lite$_{\texttt{bool}}$ fragment of $\mathcal{EL}$. Due to the lower expressivity, the obtained DL-Lite$_{\texttt{bool}}$ modules are considerably smaller than their $\mathcal{EL}$ correspondents in Table 5.2. We observe that the size of the

[2] http://www.geneontology.org/

minimal DL-Lite$_{\texttt{bool}}$ modules containing only axioms from the original ontology $\mathcal{T}$ are between 2.0 and 2.2 times larger than the corresponding general modules consisting of sub-expressions of $\mathcal{T}$ with possibly exchanged atomic concepts obtained using Rewriter. The corresponding DL-Lite$_{\texttt{bool}}$ modules obtained by the locality-based extractor are even between 34.8 and 43.2 times larger. In case of $\mathcal{EL}$ modules, the minimal justifications of the general modules computed by Rewriter are between 2.0 and 2.3 times larger, while the modules obtained by the locality-based extractor are between 11.8 and 14.6 times larger.
Concerning the computation time, we observe a significant difference between the tractable approaches (Rewriter and the locality-based extractor) and the minimal module extractor. While, for the signatures with 50 atomic concepts, the first two approaches require less than one minute, minimal module extractor required between two hours and two days depending on the signature.

5.5 Summary

Since the size of a ontology has a crucial impact on the maintenance costs and often on the performance of reasoning, it is important to keep the corresponding ontology as compact as possible. In this chapter, we discussed logic-based approaches to relevance-based revision of ontologies, i.e., approaches that guarantee a preservation of the subset of the deductive closure using only the set Σ of relevant entities. First, we show that omitting axioms based only on the absence of relevant entities can lead to a loss of relevant information.
Further, we demonstrate that, while minimal module extraction guarantees the preservation of all relevant consequences, it is based on a very strong notion of syntactic similarity not allowing for rewriting of axioms, and, therefore, is usually less effective in terms of both, signature size and module size. We show that ontology extraction gains in effectiveness in terms of ontology size, when modules are not required to be subsets of the original ontology and investigate the task of ontology extraction based on rewriting. We provide an approach to computing uniform interpolants of general $\mathcal{EL}$ terminologies based on proof theory. Moreover, we show that, if a finite uniform $\mathcal{EL}$ interpolant exists, then there exists one of at most triple exponential size in terms of the original TBox, and that, in the worst-

case, no shorter interpolant exists, thereby establishing the triple exponential tight bounds.

Further, we consider the extraction of modules that consist of concepts structurally equivalent to sub-expressions occurring in the original ontology, i.e., concepts with the same structure but possibly a different set of atomic concepts. We propose a tractable approach (referred to as Rewriter) that, in most cases, yields small ontologies, but does not guarantee the minimality of the result. As we show in our evaluation, modules extracted during our evaluation using minimal module extractor for DL-Lite$_{\texttt{bool}}$ are 2.0 to 2.2 times larger than those obtained by our approach. In case of $\mathcal{EL}$, ontologies obtained by Rewriter on average contain half as many axioms as their minimal justifications within the original ontology. In case of the locality-based module extractor, the extracted $\mathcal{EL}$ modules are on average 12 times larger than the general modules obtained by the discussed approach.

Part III

Conclusions

CHAPTER 6

Summary and Significance of Thesis' Contributions

As ontology engineering tools gain in maturity and the amount of reusable ontological data grows, the deployment of ontologies becomes feasible in increasingly powerful and critical applications. Since the development of ontologies is a highly complex and error-prone task, a stable and reliable quality assurance methodology as well as tool support are particularly important in practice. The objective of this thesis was to advance the state-of-the-art in quality assurance for ontologies with respect to accuracy and conciseness. The scope of this work is determined by the ontology development project NanOn. Within NanOn, ontology reuse and automatic knowledge acquisition tools have been applied, requiring the corresponding quality assurance. On the one hand, manual inspection of the reused and acquired ontological data was necessary to ensure a high accuracy of the resulting ontology. Thus, the first objective of this thesis was to provide a methodology and a suitable tool support for reducing the manual effort of such an inspection of ontologies with respect to accuracy. On the other hand, since the corresponding publicly available ontologies were only partially relevant within the scope of NanOn and the performance of the semantic annotation engine was highly dependent on the size of the

resulting ontology, the conciseness of the latter had to be ensured. Hence, the second objective of this thesis was to investigate the means of ensuring the conciseness of ontologies in a semantics-preserving way, i.e., without losing any information about the relevant ontology entities. In this thesis, we provided solutions for both problems including theoretical foundations, sophisticated optimizations, an implementation and comprehensive experimental results. In this chapter, we summarize the contributions of this work with respect to these two objectives and discuss its impact on the advances in ontology engineering.

6.1 Quality Assurance with Respect to Accuracy

Semantic accuracy problems are difficult to detect due to the high dimensionality of their origin and the informal nature of the application requirements. While there are approaches to detect particular types of problems automatically, the scope of such automatic methods is rather narrow. The arguably most general approach to detecting semantic accuracy problems is manual inspection of ontologies, which can reveal problems not being anticipated by ontology engineers. Manual inspection is, however, one of the most costly alternatives in quality assurance due to the high amount of required user interaction.

In this thesis, we investigated, how to reduce the manual effort of such an exhaustive manual inspection, called *ontology revision*, by employing automated reasoning. First, based on the assumption underlying standardized ontology languages that the deductive closure of the correct axioms must be disjoint from the set of incorrect axioms, we showed how to partially automate the above process by reducing the number of decisions that have to be taken by a domain expert in order to complete the inspection. An important observation is that, given the above assumption, a single decision of the domain expert can predetermine several further evaluation decisions. We developed a general framework for the corresponding reasoning support of ontology revision based on the notion of revision closure capturing such predetermined evaluation decisions.

Further, in order to ensure a decent effectiveness of the reasoning-based support, we proposed and compared various axiom ranking techniques used to determine a beneficial order of evaluation. We showed that, even though a decent effort reduc-

tion can already be achieved when axioms for each expert decision are chosen in a random way, an inspection of axioms in a more selective order can yield a higher effort reduction. We then investigated different alternatives for determining the inspection order of axioms. We introduced the notion of axiom impact, which can be used to define simple axiom ranking functions performing well for data with either a very high or a very low average accuracy. In our evaluation, we were able to reduce the number of required evaluation decisions on average by additional 19% when the statements were reviewed based on axiom impacts. To account for cases with the average accuracy being substantially different from 100% and 0%, we introduced a ranking function based on the actual estimate for the average accuracy of the ontology under revision. In our evaluation, this ranking technique almost achieved the maximum possible automation, yielding additional 11% effort reduction for datasets with a medium average accuracy.

We then showed how an estimate of the average accuracy of a datatset required for axiom ranking can be learned on-the-fly over the course of the revision. Automatic learning of an estimate for the average accuracy is very important in practice, since it deliberates the user from having to provide such an estimate. We showed that, in case of large (5,000 axioms) datasets with an unknown average accuracy, learning the average accuracy is very effective due to the law of large numbers. In our experiments, the proportion of automatically evaluated statements is nearly the same as in case where the average accuracy is known in advance. Thus, the average accuracy of an ontology does not need to be known in order to utilize axiom ranking.

Since the above reasoning support is computationally expensive, we further introduced auxiliary data structures called decision spaces that are used for keeping track of dependencies between axioms. In our evaluation, decision spaces reduced the number of reasoner calls by 75%. Moreover, we demonstrated a simple partitioning approach, which reduced the number of reasoning calls by an order of magnitude. Using these optimizations, the reasoning-based support took on average less than one second after each expert decision.

6.2 Quality Assurance with Respect to Conciseness

Due to the significant impact of the knowledge base size on the cost of reasoning and maintenance, it is crucial for most applications to keep the underlying knowledge base as concise as possible. In particular in the context of ontology reuse, quality assurance with respect to conciseness can bring about significant performance advantages. The task of improving the conciseness of an ontology while preserving the relevant information – referred to as general module extraction within this thesis– is very complex. It requires a separation of its logical consequences into the set of relevant and irrelevant ones and a subsequent computation of a new ontology that ideally entails only the relevant consequences. For most representatives of description logics underlying the standardized ontology languages and the corresponding profiles, algorithms for an automatic computation of such smaller ontologies entailing all relevant consequences have been proposed. However, for the lightweight logic $\mathcal{EL}$ underlying the OWL EL profile, the problem has only been solved partially. It has been shown that checking if a particular subontology preserves all relevant consequences for a given ontology and a relevant vocabulary subset requires exponential time. The problem of uniform interpolation – computing a knowledge base entailing only relevant consequences for a particular vocabulary subset – remained open despite the research efforts of leading description logic experts since 2008. On the one hand, there was no algorithm for computing uniform interpolants for general $\mathcal{EL}$ terminologies. On the other hand, the bound on the size of uniform $\mathcal{EL}$ interpolants remained unknown. In this thesis, we closed both gaps. We provided a worst-case optimal algorithm computing uniform interpolants for general $\mathcal{EL}$ terminologies and derived a tight, triple-exponential bound on the output size.

