Alignment Incoherence in Ontology Matching

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

vorgelegt von
Christian Meilicke

Mannheim, Juni 2011

Contents

List of Figures

List of Tables

Acknowledgement

Two weeks before writing this paragraph my nephew asked me about the day I will come back to play with him. I told him that I am very busy because I have to finish my thesis. He asked about it and I told him that it is something complicated and that I have a boss who wants me to do this thesis and some other things at the same time. He thought for a while, gave me a pack of effervescent powder, and told me that I should take some of this whenever I get tired. It will help me to continue in order to finish soon.

Thanks to the ones who helped me to finish and thanks to the ones who helped me to start. Thanks to Heiner for his trust at the beginning and for tolerating my complaints related to the sense of it all at the end. And a special thanks to Andrei who showed me the way back home.

Thanks also to Cássia, Ondrej, and Jérôme for having the chance to work with them in the context of OAEI, and thanks to Mathias and Jan for fruitful discussions related to the development of the CODI matching system.

Part I

Motivation & Foundation

Chapter 1

Introduction

That is why its name was called Babel, because there Jehovah had confused the language of all the earth, and [...] scattered them from there over all the surface of the earth (Genesis 11).

The starting quote of this thesis is taken from the story about the Tower of Babel, which can be found in the Book of Genesis. This story explains the origin of varying languages. Long time ago the whole world had one language common for all people. Due to this, the human people could communicate with each other and became skillful in handcraft and science. This enabled them to build an enormous tower, known as the Tower of Babel, which reached up to heaven. God was not pleased about the arrogance of the human people. He decided to destroy the Tower of Babel. In addition, God scattered them all over the world and made the resulting tribes talk in different languages to hinder that the same will happen again.

In this thesis we are concerned with a specific problem related to the integration of different knowledge resources. Such integration problems can typically be found in the so called Semantic Web [BLHL01]. The Semantic Web can be understood as a loosely coupled network of different knowledge resources described with different conceptualizations. These conceptualizations will be introduced as ontologies later on. The world after Gods intervention described in the story of the Tower of Babel can be seen as a metaphor for the situation in the Semantic Web. Different ontologies correspond to different languages, different knowledge resources in the World Wide Web correspond to the tribes (and their knowledge) scattered across the world, the possibility to built the Tower of Babel symbolizes the benefits and prospects that we associate with the vision of the Semantic Web.

A key to a successful integration of different knowledge resources is a dictionary that allows to translate between the underlying ontologies. Such a dictionary explains the meaning of terms from one ontology by terms of the other ontology. The process of creating such a dictionary is referred to as ontology matching [ES07]. In this thesis we are concerned with a specific topic in the field of ontology matching. This is the use of reasoning techniques and especially the role of incoherence in ontology matching. Reasoning with ontology alignments has been identified by Shvaiko and Euzenat as one of the ten outstanding challenges in ontology matching [SE08].

The remaining parts of the introduction are structured as follows. First, we introduce ontology matching as key component for the success of the Semantic Web in Section 1.1. In Section 1.2 we present a concrete ontology matching problem and illustrate the basic principle underlying the approach of this thesis. Then we present the central research questions of this thesis in Section 1.3. Finally, we outline the structure of the thesis in Section 1.4.

1.1 Ontology Matching and the Semantic Web

The notion of the Semantic Web has been coined by a famous article from Berners-Lee et al. [BLHL01]. The main idea has been taken up by many researchers and it is probably one of the most cited papers in younger computer science[1]. While the World Wide Web is meant to be used by humans, the Semantic Web is intended to be a web of data accessible and understandable by computer programs. The path to reach this goal is paved with several layers of formalizations and language recommendations. At the bottom, URIs (Uniform Resource Identifier) provide a mechanism for uniquely identifying resources and Unicode ensures a standardized format with a universal encoding. A layer upwards we find the specification of XML (eXtensible Markup Language) and RDF (Resource Description Framework). XML is used to describe data in a structured way, however, any kind of semantics is missing. Contrary to this, in RDF all statements are triples with a subject, a predicate and an object. This allows to clearly identify statements and their components.

On top of this layer we find the specification of ontologies as a formal framework. Ontologies are sets of axioms that define the relations between entities (concepts and properties) of a vocabulary. An ontology can, for example, be used to express that an *author* is a *person*, that each *person* has a *forename* and a *surname*, that everything *written by* an author is a *document*, and that there exists nothing which is both a *person* and a *document*. This vocabulary can then be used within an RDF triple to describe a piece of data . We can, for example, express that a certain book is written by a specific author and we also know that the book is a document and the author is a person with all properties attached to a person. According to this approach, resources can be described by a powerful, logic-based framework that enables to reason over the described data.

According to the vision of the Semantic Web such an approach will be used by many different users and communities to describe their data [SBL06] (e.g., governments and public authorities, companies, scientific institutions, libraries). This

[1] According to Google Scholar it has currently been cited 10821 times (04.04.2011).

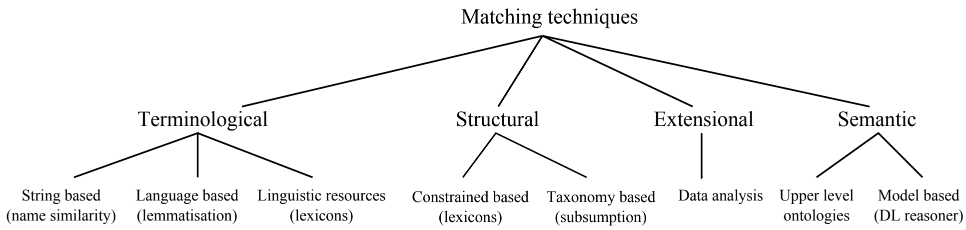

Figure 1.1: Classification of ontology matching methods. The figure shows a simplified variant of the categorization proposed [ES07]. It has been inspired by a similar classification presented in [vH08].

results in the major problem of integrating data from different sources of information. Even though two knowledge sources S_1 and S_2 are described by the use of ontologies, referred to as $\mathcal{O}_1$ and $\mathcal{O}_2$ in the following, we cannot conclude that the knowledge of S_1 is usable from the perspective of S_2 or vice versa. This can only be guaranteed

- if $\mathcal{O}_1$ and $\mathcal{O}_2$ use the same concepts and properties with the same or at least with a similar meaning, or

- if there exists a dictionary that allows to translate between the vocabulary of $\mathcal{O}_1$ and $\mathcal{O}_2$.

The first solution might be applicable in some areas, however, the Semantic Web will probably consist of knowledge sources that are described by different or only slightly overlapping ontologies. Thus, the overall success of the approach critically depends on the availability of a dictionary that allows to translate between different conceptualizations.

Such a dictionary will later be introduced formally as an ontology alignment. Ontology matching can be defined as the subdiscipline of artificial intelligence that deals with the generation of such alignments [ES07]. Generating an alignment is a challenging task. This holds both if the alignment is generated by a human expert and if it is generated by a matching algorithm in an automated setting. Note that an ontology might comprise thousands of terms that are described by a rich set of axioms, which can hardly be taken into account at once by a human expert. Other scenarios might rely on the agents metaphor, i.e., the conception that an agent searches for a specific information in different knowledge sources and combines the obtained information in a meaningful way. The agent has to use a matching algorithm to conduct the data integration without human intervention. In both scenarios it is hard to ensure that the alignment is complete and correct with respect to most of its parts.

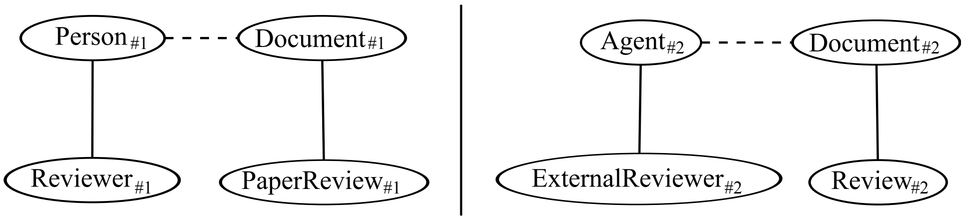

Figure 1.2: Example of a simple matching problem.

1.2 Problem Statement

A large amount of different methods to solve the problem of automatically generating an ontology alignment have been proposed. In Figure 1.1 some of these methods are presented within a rough and probably incomplete classification. See [KS03, SE05, ES07, vH08] for a detailed overview on these and other methods.

Many ontology alignment tools are based on a set of terminological methods. Basic terminological methods compare concepts and property names on the string level (e.g., computing an edit distance). Names can also be compared after applying a stemming algorithm. Moreover, lexicons can be used to detect that different strings might refer to the same concept. Other methods, described as structural methods in Figure 1.1, analyze the subsumption hierarchy. They assign a higher probability to a correspondence between two concepts iff the parent concepts of these concepts have already been matched. In a similar way other structural properties can be exploited. As already argued, the vocabulary of an ontology is used to describe concrete data instances. These instances can by analyzed within an extensional approach. Finally, there are also model-based techniques. Some of them are based on the use of a reasoner. Most of all matching systems aggregate several of these methods within different functional layers.

In this thesis we present a special kind of model-based approach, namely an approach that is concerned with the notion of alignment coherence. We will see in the following chapters, that it is not just a matching technique but rather a principle that both helps to guide the matching process and allows to judge the quality of its outcome from a logical point of view. In the following we illustrate the underlying idea by discussing a simple ontology matching problem.

Figure 1.2 shows parts from two ontologies $\mathcal{O}_1$ and $\mathcal{O}_1$ that describe the same domain with a different vocabulary. Each of these ontologies is built from four concepts. The relations between these concepts are depicted by solid and dashed lines. If two concepts are connected by a solid line, then the lower concept is more specific than the upper concept (e.g., each $Reviewer_{\#1}$ is a $Person_{\#1}$). If two concepts are connected by a dashed line, then these concepts do not overlap (e.g., there exists nothing which is both a $Person_{\#1}$ and a $Document_{\#1}$). We use subscripts $_{\#1}$ and $_{\#2}$ to distinguish between concepts from $\mathcal{O}_1$ and $\mathcal{O}_2$.

Now suppose that we run a matching system to detect which of these concepts are equivalent. Suppose the following equivalences are part of our results. We will later formally introduce the notion of a correspondence to refer to these equivalences.

$$\textit{Document}_{\#1} \textit{ is equivalent to } \textit{Document}_{\#2} \tag{1.1}$$

$$\textit{Reviewer}_{\#1} \textit{ is equivalent to } \textit{Review}_{\#2} \tag{1.2}$$

Due to the meaning that is attached to the words *'Reviewer'* and *'Review'* we know that the second correspondence is incorrect. These equivalences also result in logical problems detectable by a logical reasoner. First, it can be concluded that some x, which is a $\textit{Reviewer}_{\#1}$, is a $\textit{Review}_{\#2}$ (from 1.2). Thus, x is also a $\textit{Document}_{\#2}$ (from the second ontology). It follows that each $\textit{Reviewer}_{\#1}$ is a $\textit{Document}_{\#1}$ (from 1.2). However, at the same time it can be entailed from the first ontology that there exists no $\textit{Reviewer}_{\#1}$ that is also a $\textit{Document}_{\#1}$. This is obviously a logical contradiction and we will later introduce the technical notion of alignment incoherence to describe this situation.

In our simple example we can resolve the problem by removing one of the equivalences. Our logical analysis cannot help us to chose the right one, we have to base our decision on another source of information. Many methods generate a so called confidence value, which expresses the degree of trust in the correctness of a correspondence [ES07]. Given such a confidence value, it might make sense to remove the correspondence with lower confidence.

However, there are more complex ways in which alignments and ontologies can interfere to cause problems of that type. Moreover, often different subsets of an alignment result in logical problems for different parts of the vocabulary described in an ontology. These subsets can overlap and then there are several alternatives to resolve the incoherence. The simple heuristics proposed above is not sufficient in this case and we have to find another way to solve the problem.

1.3 Research Questions

This thesis is concerned with the impact of alignment incoherence, with the techniques to detect them, with the algorithms to resolve them, and with the effects of doing so. In particular, we try to answer the following questions.

R1 What are the consequences of using an incoherent alignment in an application?

R2 Is there an interrelation between the coherence of an alignment and its correctness or completeness?

R3 Given an incoherent alignment, how can we characterize types of coherent sub-alignments in a reasonable way?

R4 Which algorithms can be used to compute these types of coherent subalignments?

R5 Which type of coherent subalignment scores best, compared to a gold standard, if applied in an automated ontology matching scenario?

R6 How do the algorithms perform with respect to their runtime?

R7 How can alignment incoherence be exploited to support a user in revising an existing alignment?

The answers to R1 and R2 are crucial for this thesis. If we cannot show that alignment incoherence has negative consequences in an application scenario and if we do not find any interrelation between alignment coherence and and its correctness or completeness, we do not need to search for an answer to the other research questions. It does not make sense to discuss, for example, algorithms for computing coherent alignments, as long as we do not have a motivation for generating such alignments.

However, this motivation can be given in two ways, namely via an adequate answer to R1 or R2. In the context of R1 we discuss several application scenarios and we argue that an incoherent alignment results in severe problems for this kind of application. This answer is already a sufficient motivation for our approach. If we cannot present a conclusive line of reasoning with respect to R1, we will not consider alignment coherence as an end in itself. However, we can still argue in the context of R2 that we can indirectly benefit from alignment coherence as a guiding principle that enables us, given a set of matching hypotheses, to determine an alignment with a high degree of completeness and correctness. Finally, it is up to the reader if he accepts both arguments, one of them, or none.

While R1 and R2 are concerned with motivating the significance of our work, R3 and R4 are concerned with theory, methods and algorithms to resolve alignment incoherence. We place our approach within the theoretical framework of system diagnosis [Rei87]. We define a diagnosis as subset of an alignment that has to be removed such that the resulting alignment is coherent. In particular, we define two types of diagnosis. With the distinction of R3 and R4 we highlight the difference between characterizing the solution to a problem, in our case a type of diagnosis, and the algorithm to determine the solution given a concrete problem instance. For each type we present two algorithms and show that they always generate the specific type of diagnosis.

The following research questions (R5, R6, and R7) are concerned with an empirical analysis of our approach. R5 is an extrapolation of R2. While we give an answer to R2 that is more or less based on theoretical considerations, we try to answer R5 by a comprehensive set of experiments. However, these experiments are concerned with the diagnostic approach we have proposed as answer to R3. That means that our experiments are concerned with a well-founded, but specific way to resolve incoherence. R2 is, contrary to this, concerned with a general issue.

With R6 we analyze in how far our approach is applicable in realistic application scenarios. In particular, we want to find out whether the algorithms can resolve incoherence within an acceptable time frame. Finally, we have to analyze if the overall approach can also be used to support a human in the loop (R7).

1.4 Outline and Contribution

This thesis is structured in four parts. In the first part we introduce and motivate the approach presented in this thesis. In particular, we develop a formalization, which allows us to define alignment incoherence. On top of this framework we give an answer to R1 (consequences of alignment incoherence) and at least a partial answer to R2 (alignment quality in terms of correctness and completeness). The second part is concerned with methods and algorithms to detect and resolve alignment incoherence. As an answer to R3 we introduce the notion of alignment diagnosis. We define a diagnosis as a minimal subset of an alignment that has to be removed to achieve a coherent alignment. Our answer to R4 is a proposal of several algorithms to compute different types of diagnoses. In the third part we turn our head to an empirical verification of these algorithms (R5, R6, R7). In particular, we discuss and present results for several application scenarios. Finally, we give an overview on related work and end with a conclusion, where we give an explicit answer to our research questions.

Overview on Part I

In Chapter 2 we develop a theoretical framework that allows us to formally introduce the notion of alignment coherence. A precise definition requires to start first of all with a definition of syntax and semantics of ontologies (Section 2.1). Hereby, we focus on OWL-DL and formally introduce $\mathcal{SHOIN}(\mathbf{D})$. Based on this we define an alignment as a set of correspondences (Section 2.2). Again, we distinguish between syntax and semantics. Our main contribution is related to the second aspect. In particular, we reduce the semantics of a correspondence to the semantics of an axiom in an ontology that is the union of the aligned ontologies and the alignment. This approach allows to reduce alignment (in)coherence to the (in)coherence of an ontology. Thus, we can profit from a well-defined formalization.

Chapter 3 is divided in two sections. In Section 3.1 we discuss the relation between correctness and coherence of an alignment. This section is concerned with R2. It touches philosophical issues related to conceptual relativism, namely the question if there exists for two arbitrary conceptual schemes (ontologies) a coherent way to translate between them. In Section 3.2 we are back on solid grounds and discuss the problem of alignment incoherence from an application-oriented perspective. In particular, we point to problems that may result from incoherent alignments. Our insights will help us to give an answer to R1.

Overview on Part II

Chapter 4 introduces the notion of alignment diagnosis. In doing so we also adapt some notions from the field of ontology debugging to the field of alignment debugging. This is for example the notion of a minimal incoherence preserving sub-alignment (MUPS alignment). Contrary to the classical approach, in ontology matching correspondences are annotated with confidence values. Given an incoherent alignment, there are several diagnoses to solve the problem. We define a local optimal and a global optimal diagnosis as two reasonable types of diagnoses. Both take into account the role of confidence values. We illustrate the differences between these types of diagnosis at hand of several examples. This chapter is this dedicated to R3.