Further, we took a critical look on the two current formalizations of the problem of general module extraction – uniform interpolation and classical module extraction. We considered three conflicting objectives: reducing the size of the extracted general module, reducing the size of its signature and preserving the syntactic similarity of the general module and the originally given knowledge base. In most application scenarios, all three objectives are important. However, neither classical module extraction nor uniform interpolation take this into account. In Chapter 5,

we demonstrated that classical module extraction is based on a very strong notion of syntactic similarity not allowing for rewriting of axioms, and, as a result, is usually not very effective in terms of improving conciseness. Further, we showed that uniform interpolation prioritizes small signature size allowing for no compromises with respect to the other two objectives. Such an uncompromising prioritization is rarely beneficial in practice. Taking into account these shortcomings of uniform interpolation and classical module extraction, we derived an alternative formalization for the problem of general module extraction with a more balanced prioritization of objectives. We introduced a new type of general modules consisting of concepts structurally equivalent to sub-expressions occurring in the original ontology, i.e., concepts with the same structure but possibly a different set of atomic concepts. We showed how a minimal module of this type can be computed in 2EXPTIME using classical module extraction after applying a particular normalization to the originally given knowledge base.

The currently known complexity results of the approaches computing minimal general modules are negative results from practical point of view. While classical module extraction for $\mathcal{EL}$ ontologies is EXPTIME-hard, the task of uniform interpolation is even 3EXPTIME-hard with a tight triple-exponential bound on the size of uniform interpolants in case a finite result exists. The best algorithm currently known for computing minimal modules of the novel type requires double-exponential time. Given that general module extraction in ontology engineering is of a particular interest for large ontologies and that some practically relevant application scenarios involve user interaction, approaches with such a high complexity are of a limited usefulness. To enable the application of general module extraction in practice, we developed a tractable approximation of the above revised approach to general module extraction. This approximation yields, in most cases, small ontologies, but does not guarantee the minimality of the result. In our evaluation, modules extracted using classical minimal module extractor for DL-Lite$_{\texttt{bool}}$ were 2.0 to 2.2 times larger than those obtained by our tractable approximation. Also in case of $\mathcal{EL}$ ontologies, minimal justifications of modules obtained by our approach were on average twice as large a the modules themselves.

6.3 Significance of Thesis' Contributions

A successful adoption of Semantic Web technologies in many areas of application requires a stable and flexible methodology and tool support for quality assurance. Further, the increasing application of ontology reuse and automated knowledge acquisition tools in ontology engineering brings about a shift of development efforts from knowledge modeling towards quality assurance. When ontology reuse or automatic knowledge acquisition are applied, accuracy and conciseness are the two most typical quality problems taking up a large proportion of a project's budget. Yet, despite the high practical importance, there has been a substantial lack of support for essential quality assurance activities concerning these two quality dimensions. In this thesis, we made a significant step forward in ontology engineering by developing a support for two such essential quality assurance activities requiring large amount of manual effort.

In Chapter 4, we developed a methodology for partially automating the inspection of ontologies with respect to accuracy. This important method of quality assurance, not replaceable by ontology debugging or constraint formalization in professional ontology engineering projects, is usually very time-consuming. Thus, the developed reasoning support allows for a significant saving of project resources. The methodology and the implementation have been thoroughly elaborated on for a deployment in practice. The core framework is designed in a very generic way with only few restrictions, which are fulfilled by all standardized ontology modelling languages. This generality is not brought about by compromizing on the power of the framework. The latter allows for a maximum usage of reasoning for the automation of the inspection process and for a flexible choice of initial constraints for the ontology's content. Various sophisticated optimizations ensure a decent efficiency and effectiveness of the framework in practice, e.g., computational effort of the reasoning support.

In Chapter 5, we solved an intricate theoretic problem in description logics, which has a high practical importance for ensuring the conciseness of ontologies. Firstly, we obtained the exact triple-exponential bound on the size of uniform interpolants for the lightweight logic $\mathcal{EL}$ underlying the OWL EL profile. This result is very interesting from the theoretic point of view, since it proves wrong the widespread

intuition about this bound to be "only" double-exponential. This is an important foundational insight in description logics, since it reveals the effect of structure sharing in the basic logic $\mathcal{EL}$. The result is equally important for ontology engineering. While, on the one hand, this is a negative result as regards the usage of uniform interpolation for the elimination of irrelevant information from ontologies, at the same time it is a positive result, since it shows the potential of structure sharing for improving the conciseness of ontologies. By introducing a reverse operation to uniform interpolation, namely the elimination of structural redundancy from ontologies via vocabulary extension, we can "compress" ontologies in a semantics-preserving way, obtaining triple-exponentially more concise representations of $\mathcal{EL}$ ontologies in the best case. This raises a new practically relevant research question, which is particularly interesting for improving reasoning efficiency.

A further significant result of Chapter 5 is a practically motivated, novel problem formalization for the semantics-preserving improvement of ontology's conciseness. So far, the research on improving conciseness has been focusing on uniform interpolation and classical module extraction, both of which in many application scenarios do not yield the optimal result in terms of ontology's size. A further drawback of uniform interpolation is that it does not take into account the requirement of comprehensiveness for the representation of ontologies. The novel problem formalization takes this into account while allowing for a decent additional improvement in terms of ontology's conciseness and, subsequently, a substantial additional improvement of reasoning performance given the at least polynomial complexity of reasoning.

Another important contribution of Chapter 5 for ontology engineering is the tractable approach to general module extraction. Since module extraction is usually relevant for an application to large ontologies, exponential or even triple-exponential complexity is a substantial hurdle. The only currently existing tractable approach to module extraction based on syntactic locality is far from optimum in terms of effectiveness, being, however, in most cases the only feasible option. Thus, the novel tractable approach presented in Chapter 5, which has shown an improvement in terms of conciseness by an order of magnitude, is a valuable contribution to ontology engineering.

CHAPTER 6. SUMMARY AND SIGNIFICANCE OF THESIS' CONTRIBUTIONS

CHAPTER 7

Outlook

The work presented in this thesis can be extended in different directions. The reasoning-based support of the manual inspection of ontologies presented in Chapter 4 can be extended by a wide range of further optimizations that could improve its behaviour in particular scenarios. Clearly, the usability of the tool can be improved by presenting axioms in a more sophisticated way, e.g., by generating natural language sentences for each axiom. This could have a significant impact on the performance of ontology engineers. Further, more axiom ranking heuristics could be developed and implemented covering further special cases. For instance, if the overall dataset is a mixture of datasets with an average accuracy close to 100% or 0%, a different ranking strategy not considered within this thesis would be more appropriate.

Also the optimizations concerning computational effort can be further improved, e.g., the currently proposed partitioning for ABoxes. In the current version, partitions are determined at the beginning of the revision and remain unchanged during the whole revision process. However, a single partition could potentially be further divided after some evaluation decisions. For instance, after an axiom has been declined, a partition might fall apart into several smaller partitions. It would be worth investigating whether a recomputation of partitions during the revision taking into

account this possible partition refinement pays off in practice. In addition, it would be interesting to study more general partitioning methods that are also applicable to TBoxes, e.g., [KONEV et al. 2010]. To account for scenarios where partitioning is not very effective due to the high density of dependencies between axioms, a strategy for compromizing between the effectiveness of the reasoning-based support and its computational efficiency can be very important in practice. A possible solution would be to separate the ontology into parts that are not logically independent from each other based on some heuristic criteria. In this case, we might miss automatic decisions, but the potential performance gain, due to the reasoning with smaller subsets of the ontology, is likely to compensate for this drawback.

The work on ensuring the conciseness of ontologies can be extended by improving the corresponding implementation and, more importantly, by solving the theoretic problems that have emerged, but have not been solved within this thesis. Concerning the latter, we can point out three highly relevant problems that have been raised in Chapter 5.

Firstly, the formal properties of the two proposed problem definitions of general module extraction based on structural equivalence of concept expressions have not been clarified to the full extent. Among other things, the worst-case complexity of computing minimal modules of this type remains unknown. In this work, we showed that these two tasks can be performed for $\mathcal{EL}$ ontologies in EXPTIME and 2EXPTIME, respectively, which are not necessarily the corresponding lower complexity bounds. Determining the exact complexity bounds of the two tasks for $\mathcal{EL}$ and further representatives of description logics is, however, of a high practical importance. On the one hand, investigating the source of the problem's complexity can yield better approximating implementations for $\mathcal{EL}$. On the other hand, the corresponding complexity results would be important in the light of prospective implementations for more expressive description logics.

Secondly, the proposed problem definitions of general module extraction can be adapted to serve another important purpose, namely improving the efficiency of semantic applications relying on reasoning. The approach proposed in this thesis accounts for the requirement of comprehensiveness for ontologies. In case the result is not read by humans but only used for reasoning, this requirement can be lifted, allowing for further improvement with respect to conciseness. This, for

instance, would be beneficial for applications realizing ontology-based data access, where the TBox is not directly exposed to users.

Thirdly, the problem of ontology compression – a reverse problem to uniform interpolation eliminating structural redundancy from ontologies via vocabulary extension – has been motivated by the results of this thesis. From the triple-exponential bound on the size of uniform interpolants in $\mathcal{EL}$ follows that, by applying the above compression to ontologies, we obtain a triple-exponentially more concise representations in the best case. This is particularly interesting for improving reasoning efficiency.

Bibliography

[ADDICOTT et al. 2006] Addicott, R., McGivern, G., and Ferlie, E. (2006). *Networks, Organizational Learning and Knowledge Management: NHS Cancer Networks*. Public Money & Management, 26(2), pp.87–94. (Cited on page 4.)

[ALFONSECA et al. 2010] Alfonseca, E., Pasca, M., and Robledo-Arnuncio, E. (2010). *Acquisition of instance attributes via labeled and related instances*. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010)*, pp. 58–65. (Cited on page 10.)

[AMIRHOSSEINI and SALIM 2011] Amirhosseini, M. and Salim, J. (2011). *OntoAbsolute as a ontology evaluation methodology in analysis of the structural domains in upper, middle and lower level ontologies*. In *Proceedings of the International Conference on Semantic Technology and Information Retrieval (STAIR 2011)*, pp. 26 –33. (Cited on page 48.)

[ARPINAR et al. 2006] Arpinar, I., Giriloganathan, K., and Aleman-Meza, B. (2006). *Ontology quality by detection of conflicts in metadata*. In *Proceedings of the 4th International EON Workshop (EON 2006)*. (Cited on page 37.)

[ARTALE et al. 2007] Artale, A., Calvanese, D., Kontchakov, R., and Zakharyaschev, M. (2007). *DL-Lite in the Light of First-Order Logic.*. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI 2007)*, pp. 361–366. (Cited on page 28.)

[ASHBURNER 2000] Ashburner, M. (2000). *Gene Ontology: Tool for the unification of biology*. Nature Genetics, 25, pp.25–29. (Cited on pages 6 and 7.)

[BAADER et al. 2005] Baader, F., Brandt, S., and Lutz, C. (2005). *Pushing the $\mathcal{EL}$ Envelope*. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*. (Cited on pages 6 and 120.)

[BAADER et al. 2007] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. (eds.) (2007). *The Description Logic Handbook: Theory, Implementation, and Applications*: Cambridge University Press, Second ed. (Cited on pages 6 and 19.)

[BAADER et al. 2010] Baader, F., Lutz, C., and Turhan, A.-Y. (2010). *Small is Again Beautiful in Description Logics.*. Künstliche Intelligenz, 24(1), pp.25–33. (Cited on page 19.)

[BACHIR BOUIADJRA and BENSLIMANE 2011] Bachir Bouiadjra, A. and Benslimane, S. (2011). *FOEval: Full ontology evaluation*. In *Proceedings of the 7th International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE 2011)*, pp. 464 –468. (Cited on page 48.)

[BAHRAMMIRZAEE 2010] Bahrammirzaee, A. (2010). *A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems*. Neural Computing & Applications, 19, pp.1165–1195. (Cited on page 7.)

[BARRETT 2009] Barrett, D. J. (2009). *MediaWiki - Wikipedia and beyond*: O'Reilly. (Cited on page 7.)

[BAUMGARTNER et al. 2007] Baumgartner, J. A., Cohen, K. B., Fox, L. M., Acquaah-Mensah, G., and Hunter, L. (2007). *Manual curation is not sufficient for annotation of genomic databases.*. In *ISMB/ECCB (Supplement of Bioinformatics)*, pp. 41–48. (Cited on pages 7 and 10.)

[BECHINI et al. 2008] Bechini, A., Tomasi, A., and Viotto, J. (2008). *Enabling ontology-based document classification and management in ebXML registries*. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC 2008)*, pp. 1145–1150. (Cited on page 8.)

[BEHRENDT 2008] Behrendt, W. (2008). *Knowledge Based Systems in Industry – Ontology Pays Half the Rent*. In *Proceedings of the 2008 Conference on Formal Ontologies Meet Industry*, pp. 17–21. (Cited on page 7.)

[BELL et al. 2007] Bell, D., Qi, G., and Liu, W. (2007). *Approaches to Inconsistency Handling in Description-Logic Based Ontologies*. In *Proceedings of the OTM 2007 Workshops: On the Move to Meaningful Internet Systems*, pp. 1303–1311. (Cited on page 34.)

[BERNERS-LEE et al. 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). *The Semantic Web*. Scientific American, 285(5), pp. 34–43. (Cited on page 4.)

[BEYDOUN et al. 2011] Beydoun, G., Lopez-Lorca, A. A., Garcia-Sanchez, F., and Martinez-Bijar, R. (2011). *How do we measure and improve the quality of a hierarchical ontology?*. Journal of Systems and Software, 84(12), pp.2363–2373. (Cited on page 48.)

[BHOGAL et al. 2007] Bhogal, J., Macfarlane, A., and Smith, P. (2007). *A review of ontology based query expansion*. Information Processing and Management, 43(4), pp.866–886. (Cited on page 8.)

[BLOEHDORN et al. 2007] Bloehdorn, S., Cimiano, P., Duke, A., Haase, P., Heizmann, J., Thurlow, I., and Völker, J. (2007). *Ontology-Based Question Answering for Digital Libraries*. In *Proceedings of the 11th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2007)*, pp. 14–25. (Cited on page 8.)

[BOLEY et al. 2010] Boley, H., Paschke, A., and Shafiq, O. (2010). *RuleML 1.0: the overarching specification of web rules*. In *Proceedings of the 2010 International Conference on Semantic Web Rules*, pp. 162–178. (Cited on page 37.)

[BOLOTNIKOVA et al. 2011] Bolotnikova, E., Gavrilova, T., and Gorovoy, V. (2011). *To a method of evaluating ontologies*. International Journal of Computer and Systems Sciences, 50, pp.448–461. (Cited on page 11.)

[BRANDT 2004] Brandt, S. (2004). *Polynomial Time Reasoning in a Description Logic with Existential Restrictions, GCI Axioms, and - What Else?*. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2004)*, pp. 298–302. (Cited on page 6.)

[BREWSTER et al. 2004] Brewster, C., Alani, H., Dasmahapatra, S., and Wilks, Y. (2004). *Data Driven Ontology Evaluation*. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2004)*. (Cited on pages 31 and 38.)

[BUITELAAR et al. 2005] Buitelaar, P., Cimiano, P., and Magnini, B. (2005). *Ontology Learning from Text: Methods, Evaluation and Applications*: IOS Press. (Cited on page 10.)

[BURTON-JONES et al. 2005] Burton-Jones, A., Storey, V. C., Sugumaran, V., and Ahluwalia, P. (2005). *A semiotic metrics suite for assessing the quality of ontologies*. Data & Knowledge Engineering, 55(1), pp.84–102. (Cited on page 11.)

[CALVANESE et al. 2005] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. (2005). *DL-Lite: Tractable Description Logics for Ontologies*. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, pp. 602–607. (Cited on page 28.)

[CARPINETO and ROMANO 2012] Carpineto, C. and Romano, G. (2012). *A Survey of Automatic Query Expansion in Information Retrieval*. ACM Computing Surveys, 44(1), pp.1–50. (Cited on page 8.)

[CARRUTHERS et al. 2002] Carruthers, P., Stich, S., and Siegal, M. (eds.) (2002). *The Cognitive Basis of Science*: Cambridge University Press. (Cited on page 7.)

[CATE and CONRADIE 2006] Cate, B. T. and Conradie, W. (2006). *Definitorially complete description logics*. In *Proceedings of the 10th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2006)*. (Cited on page 45.)

[CEUSTERS et al. 2004] Ceusters, W., Smith, B., Kumar, A., and Dhaen, C. (2004). *Ontology-Based Error Detection in SNOMED-CT*. Studies in Health Technology and Informatics, 107, pp.482–486. (Cited on pages 33 and 38.)