In Chapter 5 we describe the building blocks of the algorithms to be introduced in the following chapter. Both sections of this chapter are concerned with reasoning techniques to decide alignment coherence, to detect unsatisfiable concepts, and to determine minimal unsatisfiability preserving alignments. However, in Section 5.1 we are concerned with complete methods, which are more or less state of the art in ontology debugging, while in Section 5.2 we propose a novel pattern based reasoning approach, that is incomplete with respect to detecting incoherence, but efficient with respect to runtime.

In Chapter 6 we show how to use the reasoning components developed in Chapter 5 to compute the diagnosis defined in Chapter 4. We propose for both local (Section 6.1) and global optimal diagnosis (Section 6.2) two algorithms: an algorithm that computes the diagnosis without using the pattern based approach and an algorithm that combines pattern based and complete reasoning methods. The challenge hereby is to design an algorithm that is efficient but still complete. These algorithms are our answer to research question R4.

Overview on Part III

In Chapter 7 we describe datasets used in the following experiments in Section 7.1. In Section 7.2 we propose a way to measure the degree of alignment incoherence. As an additional insight, closely related to R2, we point to a non-trivial relation between alignment coherence and correctness. Finally, we apply our measure in Section 7.3 to alignments generated by a rich set of systems listed in Appendix C. Our experiments show that most matching systems generate highly incoherent alignments.

We continue with two Chapters, which both focus on research question R5. In particular, we want to understand the impact of our methods on the quality of alignments in terms of precision and recall. Thereby, we focus on two different application scenarios. The first scenario (Chapter 8) is that of automatically repairing incoherent alignments generated by state of the art ontology matching systems. We study how our methods perform with respect to three important datasets used in the Ontology Alignment Evaluation Initiative [EMS$^+$11]. In addition we are in

Section 8.3 concerned with runtime efficiency and scalability (subject to R6).

In Chapter 9 we integrate our approach in the step of selecting a final alignment from a set of matching hypotheses. Thus we analyze what happens when we combine state-of-the-art extraction methods with the extraction of an optimal coherent alignment. We argue that a combined approach can increase not only precision but also recall compared to the sequential approach analyzed in the previous Chapter.

Finally, we analyze in how far the approach can be used to support a human expert in revising an automatically generated alignment in Chapter 10 (R7). In Section 10.1 we describe the approach on an abstract level. We present a revision support tool that we developed on top of our approach in Section 10.2. With the help of an example we illustrate that a user can profit from being informed about intended and unintended consequences of his decisions. In Section 10.3 we report on experiments concerned with the amount of manual effort that can be saved by the approach.

Overview on Part IV

In Chapter 11 we present related work from four different areas. In Section 11.1 we are concerned with specific approaches for alignment debugging similar to our approach. In Section 11.2 we present Dungs theory of argumentation [Dun95] and explain why we have chosen a different conceptual foundation. In Section 11.3 we present other approaches on supporting a user in the revision of an alignment. Finally, we focus on coherence-preserving techniques implemented on top of different matching systems in Section 11.4. We analyze exemplarily the relevant components of those systems that are most advanced. We conclude with a summary in Section 11.5.

Finally, we end this thesis in Chapter 12 with a conclusion. First, we summarize the main results in Section 12.1 in the light of our research questions R1-R7. We continue with a critical discussion of our approach and point to future work in Section 12.2. In Section 12.3 we end with some closing remarks.

Some parts of this thesis have already been published by the author in other places. For that reason we have added a paragraph at the end of each chapter introduction. In this paragraph we explain the origin of the material and point to completely new material explicitly. Within the main text we do not describe the origin of each line of thought, example or definition, nor do we describe minor modifications applied to it as long as they origin from the work of the author. This allows to present the contents of this thesis as a coherent piece of work with a focus on the central theme.

Chapter 2

Preliminaries

Conceptual schemes, we are told, are ways of organizing experience; they are systems of categories that give form to the data of sensation; they are points of view from which individuals, cultures, or periods survey the passing scene (Donald Davidson).[1]

In this chapter we present the formal foundations required to describe the problems and solutions presented later on. The two central notions are the notion of an ontology (Section 2.1) and the notion of an alignment (Section 2.2).

The word *'ontology'* has a multifaceted meaning. It is used in different disciplines of philosophy as well as in modern computer science. The starting quote of this section is borrowed from a philosophical essay and refers to the notion of a *'conceptual scheme'* as a way of organizing and structuring information by an individual or a collective of individuals; the notion of an ontology, as used in the context of computer science, refers to a formalized description of such a conceptual scheme. A conceptual scheme becomes also manifest in the use of the natural language, which is guided by a set of vague and implicit rules. Contrary to this, ontologies are aimed to be formal representations with a clear specification of syntactic rules and a well-defined semantics.

While an ontology is a formal representation of a conceptual scheme, an alignment allows to translate between two conceptual schemes. With regard to natural language, it can be compared to a dictionary for which each entry specifies that two words have the same or at least a similar meaning. An alignment can be defined on top of a formal framework that describes syntax and semantics in a unique and well-defined way. In particular, we introduce the notion of a reductionistic alignment semantics, which allows us to make use of the well known terminology and model theoretic semantics of description logics.

[1]The starting quote of this chapter origins from a famous philosophical article of Donald Davidson entitled "On the Very Idea of a Conceptual Scheme" [Dav74], where Davidson argues against the possibility of an untranslatable language.

This chapter is divided in two sections. In Section 2.1 a formal introduction to description logics is given. Most of the definitions can be found in slightly modified variants in many publications related to the general field of description logics (see for example *The Description Logics Handbook* [BCM$^+$03]). Contrary to this, most definitions and propositions introduced in Section 2.2 are our own contribution. They are centered around the notion of an alignment. As long as they are not marked differently, they origin from papers we previously published [MS07b, MTS07, MS08, MS09b] and are presented now in a unified fashion within this thesis.

2.1 Ontologies

In the introduction of this chapter we described ontologies as conceptual schemes. In the following we focus on their representation within a formal framework. In particular, we introduce the description logic underlying the Web Ontology Language (OWL). OWL has become a widely accepted standard to describe ontologies within the Semantic Web. According to [AvH09], some of the main requirements of an ontology language are (a) a well-defined syntax, (b) a well-defined semantics, (c) efficient reasoning support, and (d) sufficient expressive power. Requirements (c) and (d) are obviously in a conflict. The more expressive the language is, the less efficient reasoning support can be expected. For that reason three different sub languages have been defined with increasing expressive power: OWL-Lite, OWL-DL and OWL-Full. We focus in the following on OWL-DL. More precisely, we formally introduce $\mathcal{SHOIN}(\mathbf{D})$, which is the well-known Description Logics dialect underlying OWL-DL [HPS04].[2]

We have chosen OWL-DL as an example of an ontology language because it is expressive enough to show in how far specialized reasoning strategies become necessary for reasoning with ontology alignments, while it still permits efficient reasoning which makes it applicable to real word applications. Another reason for choosing a concrete ontology language is related to the point that a concrete exposition has to be preferred to a generic one from a didactical point of view. However, the overall approach proposed in this thesis is applicable to any logic that supports the most fundamental notions as satisfiability and entailment.

2.1.1 Syntax

Within natural language we use a finite vocabulary of atomic expressions and a set of rules, referred to as grammar, to construct well-formed and meaningful expressions and sentences. In the context of an ontology language the vocabulary is called signature and can be defined as follows.

[2] We have to point out that our approach, although applicable to $\mathcal{SHOIN}(\mathbf{D})$ ontologies, can only ensure coherent results for $\mathcal{SHIN}(\mathbf{D})$ ontologies. This detail will be clarified later on, when we explain the meaning of the letter $\mathcal{O}$ in $\mathcal{SHOIN}(\mathbf{D})$.

Definition 1 (Signature)**.** *A signature* $\mathbf{S}$ *is a quadruple* $\mathbf{S} = \langle \mathbf{C}, \mathbf{P}, \mathbf{R}, \mathbf{I} \rangle$ *where* $\mathbf{C}$ *is a set of concept names,* $\mathbf{P}$ *is a set of object property names,* $\mathbf{R}$ *is a set of data property names, and* $\mathbf{I}$ *is a set of individual names. The union* $\mathbf{P} \cup \mathbf{R}$ *is referred to as the set of property names.*

Definition 1 distinguishes between four types of names regarding their role in constructing complex expressions. Individual names correspond to proper names in natural language expressions, while concept and property names correspond to monadic and binary predicates. Sentences 2.1 and 2.2 contain phrases that correspond to individual names (underlined), monadic predicates (in 2.1, not underlined) and binary predicates (in 2.2, not underlined).

Black Beauty is a racehorse. (2.1)

Bucephalus is the horse of Alexander the Great. (2.2)

Property names can be divided in names of object and data properties (sometimes also referred to as abstract and concrete roles). The letter $\mathbf{D}$ in the abbreviation $\mathcal{SHOIN}(\mathbf{D})$ refers to the data theory required to define data properties. A data theory is a mapping from a datatype to sets of values and a mapping from a set of constants to the values denoted by these constants [HPS04]. While an object property relates instances from the abstract domain with each other, a data property relates instances from the abstract domain with concrete data values (e.g., String, or Integer). Example (2.3) is a sentence that can be translated into DL by the use of a data property.

Alexander the Great is entitled "Alexander III of Macedon". (2.3)

Within DL complex expressions can be constructed from the atomic names of the signature. These expressions are referred to as descriptions. Each letter in $\mathcal{SHOIN}$ stands for specific combination rules for constructing complex concepts and properties.[3] An example is the letter $\mathcal{I}$, which refers to constructing the inverse property from a property name by adding a superscript $^{-1}$ to the property name. The following definition describes how to construct property descriptions.

Definition 2 (Property description)**.** *Given a signature* $\mathbf{S} = \langle \mathbf{C}, \mathbf{P}, \mathbf{R}, \mathbf{I} \rangle$*. An object property description in* $\mathbf{S}$ *is an object property name* $P \in \mathbf{P}$ *or an inverse property* P^{-1} *with* $P \in \mathbf{P}$*. A data property description in* $\mathbf{S}$ *is a data property name* $R \in \mathbf{P}$*. A property description is either an object property description or a data property description.*

In natural language, applying the superscript $^{-1}$ has its counterpart in the change from active to passive voice. Constructing complex property descriptions is limited to constructing inverse properties. Contrary to this, $\mathcal{SHOIN}(\mathbf{D})$ offers

[3]Dialects of description logics differ in the way in which atomic entities can be combined to build up complex concepts and properties. An introduction to the members of the description logic family is given in [BN03].

a wide range of possibilities to construct complex concept descriptions as listed in the following definition.

Definition 3 (Concept description)**.** *Given a signature* $\mathbf{S} = \langle \mathbf{C}, \mathbf{P}, \mathbf{R}, \mathbf{I} \rangle$ *and a datatype theory* $\mathbf{D}$*. A concept description in* $\mathbf{S}$ *is a concept name* $A \in \mathbf{C}$*, or the top symbol* $\top$*, or the bottom symbol* $\bot$*, or a complex concept description*

$$
\begin{array}{rl}
\neg C & \textit{(atomic negation)} \\
B \sqcap C & \textit{(conjunction)} \\
B \sqcup C & \textit{(disjunction)} \\
\{o_1, \ldots, o_n\} & \textit{(one of)} \\
\exists P.C & \textit{(exists restriction)} \\
\forall P.C & \textit{(value restriction)} \\
\exists_{\leq n} P & \textit{(at least restriction)} \\
\exists_{\geq n} P & \textit{(at most restriction)} \\
\exists R.D & \textit{(data exists restriction)} \\
\forall R.D & \textit{(data value restriction)} \\
\exists_{\leq n} R & \textit{(data at least restriction)} \\
\exists_{\geq n} R & \textit{(data at most restriction)}
\end{array}
$$

where B *and* C *are concept descriptions in* $\mathbf{S}$*,* D *is a datatype defined in* $\mathbf{D}$*,* P *is an object property description in* $\mathbf{S}$*,* R *is a data property description in* $\mathbf{S}$*,* $n \in \mathbb{R}^+$*, and* $o_1, \ldots, o_n \in \mathbf{I}$ *are individual names.*

This definition comprises a recursive element. Complex concept descriptions can be constructed from atomic and complex concept and property descriptions. There is one exception which is referred to as nominal description (listed with the adjunct '*one of*'). It is the definition of a concept by enumerating its instances. The letter $\mathcal{O}$ in $\mathcal{SHOIN}(\mathbf{D})$ refers to this type of definition. In the next section we formalize how the meaning of a complex concept is defined by the meaning of its constituents.

Concepts and property descriptions can be used within two types of statements. Statements of the first type are called axioms. A set of axioms is called a TBox.

Definition 4 (TBox)**.** *Given a signature* $\mathbf{S} = \langle \mathbf{C}, \mathbf{P}, \mathbf{R}, \mathbf{I} \rangle$ *as well as concept descriptions* B *and* C *in* $\mathbf{S}$*, object property descriptions* P *and* Q *in* $\mathbf{S}$*, and data property description* R *and* S *in* $\mathbf{S}$*. An axiom in* $\mathbf{S}$ *is of the form*

$$
\begin{array}{rl}
B \sqsubseteq C, B \equiv C & \textit{(concept inclusion, equivalence)} \\
P \sqsubseteq Q, P \equiv Q & \textit{(object property inclusion, equivalence)} \\
R \sqsubseteq S, R \equiv S & \textit{(data property inclusion, equivalence)} \\
\textit{trans}(P) & \textit{(object property transitivity).}
\end{array}
$$

A finite set of axioms in $\mathbf{S}$ *is called a TBox in* $\mathbf{S}$.

The letter *"T"* in *"TBox"* points to the fact that the TBox contains terminological axioms. In most cases these axioms are used to define the meaning of a named concept or property by clarifying its relations to the other concepts in the ontology.

The second type of statement is called assertion. Opposed to an axiom, an assertion is used to make a statement about an instance by describing its qualities in terms of concept membership and relations to other instances.

Definition 5 (ABox)**.** *Given a signature $\mathbf{S} = \langle \mathbf{C}, \mathbf{P}, \mathbf{R}, \mathbf{I} \rangle$ and a datatype theory $\mathbf{D}$. Further, let C by a concept description in $\mathbf{S}$, let P be an object property description in $\mathbf{S}$, and let R and S be data property description in $\mathbf{S}$. An assertion in $\mathbf{S}$ is of the form*

$$\begin{array}{rl} C(a) & \textit{(concept assertion)} \\ P(a,b) & \textit{(object property assertion)} \\ R(a,d) & \textit{(data property assertion)} \\ a = b & \textit{(equality)} \\ a \neq b & \textit{(inequality)} \end{array}$$

where d is a concrete data value defined in $\mathbf{D}$. A finite set of assertions in $\mathbf{S}$ is called an ABox in $\mathbf{S}$.

An ontology consists of both terminological axioms and assertions. However, we might also have cases where ABox or TBox are missing, i.e., are empty sets.

Definition 6 (Ontology)**.** *Given an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ in $\mathbf{S} = \langle \mathbf{C}, \mathbf{P}, \mathbf{R}, \mathbf{I} \rangle$. The union $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ is called an ontology in $\mathbf{S}$. $\mathbf{S}$ is called the signature of $\mathcal{O}$ if there exists no $\mathbf{S}' = \langle \mathbf{C}', \mathbf{P}', \mathbf{R}', \mathbf{I}' \rangle$ such that (1) $\mathcal{O}$ is in $\mathbf{S}'$ and (2) $\mathbf{C}' \subset \mathbf{C}$ or $\mathbf{P}' \subset \mathbf{P}'$ or $\mathbf{R}' \subset \mathbf{R}'$ or $\mathbf{I}' \subset \mathbf{I}'$.*

The second part of this definition ensures the uniqueness of an ontologies signature, i.e., that a signature $\mathbf{S}$ is the signature of an ontology $\mathcal{O}$ iff $\mathbf{S}$ is the minimal signature such that $\mathcal{O}$ is in $\mathbf{S}$.

2.1.2 Semantics

So far we understand how to create complex expressions, axioms and assertions from the typed vocabulary called signature. However, we do not understand how the semantics of a complex expression is determined by the semantics of its components. This issue is covered in the following definition of an interpretation.