[COLOMB 2002] Colomb, R. (2002). *Quality of Ontologies in Interoperating Information Systems*. Technical Report, ISIB-CNR. (Cited on pages 7 and 11.)

[COMON et al. 2008] Comon, H., Jacquemard, F., Dauchet, M., Gilleron, R., Lugiez, D., Loding, C., Tison, S., and Tommasi, M. (2008). *Tree Automata Techniques and Applications*. (Cited on page 118.)

[CORCHO et al. 2009] Corcho, O., Roussey, C., Blazquez, L. M. V., and Perez, I. (2009). *Pattern-based OWL Ontology Debugging Guidelines*. In *Proceedings of the Workshop on Ontology Patterns (WOP 2009)*. (Cited on page 31.)

[CROSS and HU 2011] Cross, V. and Hu, X. (2011). *Using semantic similarity in ontology alignment.*. In *Proceedings of the 6th International Workshop on Ontology Matching (OM 2011)*. (Cited on page 10.)

[DEGTYARENKO et al. 2008] Degtyarenko, K., de Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcntara, R., Darsow, M., Guedj, M., and Ashburner, M. (2008). *ChEBI: a database and ontology for chemical entities of biological interest*. Nucleic Acids Research, 36, pp.D344–D350. (Cited on pages 6, 7 and 12.)

[DEMIR et al. 2010] Demir, E., Cary, M. P., Paley, S., Fukuda, K., Lemer, C., Vastrik, I., Wu, G., D'Eustachio, P., Schaefer, C., Luciano, J., Schacherer, F., Martinez-Flores, I., Hu, Z., Jimenez-Jacinto, V., Joshi-Tope, G., Kandasamy, K., Lopez-Fuentes, A. C., Mi, H., Pichler, E., Rodchenkov, I., Splendiani, A., Tkachev, S., Zucker, J., Gopinath, G., Rajasimha, H., Ramakrishnan, R., Shah, I., Syed, M., Anwar, N., Babur, O., Blinov, M., Brauner, E., Corwin, D., Donaldson, S., Gibbons, F., Goldberg, R., Hornbeck, P., Luna, A., Murray-Rust, P., Neumann, E., Ruebenacker, O., Samwald, M., van Iersel, M., Wimalaratne, S., Allen, K., Braun, B., Whirl-Carrillo, M., Cheung, K.-H., Dahlquist, K., Finney, A., Gillespie, M., Glass, E., Gong, L., Haw, R., Honig, M., Hubaut, O., Kane, D., Krupa, S., Kutmon, M., Leonard, J., Marks, D., Merberg, D., Petri, V., Pico, A., Ravenscroft, D., Ren, L., Shah, N., Sunshine, M., Tang, R., Whaley, R., Letovksy, S., Buetow, K. H., Rzhetsky, A., Schachter, V., Sobral, B. S., Dogrusoz, U., McWeeney, S., Aladjem, M., Birney, E., Collado-Vides, J., Goto, S., Hucka, M., Le Novere, N., Maltsev, N., Pandey, A., Thomas, P., Wingender, E., Karp, P. D., Sander, C., and Bader, G. D. (2010). *The BioPAX community standard for pathway data sharing*. Nature Biotechnology, 28(9), pp.935–942. (Cited on page 7.)

[DÍAZ-GALIANO et al. 2009] Díaz-Galiano, M. C., Martín-Valdivia, M., and Ureña López, L. A. (2009). *Query expansion with a medical ontology to improve a multimodal information retrieval system*. Computers in Biology and Medicine, 39(4), pp.396–403. (Cited on page 8.)

[DONINI and MASSACCI 2000] Donini, F. M. and Massacci, F. (2000). *EXPTIME tableaux for ALC*. Artificial Intelligence, 124(1), pp.87–138. (Cited on page 29.)

[DU and SHEN 2008] Du, J. and Shen, Y.-D. (2008). *Computing minimum cost diagnoses to repair populated DL-based ontologies*. In *Proceedings of the 17th International Conference on World Wide Web (WWW 2008)*, pp. 565–574. (Cited on page 34.)

[DUONG et al. 2010] Duong, T. H., Cha, S.-J., and Jo, G. S. (2010). *An Effective Method for Ontology Integration by Propagating Inconsistency*. In *Proceedings of the 2nd International Conference on Knowledge and Systems Engineering (KSE 2010)*, pp. 41 –46. (Cited on page 32.)

[ELHADAD et al. 2009] Elhadad, M., Gabay, D., and Netzer, Y. (2009). *Automatic Evaluation of Search Ontologies in the Entertainment Domain using Natural Language Processing*. (Cited on page 31.)

[EUZENAT et al. 2011] Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., and Trojahn, C. (2011). *Ontology Alignment Evaluation Initiative: Six Years of Experience*. Journal on Data Semantics, 15, pp.158–192. (Cited on page 10.)

[FELDMAN et al. 2009] Feldman, A., Provan, G., and Van Gemund, A. (2009). *FRACTAL: efficient fault isolation using active testing*. In *Proceedings of the 21th International Joint Conference on Artifical Intelligence (IJCAI 2009)*, pp. 778–784. (Cited on page 32.)

[FELFERNIG et al. 2009] Felfernig, A., Friedrich, G., Schubert, M., Mandl, M., Mairitsch, M., and Teppan, E. (2009). *Plausible repairs for inconsistent requirements*. In *Proceedings of the 21th International Joint Conference on Artifical Intelligence (IJCAI 2009)*, pp. 791–796. (Cited on page 32.)

[FELLMANN et al. 2011] Fellmann, M., Thomas, O., and Busch, B. (2011). *A Query-Driven Approach for Checking the Semantic Correctness of Ontology-Based Process Representations*. In *Proceedings of the 14th International Conference on Business Information Systems (BIS 2011)*, pp. 62–73. (Cited on page 36.)

[FERRÁNDEZ et al. 2009] Ferrández, Ó., Izquierdo, R., Ferrández, S., and Vicedo, J. L. (2009). *Addressing ontology-based question answering with collections of user queries*. Information Processing & Management, 45(2), pp.175–188. (Cited on pages 8 and 11.)

[FOX et al. 2009] Fox, P., McGuinness, D. L., Cinquini, L., West, P., Garcia, J., Benedict, J. L., and Middleton, D. (2009). *Ontology-supported scientific data frameworks: The Virtual Solar-Terrestrial Observatory experience*. Computers & Geosciences, 35(4), pp.724–738. (Cited on page 6.)

[FU et al. 2009] Fu, J., Xu, J., and Jia, K. (2009). *Domain Ontology Based Automatic Question Answering*. In *Proceedings of the International Conference on Computer Engineering and Technology (ICCET 2009)*, pp. 346 –349. (Cited on page 8.)

[FÜRBER and HEPP 2011] Fürber, C. and Hepp, M. (2011). *Towards a vocabulary for data quality management in semantic web architectures*. In *Proceedings of the 1st International Workshop on Linked Web Data Management (LWDM 2011)*, pp. 1–8. (Cited on page 37.)

[GANGEMI et al. 2006] Gangemi, A., Catenacci, C., Ciaramita, M., and Lehmann, J. (2006). *Modelling ontology evaluation and validation*. In *Proceedings of the 3rd European Semantic Web Conference (ESWC 2006)*. (Cited on page 48.)

[GANGEMI et al. 2002] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. (2002). *Sweetening Ontologies with DOLCE*. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pp. 166–181. (Cited on page 5.)

[GARSON 1989] Garson, J. (1989). *Modularity and Relevant Logic*. Notre Dame Journal of Formal Logic, 30(2), pp.207–223. (Cited on page 42.)

[GHILARDI et al. 2006] Ghilardi, S., Lutz, C., and Wolter, F. (2006). *Did I Damage my Ontology? A Case for Conservative Extensions in Description Logics*. In *Proceedings of the 10th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2006)*, pp. 187–197. (Cited on pages 42 and 45.)

[GIACOMELLI et al. 2012] Giacomelli, P., Munaro, G., and Rosso, R. (2012). *Can an Ad-hoc ontology Beat a Medical Search Engine? The Chronious Search Engine case*. Computing Research Repository, abs/1203.4494. (Cited on page 12.)

[GKOUTOS et al. 2005] Gkoutos, G. V., Green, E. C., Mallon, A.-M. M., Hancock, J. M., and Davidson, D. (2005). *Using ontologies to describe mouse phenotypes.*. Genome Biology, 6(1). (Cited on page 6.)

[GLIMM et al. 2010] Glimm, B., Rudolph, S., and Völker, J. (2010). *Integrated Metamodeling and Diagnosis in OWL 2*. In *Proceedings of the 9th International Semantic Web Conference (ISWC 2010)*, pp. 257–272. (Cited on page 36.)

[GOLBREICH et al. 2006] Golbreich, C., Zhang, S., and Bodenreider, O. (2006). *The foundational model of anatomy in OWL: Experience and perspectives*. Web Semantics: Science, Services and Agents on the World Wide Web, 4, pp.181–195. (Cited on page 6.)

[GÓMEZ-PÉREZ 2004] Gómez-Pérez, A. (2004). *Ontology Evaluation*. In Staab, S. and Studer, R. (eds.): *Handbook on Ontologies in Information Systems, First Edition*, pp. 251–274: Springer. (Cited on page 11.)

[GOOCH 2012] Gooch, P. (2012). *Systematic identification and correction of spelling errors in the foundational model of anatomy*. In *Proceedings of the 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences (SWAT4LS 2011)*, pp. 34–35. (Cited on page 31.)

[GOODWIN 2005] Goodwin, J. (2005). *Experiences of Using OWL at the Ordnance Survey*. In *Proceedings of OWL: Experiences and Directions (OWLED 2005)*. (Cited on page 6.)

[GRAU et al. 2007a] Grau, B. C., Horrocks, I., Kazakov, Y., and Sattler, U. (2007a). *Extracting Modules From Ontologies: A Logic-based Approach*. In *Proceedings of OWL: Experiences and Directions (OWLED 2007)*. (Cited on page 44.)

[GRAU et al. 2007b] Grau, B. C., Horrocks, I., Kazakov, Y., and Sattler, U. (2007b). *Just the right amount: extracting modules from ontologies*. In *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*, pp. 717–726. (Cited on pages 95 and 149.)

[GRAU et al. 2007c] Grau, B. C., Horrocks, I., Kazakov, Y., and Sattler, U. (2007c). *A logical framework for modularity of ontologies*. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI 2007)*, pp. 298–304. (Cited on pages 43, 44 and 81.)

[GRAU et al. 2006] Grau, B. C., Parsia, B., Sirin, E., and Kalyanpur, A. (2006). *Modularity and Web Ontologies*. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)*, pp. 198–209. (Cited on page 43.)

[GRAUPMANN et al. 2005] Graupmann, J., Schenkel, R., and Weikum, G. (2005). *The SphereSearch Engine for Unified Ranked Retrieval of Heterogeneous XML and Web Documents*. In *Proceedings of the 31st International Conference on Very Large Databases (VLDB 2005)*, pp. 529–540. (Cited on page 8.)

[GRIMM and MOTIK 2005] Grimm, S. and Motik, B. (2005). *Closed World Reasoning in the Semantic Web through Epistemic Operators.*. In *Proceedings of OWL: Experiences and Directions (OWLED 2005)*. (Cited on page 37.)

[GRUBER 1993] Gruber, T. R. (1993). *A translation approach to portable ontology specifications.*. Knowledge Acquisition, 5, pp. 199 – 220. (Cited on page 5.)

[GRUBER 1995] Gruber, T. R. (1995). *Toward principles for the design of ontologies used for knowledge sharing*. International Journal on Human-Computer Studies, 43(5-6), pp.907–928. (Cited on page 11.)

[GRUBER 2007] Gruber, T. R. (2007). *Automatically Integrating Heterogeneous Ontologies from Structured Web Pages*. International Journal on Semantic Web and Information Systems, 3(1), pp.1–11. (Cited on page 10.)

[GRÜNINGER and FOX 1995] Grüninger, M. and Fox, M. S. (1995). *Methodology for the design and evaluation of ontologies*. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing*. (Cited on page 11.)

[GUARINO and WELTY 2002] Guarino, N. and Welty, C. (2002). *Evaluating ontological decisions with OntoClean*. Communications of the ACM, 45(2), pp.61–65. (Cited on page 35.)

[HAASE and QI 2007] Haase, P. and Qi, G. (2007). *An analysis of approaches to resolving inconsistencies in DL-based ontologies*. In *Proceedings of the International Workshop on Ontology Dynamics (IWOD 2007)*. (Cited on page 34.)

[HELBIG 2005] Helbig, H. (2005). *Knowledge Representation and the Semantics of Natural Language*: Springer. (Cited on page 37.)

[HERRE 2010] Herre, H. (2010). *General Formal Ontology (GFO) : A Foundational Ontology for Conceptual Modelling*. In Poli, R. and Obrst, L. (eds.): *Theory and Applications of Ontology*: Springer. (Cited on page 5.)

[HORROCKS et al. 2006] Horrocks, I., Kutz, O., and Sattler, U. (2006). *The Even More Irresistible SROIQ*. In *Proceedings of the 10th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2006)*, pp. 57–67. (Cited on pages 19 and 27.)

[HORROCKS et al. 2004] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., and Dean, M. (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. (Cited on page 37.)

[HUANG et al. 2012] Huang, H., Stvilia, B., Jorgensen, C., and Bass, H. W. (2012). *Prioritization of data quality dimensions and skills requirements in genome annotation work*. Journal of the American Society for Information Science and Technology, 63(1), pp.195–207. (Cited on pages 11 and 13.)

[IFRIM and WEIKUM 2006] Ifrim, G. and Weikum, G. (2006). *Transductive learning for text classification using explicit knowledge models*. In *Proceedings of the 10th European Conference on Principle and Practice of Knowledge Discovery in Databases (ECML-PKDD 2006)*, pp. 223–234. (Cited on page 8.)

[JACKSON 1999] Jackson, P. (1999). *Introduction to Expert Systems, 3rd Edition*: Addison-Wesley. (Cited on page 5.)

[JANOWICZ et al. 2008] Janowicz, K., Maué, P., Wilkes, M., Schade, S., Scherer, F., Braun, M., Dupke, S., and Kuhn, W. (2008). *Similarity as a Quality Indicator in Ontology Engineering*. In *Proceedings of the 5th International Conference on Formal Ontology in Information Systems (FOIS 2008)*, pp. 92–105. (Cited on page 48.)

[JI et al. 2009] Ji, Q., Haase, P., Qi, G., Hitzler, P., and Stadtmüller, S. (2009). *RaDON - Repair and Diagnosis in Ontology Networks.*. In *Proceedings of the 6th European Semantic Web Conference (ESWC 2009)*, pp. 863–867. (Cited on page 34.)

[JIMÉNEZ-RUIZ et al. 2009a] Jiménez-Ruiz, E., Grau, B. C., Horrocks, I., and Llavori, R. B. (2009a). *Building Ontologies Collaboratively Using ContentCVS*. In *Proceedings of the 22nd International Workshop on Description Logics (DL 2009)*. (Cited on pages 39, 40 and 90.)

[JIMÉNEZ-RUIZ et al. 2009b] Jiménez-Ruiz, E., Grau, B. C., Horrocks, I., and Llavori, R. B. (2009b). *Ontology Integration Using Mappings: Towards Getting the Right Logical Consequences*. In *Proceedings of the 6th European Semantic Web Conference (ESWC 2009)*, pp. 173–187. (Cited on pages 39, 40 and 90.)

[KANEHISA and GOTO 2000] Kanehisa, M. and Goto, S. (2000). *KEGG: Kyoto Encyclopedia of Genes and Genomes*. Nucleic Acids Research, 28(1), pp.27–30. (Cited on page 7.)

[KAZAKOV 2008] Kazakov, Y. (2008). *$\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are Harder than $\mathcal{SHOIQ}$*. In *Proceedings of the 11th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2008)*, pp. 274–284. (Cited on page 80.)

[KERRIEN et al. 2007] Kerrien, S., Orchard, S., Montecchi-Palazzi, L., Aranda, B., Quinn, A. F., Vinod, N., Bader, G. D., Xenarios, I., Wojcik, J., Sherman, D., Tyers, M., Salama, J. J., Moore, S., Ceol, A., Chatr-Aryamontri, A., Oesterheld, M., Stümpflen, V., Salwinski, L., Nerothin, J., Cerami, E., Cusick, M. E., Vidal, M., Gilson, M., Armstrong, J., Woollard, P., Hogue, C., Eisenberg, D., Cesareni, G., Apweiler, R., and Hermjakob, H. (2007). *Broadening the horizon–level 2.5 of the HUPO-PSI format for molecular interactions.*. BMC Biology, 5(1), pp.44+. (Cited on page 7.)

[KOENIG 1995] Koenig, A. (1995). *Patterns and Antipatterns*. Journal of Object-Oriented Programming, 8(1), pp.46–48. (Cited on page 36.)

[KÖHLER et al. 2011] Köhler, S., Bauer, S., Mungall, C. J., Carletti, G., Smith, C. L., Schofield, P., Gkoutos, G. V., and Robinson, P. N. (2011). *Improving ontologies by automatic reasoning and evaluation of logical definitions*. BMC Bioinformatics, 12, pp.418–418. (Cited on page 38.)