Definition 7 (Interpretation)**.** *Given a signature $\mathbf{S} = \langle \mathbf{C}, \mathbf{P}, \mathbf{R}, \mathbf{I} \rangle$ and a datatype theory $\mathbf{D}$. An interpretation $\mathcal{I} = \left\langle \Delta^{\mathcal{I}}, \Delta^{\mathcal{I}}_{D}, \cdot^{\mathcal{I}} \right\rangle$ consists of a set $\Delta^{\mathcal{I}}$, which is the abstract domain; a set $\Delta^{\mathcal{I}}_{D}$, which is the concrete domain (concrete data values); and a function $\cdot^{\mathcal{I}}$ that maps every concept name in $\mathbf{C}$ to a subset of $\Delta^{\mathcal{I}}$, every object property name in $\mathbf{P}$ to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, every data property name* *in* $\mathbf{R}$ *to a subset of* $\Delta^{\mathcal{I}} \times \Delta_D^{\mathcal{I}}$*, every individual name in* $\mathbf{I}$ *to an element of* $\Delta^{\mathcal{I}}$*, every datatype in* $\mathbf{D}$ *to a subset of* $\Delta_D^{\mathcal{I}}$*, and every data constant to a value in* $\Delta_D^{\mathcal{I}}$*. Furthermore,*

$$\begin{array}{rcl}
\top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} & = & \emptyset \\
(\neg C)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(B \sqcap C)^{\mathcal{I}} & = & B^{\mathcal{I}} \cap C^{\mathcal{I}} \\
(B \sqcup C)^{\mathcal{I}} & = & B^{\mathcal{I}} \cup C^{\mathcal{I}} \\
\{o_1, \ldots, o_n\}^{\mathcal{I}} & = & \{o_1^{\mathcal{I}}, \ldots, o_n^{\mathcal{I}}\} \\
(\exists P.C)^{\mathcal{I}} & = & \{x \mid \exists y \, \langle x, y \rangle \in P^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \\
(\forall P.C)^{\mathcal{I}} & = & \{x \mid \forall y \, \langle x, y \rangle \in P^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\} \\
(\exists_{\leq n} P)^{\mathcal{I}} & = & \{x \mid \#\{\langle x, y \rangle \in P^{\mathcal{I}}\} \leq n\} \\
(\exists_{\geq n} P)^{\mathcal{I}} & = & \{x \mid \#\{\langle x, y \rangle \in P^{\mathcal{I}}\} \geq n\} \\
(\exists R.D)^{\mathcal{I}} & = & \{x \mid \exists y \, \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in D^{\mathcal{I}}\} \\
\forall R.D)^{\mathcal{I}} & = & \{x \mid \forall y \, \langle x, y \rangle \in R^{\mathcal{I}} \rightarrow y \in D^{\mathcal{I}}\} \\
(\exists_{\leq n} R)^{\mathcal{I}} & = & \{x \mid \#\{\langle x, y \rangle \in P^{\mathcal{I}}\} \leq n\} \\
(\exists_{\geq n} R)^{\mathcal{I}} & = & \{x \mid \#\{\langle x, y \rangle \in P^{\mathcal{I}}\} \geq n\}
\end{array}$$

where B *and* C *are concept descriptions in* $\mathbf{S}$*,* D *is a datatype defined in* $\mathbf{D}$*,* P *is an object property description in* $\mathbf{S}$*,* R *is a data property description in* $\mathbf{S}$*,* $n \in \mathbb{R}^+$*, and* $o_1, \ldots, o_n \in \mathbf{I}$ *are individual names.*

The constraints listed in Definition 7 specify the meaning of a complex expression in terms of the meaning of its constituents according to the type of combining the components to a new expression. Notice that the meaning of a natural language expression is in a very similar way defined by the meaning of its constituents. The general principle is known as the principle of compositionality. It is the principle that the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them.[4]

The assertions and axioms of an ontology put further constraints on the set of interpretations. Thus, we say that an interpretation satisfies (or does not satisfy) an axiom or assertion.

Definition 8 (Satisfiability)**.** *Given interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \Delta_D^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$*.* $\mathcal{I}$ *satisfies an axiom*

$$\begin{array}{rcl}
B \sqsubseteq C & \textit{iff} & B^{\mathcal{I}} \subseteq C^{\mathcal{I}} \\
P \sqsubseteq Q & \textit{iff} & P^{\mathcal{I}} \subseteq Q^{\mathcal{I}} \\
R \sqsubseteq S & \textit{iff} & R^{\mathcal{I}} \subseteq S^{\mathcal{I}} \\
\textit{trans}(P) & \textit{iff} & \langle x, y \rangle \in R^{\mathcal{I}} \wedge \langle y, z \rangle \in R^{\mathcal{I}} \rightarrow \langle x, z \rangle \in R^{\mathcal{I}}
\end{array}$$

[4]The first modern formulation has already been given by Gottlob Frege in the year 1884 [Fre84].

where B and C are concept descriptions, P and Q are object property descriptions, and R and S are data properties. $\mathcal{I}$ satisfies an equivalence axiom $X \equiv Y$ iff $\mathcal{I}$ satisfies $X \sqsubseteq Y$ and $\mathcal{I}$ satisfies $Y \sqsubseteq X$. Furthermore, $\mathcal{I}$ satisfies an assertion

$$\begin{array}{rcl} C(a) & \textit{iff} & a^{\mathcal{I}} \in C^{\mathcal{I}} \\ P(a,b) & \textit{iff} & \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in P^{\mathcal{I}} \\ R(a,d) & \textit{iff} & \langle a^{\mathcal{I}}, d^{\mathcal{I}} \rangle \in R^{\mathcal{I}} \\ a = b & \textit{iff} & a^{\mathcal{I}} = b^{\mathcal{I}} \\ a \neq b & \textit{iff} & a^{\mathcal{I}} \neq b^{\mathcal{I}} \end{array}$$

where C is a concept description, P is an object property description, R is a data property, a and b are individuals, and d is a data value.

Due to Definition 8, an ontology divides the set of interpretations into those interpretations that do not satisfy the ontology and those interpretations that satisfy the ontology. The latter ones are called models of the ontology.

Definition 9 (Model)**.** *An interpretation $\mathcal{I}$ is a model for an ontology $\mathcal{O}$, iff $\mathcal{I}$ satisfies each axiom and each assertion in $\mathcal{O}$.*

On top of the notion of a model we introduce the notion of entailment. It describes the relation between an ontology $\mathcal{O}$ (= set of axioms and assertions) and a single axiom or assertion a. If $\mathcal{O}$ entails a, we also say that a follows from $\mathcal{O}$.

Definition 10 (Entailment)**.** *An ontology $\mathcal{O}$ entails an assertion or axiom a, iff each model for $\mathcal{O}$ is also a model for a. An ontology $\mathcal{O}$ entails a set of assertions or axioms A, iff each model for $\mathcal{O}$ is also a model for each $a \in A$. We write $\mathcal{O} \models a$ if $\mathcal{O}$ entails a; if $\mathcal{O}$ does not entail a we write $\mathcal{O} \not\models a$.*

To decide whether an axiom or assertion a is entailed is a typical reasoning task to be solved by a DL reasoning system such as Racer [MH03] or Pellet [SPG$^+$07]. We will see later, that one source of complexity for the algorithms to be introduced is the complexity of the reasoning involved.

Finally, we have to define the notion of concept and property (un)satisfiability as well as the notion of ontology (in)coherence. A concept C is defined to be unsatisfiable iff each model $\mathcal{I}$ of $\mathcal{O}$ maps C to the empty set, i.e., an instance of C cannot exist for logical reasons. Notice that, in the following, we extend the standard definition of concept unsatisfiability (as given in [HQ07]) and include additionally the unsatisfiability of properties.

Definition 11 (Concept/Property Unsatisfiability)**.** *Given an ontology $\mathcal{O}$ and its signature* $\mathbf{S}$*. A concept description C (property description P) in* $\mathbf{S}$ *is unsatisfiable in $\mathcal{O}$ iff for each model $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \Delta^{\mathcal{I}}_D, \cdot^{\mathcal{I}} \rangle$ of $\mathcal{O}$ we have $C^{\mathcal{I}} = \emptyset$ ($P^{\mathcal{I}} = \emptyset$).*

The unsatisfiability of a named concept (or property) is categorized as evidence for a modeling mistake or flaw in the ontology [SC03]. This is based on the basic consideration that a concept or property is introduced in an ontology to make statements about individuals, i.e., the concept/property is in principle intended to be used in the ABox of an ontology. This consideration is not refuted by the fact that many ontologies in the Semantic Web miss an ABox.[5]

The satisfiability of properties and concepts can be checked by any standard reasoner. We can exploit the following proposition.

Proposition 1 (Satisfiability and Subsumption)**.** *Given an ontology $\mathcal{O}$ and its signature $\mathbf{S}$. A concept description C in $\mathbf{S}$ is unsatisfiable in $\mathcal{O}$, iff $\mathcal{O} \models C \sqsubseteq \bot$. A property description P in $\mathbf{S}$ is unsatisfiable in $\mathcal{O}$, iff $\mathcal{O} \models \exists P.\top \sqsubseteq \bot$.*

Proof. The correctness of Proposition 1 follows from the fact that $\bot$ is by definition unsatisfiable, and thus each subconcept of $\bot$ is also unsatisfiable. With respect to the unsatisfiability of a property we can conclude from $\{x \mid \exists y \, \langle x, y\rangle \in P^{\mathcal{I}}\} = (\exists P.\top)^{\mathcal{I}} \subseteq \bot^{\mathcal{I}} = \emptyset$ that $P^{\mathcal{I}} = \emptyset$. □

Since the unsatisfiability of a concept (or property) is a symptom for a logical problem, an ontology is called incoherent when there exists an unsatisfiable named concept or property; otherwise the ontology is called coherent.

Definition 12 (Incoherence)**.** *Given an ontology $\mathcal{O}$ and its signature $\mathbf{S} = \langle \mathbf{C}, \mathbf{P}, \mathbf{R}, \mathbf{I}\rangle$. $\mathcal{O}$ is incoherent iff there exists an unsatisfiable $C \in \mathbf{C}$, $P \in \mathbf{P}$ or $R \in \mathbf{R}$. Otherwise $\mathcal{O}$ is called coherent.*

There exists a strong dependence between incoherence and inconsistency, both attributes of an ontology should not be mixed up (compare Definition 12 above and Definition 22). The notion of incoherence and in particular its extension to the notion of alignment incoherence, introduced at the end of the following section, forms the basis of the diagnostic approach central to this thesis.

2.2 Alignments

In the introductory section of this chapter we compared an alignment with a dictionary. We already explained how conceptual schemes are externalized as terminological axioms of an ontology. We now formally introduce the notion of an alignment. Again, we have to distinguish between syntax and semantics.

2.2.1 Syntax

Similar to an ontology, which is a set of axioms and assertions, an alignment is a set of correspondences. A correspondence can be understood as link between at

[5] One might argue against this position as follows: An axiom as $A \sqsubseteq B$ defines the semantics of A by explaining its relation to B. This explanation does not include nor require an ABox with individuals. However, this explanation means that each instance of A is an instance of B. An exception from this general principle can be found in ontologies that describe relations between abstract entities e.g., mathematical ontologies, where the unsatisfiability of a concept might be a non trivial and correct reasoning result.

least two entities (individuals, concepts, properties) from two different ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. We first give a very general definition of a correspondence followed by a specification of different subtypes.

Definition 13 (Correspondence). *Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ and their signature $\mathbf{S}_1 = \langle \mathbf{C}_1, \mathbf{P}_1, \mathbf{R}_1, \mathbf{I}_1 \rangle$ and $\mathbf{S}_2 = \langle \mathbf{C}_1, \mathbf{P}_2, \mathbf{R}_2, \mathbf{I}_1 \rangle$. A correspondence between $\mathcal{O}_1$ and $\mathcal{O}_2$ is a triple $\langle x, y, r \rangle$ where*

(1) x and y are, according to Definition 3, concept descriptions in $\mathbf{S}_1$ and $\mathbf{S}_2$ with r $\in \{\sqsubseteq, \sqsupseteq, \equiv\}$; or

(2) x and y are, according to Definition 2, property descriptions in $\mathbf{S}_1$ and $\mathbf{S}_2$ with r $\in \{\sqsubseteq, \sqsupseteq, \equiv\}$; or

(3) x and y are individuals in $\mathbf{I}_1$ and $\mathbf{I}_2$ with r $\in \{=, \neq\}$.

A correspondence of type (1) and (2) is called terminological correspondence, a correspondence of type (3) is called instance correspondence.

This definition is based on the definition given by Euzenat and Shvaiko [ES07]. Contrary to our approach, Euzenat and Shvaiko define a correspondence as 5-tuple, which contains additionally an unique id and a confidence value. The degree of confidence (often expressed as a value in the range $[0, 1]$) is defined to be a measure of trust in the fact that the correspondence holds. Given a correspondence c we use a notation $\alpha(c) = 0.6$ to express that the degree of confidence is 0.6, where α might refer to a matching system or any other source of evidence. We also omit to introduce an explicit id. Opposed to this, we define two correspondences $\langle x, y, r \rangle$ and $\langle x', y', r' \rangle$ to be the same correspondences, iff $x = x'$, $y = y'$ and $r = r'$. Thus, we can for example easily say that two matching systems α and β generate the same correspondence c with different confidences $\alpha(c)$ and $\beta(c)$.

We specified the entities that can appear in a correspondence by pointing back to Definitions 2 and 3. Thus, we can also distinguish between different types of correspondences regarding the complexity of the involved entities.

Definition 14 (Complexity of a Correspondence). *Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. A correspondence $c = \langle x, y, r \rangle$ between $\mathcal{O}_1$ and $\mathcal{O}_2$ is non-complex iff (i) x and y are concept names or (ii) x and y are property names or (iii) x and y are individuals; otherwise c is complex.*

Correspondences between individuals are always non-complex. Their exists no rule for constructing complex individual descriptions in $\mathcal{SHOIN}(\mathbf{D})$. According to the following definition, a set of correspondences is referred to as alignment. We further declare that an alignment has a certain characteristic, if all correspondences of the alignment have this characteristic.

Definition 15 (Alignment). *Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. A set of correspondences $\mathcal{A}$ between $\mathcal{O}_1$ and $\mathcal{O}_2$ is called an alignment between $\mathcal{O}_1$ and $\mathcal{O}_2$. If all* *correspondences in $\mathcal{A}$ are terminological correspondences, then $\mathcal{A}$ is a terminological alignment. If all correspondences in $\mathcal{A}$ are instance correspondences, then $\mathcal{A}$ is an instance alignment. If all correspondences in $\mathcal{A}$ are non-complex correspondences, then $\mathcal{A}$ is a non-complex alignment; otherwise $\mathcal{A}$ is a complex alignment.*

So far we described correspondences as triples and alignments as sets of triples. This formalization is well suited for our purpose and in the next section we show how to define the semantics for this syntactical representation.

There exists also a rdf-serialization for alignments that is the input and output format supported by the Alignment API [Euz04], a Java API and implementation for expressing and sharing ontology alignments. The expressivity supported by the Alignments API was in the past restricted to non-complex correspondences and has recently been extended to a more expressive language referred to as EDOAL (Expressive and Declarative Ontology Alignment Language).[6] First approaches to define an expressive alignment language have been made by Franois Scharffe in the context of the SEKT project [SB05]. Further need for expressive alignments has been emphasized by pointing to typical patterns of correspondence that can only be described by the use of an expressive alignment language [SF08]. These approaches have recently been redesigned and reimplemented under the name of EDOAL as part of the Alignment API [DESdS11].

The rdf-syntax of the Alignment API (mainly intended as for automatic processing) is not committed to a certain semantics.[7] An alignment language which differs in this regard is C-OWL (Context OWL, [BGvH+04]), a language whose syntax and semantics have been obtained by extending the OWL syntax and semantics to allow for the representation of contextual ontologies, i.e., ontologies that are linked with each other by so called bridge rules. A the end of the following section we have a clear understanding what it means to define the semantics of an alignment language. In particular, we will see in how far notions as entailment and coherenence depend on such a semantics. Although we propose a concrete family of semantics, we will also see in how far the underlying semantics can be replaced by a different one. This makes our approach applicable to all kinds of alignment semantics that share certain properties.

2.2.2 Semantics

We introduce a semantics based on the approach we already proposed in [MS07b]. Later we refined and modified this approach in [MS09b]. Now we extend our approach to a wider set of alignments. In particular, we show how to apply the approach to complex correspondences.

[6]The documentation of the EDOAL language can be found at `http://alignapi.gforge.inria.fr/edoal.html`.

[7]Note that for our experiments we make use of the Alignment API format, since this is supported by most matching system and has been used successfully in many evaluation scenarios [EIM+07, CEH+08, EFH+09].

The semantics we propose is referred to as reductionistic alignment semantics [MS09b]. A reductionistic semantics reduces the semantics of an alignment to the standard model-theoretic semantics of axioms and assertions in an ontology. Hence, a reductionistic alignment semantics requires to map an alignment and the matched ontologies to a set of axioms and assertions. In the following definition we propose to split this mapping in two functions ζ and ξ referred to as *'transformation'* and *'translation'* in the following.

Definition 16 (Reductionistic Semantics)**.** *Given an alignment $\mathcal{A}$ between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ with signature $\mathbf{S}_1$ and $\mathbf{S}_2$. A reductionistic alignment semantics $\mathcal{S} = \langle \zeta, \xi \rangle$ is a pair of functions where ζ maps an ontology $\mathcal{O}$ to an ontology $\zeta(\mathcal{O})$ ('transformation') and ξ maps a correspondence between $\mathcal{O}_1$ and $\mathcal{O}_2$ to an ontology in the signature $\mathbf{S}_1 \cup \mathbf{S}_2$ ('translation').*

The result of applying a reductionistic semantics is called the aligned ontology. The transformation function ζ is applied to both $\mathcal{O}_1$ and $\mathcal{O}_2$, while the translation function ξ maps each correspondence in $\mathcal{A}$ to a set of axioms and/or assertions. Notice that the domain of ξ is the set of correspondences between $\mathcal{O}_1$ and $\mathcal{O}_2$ and not the set of alignments. This will be important when it comes to the definition of alignment entailment. The union of applying ζ and ξ is called the $\mathcal{S}$-aligned ontology.

Definition 17 (Aligned Ontology)**.** *Given an alignment $\mathcal{A}$ between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ and a reductionistic alignment semantics $\mathcal{S} = \langle \zeta, \xi \rangle$. The $\mathcal{S}$-aligned ontology of $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{A}$ is defined as $\mathcal{A}_{\mathcal{S}}(\mathcal{O}_1, \mathcal{O}_2) = \zeta(\mathcal{O}_1) \cup \zeta(\mathcal{O}_2) \cup \bigcup_{c \in \mathcal{A}} \xi(c)$.*

Before we continue with the general notions of alignment entailment and incoherence, we introduce a concrete instantiation of a reductionistic alignment. The natural alignment semantics is the simplest way to interpret an alignment. It is based on a 1:1 translation into assertions and axioms whereas the transformation function is the identity function.

Definition 18 (Natural Semantics)**.** *Given a correspondence $\langle X, Y, r \rangle$ and an ontology $\mathcal{O}$. The natural semantics $\mathcal{S}_n = \langle \zeta_n, \xi_n \rangle$ is defined by a specification of its components $\zeta_n(\mathcal{O}) \mapsto \mathcal{O}$ and $\xi_n(\langle X, Y, r \rangle) \mapsto \{X \; r \; Y\}$.*

If we apply, for example, the natural semantics $\mathcal{S}_n$ on the alignment $\mathcal{A} = \{\langle A, B, \sqsubseteq \rangle, \langle P, Q, \equiv \rangle, \langle a, b, \neq \rangle\}$ between $\mathcal{O}_1$ and $\mathcal{O}_2$, we will have $\mathcal{A}_{\mathcal{S}}(\mathcal{O}_1, \mathcal{O}_2) = \mathcal{O}_1 \cup \mathcal{O}_2 \cup \{A \sqsubseteq B, P \equiv Q, a \neq b\}$.

Note that it is not complicated to define other semantics within this framework. Distributed Description Logics (DDL) is an interesting example. DDL is a formalism intended especially to enable reasoning between multiple ontologies connected by directional semantic alignments [BS03, SBT05]. It differs from our reductionistic approach by introducing additional model-theoretic elements like disjoint domains and a relation between their elements. As a result, DDL is a specific semantics that is designed for reasoning with and about ontologies linked via alignment. However, the authors show also how to map DDL to DL. This mapping requires a non-trivial transformation function ζ_{DDL} to express that concepts from $\mathcal{O}_1$ and concepts from $\mathcal{O}_2$ are disjoint be default. Details can be found in Chapter 7 of [BS03].

Let us continue with the definition of entailment, satisfiability and incoherence. Note that these definition are valid for any kind of reductionistic alignment semantics. Each particular correspondence in $\mathcal{A}$ is translated on its own by ξ. This allows to identify a counterpart for each correspondence in the aligned ontology. Thus, we can define the notion of entailment with respect to correspondences and alignments analog to Definition 10.

Definition 19 (Alignment Entailment). *Given an alignment $\mathcal{A}$ and a correspondence c between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ as well as a reductionistic alignment semantics $\mathcal{S} = \langle \zeta, \xi \rangle$. Correspondence c is $\mathcal{S}$-entailed by $\mathcal{A}$ with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ iff each model for $\mathcal{A}_\mathcal{S}(\mathcal{O}_1, \mathcal{O}_2)$ is also a model for $\xi(c)$. If c is entailed we shortly write $\mathcal{A} \models^{\mathcal{S}}_{\mathcal{O}_1,\mathcal{O}_2} c$, and $\mathcal{A} \not\models^{\mathcal{S}}_{\mathcal{O}_1,\mathcal{O}_2} c$ otherwise. An alignment is entailed iff each of its correspondences is entailed.*

This definition reduces alignment entailment to the standard notion of entailment. Any reasoner that is applicable to check entailment of an axiom/assertion with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ can thus be used to check whether a correspondence is entailed by an alignment with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$. However, the correctness of this statement depends on the expressivity of the aligned ontology. In particular, we have to ensure that the expressivity of $\mathcal{A}_\mathcal{S}(\mathcal{O}_1, \mathcal{O}_2)$ does not exceed the expressivity of $\mathcal{O}_1 \cup \mathcal{O}_2$. The following proposition guarantees the applicability of any $\mathcal{SHOIN}(\mathbf{D})$-reasoner to ontology alignments interpreted by the natural alignment semantics $\mathcal{S}_n$ given the matched ontologies are in $\mathcal{SHOIN}(\mathbf{D})$.

Proposition 2 (Expressivity of the $\mathcal{S}_n$-aligned Ontology). *Given an alignment $\mathcal{A}$ between $\mathcal{SHOIN}(\mathbf{D})$ ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, which are both in $\mathcal{SHOIN}(\mathbf{D})$. The aligned ontology $\mathcal{A}_{\mathcal{S}_n}(\mathcal{O}_1, \mathcal{O}_2)$ is a $\mathcal{SHOIN}(\mathbf{D})$ ontology.*

Proof. Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be in $\mathcal{SHOIN}(\mathbf{D})$. It follows that $\zeta_n(\mathcal{O}_1) \cup \zeta_n(\mathcal{O}_2) = \mathcal{O}_2 \cup \mathcal{O}_2$ is in $\mathcal{SHOIN}(\mathbf{D})$. Thus, we have to show that $\xi_n(c)$ is in $\mathcal{SHOIN}(\mathbf{D})$ for each $c \in \mathcal{A}$. Due to the definition of a correspondence (Definition 13), a terminological correspondence expresses an equivalence or subsumption relation between concepts or property descriptions, while an instance correspondence expresses (in)equality between individuals. Due to the definition of the natural semantics (Definition 18), each correspondence is translated as single axiom in a straight forward way. As a result $\bigcup_{c \in \mathcal{A}} \xi_n(c)$ will only contain axioms listed in Definition 4 and 5. Thus, $\mathcal{A}_{\mathcal{S}_n}(\mathcal{O}_1, \mathcal{O}_2)$ is in $\mathcal{SHOIN}(\mathbf{D})$. □

In the following we need to talk frequently about concept descriptions, property descriptions, and individuals that can be constructed from the signature $\mathbf{S}_n$ of an ontology $\mathcal{O}_n$. Such entities are referred to by a subscript notation $..._{\#n}$ to denotate their origin.

We already introduced the notion of an unsatisfiable concept/property (Definition 11). Obviously, an aligned ontology, as any kind of ontology, can contain unsatisfiable concepts or properties. Suppose for example that $C_{\#i}$ with $i \in \{1,2\}$ is unsatisfiable in $\mathcal{A}_{\mathcal{S}_n}(\mathcal{O}_1, \mathcal{O}_2)$. We distinguish between two cases: $C_{\#i}$ is also unsatisfiable in $\mathcal{O}_i$ or $C_{\#i}$ is satisfiable in $\mathcal{O}_i$. In the second case we can conclude that the unsatisfiability has been caused by $\mathcal{A}$. We call such a concept an aligned unsatisfiable concept.

Definition 20 (Aligned Concept/Property Unsatisfiability)**.** *Given an alignment $\mathcal{A}$ between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ as well as a reductionistic alignment semantics $\mathcal{S}$. A named concept or property $C_{\#i}$ with $i \in \{1,2\}$ is $\mathcal{S}$-unsatisfiable due to $\mathcal{A}$ with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ iff $C_{\#i}$ is satisfiable in $\mathcal{O}_i$ and unsatisfiable in $\mathcal{A}_{\mathcal{S}_n}(\mathcal{O}_1, \mathcal{O}_2)$.*

Later on we will see that the existence of an aligned unsatisfiable concept or property can be considered as symptom for an erroneous alignment and results in unintended consequences. Analogous to the classical notion of ontology incoherence (compare Definition 12), we introduce the notion of alignment incoherence as follows.

Definition 21 (Alignment Incoherence)**.** *Given an alignment $\mathcal{A}$ between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ with signatures $\mathbf{S}_1 = \langle \mathbf{C}_1, \mathbf{P}_1, \mathbf{R}_1, \mathbf{I}_1 \rangle$ and $\mathbf{S}_2 = \langle \mathbf{C}_2, \mathbf{P}_2, \mathbf{R}_2, \mathbf{I}_2 \rangle$ as well as a reductionistic alignment semantics $\mathcal{S}$. $\mathcal{A}$ is $\mathcal{S}$-incoherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ iff there exists $C_{\#i} \in \mathbf{C}_i \cup \mathbf{P}_i \cup \mathbf{R}_i$ with $i \in \{1,2\}$ that is $\mathcal{S}$-unsatisfiable due to $\mathcal{A}$ with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$. Otherwise $\mathcal{A}$ is $\mathcal{S}$-coherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$.*[8]

With Definition 21 we introduce the central notion of this thesis. Notice that in Proposition 1 we showed how satisfiability and entailment are related. The definition of alignment incoherence is thus solely based on the notion of alignment entailment. Although we gave a definition of alignment entailment on top of a reductionistic semantics, any other approach that specifies the notion of alignment entailment is a sufficient fundament for the theory of alignment diagnosis introduced later. In particular, different parts of this thesis rely on different levels of abstraction.

- Chapter 4 solely relies on the notion of alignment incoherence. The theoretical framework presented in this chapter holds for **any alignment semantics that supports the notion of entailment**.
- The complete reasoning methods presented in Section 5.1 and the algorithms that solely rely on these techniques (Algorithm 6 and 9 presented in Chapter 6) can be applied to **any reductionistic alignment semantics**.

[8]Note that we do not assume that the elements of $\mathbf{S}_1$ and $\mathbf{S}_2$ are disjoint. For example, we do not exclude that there are some concepts names shared by both $\mathbf{C}_1$ and $\mathbf{C}_2$. This means that dependencies between those entities described in $\mathcal{O}_1$ and those described in $\mathcal{O}_2$ might not only be related to $\mathcal{A}$, but can also by caused implicitly by using the same vocabulary.

- Improvements of the core algorithms are suggested (Algorithm 8 and 10) in Chapter 6 using additionally the efficient reasoning methods proposed in Section 5.2. They are tailored to **the natural reductionistic alignment semantics** and are thus not applicable other reductionistic semantics.

Before we turn our head to the notion of alignment diagnosis, we continue with a section that is concerned with the motivation of our approach. In particular, we explain why to use alignment incoherence as a guiding principle in Ontology Matching. Thus, we are in the following section concerned with research questions R1 and R2.

Chapter 3

Motivation

Thinking and being are governed by contradiction (Aristotle).

In the previous chapter we constructed a formal framework to introduce the notion of alignment incoherence. Based on this framework, we will define and solve the research problem of this thesis. However, before we continue with the central theme, we have to understand the significance of alignment coherence to its full extent. We already motivated our approach roughly in the introduction of this thesis. However, we are now able to describe and explain the relevance of alignment coherence in detail

We motivate our approach from two different points of view. In Section 3.1 we first present an uncommon example, which is an instance of a general problem referred to as radical translation. At first glimpse it seems to be a problem that is not related with the topic of this thesis, even though it is well known in the philosophy of language [Cri98]. However, it will turn out that the problem of ontology matching is a special case of radical translation. Based on this example we argue that it makes sense to assume that a correct alignment will always be a coherent alignment. One of the main principles for constructing a correct alignment, should thus always be the principle to construct a coherent alignment.

In Section 3.2 we tackle the same issue from a different perspective. We discuss the relevance of alignment coherence from an application oriented perspective: We explain the effects of an incoherent alignment in the context of reasoning with a merged ontology (Section 3.2.1), we show what happens if an incoherent alignment is used for migrating instances from one ontology to another (Section 3.2.2), and we discuss a scenario, where the alignment is used to rewrite and process a query (Section 3.2.3).

Most of the content presented in this chapter origins from our previously published papers. The first part of Section 3.1 was presented as motivating example in [MS07b]. We discussed some of the considerations exposed at the end of this section first in [MS08], however, the main line of reasoning we present for the first time in this thesis. The content of Section 3.2 is a slightly extended version of the pragmatic motivation for alignment coherence we discussed in [MS08].

3.1 Truth and Coherence

Suppose that a linguist wants to explore the unknown language L of some people that have not been in contact to human civilization yet. The native people accept the researcher and let him be part of their daily life. At the first stage of his project the linguist observes the linguistic behavior of the natives and establishes some hypothesis about the meaning of the words that are uttered by the natives. The following could be a typical example for such a situation.

Example 1. *The linguist and a native are standing in front of an oak tree. A rabbit is sitting close to the tree. The native points at the direction of the tree and utters the word "Gavagai!". The linguist considers two possible hypothesis about the meaning of the word. The word might on the one hand refer to oak or might on the other hand refer to rabbit. He writes down both hypothesis and marks them with subscript $_q$ as questionable.*

As time goes by, the linguist is able to utter simple sentences in L. He also finds out which words and gestures mean approval and rejection. After a while he also manages to ask questions of the form *"Are all x y?"* in L. This enables him to apply an elaborated strategy.

Example 2. *(continued) From time to time the linguist cleans up the entries in his dictionary. He finds, amongst others, the following three entries.*

$$gavagai = rabbit_q \tag{3.1}$$

$$gavagai = oak_q \tag{3.2}$$

$$snok = tree \tag{3.3}$$

In order to find out if the first or the second entry has to be removed he asks the native "Are all gavagais snoks?". The bemused native denies the question. For that reason the linguist removes the second entry and keeps the first one.

The linguists decision is based in the following line of reasoning. If *"gavagai"* means the same as *"oak"* and *"snok"* means the same as *"tree"* then everything that is a gavagai also has to be a snok, because the linguist knows that an oak is a special kind of a tree. He transfers this subsumption relation to the concepts gavagai and snok. By asking the question *"are all gavagais snoks?"*, the linguist checks if this entailment is accepted by the native. The native denies this question and therefore the linguist is justified in removing the second or the third entry. Since he has marked the second entry as questionable he decides to remove it instead of removing the third entry.[1]

[1] Similar examples have been discussed by Quine [Qui73] with regard to developing a theory of meaning. We first mentioned this concrete example in the context of ontology matching in [MS07b].

Alignment (based on the dictionary)	
(1) $\langle Gavagai_{\#1}, Oak_{\#2}, \equiv \rangle$	*corresponds to entry 3.2*
(2) $\langle Snok_{\#1}, Tree_{\#2}, \equiv \rangle$	*corresponds to entry 3.3*
Axioms of ontology #1 (Native)	Axioms of ontology #2 (Linguist)
(3) $Gavagai \sqsubseteq \neg Snok$	(4) $Rabbit \sqsubseteq \neg Tree$
	(5) $Oak \sqsubseteq Tree$
Entailments (made by the linguist)	
(6) $Gavagai_{\#1} \equiv Oak_{\#2}$	*from (1)*
(7) $Snok_{\#1} \equiv Tree_{\#2}$	*from (2)*
(8) $Gavagai_{\#1} \sqsubseteq Snok_{\#1}$	*from (5), (6) and (7)*
(9) $Gavagai_{\#1} \sqsubseteq \bot$	*from (3) and (8)*

Table 3.1: Correspondences, axioms and entailments resulting in the unsatisfiability of $Gavagai_{\#1}$.

First of all, both speakers are committed to a specific conceptual scheme. In particular, we can identify a hierarchy of concepts which is relevant for both speakers with regard to questions of the form *"are all gavagais snoks?"*. It is obvious that the conceptual scheme of both speakers is not limited to a concept hierarchy, but will also comprise the meaning of functional terms as *"father of ..."* or binary predicates as *"... is faster than ..."* as well as relations amongst them. The language (and the linguistic behavior) of both speakers is an implicit and vague representation of the underlying conceptual scheme.

An ontology, on the other hand, we introduced as an explicit and formal representation of a conceptual scheme. Moreover, we can now use the theoretical apparatus developed in the previous section to rephrase Example 2 formally. In particular, we have to specify the conceptual scheme of both the native and the linguist as ontology. The dictionary of the linguist has to be interpreted as alignment. The resulting ontologies, the alignment, and the reasoning of the linguist are listed in Table 3.1.

We can now very precisely argue why the linguist decides to remove one of the Correspondences 1 and 2 in Table 3.1: these correspondences form an incoherent alignment. If both correspondences are correct, the native would use a concept that is for logical reasons empty.

The assumption underlying the approach of the linguist can be summarized as follows: *The entries of a dictionary are either correct or incorrect. If a dictionary results in a logical contradiction, at least one of its entries has to be incorrect.* This assumption corresponds to the main principle underlying the approach of this thesis. We phrase it as Presupposition 1.

Premise 1. *Given an alignment $\mathcal{A}$ between $\mathcal{O}_1$ and $\mathcal{O}_2$. A correspondence $c \in \mathcal{A}$ is either correct or incorrect. If $\mathcal{A}$ is incoherent then one of its correspondences is incorrect.*

Although, we are not aware of a publication that attacks our premise explicitly, we were suprised that in several face-to-face discussions it has been argued against it. In the following we discuss the most important ones. We start with the following line of reasoning

Argument 1. *Modeling an ontology is a highly subjective decision. Two different persons will probably design ontologies that differ to a large degree. Due to this heterogeneity it cannot be assured that there exists a unique alignment between them, nor will an acceptable alignment necessarily be coherent. What holds for modeling ontologies, holds in particular for modeling alignments: It is not appropriate to argue that an ontology is incorrect; and more than ever it is senseless to argue that an incoherent alignment is incorrect.*

This argument is based on a misconception, that has also been a topic of philosophical discussions. An example can be found in Donalds Davidsons article *'On the very idea of a conceptual scheme'*, where he argues against the possibility of an untranslatable language. He attacks a more general line of reasoning as the one we presented in Argument 1. However, Davidson's explanation is also helpful with regard to our discussion.