[KONEV et al. 2010] Konev, B., Lutz, C., Ponomaryov, D., and Wolter, F. (2010). *Decomposing Description Logic Ontologies*. In *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010)*. (Cited on pages 81 and 164.)

[KONEV et al. 2008] Konev, B., Lutz, C., Walther, D., and Wolter, F. (2008). *Semantic Modularity and Module Extraction in Description Logics*. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*, pp. 55–59. (Cited on page 42.)

[KONEV et al. 2009a] Konev, B., Lutz, C., Walther, D., and Wolter, F. (2009a). *Formal properties of modularisation*. In Stuckenschmidt, H., Parent, C., and Spaccapietra, S. (eds.): *Modular Ontologies*, pp. 25–66: Springer-Verlag. (Cited on page 43.)

[KONEV et al. 2009b] Konev, B., Walther, D., and Wolter, F. (2009b). *Forgetting and uniform interpolation in large-scale description logic terminologies*. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence (IJCAI 2009)*, pp. 830–835. (Cited on pages 46, 95 and 114.)

[KONTCHAKOV et al. 2008] Kontchakov, R., Wolter, F., and Zakharyaschev, M. (2008). *Can You Tell the Difference Between DL-Lite Ontologies?*. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008)*, pp. 285–295. (Cited on pages 43 and 44.)

[KONTCHAKOV et al. 2010] Kontchakov, R., Wolter, F., and Zakharyaschev, M. (2010). *Logic-based ontology comparison and module extraction, with an application to DL-Lite*. Artificial Intelligence, 174, pp.1093–1141. (Cited on pages 95 and 149.)

[KRÖTZSCH and VRANDECIC 2011] Krötzsch, M. and Vrandecic, D. (2011). *Semantic MediaWiki*. In *Foundations for the Web of Information and Services*, pp. 311–326: Springer. (Cited on page 7.)

[LEGG and SARJANT 2012] Legg, C. and Sarjant, S. (2012). *Bill Gates is not a parking meter: Philosophical quality control in automated ontology building*. In *Proceedings of the Symposium on Computational Philosophy*. (Cited on page 38.)

[LEI et al. 2007] Lei, Y., Uren, V., and Motta, E. (2007). *A framework for evaluating semantic metadata*. In *Proceedings of the 4th International Conference on Knowledge Capture (K-CAP 2007)*, pp. 135–142. (Cited on pages 11 and 48.)

[LENAT and GUHA 1989] Lenat, D. B. and Guha, R. V. (1989). *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*: Addison-Wesley Longman Publishing Co., Inc., 1st ed. (Cited on pages 5 and 9.)

[LEWEN and D'AQUIN 2010] Lewen, H. and d'Aquin, M. (2010). *Extending Open Rating Systems for Ontology Ranking and Reuse*. In *Proceedings of the 17th International Conference on Knowledge Engineering and Management by the Masses (EKAW 2010)*, pp. 441–450. (Cited on page 47.)

[LIEBMAN and MOLINARO 2011] Liebman, M. and Molinaro, S. (2011). *Hypothesis Generation and Evaluation in Clinical Trial Design*. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2011)*, pp. 645 –651. (Cited on page 7.)

[LIU et al. 2004] Liu, S., Liu, F., Yu, C., and Meng, W. (2004). *An effective approach to document retrieval via utilizing WordNet and recognizing phrases*. In *Proceeding of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, pp. 266–272. (Cited on page 8.)

[LOZANO-TELLO and GÓMEZ-PÉREZ 2004] Lozano-Tello, A. and Gómez-Pérez, A. (2004). *ONTOMETRIC: A Method to Choose the Appropriate Ontology*. Journal of Database Management, 15(2), pp. 1–18. (Cited on page 48.)

[LUTZ et al. 2010] Lutz, C., Piro, R., and Wolter, F. (2010). *Enriching $\mathcal{EL}$-Concepts with Greatest Fixpoints*. In *Proceedings of the19th European Conference on Artificial Intelligence (ECAI 2010)*, pp. 41–46. (Cited on pages 27 and 28.)

[LUTZ et al. 2012] Lutz, C., Seylan, I., and Wolter, F. (2012). *An Automata-Theoretic Approach to Uniform Interpolation and Apprximation in the Description Logic EL*. In *Proceedings of the 13th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2012)*. (Cited on pages 46, 95, 115, 116 and 129.)

[LUTZ et al. 2007] Lutz, C., Walther, D., and Wolter, F. (2007). *Conservative extensions in expressive description logics*. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI 2007)*, pp. 453–458. (Cited on page 42.)

[LUTZ and WOLTER 2010] Lutz, C. and Wolter, F. (2010). *Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$*. Journal of Symbolic Computation, 45(2), pp.194–228. (Cited on pages 43, 107 and 129.)

[LUTZ and WOLTER 2011] Lutz, C. and Wolter, F. (2011). *Foundations for Uniform Interpolation and Forgetting in Expressive Description Logics*. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*. (Cited on pages 46, 114 and 129.)

[MAEDCHE and STAAB 2002] Maedche, E. and Staab, S. (2002). *Measuring Similarity between Ontologies*. In *Proceedings of the European Conference on Knowledge Acquisition and Management (EKAW 2002)*, pp. 251–263. (Cited on page 31.)

[MANOLA and MILLER 2004] Manola, F. and Miller, E. (2004). *RDF Primer*. (Cited on page 6.)

[MARTINS and SILVA 2011] Martins, H. and Silva, N. (2011). *A Generic Recommendation System based on Inference and Combination of OWL-DL Ontologies*. Computational Intelligence for Engineering Systems Emergent Applications, 46, pp.134–146. (Cited on page 8.)

[MATKAR and PARAB 2011] Matkar, R. and Parab, A. (2011). *Ontology based expert systems – replication of human learning*. In *Proceedings of the 1st International Conference on Contours of Computing Technology (Thinkquest 2010)*, pp. 43–47. (Cited on page 5.)

[MCCARTHY and HAYES 1987] McCarthy, J. and Hayes, P. J. (1987). *Some philosophical problems from the standpoint of artificial intelligence*. In Ginsberg, M. L. (ed.): *Readings in nonmonotonic reasoning*, pp. 26–45: Morgan Kaufmann Publishers Inc. (Cited on page 9.)

[MCGUINNESS and VAN HARMELEN 2004] Mcguinness, D. L. and van Harmelen, F. (2004). *OWL Web Ontology Language Overview*. W3C Recommendation, W3C. (Cited on page 6.)

[MEILICKE et al. 2007] Meilicke, C., Stuckenschmidt, H., and Tamilin, A. (2007). *Repairing Ontology Mappings*. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI 2007)*, pp. 1408–1413. (Cited on page 34.)

[MEILICKE et al. 2008] Meilicke, C., Stuckenschmidt, H., and Tamilin, A. (2008). *Supporting Manual Mapping Revision using Logical Reasoning*. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI 2008)*, pp. 1213–1218. (Cited on pages 39 and 90.)

[MIDDLETON et al. 2009] Middleton, S. E., Roure, D. D., and Shadbolt, N. R. (2009). *Ontology-Based Recommender Systems*. In Staab, S. and Rudi Studer,

D. (eds.): *Handbook on Ontologies*, International Handbooks on Information Systems, pp. 779–796: Springer Berlin Heidelberg. (Cited on page 8.)

[MIREL 2009] Mirel, B. (2009). *Supporting cognition in systems biology analysis: findings on users' processes and design implications*. Journal of Biomedical Discovery and Collaboration, 4(1), pp.2+. (Cited on page 7.)

[MOTIK et al. 27 October 2009] Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., and Lutz, C. (eds.) (27 October 2009). *OWL 2 Web Ontology Language: Profiles*: W3C Recommendation. Available at `http://www.w3.org/TR/owl2-profiles/`. (Cited on pages 6 and 19.)

[NATALE et al. 2011] Natale, D. A., Arighi, C. N., Barker, W. C., Blake, J. A., Bult, C. J., Caudy, M., Drabkin, H. J., D'Eustachio, P., Evsikov, A. V., Huang, H., Nchoutmboube, J., Roberts, N. V., Smith, B., Zhang, J., and Wu, C. H. (2011). *The Protein Ontology: a structured representation of protein forms and complexes*. Nucleic Acids Research, 39, pp.D539–D545. (Cited on page 6.)

[NIKITINA 2010] Nikitina, N. (2010). *Semi-Automatic Revision of Formalized Knowledge*. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, pp. 1097–1098. (Cited on page 17.)

[NIKITINA 2011] Nikitina, N. (2011). *Forgetting in General EL Terminologies*. In *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*. (Cited on pages 17 and 133.)

[NIKITINA 2012] Nikitina, N. (2012). *OBA: Supporting Ontology-Based Annotation of Natural Language Resources*. In *Proceedings of the 9th Extended Semantic Web Conference (ESWC 2012)*. (Cited on page 11.)

[NIKITINA and GLIMM 2012] Nikitina, N. and Glimm, B. (2012). *Hitting the Sweetspot: Economic Rewriting of Knowledge Bases*. In *Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*. (To Appear). (Cited on page 17.)

[NIKITINA et al. 2011a] Nikitina, N., Glimm, B., and Rudolph, S. (2011a). *Wheat and Chaff – Practically Feasible Interactive Ontology Revision*. In *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*, pp. 487–503. (Cited on page 17.)

[NIKITINA and RUDOLPH 2012] Nikitina, N. and Rudolph, S. (2012). *ExpExpExplosion: Uniform Interpolation in General EL Terminologies*. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*. (Shortlisted for best paper award). (Cited on pages 17, 134 and 149.)

[NIKITINA et al. 2011b] Nikitina, N., Rudolph, S., and Glimm, B. (2011b). *Reasoning-Supported Interactive Revision of Knowledge Bases*. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pp. 1027–1032. (Cited on page 17.)

[NIKITINA et al. 2011c] Nikitina, N., Rudolph, S., and Glimm, B. (2011c). *Reasoning-Supported Interactive Revision of Knowledge Bases*. In *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*. (Cited on page 17.)

[NIKITINA et al. 2012] Nikitina, N., Rudolph, S., and Glimm, B. (2012). *Reasoning-Supported Interactive Revision of Ontologies*. Web Semantics: Science, Services and Agents on the World Wide Web, Special Issue on Reasoning with Context in the Semantic Web, pp.118–130. (Cited on page 17.)

[NOY and MUSEN 2003] Noy, N. F. and Musen, M. A. (2003). *The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping*. International Journal of Human-Computer Studies, 59. (Cited on page 40.)

[OBRST et al. 2007] Obrst, L., Ceusters, W., Mani, I., Ray, S., and Smith, B. (2007). *The evaluation of ontologies*. In Baker, C. J. and Cheung, K.-H. (eds.): *Revolutionizing Knowledge Discovery in the Life Sciences*, pp. 139–158: Springer. (Cited on page 11.)

[OUYANG et al. 2011] Ouyang, L., Zou, B., Qu, M., and Zhang, C. (2011). *A method of ontology evaluation based on coverage, cohesion and coupling*. In *Proceedings of the 8th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2011)*, pp. 2451 –2455. (Cited on page 31.)

[OWL WORKING GROUP 27 October 2009] OWL Working Group, W. (27 October 2009). *OWL 2 Web Ontology Language: Document Overview*: W3C Recommendation. Available at `http://www.w3.org/TR/owl2-overview/`. (Cited on pages 6, 19 and 82.)

[OZDIKIS et al. 2011] Ozdikis, O., Orhan, F., and Danismaz, F. (2011). *Ontology-based recommendation for points of interest retrieved from multiple data sources*. In *Proceedings of the International Workshop on Semantic Web Information Management (SWIM 2011)*, pp. 1–6. (Cited on page 8.)

[PAK and ZHOU 2011] Pak, J. and Zhou, L. (2011). *A Framework for Ontology Evaluation*. In *Proceedings of the Workshop on Exploring the Grand Challenges for Next Generation E-Business*, pp. 10–18. (Cited on page 11.)

[PARK et al. 2011] Park, Y. R., Kim, J., Lee, H. W., Yoon, Y. J., and Kim, J. H. (2011). *GOChase-II: correcting semantic inconsistencies from Gene Ontology-based annotations for gene products.*. BMC Bioinformatics, 12, pp.40+. (Cited on pages 31 and 38.)

[PIERKOT et al. 2011] Pierkot, C., Zimanyi, E., Lin, Y., and Libourel, T. (2011). *Advocacy for External Quality in GIS*. In *Proceedings of the 4th International Conference on GeoSpatial Semantics (GeoS 2011)*, pp. 151–165. (Cited on page 47.)

[PRUD'HOMMEAUX and SEABORNE 2008] Prud'hommeaux, E. and Seaborne, A. (2008). *SPARQL Query Language for RDF*. W3C Recommendation, 4, pp.1–106. (Cited on page 36.)

[QI and YANG 2008] Qi, G. and Yang, F. (2008). *A Survey of Revision Approaches in Description Logics*. In *Proceedings of the 21st International Workshop on Description Logics (DL 2008)*. (Cited on page 53.)

[RADULOVIC and GARCIA-CASTRO 2011] Radulovic, F. and Garcia-Castro, R. (2011). *Towards a Quality Model for Semantic Technologies*. In *Proceedings of the 5th International Conference on Computational Science and Its Applications (ICCSA 2011)*, pp. 244–256. (Cited on page 11.)

[RASKIN and PAN 2005] Raskin, R. G. and Pan, M. J. (2005). *Knowledge representation in the semantic web for Earth and environmental terminology (SWEET)*. Computers & Geosciences, 31(9), pp.1119–1125. (Cited on page 6.)

[RECTOR et al. 1994] Rector, A., Gangemi, A., Galeazzi, E., Glowinski, A., and Rossi-Mori, A. (1994). *The GALEN CORE Model Schemata for Anatomy: Towards a Re-usable Application-Independent Model of Medical Concepts*. In *Proceedings of the 12th International Congress of the European Federation for Medical Informatics (MIE 1994)*, pp. 229–233. (Cited on page 6.)

[RECTOR et al. 2011] Rector, A. L., Iannone, L., and Stevens, R. (2011). *Quality assurance of the content of a large DL-based terminology using mixed lexical and semantic criteria: experience with SNOMED CT*. In *Proceedings of the 6th International Conference on Knowledge Capture (K-CAP 2011)*, pp. 57–64. (Cited on page 31.)

[RETTINGER et al. 2012] Rettinger, A., Lösch, U., Tresp, V., d'Amato, C., and Fanizzi, N. (2012). *Mining the Semantic Web - Statistical learning for next generation knowledge bases*. Data Mining and Knowledge Discovery, 24, pp.613–662. (Cited on page 10.)

[ROGERS et al. 1998] Rogers, J., Price, C., Rector, A., Solomon, W., and Smejko, N. (1998). *Validating Clinical Terminology Structures: Integration and Cross-Validation of Read Thesaurus and GALEN*. In *Proceedings of the AMIA Fall Symposium*, pp. 845–849. (Cited on page 38.)

[ROGOZAN and PAQUETTE 2005] Rogozan, D. and Paquette, G. (2005). *Managing Ontology Changes on the Semantic Web*. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI 2005)*, pp. 430–433. (Cited on page 31.)

[ROYAL SOCIETY OF CHEMISTRY 2012] Royal Society of Chemistry (2012). *Chemical Methods Ontology (CMO)*. http://www.rsc.org/ontologies/CMO/index.asp. (Cited on page 12.)

[RUSSELL and NORVIG 2002] Russell, S. J. and Norvig, P. (2002). *Artificial Intelligence: A Modern Approach (2nd Edition)*: Prentice Hall. (Cited on page 4.)

[SATOH 1988] Satoh, K. (1988). *Nonmonotonic Reasoning by Minimal Belief Revision*. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pp. 455–462. (Cited on page 53.)

[SCHLOBACH and CORNET 2003] Schlobach, S. and Cornet, R. (2003). *Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies*. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pp. 355–362. (Cited on page 53.)

[SCHNOES et al. 2009] Schnoes, A. M., Brown, S. D., Dodevski, I., and Babbitt, P. C. (2009). *Annotation Error in Public Databases: Misannotation of Molecular Function in Enzyme Superfamilies*. PLoS Computational Biology, 5(12), pp.e1000605. (Cited on page 32.)

[SCHOCKAERT and PRADE 2010] Schockaert, S. and Prade, H. (2010). *An Inconsistency-Tolerant Approach to Information Merging Based on Proposition Relaxation*. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI 2010)*. (Cited on page 35.)

[SCHRIML et al. 2012] Schriml, L. M. M., Arze, C., Nadendla, S., Chang, Y.-W. W. W., Mazaitis, M., Felix, V., Feng, G., and Kibbe, W. A. A. (2012). *Disease Ontology: a backbone for disease semantic integration.*. Nucleic Acids Research, 40(Database issue), pp.D940–D946. (Cited on page 7.)

[SEIDENBERG and RECTOR 2006] Seidenberg, J. and Rector, A. L. (2006). *Web ontology segmentation: analysis, classification and use*. In *Proceedings of the*

15th International Conference on World Wide Web (WWW 2006), pp. 13–22. (Cited on page 40.)