> *The dominant metaphor of conceptual relativism, that of differing points of view, seems to betray an underlying paradox. Different points of view make sense, but only if there is a common coordinate system on which to plot them; yet the existence of a common system belies the claim of dramatic incomparability [Dav74].*

Whenever someone argues that two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ are heterogeneous descriptions of the same domain, we can ask her to explain the differences between $\mathcal{O}_1$ and $\mathcal{O}_2$. Without such an additional explanation, we can reject the claim of heterogeneity as insubstantial. However, the required explanation is already a partial alignment, which supports and contradicts the claim at the same time. It will contain statements like *'concept A in $\mathcal{O}_1$ has a slightly different meaning than concept A' in $\mathcal{O}_2$'*. If we recognize two ontologies as heterogeneous, we must have a clear understanding of their differences regarding the meaning of their concepts and properties. This belies the claim that it is inappropriate to distinguish between correct and incorrect correspondences.

Sometimes it is not possible to bridge the gap between $\mathcal{O}_1$ and $\mathcal{O}_2$ with simple equivalence or subsumption correspondences. It might even be the case that the apparatus developed in the last Chapter is not expressive enough to express the relation between A and A' by a correspondence. However, this is a different issue, which is not a counterargument to Premise 1.

Another argument is based on the problem of deciding the correctness of a correspondences. It can be argued as follows.

Argument 2. *It is often very hard or even impossible to decide whether a correspondence is correct or incorrect. Even experts carefully analyzing $\mathcal{O}_1$ and $\mathcal{O}_2$*

cannot always agree on the correctness of certain critical correspondences. For that reason it makes no sense to distinguish between correct and incorrect correspondences in a strict sense.

This line of reasoning is based on a missing distinction between the genesis of knowledge and its validity. Even though we might have no sufficient argument to decide whether a or $\neg a$ is the case, we nevertheless know that one of a or $\neg a$ must be the case. Whether a or $\neg a$ is the case is independent from our perception, knowledge, or justification of/for a. The same holds for the correspondences of an alignment.

The final argument that leads over to the next section. It emphasizes a pragmatic point of view.

Argument 3. *Correctness or incorrectness is not really important when an alignment is used in an application scenario. The relevant aspect of a correspondence is its usefulness. Questions like 'What are the benefits for a user?' should be taken into account. Correctness and its relation to coherence is, finally, an irrelevant aspect of an alignment.*

This argument contains an element of truth, however, its conclusion is not acceptable. For example, van Hage et. al [vHKS08] have proposed an evaluation approach focusing on the benefit of a correspondence from an application-oriented perspective. The authors emphasize that the set of correct correspondences contains correspondences that are more or less relevant for the user. However, they do not argue to evaluate the relevance of an incorrect correspondence. The notion of relevance can thus be a useful supplement but not a replacement for correctness.

In the introduction we have asked in research question R2 about the interrelation between alignment coherence and the quality of the alignment in terms of precision and recall. In this section, we have presented the premise that an incoherent alignment must contain at least one incorrect correspondence. We have illustrated a reasonable example to motivate this premise and its usage in the context of radical translation. We have discussed several arguments against the premise and conclude that none of these arguments is conclusive. On the other hand we admit that we cannot give a proof for the correctness of our premise. Such a proof would touch general philosophical issues that cannot be answered within this thesis. At least, we have presented some theoretical considerations for the assumption that incoherence is a clear symptom for incorrectness.

We will revisit research question R2 again in Section 7.2 where we show – based on Premise 1 – that the incoherence of an alignment can be used as an upper bound for its precision. The concretion of R2 is R5, which deals with the effects of different ways to resolve incoherence on precision and recall. This question will be the guideline in Chapter 8 and Chapter 9 where we present a rich set of experiments analyzing the applicability of our premise in different scenarios.

3.2 Impact of Alignment Incoherence

In the previous section we discussed several counterarguments for our premise. One of these arguments emphasized the importance of a pragmatical approach. While we do not accept the conclusion, we pick up its motivation and focus in the following on the impact of alignment incoherence from an application oriented perspective. In particular, we show that the incoherence of an alignment has a negative effects on several important types of applications. We are thus concerned with research question R1.

In [NS05], the following four different purposes of using ontology alignments have been proposed. A more fine-grained distinction has been suggested in Euzenat and Shvaiko [ES07], however, most of the scenarios proposed there can also be subsumed under one of the following categories.

Frameworks Alignments are described in frameworks on an abstract level independent of an intended use.

Terminological Reasoning Alignments are used to perform reasoning tasks (e.g., entailment of a concept subsumption) across aligned ontologies.

Data Transformation Data from one ontology is transferred into the terminology of another ontology based on the knowledge encoded in an alignment.

Query Processing Queries formulated with respect to a certain ontology are translated into the terminology of a different ontology with the help of an alignment.

Since we try to argue for the relevance of alignment coherence in a practical context, abstract frameworks are of minor interest for us. We discuss each of the remaining scenarios in one of the following subsections.

3.2.1 Terminological Reasoning

Throughout this and the following subsections we refer to Figure 3.1. This figure depicts fragments of two ontologies $\mathcal{O}_1$ (on the left) and $\mathcal{O}_2$ (on the right). A square represents a concept, an ellipse a property, subsumption is represented by indentation. Domain and range of a property are restricted to be the concepts connected by the accordant arrow. Dashed horizontal lines represent disjointness between concepts. We suppose in the following that there are two companies C_1 and C_2, that use these ontologies to describe human resources and related topics.

In our first scenario we assume that the owners of the companies decide to join each other. The chief information officers create a comprehensive ontology that contains the axioms of both $\mathcal{O}_1$ and $\mathcal{O}_2$. On top of this, the IT systems of both subcontractors can still be used in the future and the integration can be conducted smoothly. Obviously, an alignment $\mathcal{A}$ is the key to this approach and the

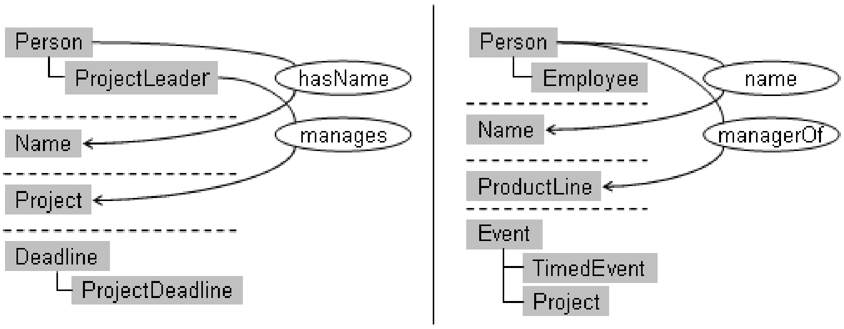

Figure 3.1: Fragments of two ontologies from the domain of management.

new ontology is the aligned ontology $\mathcal{A}_{\mathcal{S}}(\mathcal{O}_1, \mathcal{O}_2)$ that results from applying the alignment.

A generic description of a 'terminological reasoning' scenario is obviously not committed to a certain semantic. However, it can be supposed that the reasoning tasks to be performed are based on a some model-theoretic semantics that uses notions as model and entailment. Such a semantics allows the definition of an inconsistent ontology as follows.

Definition 22 (Inconsistency)**.** *An ontology $\mathcal{O}$ is inconsistent iff there exist no model for $\mathcal{O}$, otherwise $\mathcal{O}$ is consistent.*

Although incoherence and inconsistency are closely related, they are different notions [HQ07]. There are ontologies that are incoherent but consistent and ontologies that are coherent but inconsistent. The set of axioms $\mathcal{O}' = \{C \sqsubseteq D, C \sqsubseteq \neg D\}$ is an example for the first type, because C is an unsatisfiable concept. The set of assertions $\mathcal{O}'' = \{a \neq b, a = b\}$ is an example for the second type.

An incoherent ontology can easily be turned into an inconsistent ontology by adding a single assertion. For our small example we have to add the assertion $C(a)$. In general, an incoherent ontology is an inconsistent ontology if one of the unsatisfiable concepts has an individual as instance [FHP$^+$06]. In our example, this will be the case for most or all of the concepts, because they have obviously been introduced to describe the stored data of the companies.

Now suppose that we have an incoherent alignment $\mathcal{A}$. This results in an incoherent aligned ontology, i.e., at least one of the concepts in the aligned ontology is unsatisfiable. Any assertion in the ABox that uses such a concept results directly in the inconsistency of the aligned ontology. As a result it is not possible to reason with standard methods in the aligned ontology. This is related to the definition of entailment (Definition 10). An assertion $C(a)$ is entailed by $\mathcal{A}_{\mathcal{S}}(\mathcal{O}_1, \mathcal{O}_2)$ iff each model of $\mathcal{A}_{\mathcal{S}}(\mathcal{O}_1, \mathcal{O}_2)$ is also a model for $C(a)$. Given the inconsistency of $\mathcal{A}_{\mathcal{S}}(\mathcal{O}_1, \mathcal{O}_2)$, this is the case for each arbitrary $C(a)$. This means that every kind of assertion follows from the aligned ontology. The information stored in the databases of C_1 and C_2 has become completely useless.

There are two possible ways to cope with inconsistencies. On the one hand one can try to repair inconsistent ontologies by removing erroneous axioms and/or assertions. This approach has been suggested by Schlobach and Cornet [SC03] and the same approach is the underlying principle applied in this thesis. However, we repair the alignment that is the root of the problem before we add it to the merged ontology. On the other hand one can try to reason with inconsistent ontologies. For this purpose different techniques have been suggested (see for example [HvHtT05]). However, to our knowledge none of these techniques has been implemented as part of semantic tools used by a larger community.

3.2.2 Data Transformation

Let us consider a concrete example to understand the effects of incoherence in the context of data transformation. This time C_2 takes over C_1. The CIO of C_2 decides to migrate all instance data of $\mathcal{O}_1$ into $\mathcal{O}_2$. $\mathcal{O}_1$ will no longer be maintained. A terminological alignment $\mathcal{A}$ between $\mathcal{O}_1$ and $\mathcal{O}_2$ has to be created to migrate the instances of $\mathcal{O}_1$ to $\mathcal{O}_2$ in a fully automated way.

Given a terminological alignment $\mathcal{A}$ between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ with signature $\mathbf{S}_1 = \langle \mathbf{C}_1, \mathbf{P}_1, \mathbf{R}_1, \mathbf{I}_1 \rangle$ and $\mathbf{S}_2 = \langle \mathbf{C}_2, \mathbf{P}_2, \mathbf{R}_2, \mathbf{I}_2 \rangle$, the process of data transformation contains at least the following steps.

1. For all concept correspondences $\langle C_{\#1}, D_{\#2}, r \rangle \in \mathcal{A}$ with $r \in \{\equiv, \sqsubseteq\}$ and for all $a \in \mathbf{I}_1$ with $\mathcal{O}_1 \models C_{\#1}(a)$ add the assertion $D_{\#2}(a)$ to $\mathcal{O}_2$.

2. For all correspondences $\left\langle P_{\#1}, Q_{\#2}, r \right\rangle \in \mathcal{A}$ with $r \in \{\equiv, \sqsubseteq\}$ where $P_{\#1}$ and $Q_{\#2}$ denote object property descriptions and for all $a, b \in \mathbf{I}_1$ with $\mathcal{O}_1 \models P_{\#1}(a, b)$ add the assertion $Q_{\#2}(a, b')$ to $\mathcal{O}_2$.

3. For all correspondences $\langle R_{\#1}, S_{\#2}, r \rangle \in \mathcal{A}$ with $r \in \{\equiv, \sqsubseteq\}$ where $R_{\#1}$ and $S_{\#2}$ denote data properties and for all data values d as well as for all $a \in \mathbf{I}_1$ with $\mathcal{O}_1 \models R_{\#1}(a, d)$ add the assertion $S_{\#2}(a, d)$ to $\mathcal{O}_2$.

In the following we refer to the ontology that results from applying this minimal set of migration rules as $\mathcal{A}(\mathcal{O}_1) \rightarrow \mathcal{O}_2$. The proposed procedure should be acceptable with regard to any reasonable alignment semantics. In particular, it is weaker than the natural semantics, i.e., for each assertion α with $\mathcal{A}(\mathcal{O}_1) \rightarrow \mathcal{O}_2 \models \alpha$ we have $\mathcal{A}_{\mathcal{S}_n}(\mathcal{O}_1, \mathcal{O}_2) \models \alpha$, while the inverse assumption does not hold.

At first sight, alignment coherence seems to be irrelevant with respect to this use case, because we do not copy any of the terminological axioms. The TBox of ontology $\mathcal{O}_2$ is not affected by the data migration. Consider the following correspondences to understand why this impression is deceptive.

$$\langle \mathit{Person}_{\#1}, \mathit{Person}_{\#2}, \equiv \rangle \tag{3.4}$$

$$\left\langle \mathit{ProjectLeader}_{\#1}, \mathit{Project}_{\#2}, \sqsubseteq \right\rangle \tag{3.5}$$

Let now $\mathcal{A}$ contain correspondences (3.4) and (3.5). Suppose that $\mathit{Project}_{\#2}$ and $\mathit{Person}_{\#2}$ are disjoint concepts. Further, let $\mathit{ProjectLeader}_{\#2}$ – directly or entailed

– be a subclass of $Person_{\#2}$. Due to the alignment $ProjectLeader_{\#1}$ is subsumed by both $Person_{\#2}$ and $Project_{\#2}$. $\mathcal{A}$ is thus incoherent because $ProjectLeader_{\#1}$ becomes unsatisfiable in $\mathcal{A}_{\mathcal{S}_n}(\mathcal{O}_1, \mathcal{O}_2)$.

Suppose now, there exists an instance a with $ProjectLeader_{\#1}(a)$. Applying the first migration rule results in both $Project_{\#2}(a)$ and $Person_{\#2}(a)$. Due to the disjointness of $Project_{\#2}$ and $Person_{\#2}$ there exists no model for $\mathcal{A}(\mathcal{O}_1) \rightarrow \mathcal{O}_2$ and thus $\mathcal{A}(\mathcal{O}_1) \rightarrow \mathcal{O}_2$ is an inconsistent ontology.

Opposed to our first impression there seems to be a tight link between the incoherence of $\mathcal{A}$ and the inconsistency of $\mathcal{O}_2(\mathcal{A}) \rightarrow \mathcal{O}_1$. The resulting problems are the same that we already described in the previous subsection. Standard reasoning tasks cannot be solved anymore. Moreover, the resulting problems are not only relevant for the newly added instances. The logical inconsistencies affect reasoning tasks related to the old instances in the same way.

3.2.3 Query Processing

In the following we revisit a variant of the example given above to explain the use case of query processing. Again, company C_2 takes over C_1. But this time both $\mathcal{O}_1$ and $\mathcal{O}_2$ are maintained. Instead of migrating all instances from $\mathcal{O}_1$ to $\mathcal{O}_2$, queries are rewritten at runtime to enable information integration between $\mathcal{O}_1$ and $\mathcal{O}_2$. A terminological alignment is the key for information integration. It is used for processing queries to generate result sets that contain data from both ontologies.

Solving the problem of query processing and its inherent task of translating a query, requires to choose an appropriate query language. As we are not concerned with solving this specific problem, we argue on an abstract level instead of discussing, e.g., characteristics of a SPARQL implementation (see [EPS08] and [CSM$^+$10] as an example for a concrete proposal). A query language for $\mathcal{SHOIN}(\mathbf{D})$ ontologies should at least support instance retrieval for complex concept descriptions. Depending on the concrete query language there might be a complex set of rewriting rules. Similar as above, we formalize a minimal set of rules. Given a query q, we suggest two simple rules.

R_1: If $\mathcal{O}_i \models C_{\#i} \equiv D_{\#i}$, then q can be transformed into an equivalent query by replacing all occurrences of $C_{\#i}$ by $D_{\#i}$.

R_2: If there exists a correspondence $\langle C_{\#i}, D_{\#j}, \equiv \rangle \in \mathcal{A}$, then q can be transformed into an equivalent query by replacing all occurrences of $C_{\#i}$ by $D_{\#j}$.

Both are simple substitution rules that allow to replace a concept or property by an equivalent concept or property. The equivalence can be derived from one of the ontologies (R_1) or can be stated in the alignment (R_2).