[SERAFINI and TAMILIN 2005] Serafini, L. and Tamilin, A. (2005). *DRAGO: distributed reasoning architecture for the semantic web*. In *Proceedings of the 2nd European Semantic Web Conference (ESWC 2005)*, pp. 361–376. (Cited on pages 39 and 90.)

[SHCHEKOTYKHIN et al. 2012] Shchekotykhin, K., Friedrich, G., Fleiss, P., and Rodler, P. (2012). *Interactive ontology debugging: Two query strategies for efficient fault localization*. Web Semantics: Science, Services and Agents on the World Wide Web, 12(0), pp.88–103. (Cited on page 35.)

[SHEARER and HORROCKS 2009] Shearer, R. and Horrocks, I. (2009). *Exploiting Partial Information in Taxonomy Construction*. In *Proceedings of the 8th International Semantic Web Conference (ISWC 2009)*, pp. 569–584. (Cited on page 57.)

[SIOUTOS et al. 2007] Sioutos, N., Coronado, S. d., Haber, M. W., Hartel, F. W., Shaiu, W.-L., and Wright, L. W. (2007). *NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information*. Journal of Biomedical Informatics, 40(1), pp.30–43. (Cited on page 6.)

[SMITH et al. 2007] Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungall, C. J., Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.-A., Scheuermann, R. H., Shah, N., Whetzel, P. L., and Lewis, S. (2007). *The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration*. Nature Biotechnology, 25(11), pp.1251–1255. (Cited on page 6.)

[SMITH 2011] Smith, L. (2011). *Applying semantic search and ontology to the travel industry*. (Cited on page 7.)

[SONG et al. 2005] Song, M.-H., Lim, S.-Y., Kang, D.-J., and Lee, S.-J. (2005). *Automatic Classification of Web Pages based on the Concept of Domain Ontology*. In *Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC 2005)*, pp. 645–651. (Cited on page 8.)

[SOWA 2000] Sowa, J. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*: Course Technology. (Cited on page 5.)

[SPACKMAN et al. 1997] Spackman, K. A., Campbell, K. E., and Cote, R. A. (1997). *SNOMED RT: A reference terminology for health care*. In *Proceedings of the AIMA Fall Symposium*, pp. 640–644. (Cited on page 6.)

[STEVENS and LORD 2009] Stevens, R. and Lord, P. (2009). *Application of Ontologies in Bioinformatics*. In Staab, S. and Studer, R. (eds.): *Handbook on Ontologies*, International Handbooks Information System, pp. 735–756: Springer Berlin Heidelberg. (Cited on page 7.)

[STUCKENSCHMIDT et al. 2009] Stuckenschmidt, H., Parent, C., and Spaccapietra, S. (eds.) (2009). *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*: Springer Berlin Heidelberg. (Cited on page 41.)

[STVILIA 2007] Stvilia, B. (2007). *A model for ontology quality evaluation*. First Monday, 12. (Cited on pages 11 and 48.)

[SUCHANEK et al. 2008] Suchanek, F. M., Kasneci, G., and Weikum, G. (2008). *YAGO: A Large Ontology from Wikipedia and WordNet*. Web Semantics: Science, Services and Agents on the World Wide Web, 6, pp. 203–217. (Cited on page 10.)

[SUPEKAR 2005] Supekar, K. (2005). *A peer-review approach for ontology evaluation*. In *Proceedings of the 8th International Protégé Conference*. (Cited on page 47.)

[SUPEKAR 2004] Supekar, K. (2004). *Characterizing quality of knowledge on semantic web*. In *Proceedings of the AAAI Florida AI Research Symposium (FLAIRS 2004)*. (Cited on page 48.)

[SUREEPHONG et al. 2008] Sureephong, P., Chakpitak, N., Ouzrout, Y., and Bouras, A. (2008). *An Ontology-based Knowledge Management System for Industry Clusters*. Computing Research Repository, abs/0806.0526. (Cited on page 7.)

[TARTIR et al. 2005] Tartir, S., Arpinar, I. B., Moore, M., Sheth, A. P., and Aleman-meza, B. (2005). *OntoQA: Metric-based ontology quality analysis*. In *Proceedings of the IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*. (Cited on page 48.)

[TARTIR et al. 2009] Tartir, S., McKnight, B., and Arpinar, I. B. (2009). *SemanticQA: web-based ontology-driven question answering*. In *Proceedings of the 2009 ACM Symposium on Applied Computing*, pp. 1275–1276. (Cited on page 8.)

[TIPNEY and HUNTER 2010] Tipney, H. and Hunter, L. (2010). *Knowledge-Driven Approaches to Genome-Scale Analysis*: John Wiley & Sons, Ltd. (Cited on page 7.)

[TOBIES 2001] Tobies, S. (2001). *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen. (Cited on page 29.)

[UNGER et al. 2012] Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.-C., Gerber, D., and Cimiano, P. (2012). *Template-based question answering over RDF data*. In *Proceedings of the 21th International Conference on World Wide Web (WWW 2012)*, pp. 639–648. (Cited on page 8.)

[VARGAS-VERA et al. 2003] Vargas-Vera, M., Motta, E., and Domingue, J. (2003). *AQUA: an ontology driven question answering system*. In *Proceedings of the AAAI Spring Symposium, New Directions in Question Answering*. (Cited on page 8.)

[VERSPOOR et al. 2009] Verspoor, K., Dvorkin, D., Cohen, K. B., and Hunter, L. (2009). *Ontology quality assurance through analysis of term transformations*. Bioinformatics, 25(12), pp.77–84. (Cited on page 31.)

[VÖLKER 2009] Völker, J. (2009). *Learning Expressive Ontologies*. Studies on the Semantic Web: Aka. (Cited on page 10.)

[VÖLKER et al. 2005] Völker, J., Vrandečić, D., and Sure, Y. (2005). *Automatic evaluation of ontologies (AEON)*. In *Proceedings of the 4th European Semantic Web Conference (ESWC 2005)*, pp. 716–731. (Cited on page 36.)

[VRANDEČIĆ 2010] Vrandečić, D. (2010). *Ontology Evaluation*. Phdthesis, KIT, Fakultät für Wirtschaftswissenschaften, Karlsruhe. (Cited on pages 11 and 36.)

[VRANDEČIĆ and GANGEMI 2006] Vrandečić, D. and Gangemi, A. (2006). *Unit tests for ontologies*. In *Proceedings of the 1st International Workshop on Ontology Content and Evaluation in Enterprise*. (Cited on page 36.)

[WANG et al. 2009a] Wang, K., Wang, Z., Topor, R., Pan, J. Z., and Antoniou, G. (2009a). *Concept and Role Forgetting in ALC Ontologies*. In *Proceedings of the 8th International Semantic Web Conference (ISWC 2009)*, pp. 666–681. (Cited on page 45.)

[WANG et al. 2009b] Wang, Z., Wang, K., Topor, R., Pan, J. Z., and Antoniou, G. (2009b). *Uniform Interpolation for ALC Revisited*. In *Proceedings of the 22nd Australasian Joint Conference on Advances in Artificial Intelligence (AI 2009)*, pp. 528–537. (Cited on page 45.)

[WANG et al. 2010] Wang, Z., Wang, K., Topor, R., and Zhang, X. (2010). *Tableau-based Forgetting in ALC Ontologies*. In *Proceedings of the19th European Conference on Artificial Intelligence (ECAI 2010)*, pp. 47–52. (Cited on page 45.)

[WANG et al. 2008] Wang, Z., Wang, K., Topor, R. W., and Pan, J. Z. (2008). *Forgetting Concepts in DL-Lite*. In *Proceedings of the 5th European Semantic Web Conference (ESWC 2008)*, pp. 245–257. (Cited on page 46.)

[WHEELER et al. 2000] Wheeler, D. L., Chappey, C., Lash, A. E., Leipe, D. D., Madden, T. L., Schuler, G. D., Tatusova, T. A., and Rapp, B. A. (2000). *Database resources of the National Center for Biotechnology Information.*. Nucleic Acids Research, 28(1), pp.10–14. (Cited on page 7.)

[XIE and BURSTEIN 2011] Xie, J. and Burstein, F. (2011). *Using Machine Learning to Support Resource Quality Assessment: An Adaptive Attribute-Based Approach for Health Information Portals*. In *Proceedings of the 16th International Conference on Database Systems for Advanced Applications (DASFAA 2011)*, pp. 526–537. (Cited on page 47.)

[YEH et al. 2011] Yeh, P. Z., Puri, C. A., Wagman, M., and Easo, A. K. (2011). *Accelerating the Discovery of Data Quality Rules: A Case Study.*. In *Proceedings of the 23rd Innovative Applications of Artificial Intelligence Conference (IAAI 2011)*. (Cited on page 37.)