Suppose we query for the name of all project leaders, formally speaking we are interested in the instances of $\exists hasName^{-1}_{\#1}.ProjectLeader_{\#1}$. To receive instances of both $\mathcal{O}_1$ and $\mathcal{O}_2$ we have to rewrite the query for $\mathcal{O}_2$. Now let $\mathcal{A}$ contain correspondences (3.6), (3.7), and (3.8).

$$\langle \mathit{hasName}_{\#1}, \mathit{name}_{\#2}, \equiv \rangle \tag{3.6}$$

$$\left\langle \mathit{Project}_{\#1}, \mathit{Project}_{\#2}, \equiv \right\rangle \tag{3.7}$$

$$\left\langle \mathit{manages}_{\#1}, \mathit{managerOf}_{\#2}, \equiv \right\rangle \tag{3.8}$$

Suppose that $\mathcal{O}_1$ contains axiom $\mathit{ProjectLeader}_{\#1} \equiv \exists \mathit{manages}_{\#1}.\mathit{Project}_{\#1}$. We exploit this axiom by applying R_1. Now for every concept and property name that occurs in $\exists \mathit{hasName}^{-1}_{\#1}.\exists \mathit{manages}_{\#1}.\mathit{Project}_{\#1}$, there exists a direct counterpart in $\mathcal{O}_2$ specified by $\mathcal{A}$. By applying R_2 we thus finally end with a concept description based on the signature of $\mathcal{O}_2$.

$$\exists \mathit{hasName}^{-1}_{\#1}.\mathit{ProjectLeader}_{\#1} \tag{3.9}$$

$$\stackrel{R_1}{\Longleftrightarrow} \quad \exists \mathit{hasName}^{-1}_{\#1}.\exists \mathit{manages}_{\#1}.\mathit{Project}_{\#1} \tag{3.10}$$

$$\stackrel{R_2}{\Longleftrightarrow} \quad \exists \mathit{name}^{-1}_{\#2}.\exists \mathit{managerOf}_{\#2}.\mathit{Project}_{\#2} \tag{3.11}$$

What happens if we process the query based on this concept description to $\mathcal{O}_2$? As result we receive the empty set. The range of $\mathit{managerOf}_{\#2}$ is concept $\mathit{ProductLine}_{\#2}$, and $\mathit{ProductLine}_{\#2}$ is defined to be disjoint with $\mathit{Project}_{\#2}$. Thus, for logical reasons there exists no instance of concept description (3.11) in $\mathcal{O}_2$.

This problem is caused by the incorrectness of correspondence (3.8). But the incorrectness of (3.8) does not only affect the query under discussion. It also causes $\mathcal{A}$ to become incoherent, because in the aligned ontology concept $\mathit{ProjectLeader}_{\#1}$ becomes unsatisfiable due to its equivalence with concept description $\exists \mathit{manages}_{\#1}.\mathit{Project}_{\#1}$. This time we find a strong link between alignment incoherence and the emptiness of the result set when the alignment is used for rewriting the query.

Overall, we conclude that alignment incoherence results in three important scenarios in severe problems. We reported about failures to generate the requested results or, even more, meaningful results cannot be generated at all. In the introduction we raised the question about the consequences of using an incoherent alignment in an application (R1). Within the previous three subsections we gave an answer to this question.

Part II

Methods

Chapter 4

Alignment Diagnosis

> *How often have I said to you that when you have eliminated the impossible, whatever remains, however improbable, must be the truth (Arthur Conan Doyle).*

This chapter starts with a quote of a Sherlock Holmes novel [Doy90] where Mr. Holmes explains the principle of his approach to Dr. Watson. The principle underlying this thesis is very similar. Mr. Holmes eliminates the impossible to find the truth given a set of hypotheses concerned with the circumstances of a crime. We eliminate incoherent combinations of correspondences in an alignment to find a correct subset of the alignment. Mr. Holmes believes that his approach will finally lead him to the truth, i.e., a correct description of a sequence of events related to the crime. Indeed, we know that our approach will (in most cases) not lead us to a perfect alignment, however, we expect the resulting alignment to contain less incorrect correspondences compared to the incoherent alignment.

A crucial difference between the approach of Mr. Holmes and the approach proposed in this thesis is related to the *'elimination of the impossible'*. With regard to alignment incoherence, an incoherence is resolved by identifying combinations of correspondences resulting in an incoherent alignment. For each of these combination we have to remove at least one correspondence to arrive at a coherent subset of the alignment. Contrary to Mr. Holmes, we cannot *'remove the impossible'* itself, but have to remove one of its causes. Thus, there exists not a single solution but there are competing solutions to the same problem.

In Section 4.1 we describe our approach as a special kind of diagnostic method. Reiter [Rei87] coined the notion of diagnosis and used it in the context of reasoning-based system analysis. We pick up his terminology and show how to apply it to the problem of diagnosing alignment incoherence. Additionally, we have to define formally what we described so far as incoherent combination of correspondences. For this purpose we pick up a terminology proposed by Schlobach and Cornet in [SC03] and introduce the notion of a minimal incoherence preserving subalignment.

As already argued, there are often several coherent subalignments of the same incoherent alignment, i.e., we have several diagnosis for the same problem. In Section 4.2 we define two specific types of diagnosis. The first type will be referred to as local optimal diagnosis. It can be understood as a diagnosis that is optimal as long as we are concerned with each incoherent combination of correspondences on its own. The second type of diagnosis is optimal from a global point of view. i.e., it is the diagnosis that exceeds all other diagnoses with respect to a certain criteria.

We first discussed the problem of alignment incoherence as diagnostic problem in [MST06] and later on in [MTS07]. The terminology of incoherence preserving subalignments is based on the work of Schlobach [SC03]. We first adapted it to the alignment diagnosis problem in [MS09b]. The definition of a local optimal diagnosis as well as the propositions and proofs of Section 4.2.1 we first presented in [MS09b].

4.1 A Theory of Diagnosis

Throughout this chapter we use $\mathcal{A}$ to refer to an alignment between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. Furthermore, we use $\mathcal{O}$ with or without subscript to refer to an ontology. For the sake of conciseness we omit the prefix $\mathcal{S}$ in expressions as *'$\mathcal{S}$-unsatisfiable'* and *'$\mathcal{S}$-incoherent'*. For all of the following definitions we thus have to keep in mind an implicit reference to a reductionistic alignment semantics $\mathcal{S}$.

As already indicated in [MTS07] and later on in [MS09b], the problem of resolving incoherences in ontology alignments can be understood as a specific instance of a problem generally described by Reiters theory of diagnosis [Rei87]. Reiter describes a diagnostic problem in terms of a system and its components. The need for a diagnosis arises, when the observed system behavior differs from the expected behaviour. According to Reiter, the diagnostic problem is to determine a set of those system components which, when assumed to be functioning abnormally, explain the discrepancy between observed and correct behaviour. If this set of components is minimal, it is referred to as diagnosis Δ. In our context a system is a tuple $\langle \mathcal{A}, \mathcal{O}_1, \mathcal{O}_2, \mathcal{S} \rangle$. The discrepancies between observed and correct behaviour are the terminological entities that were satisfiable in $\mathcal{O}_1$ and $\mathcal{O}_2$ and have become unsatisfiable in $\mathcal{A}_{\mathcal{S}}(\mathcal{O}_1, \mathcal{O}_2)$. The components of the system are the axioms of $\mathcal{O}_1$ and $\mathcal{O}_2$ as well as the correspondences of $\mathcal{A}$. However, with respect to our problem the set of possibly erroneous components is restricted to the correspondences of $\mathcal{A}$. Thus, an alignment diagnosis is defined as a minimal set $\Delta \subseteq \mathcal{A}$ such that $\mathcal{A} \setminus \Delta$ is coherent.

Definition 23 (Alignment Diagnosis)**.** *$\Delta \subseteq \mathcal{A}$ is a diagnosis for $\mathcal{A}$ with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ iff $\mathcal{A} \setminus \Delta$ is coherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ and for each $\Delta' \subset \Delta$ alignment $\mathcal{A} \setminus \Delta'$ is incoherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$.*

However, an alignment diagnosis can also be characterized in a different way, which sheds more light on the problem of computing the diagnosis. To understand the alternative specification of a diagnosis, which will be presented in Proposition 4, we have to recall and adopt some notions widely used in the field of ontology debugging. We start with the definition of a minimal incoherence preserving sub-TBox (MIPS). Note that a MIPS is referred to as minimal conflict set in terms of Reiters theory. Given an incoherent ontology $\mathcal{O}$, a MIPS is an incoherent set of axioms $\mathcal{M} \subseteq \mathcal{O}$ such that any proper subset $\mathcal{M}' \subset \mathcal{M}$ is coherent. This definition has been given by Schlobach and Cornet in [SC03]. We give a similar definition to characterize minimal incoherent subsets of an incoherent alignment.

Definition 24 (MIPS Alignment)**.** *$\mathcal{M} \subseteq \mathcal{A}$ is a minimal incoherence preserving subalignment (MIPS alignment) of $\mathcal{A}$, iff $\mathcal{M}$ is incoherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ and there exists no $\mathcal{M}' \subset \mathcal{M}$ such that $\mathcal{M}'$ is incoherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$. The collection of all MIPS alignments is referred to as MIPS $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$.*

Compared to a MIPS, we define a MUPS alignment as a minimal unsatisfiability preserving subalignment. Thus, a MUPS alignment is always related to some specific aligned unsatisfiable concept or property. The notion of a MUPS has first been introduced for incoherent ontologies in [SC03].

Definition 25 (MUPS Alignment)**.** *Given a concept or property $C_{\#i}$ with $i \in \{1, 2\}$ unsatisfiable due to $\mathcal{A}$. $\mathcal{M} \subseteq \mathcal{A}$ is a minimal unsatisfiability preserving subalignment (MUPS alignment) with respect to $C_{\#i}$, iff $C_{\#i}$ is unsatisfiable due to $\mathcal{M}$ and there exists no $\mathcal{M}' \subset \mathcal{M}$ such that $C_{\#i}$ is unsatisfiable due to $\mathcal{M}'$. The collection of all MUPS alignments is referred to as MUPS $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$.*

The notions of MIPS and MUPS are obviously closely related. The following proposition describes the interrelation between MIPS and MUPS.

Proposition 3 (MIPS and MUPS)**.** *For each $M \in$ MUPS $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ there exists $M' \in$ MIPS $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ such that $M' \subseteq M$ and for each $M' \in$ MIPS $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ we have $M' \in$ MUPS $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$, i.e., MIPS $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2) \subseteq$ MUPS $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$.*

Proof. Given an alignment M, which is a MUPS with respect to $C_{\#i}$, If we remove one of its correspondences, M will no longer be a MUPS for $C_{\#i}$. However, M can still be an incoherent alignment due to the unsatisfiability of some other concept or property. This means that M contains a MIPS as subalignment or is already a MIPS. On the other hand we know that each MIPS causes the unsatisfiability of at least one entity and it is thus a MUPS for that entity. □

Reiter argues that a diagnosis is a minimal hitting-set over the set of all minimal conflict sets. Let us recall the general notion of a hitting-set from set theory.

Definition 26 (Hitting-set)**.** *Given a set T and a collection $\mathcal{S} = \{S_1, \ldots, S_n\}$ with $S_i \subseteq T$ for $i = 1 \ldots n$. $H \subseteq T$ is a hitting-set for $\mathcal{S}$ iff $H \cap S_i \neq \emptyset$ for $i = 1 \ldots n$. $H \subseteq T$ is a minimal hitting-set for $\mathcal{S}$ iff H is a hitting-set for $\mathcal{S}$ and there exists no $H' \subset H$ such that H' is a hitting-set for $\mathcal{S}$.*

As explained above, a minimal conflict set in the general theory of diagnosis is equivalent to a MIPS in the context of diagnosing ontology alignments. A diagnosis for an incoherent alignment $\mathcal{A}$ is thus a minimal hitting-set for *MIPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$.

Proposition 4 (Diagnosis and Minimal Hitting-set)**.** $\Delta \subseteq \mathcal{A}$ *is a diagnosis for* $\mathcal{A}$ *with respect to* $\mathcal{O}_1$ *and* $\mathcal{O}_2$*, iff* Δ *is a minimal hitting-set for MIPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$.

Proof. Proposition 4 is a special case of corollary 4.5 in [Rei87] where Reiter gives an accordant proof. □

A family of approaches for computing a diagnosis is based on Proposition 4. The origin of these algorithms can be found in Reiters publication [Rei87]. An example for its application to ontology debugging is given in [SHCvH07]. For alignment debugging a similar approach has been described by Qi et al. in [QHH$^+$08]. All of these approaches have in common to compute a hitting-set over all MIPS using a variant of the hitting-set-tree algorithm. Details can be found in the chapter on related work (Chapter 11).

We will see that our approach differs from the commonly used strategy. In the next section we propose two types of diagnosis that single out a reasonable subset from the set of all possible diagnosis. However, it will turn out that both approaches are not only differently motivated, but will also result for many concrete cases in different solutions and require specific methods to be computed.

4.2 Types of Diagnosis

In the following we present an example that helps us to understand why to prefer certain diagnoses over their alternatives. Figure 4.1 depicts four different allocations of *MIPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_1)$ for an incoherent alignment $\mathcal{A}$ with five correspondences. According to Definition 23, the following enumeration is a complete listing of all possible diagnoses.

$$\begin{aligned}
\{a\}, \{d\}, \{e\} & \qquad \textit{(Subfigure I)} \\
\{a, c\}, \{d\}, \{e, c\} & \qquad \textit{(Subfigure II)} \\
\{b, d\}, \{c, e\} & \qquad \textit{(Subfigure III)} \\
\{a, b, e\}, \{c, d\}, \{a, c, e\}, \{a, b, d\} & \qquad \textit{(Subfigure IV)}
\end{aligned}$$

Without any additional information, our criteria for judging different diagnosis are quite limited. A prominent approach requests us to chose the smallest diagnosis from the set of all possible diagnoses. This approach is substantiated by the principle of minimal change, an important principle in the AGM theory of belief revision [Gae92]. Based on this principle we can rule out some diagnosis (see Subfigure II and IV). Nevertheless, for the alignment depicted in Subfigure I all diagnosis contain exactly one correspondence. Similar for Subfigure III, here we

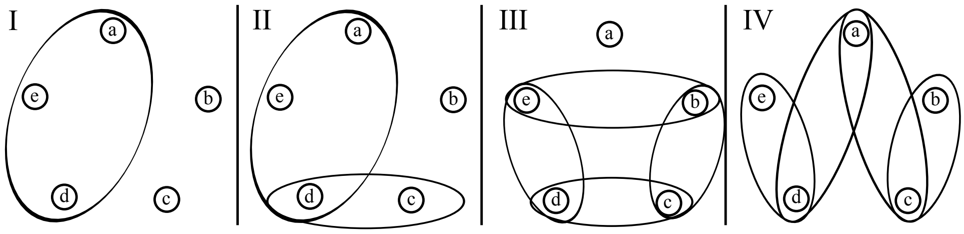

Figure 4.1: Four examples of an incoherent alignment that consists of five correspondences a, b, c, d and e. For each of the four alignments all MIPS alignments *MIPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_1)$ are depicted as ellipses.

have two diagnosis and both consist of two correspondences. A principle of minimal change that is based on the number of correspondences to be eliminated is thus useless in many situations.

However, we have not yet taken into account an important source of evidence. In many cases automatically generated alignments are additionally annotated by confidence values. These values are computed from the outcome of several algorithms that are concerned with a variety of different ontological aspects. Confidence values can also be added by human experts. Remember that a confidence value is used to express the degree of trust in the correctness of a correspondence. This additional source of evidence is crucial for finding a reasonable diagnosis. In the following we show how it can be taken into account in an appropriate way.

4.2.1 Local Optimal Diagnosis

In this section we formally define the notion of a local optimal diagnosis and motivate its appropriateness by discussing the examples depicted in Figure 4.1. We refer to the four alignments of subfigures I to IV as alignment $\mathcal{A}_I$, $\mathcal{A}_{II}$, $\mathcal{A}_{III}$, and $\mathcal{A}_{IV}$. Further we use $\alpha : \mathcal{A} \rightarrow [0, 1]$ to refer to a function that assigns a confidence value to each correspondence in an alignment.

We start our considerations with a simple principle. This principle will be refined later on step by step. For $\mathcal{A}_I$ there are three diagnoses. Suppose now that we have $\alpha(a) = 0.9$, $\alpha(b) = 0.8$, $\alpha(c) = 0.7$, $\alpha(d) = 0.6$, and $\alpha(e) = 0.5$. Knowing that we have to remove one of a, d, e and given the knowledge of α, we would choose $\{e\}$ as diagnosis from the set of possible diagnoses listed above. According to this choice, we solve the underlying problem be removing the correspondence with lowest confidence from the alignment. Our choice is based on the general rule of thumb *'From each MIPS $\mathcal{M}$ remove the correspondence with lowest confidence'*. However, this principle needs to be refined in order to be applicable to overlapping conflicts. We have to add the rider *'... unless another correspondence in $\mathcal{M}$ has not yet been removed'*.

In [MST06] we described an algorithm that is based on a slightly modified version of this principle. It was our first approach to resolve alignment incoherence based on reasoning in DDL. The algorithm randomly choses a conflicting set of correspondences and removes the correspondence with lowest confidence. It terminates until no further conflicts can be found. However, this algorithm leads sometimes to unreasonable choices. Moreover, the result of the algorithm depends on the order in which conflicts are processed. An example arises from $\mathcal{A}_{II}$ and the confidence allocation described above. If we process $\{c, d\}$ at first, we have to remove d. With this choice we resolve the second MIPS $\{a, d, e\}$ at the same time and our final diagnosis is $\{d\}$. If we start with $\{a, d, e\}$, we will first remove $\{e\}$. We continue with $\{c, d\}$ and are forced to remove d. As a result of the approach we have removed $\{d, e\}$ and have to discover that we have removed a non-minimal hitting set, i.e., we have not constructed a diagnosis.

The crucial point is to model the interdependencies between candidates for a removal in an appropriate way. The following recursive definition introduces the notion of an accused correspondence to solve this problem.

Definition 27 (Accused Correspondence)**.** *Correspondence $c \in \mathcal{A}$ is accused by $\mathcal{A}$ with respect to $\mathcal{O}_1$, $\mathcal{O}_2$ and α iff there exists some $\mathcal{M} \in MIPS\,(\mathcal{A}, \mathcal{O}_2, \mathcal{O}_2)$ with $c \in \mathcal{M}$ such that for all $c' \in \mathcal{M} \setminus \{c\}$ we have*

(1) $\alpha(c') > \alpha(c)$ and

(2) c' is not accused by $\mathcal{A}$ with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$.

We have chosen the term 'accused correspondence' because the correspondence with lowest confidence in a MIPS alignment $\mathcal{M}$ is 'accused' to cause the problem. This charge will be rebutted if one of the other correspondences in $\mathcal{M}$ is already accused due to the existence of another MIPS alignment.

The notion of an accused correspondence reminds of Dung argumentation framework [Dun95], that is based on an attack-relation defined on a set of arguments. Indeed, we show in Section 11.2 how to define the attack relation for analyzing incoherent alignments in such a way that the set of accused correspondences is a preferred extension in Dungs framework.

Problems emerge if we are concerned with an alignment that contains a MIPS $\mathcal{M}$ with $c \neq c' \in \mathcal{M}$ and $\alpha(c) = \alpha(c') = argmin_{c \in \mathcal{M}} \alpha(c)$. According to Definition 27 none of the correspondences in $\mathcal{M}$ is accused. For that reason, we demand in the following that α imposes a strict order on $\mathcal{A}$, i.e., $\alpha(c) < \alpha(d) \vee \alpha(c) > \alpha(d)$ for each $c \neq d \in \mathcal{A}$. The requirement is not realistic for many matching systems and the confidences α that they generate. In its practical application we have to fall back to an additional criteria α' (or even a set of criteria $\alpha', \alpha'', \ldots$) with $\alpha'(c) \neq \alpha'(d)$. However, in the following we neglect this aspect for the sake of simplicity and treat α as a confidence function which imposes a strict order on $\mathcal{A}$.

The recursive character of Definition 27 allows us to infer the following proposition. We will later propose an algorithm for computing the set of accused correspondences. Proposition 5 is crucial for the correctness of this algorithm.

Proposition 5. *Let $\mathcal{A}' \dot\cup \mathcal{A}''$ be a disjoint union of $\mathcal{A}$ with $argmin_{c\in\mathcal{A}'}\alpha(c) > argmax_{c\in\mathcal{A}''}\alpha(c)$. Correspondence $c \in \mathcal{A}'$ is accused by $\mathcal{A}'$ iff c is accused by $\mathcal{A}$.*

Proof. Suppose that Proposition 5 is incorrect and let $\mathcal{A}^*$ and $\mathcal{A}^\dagger$ be defined as $\mathcal{A}^* = \{c \in \mathcal{A}' \mid c \text{ is accused by } \mathcal{A}' \text{ and is not accused by } \mathcal{A}\}$ and $\mathcal{A}^\dagger = \{c \in \mathcal{A}' \mid c \text{ is accused by } \mathcal{A} \text{ and is not accused by } \mathcal{A}'\}$. It follows that $\mathcal{A}^* \cup \mathcal{A}^\dagger \neq \emptyset$, and in particular that there exists a correspondence $\tilde{c} = argmax_{c\in\mathcal{A}^*\cup\mathcal{A}^\dagger}\alpha(c)$. In the following we show that there exists no such $\tilde{c}$ and thus we indirectly prove the correctness of Proposition 5. First suppose that $\tilde{c} \in \mathcal{A}^*$ and $\tilde{c} \notin \mathcal{A}^\dagger$. It follows that there exists $\mathcal{M} \in \textit{MIPS}\,(\mathcal{S}, \mathcal{A}', \mathcal{O}_1)\,\mathcal{O}_2$ such that $\tilde{c} = argmin_{c\in\mathcal{M}}\alpha(c)$ and all $c \in \mathcal{M} \setminus \{\tilde{c}\}$ are not accused by $\mathcal{A}'$. We also know that $\textit{MIPS}\,(\mathcal{S}, \mathcal{A}', \mathcal{O}_1)\,\mathcal{O}_2 \subseteq \textit{MIPS}\,(\mathcal{S}, \mathcal{A}, \mathcal{O}_1)\,\mathcal{O}_2$ and thus $\mathcal{M} \in \textit{MIPS}\,(\mathcal{S}, \mathcal{A}, \mathcal{O}_1)\,\mathcal{O}_2$. Since $\tilde{c}$ is not accused by $\mathcal{A}$ it follows that there exists $c^\dagger \in \mathcal{M} \setminus \{\tilde{c}\}$ with $\alpha(\tilde{c}) < \alpha(c^\dagger)$ which is accused by $\mathcal{A}$ and not accused by $\mathcal{A}'$. Thus, $\alpha(\tilde{c}) < \alpha(c^\dagger)$ and $\alpha(c^\dagger) \in \mathcal{A}^\dagger \subseteq \mathcal{A}^* \cup \mathcal{A}^\dagger$ contradicting our assumption. Now suppose that $\tilde{c} \notin \mathcal{A}^*$ and $\tilde{c} \in \mathcal{A}^\dagger$. Again, it follows that there exists $\mathcal{M} \in \textit{MIPS}\,(\mathcal{S}, \mathcal{A}, \mathcal{O}_1)\,\mathcal{O}_2$ such that $\tilde{c} = argmin_{c\in\mathcal{M}}\alpha(c)$ and all $c \in \mathcal{M} \setminus \{\tilde{c}\}$ are not accused by $\mathcal{A}$. We also know that $\mathcal{M} \in \textit{MIPS}\,(\mathcal{S}, \mathcal{A}', \mathcal{O}_1)\,\mathcal{O}_2$ since $\tilde{c} \in \mathcal{A}'$ and $\alpha(c) \geq \alpha(\tilde{c})$ for all $c \in \mathcal{M}$. Since $\tilde{c}$ is not accused by $\mathcal{A}'$ it follows that there exists $c^* \in \mathcal{M} \setminus \{\tilde{c}\}$ which is accused by $\mathcal{A}'$ and not accused by $\mathcal{A}$. Thus, $\alpha(\tilde{c}) < \alpha(c^*)$ and $\alpha(c^*) \in \mathcal{A}^* \subseteq \mathcal{A}^* \cup \mathcal{A}^\dagger$ again contradicting our assumption that there exists an element in $\mathcal{A}^* \cup \mathcal{A}^\dagger$ with highest confidence.[1] □

The following proposition states that the set of all accused correspondences forms a diagnosis, i.e., is a minimal hitting set over $\textit{MIPS}\,(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$. We give an explicit proof.

Proposition 6. *$\Delta = \{c \in \mathcal{A} \mid c \text{ is accused by } \mathcal{A} \text{ with respect to } \mathcal{O}_1 \text{ and } \mathcal{O}_2\}$ is a diagnosis for $\mathcal{A}$ with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$.*

Proof. Let Δ be the alignment which consists of those and only those correspondences accused by $\mathcal{A}$ with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$. Further let $\mathcal{M} \in \textit{MIPS}\,(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ be an arbitrarily chosen MIPS alignment and let $c^* = argmin_{c\in\mathcal{M}}\alpha(c)$ be the correspondence with lowest confidence in $\mathcal{M}$. Due to Definition 27 we know that c^* is either accused by $\mathcal{A}$ or there exists some $c' \neq c^* \in \mathcal{M}$ which is accused by $\mathcal{A}$. Thus, for each $\mathcal{M} \in \textit{MIPS}\,(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ there exists a correspondence $c \in \mathcal{M}$ such that $c \in \Delta$. We conclude that Δ' is a hitting set for $\textit{MIPS}\,(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$. Let now $\tilde{c}$ be an arbitrarily chosen element from Δ'. Due to Definition 27 there exists a MIPS $\mathcal{M} \in \textit{MIPS}\,(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ with $\mathcal{M} \cap \Delta' = \{\tilde{c}\}$. Thus, $\mathcal{A}' \setminus \tilde{c}$ is no hitting set for $\textit{MIPS}\,(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ for any $\tilde{c} \in \mathcal{A}'$ which means that Δ is a minimal hitting set. Based on proposition 4 we conclude that $\mathcal{A}'$ is a diagnosis. □

According to the notion of an accused correspondences, the whole collection $\textit{MIPS}\,(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ is not taken into account from a global point of view. Each

[1] We would like to thank Anne Schlicht for proposing a sketch of this proof.

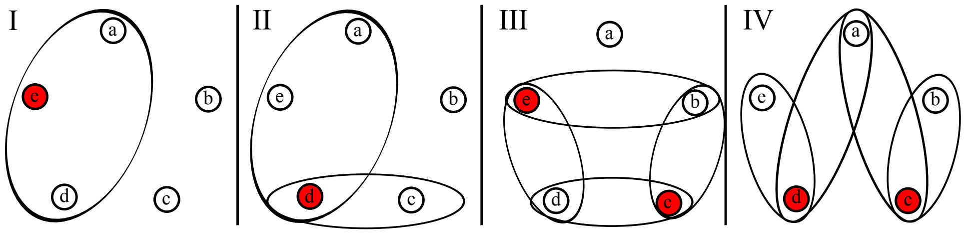

Figure 4.2: Given a confidence distribution α with $\alpha(a) = 0.9$, $\alpha(b) = 0.8$, $\alpha(c) = 0.7$, $\alpha(d) = 0.6$, and $\alpha(e) = 0.5$, local optimal diagnoses are marked as filled circles.

removal decision is the optimal choice with respect to the concrete MIPS under discussion. Therefore, we define the resulting set of correspondences as local optimal diagnosis.

Definition 28 (Local Optimal Diagnosis)**.** *A diagnosis Δ such that all $c \in \Delta$ are accused by $\mathcal{A}$ with respect to $\mathcal{O}_1$, $\mathcal{O}_2$ and α is referred to as local optimal diagnosis.*

Let us take a look at the examples introduced in Figure 4.1. In Figure 4.2 we show the same alignments and have additionally marked the correspondences belonging to the local optimal diagnosis as filled circles. These diagnoses result from a confidence distribution α already specified at the beginning, i.e., $\alpha(a) = 0.9$, $\alpha(b) = 0.8$, $\alpha(c) = 0.7$, $\alpha(d) = 0.6$, and $\alpha(e) = 0.5$. In the following we discuss these examples in detail.

Subfigure I: The local optimal diagnosis is $\Delta = \{e\}$. Based on the assumption that the confidence values impose a correct order of correspondences, it is the most reasonable choice to remove e from $\mathcal{A}$.

Subfigure II: Although we have $e = argmin_{x\in\{c,d,e\}}\alpha(x)$, e is nevertheless not accused. This is based on the fact that d is already accused due to $\{c, d\} \in$ *MIPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ whereas c is not accused. Notice that c cannot be accused, because there exists no MIPS $\mathcal{M}$ such that $c = argmin_{x\in\mathcal{M}}\alpha(x)$. Thus, $\Delta = \{d\}$ is the local optimal diagnosis. Again, there exists no reason for choosing one of the other diagnosis $\{c, e\}$ or $\{a, c\}$.

Subfigure III: The local optimal diagnosis is $\Delta = \{c, e\}$. We prefer it over $\{b, d\}$ because $\alpha(b) > \alpha(c)$ and $\alpha(d) > \alpha(e)$. Again, we find the local optimal diagnosis by first determining that b is not accused. This leads to the conclusion that c is accused. Therefore, d is not accused and finally e is accused because of the remaining MIPS $\{d, e\}$ and $\{b, e\}$.

Subfigure IV: We first notice that $a = argmax_{c\in\mathcal{A}}\alpha(c)$ cannot be accused. It follows that both c and d are accused, because they are the *'weakest'* correspondences in a MIPS where none of the other correspondences is accused.

Since $\{c, d\}$ forms a diagnosis, we can conclude that $\Delta = \{c, d\}$ is already a local optimal diagnosis.

Notice that we implicitly used Proposition 5 in our considerations. Look for example at Subfigure IV. We started our consideration with the statement that a is not accused. This insight is based on dividing $\mathcal{A}$ in $\mathcal{A}' = \{a\}$ and $\mathcal{A}'' = \{b, c, d, e\}$. Obviously a is not accused by $\mathcal{A}'$, since $\mathcal{A}'$ is not even incoherent. Based on Proposition 5 we are justified to conclude that a is also not accused by $\mathcal{A} = \mathcal{A}' \cup \mathcal{A}''$. This train of thought is the basis for Algorithm 6, which will be introduced in Section 6.1.

So far, we argued with regard to all four examples that the local optimal diagnosis is the most reasonable choice. However, the quality of the local optimal diagnosis depends on the confidence values. For the same alignments there are confidence distributions resulting in a local optimal diagnosis that is not the best choice. In answer to this we introduce the notion of a global optimal diagnosis. Hereby we will revisit some of the examples discussing different confidence distributions.

4.2.2 Global Optimal Diagnosis

A global optimal diagnosis Δ is a diagnosis that exceeds all other correspondences with regard to a certain criteria. This criteria is the sum of confidence values of the correspondences in Δ. In particular, a global optimal diagnosis is the diagnosis with the lowest sum of confidences compared to any other diagnosis. Formally, it is defined as follows.

Definition 29. *A diagnosis Δ for $\mathcal{A}$ with respect to $\mathcal{O}_1$, $\mathcal{O}_2$ and α is a global optimal diagnosis iff there exists no diagnosis Δ' for $\mathcal{A}$ with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ such that $\sum_{c \in \Delta'} \alpha(c) < \sum_{c \in \Delta} \alpha(c)$.*

This definition is based on the idea to remove as less correspondences as possible, whereas each correspondence is weighted via its confidence. The problem of finding a global optimal diagnosis is equivalent to the problem of solving the weighted variant of the hitting set problem. This problem as well as its unweighted counterpart is known to be NP-complete [GJ79]. For that reason we can also describe a global optimal diagnosis as follows.

Proposition 7. *A diagnosis Δ for $\mathcal{A}$ with respect to $\mathcal{O}_1$, $\mathcal{O}_2$ and α is a global optimal diagnosis iff there exists no minimal hitting-set Δ' for MIPS $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ such that $\sum_{c \in \Delta'} \alpha(c) < \sum_{c \in \Delta} \alpha(c)$*

Proof. The correctness of this proposition follows directly from Proposition 4 applied to Definition 23. Proposition 4 says that each diagnosis is a minimal hitting set over *MIPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$. □

The algorithm that we finally propose computes a global optimal diagnosis not as a hitting set over *MIPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$. Instead of that, we implicitly compute a

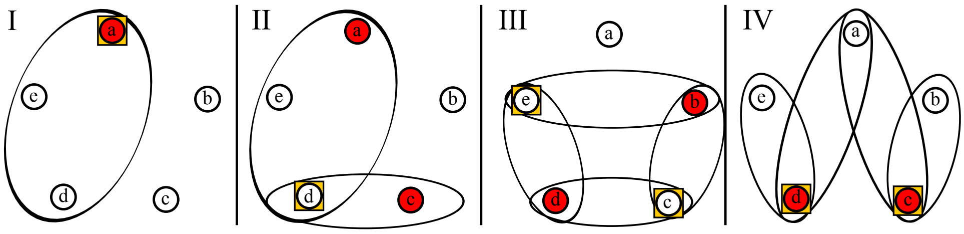

Figure 4.3: Given a confidence distribution α with $\alpha(a) = 0.6$, $\alpha(b) = 0.8$, $\alpha(c) = 0.5$, $\alpha(d) = 0.7$, and $\alpha(e) = 0.9$, local optimal diagnoses are marked by red circles and global optimal diagnoses are marked by yellow rectangles.

hitting set over *MUPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$. For that reason the following proposition will be required in Chapter 6.

Proposition 8. *A diagnosis Δ for $\mathcal{A}$ with respect to $\mathcal{O}_1$, $\mathcal{O}_2$ and α is a global optimal diagnosis iff there exists no minimal hitting-set Δ' for MUPS $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ such that $\sum_{c \in \Delta'} \alpha(c) < \sum_{c \in \Delta} \alpha(c)$.*

Proof. From Proposition 3 we know that *MIPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2) \subseteq$ *MUPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$. Thus, a minimal hitting set over *MUPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ is also a minimal hitting set over *MIPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$. We now have to show that the *'smallest weighted'* minimal hitting set over *MUPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ is also the *'smallest weighted'* minimal hitting set over *MIPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$, because the best solution for a MUPS hitting set might be worse than the best solution for a MIPS hitting set. Hover, this is not possible according to Proposition 3, because for each $M \in$ *MUPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ there exists $M' \in$ *MIPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ such that $M' \subseteq M$, which means that each hitting set over *MIPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ is also a hitting set over *MUPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$. $\square$

Let us consider the differences between a local optimal and a global optimal diagnosis. First of all, in many cases both types of diagnosis are the same. Examples can be found in Figure 4.2. For all of the four examples there is no difference between local and optimal diagnosis. However, changing the confidence distribution to $\alpha(a) = 0.6$, $\alpha(b) = 0.8$, $\alpha(c) = 0.5$, $\alpha(d) = 0.7$, and $\alpha(e) = 0.9$ we obtain different results for Subfigures II and III. The resulting diagnoses are depicted in Figure 4.3. Local optimal diagnoses are depicted by circles, global optimal diagnoses are depicted by rectangles.

It is obvious that in the special case of non-overlapping MIPS, there is no difference between local and global diagnoses. Subfigure I is a very simple example. Subfigure II illustrates a typical example for a difference between between local and global optimal diagnosis. Due to the definition of an accused correspondence (see Definition 27) a correspondence x is only accused if there exists some MIPS $\mathcal{M}$ such that $c = argmin_{x \in \mathcal{M}} \alpha(c)$. That means that many correspondences are excluded a priori as candidates for a local optimal diagnosis. A correspondence of this type can, nevertheless, be part of a global optimal diagnosis. A global optimal diagnosis contains d with $\alpha(d) = 0.7 < 1.1 = \alpha(a) + \alpha(c)$. Similar for Subfigure III, the confidence score of the competing diagnosis $\{b, d\}$ and $\{c, e\}$ is 1.5 and 1.4, respectively. The global optimal diagnosis for the diagnostic problem is thus $\{c, e\}$.

Note that the outcome of computing a local optimal diagnosis is highly sensitive to the order of confidence values. Similar to Subfigure II, we can easily construct an example where a local optimal diagnosis Δ_L for $\mathcal{A}$ consist of $|\mathcal{A}| - 1$ correspondences, while the global optimal diagnosis Δ_G for the same problem consist of exactly one correspondence such that $\Delta_L \cup \Delta_G = \mathcal{A}$. Although such MIPS structures are expected to occur rarely, this consideration shows the weakness of a local optimal diagnosis that we revisit in the context of our experiments in Chapter 9.

These kind of problems are avoided by computing a global optimal diagnosis, which is determined by an aggregation of confidence values. Computing the total of confidences, however, is based on an underlying semantics that we ascribe to the confidence value of a correspondence. Due to this semantics, we assume that it is appropriate to compare our trust in an alignments by comparing the confidence-total of its correspondences. Note that such a semantics for the numbers attached to correspondences, which we introduced as *'confidence values'*, goes beyond the weak semantics of an ordinal scale expressed by *'the higher the confidence value of* c*, the higher is our trust in* c*'*.

4.3 Summary

Within this chapter we have analyzed possible answers to R3: given an incoherent alignment $\mathcal{A}$, how can we characterize a coherent sub-alignment Δ in a reasonable way? First of all, we introduced the theory of diagnosis. On top of this theory we argued that such a subalignment has to be a diagnosis, which is defined as minimal set of correspondence Δ that resolves the problem if we retract Δ from $\mathcal{A}$. It is minimal in the sense that the removal of each proper subset of Δ is not sufficient.

However, for a concrete instance of a diagnostic problem there are often several competing diagnoses. Even though all of them are minimal, they can vary in number of correspondences and with respect to the confidence values attached to their elements. We have proposed two special types of diagnoses, introduced as local optimal and global optimal diagnosis. We have argued with the help of several examples that both types of diagnoses can be reasonable choices that should be preferred compared to other diagnoses.

The local optimal diagnosis builds on the principle to trust always in correspondences with higher confidence in case of a conflicts. In particular, it makes no difference if such a correspondence conflicts with one or several correspondences. This is different for the global optimal diagnosis. It is defined as the diagnosis with the lowest total of confidences compared to all other possible diagnosis. Note that it is not possible to proof that the global optimal diagnosis is always the perfect choice. For that reason we tried to illustrate with some examples that it is the most reasonable choice in many situations. Finally, we have to proof our assumption by an empirical analysis.

Chapter 5

Detecting and Isolating Incoherence

> *Audiences are always better pleased with a smart retort, some joke or epigram, than with any amount of reasoning (Charlotte Perkins Gilman).*[1]

In this chapter we describe the reasoning techniques used by the algorithms presented later on in Chapter 6. These techniques are designed to decide whether an alignment is incoherent or to detect a single MUPS or MIPS alignment in an incoherent alignment. Notice that we do not discuss algorithms for detecting all MUPS or MIPS in an incoherent alignment, which might be surprising at first sight. We will later see that our algorithms do not require such an input. This is one of the main differences to alternative approaches in the field of alignment debugging.

In Section 5.1 we present several straight forward techniques to reason about alignment coherence as well as a simplified variant of the *'expand-and-shrink-algorithm'* described in [Kal06]. We show how to apply this algorithm to the problem of detecting a single MUPS alignment. In addition, we present several interesting MUPS alignments based on correspondences generated for a concrete ontology matching problem. These MUPS alignments enable a better understanding of the complexity of the underlying reasoning problem.

In Section 5.2 we propose an incomplete reasoning technique to decide whether a pair of correspondences is incoherent. This reasoning technique is based on detecting certain patterns that result in incoherences. Our experimental results will later on show that even the very limited set of pattern we propose is sufficient to detect a large number of MUPS alignments.

[1] So here is the joke: Two Muffins were baking in an oven. One muffin turns to the other and says, *'Holy Shit it's hot in here!'* The other muffin says, *'Holy Shit... A talking muffin!'*

The reasoning procedures described in Section 5.1 are directly adapted from the ontology debugging approaches described for example in [SC03, SH05, Kal06]. The basic approach can also be found in the work of Qi et al. [QHH$^+$08]. Contrary to this, the use of incomplete but efficient pattern based reasoning techniques in the context of alignment debugging has been first developed as part of our work. In particular, we used this technique in [MS07b], later on in the context of manual revision of alignments [MSSZ09], and described it more elaborately in [MS09b]. Although similar techniques have partially been implemented in some matching systems – examples are the systems LILY [WX08a] and ASMOV [JMSK09] – an analysis of completeness and a relation to a well founded theory is to our knowledge missing.

5.1 Complete Reasoning Components

We require only a limited set of complete reasoning procedures as building blocks for the algorithms presented later on. We describe these reasoning procedures as complete because they never miss, for example, a MUPS alignment whenever there exists such a MUPS. This is not the case for the methods presented in the next section. In particular, the following reasoning procedures are required.

ISCOHERENT$_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$. Returns *true* if $\mathcal{A}$ is coherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$, *false* otherwise.

GETALIGNEDUSATENTITIES$_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$. Returns the set of all aligned unsatisfiable concepts and properties.

GETSOMEALIGNEDUSATENTITY$_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$. Returns some aligned unsatisfiable concept or property.

ISUSATENTITY$_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2, C_{\#i})$ with $i \in \{1, 2\}$. Returns *true* if $C_{\#i}$ is unsatisfiable, *false* otherwise.

MUPSWALK$_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2, C_{\#i})$ with $i \in \{1, 2\}$ and $C_{\#i}$ being an unsatisfiable concept or property. Returns a MUPS for $C_{\#i}$.

Prior to running any of these algorithms, we have to ensure that the aligned ontology – which can be coherent or incoherent – will never become inconsistent. Pay attention to the fact that for an inconsistent ontology we cannot apply standard reasoning techniques as argued in Section 3.2.1. An incoherent but consistent ontology, however, is not subject to these restrictions. For that reason we have to take care that the incoherence of the aligned ontology does not result in its inconsistency. We do so by applying a preprocessing step prior to any reasoning activities by removing the ABox of $\mathcal{O}_1$ and $\mathcal{O}_2$. Additionally, we replace every nominal concept description, listed as *(one of)* in Definition 3, by a new concept name.[2]

[2]Notice that for this reason our approach can only ensure the coherence of alignments between $\mathcal{SHIN}(\mathbf{D})$ TBoxes. Even though it will in most cases result in coherent alignments for

Algorithm 1 Checks the aligned unsatisfiability of a concept or property.

$\textsc{IsUSatEntity}_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2, C_{\#i})$

$\langle \mathbf{C}, \mathbf{P}, \mathbf{R}, \mathbf{I} \rangle \leftarrow$ *signature of* $\mathcal{O}_1 \cup \mathcal{O}_2$
if $C_{\#i} \in \mathbf{C}$ **then**
  if $\mathcal{A}_{\mathcal{S}}(\mathcal{O}_1, \mathcal{O}_2) \models C_{\#i} \sqsubseteq \bot$ **and** $\mathcal{O}_i \not\models C_{\#i} \sqsubseteq \bot$ **then**
    return *true*
  end if
end if
if $C_{\#i} \in \mathbf{P} \cup \mathbf{R}$ **then**
  if $\mathcal{A}_{\mathcal{S}}(\mathcal{O}_1, \mathcal{O}_2) \models \exists C_{\#i}.\top \sqsubseteq \bot$ **and** $\mathcal{O}_i \not\models \exists C_{\#i}.\top \sqsubseteq \bot$ **then**
    return *true*
  end if
end if
return *false*

The method $\textsc{IsCoherent}_{\mathcal{S}}$ is only mentioned for the sake of convenience. We introduced it to increase the readability of the complex algorithms presented later on. It is implemented on top of the method $\textsc{GetSomeAlignedUSatEntity}_{\mathcal{S}}$. If the latter returns an unsatisfiable entity, we know that the alignment under discussion is incoherent. The algorithm $\textsc{GetAlignedUSatEntities}_{\mathcal{S}}$ for computing the set of unsatisfiable entities is a very simple approach that iterates over all named concepts and properties each time checking whether the entity is unsatisfiable in the aligned ontology. We have not depicted the pseudocode for $\textsc{GetSomeAlignedUSatEntity}_{\mathcal{S}}$ as well, because it works similar, however, it stops the iteration when the first unsatisfiable entity has been found and returns $\textsc{Nil}$ otherwise. Internally both algorithms use $\textsc{IsUSatEntity}_{\mathcal{S}}$, which is shown in the pseudocode of Algorithm 1. Again, we used a straightforward implementation on top of Definition 20. Notice that we use Proposition 1 for checking the unsatisfiability of a property via checking the unsatisfiability of its domain.

Method $\textsc{MUPSWalk}_{\mathcal{S}}$ (Algorithm+2) is a simplified variant of the *'expand-and-shrink-algorithm'* proposed in [Kal06] for debugging incoherent alignments. The basic idea of the algorithm is the following one: Given an incoherent set of axioms $\mathcal{A}$, start with an empty set of axioms and add step by step subsets of $\mathcal{A}$ until an incoherent set of axioms has been constructed. Afterwards remove step by step single axioms until the set becomes again coherent. This procedure constructs a MIPS. Alternatively, it can be used to construct a MUPS by taking unsatisfiability of a specific concept into account instead of incoherence.

Our variant of the algorithm is only based on the *'shrink-part'* of the algorithm, while we omit the *'expand-part'*. Given an unsatisfiable concept or property $C_{\#i}$, we start with the complete set of correspondences and remove step by step a single

$\mathcal{SHOIN}(\mathbf{D})$ ontologies, alignment coherence cannot be guaranteed in general for $\mathcal{SHOIN}(\mathbf{D})$ ontologies due to the required preprocessing.

correspondence c. In each step we remove c from the MUPS $\mathcal{M}$ to be constructed. If the unsatisfiability of $C_{\#i}$ is resolved, we add c again to $\mathcal{M}$, because we know that $\mathcal{M}$ does not contain a MUPS for $C_{\#i}$ anymore. Otherwise we know that we correctly removed c because there is still one MUPS contained in the reduced $\mathcal{M}$. Due to this strategy $\mathcal{M}$ will finally be a MUPS for $C_{\#i}$.

Algorithm 2 Constructs a minimal unsatisfiability preserving subalignment.

$\text{MUPSWALK}_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2, C_{\#i})$

$\mathcal{M} = \mathcal{A}$
for all $c \in \mathcal{A}$ **do**
    $\mathcal{M} \leftarrow \mathcal{M} \setminus \{c\}$
    if $\text{ISUSATENTITY}_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2, C_{\#i})$ **then**
        $\mathcal{M} \leftarrow \mathcal{M} \cup \{c\}$
    end if
end for
return $\mathcal{M}$

We do not apply the original *'expand-and-shrink-algorithm'* proposed for ontology debugging for the following reason. This algorithm – if we would adapt it to incoherences in ontology alignments – starts with an empty set $\mathcal{M} = \emptyset$. The expansion phase is determined by a a heuristic that adds first those axioms – in our case correspondences – for which it is more probable that they are involved in the unsatisfiability. However, in the following section we will define a set of specialized algorithms that directly check combinations of these correspondences for incoherence. These combinations will in particular consist of those correspondences that we would add in the first expansion steps. We will later on explain, that in the overall context we never apply $\text{MUPSWALK}_{\mathcal{S}}$ on an alignment that has not already been checked by these specialized methods. The use of a stepwise expansion will thus be of very limited benefit.

In the following we discuss an example of an incoherent alignment and apply some of the algorithms presented above to it. This gives us on the one hand an insight in the complexity of the reasoning involved and deepens our understanding of the algorithms presented so far. Notice that our example is not artificially constructed, but is based on correspondences generated by concrete matching systems on a realistic matching task. The correspondences discussed in the following have been generated by the matching systems participating in the OAEI 2009 CONFERENCE track. We refer the reader to Section 7.1 for a detailed description of the

OAEI and the relevant dataset.[3]

$$\langle \mathit{rejectPaper}_{\#1}, \mathit{reviewerOfPaper}_{\#2}, \equiv \rangle \tag{5.1}$$

$$\langle \mathit{Acceptance}_{\#1}, \mathit{AcceptedPaper}_{\#2}, \equiv \rangle \tag{5.2}$$

$$\langle \mathit{ConferenceMember}_{\#1}, \mathit{ConferenceParticipant}_{\#2}, \equiv \rangle \tag{5.3}$$

$$\langle \mathit{Paper}_{\#1}, \mathit{Paper}_{\#2}, \equiv \rangle \tag{5.4}$$

$$\langle \mathit{Person}_{\#1}, \mathit{Person}_{\#2}, \equiv \rangle \tag{5.5}$$

If we apply GETALIGNEDUSATENTITIES$_{\mathcal{S}}$ on this alignment, it detects five unsatisfiable concepts. In particular, the algorithm returns as result $\{\mathit{Acceptance}_{\#1}$, $\mathit{AssignedPaper}_{\#2}$, $\mathit{EvaluatedPaper}_{\#2}$, $\mathit{AcceptedPaper}_{\#2}$, $\mathit{RejectedPaper}_{\#2}\}$. Notice first of all that some of the unsatisfiable concepts do not at all occur in the correspondences. The unsatisfiability of these five concepts is based on two MUPS alignments. They are depicted and explained, together with the relevant axioms of the ontologies, in Tables 5.1 and 5.2.

Alignment	
(1) $\langle \mathit{Acceptance}_{\#1}, \mathit{AcceptedPaper}_{\#2}, \equiv \rangle$	
(2) $\langle \mathit{Paper}_{\#1}, \mathit{Paper}_{\#2}, \equiv \rangle$	
Axioms of ontology #1	**Axioms of ontology #2**
(3) $\mathit{Paper} \sqsubseteq \mathit{Document}$	(6) $\mathit{AcceptedPaper} \sqsubseteq \mathit{EvaluatedPaper}$
(4) $\mathit{Acceptance} \sqsubseteq \mathit{Decision}$	(7) $\mathit{EvaluatedPaper} \sqsubseteq \mathit{AssignedPaper}$
(5) $\mathit{Document} \sqsubseteq \neg\mathit{Decision}$	(8) $\mathit{AssignedPaper} \sqsubseteq \mathit{SubmittedPaper}$
	(9) $\mathit{SubmittedPaper} \sqsubseteq \mathit{Paper}$
Entailments	
(10) $\mathit{Acceptance}_{\#1} \equiv \mathit{AcceptedPaper}_{\#2}$	*from (1)*
(11) $\mathit{Paper}_{\#1} \equiv \mathit{Paper}_{\#2}$	*from (2)*
(12) $\mathit{AcceptedPaper}_{\#2} \sqsubseteq \mathit{Paper}_{\#2}$	*from (6), (7), (8) and (9)*
(13) $\mathit{Acceptance}_{\#1} \sqsubseteq \mathit{Paper}_{\#2}$	*from (10) and (12)*
(14) $\mathit{Acceptance}_{\#1} \sqsubseteq \mathit{Paper}_{\#1}$	*from (11) and (13)*
(15) $\mathit{Acceptance}_{\#1} \sqsubseteq \mathit{Document}_{\#1}$	*from (3) and (14)*
(16) $\mathit{Acceptance}_{\#1} \sqsubseteq \neg\mathit{Decision}_{\#1}$	*from (5) and (15)*
(17) $\mathit{Acceptance}_{\#1} \sqsubseteq \bot$	*from (4) and (16)*

Table 5.1: Alignment between concepts that is incoherent due to a conflict between a chain of subsumption axioms and a disjointness axiom.

Our first example of a MUPS alignment (Table 5.1) consists of two correspondences, namely Correspondences 5.2 and 5.4 of the listing presented above. It is

[3]This example contains correspondences that have been detected with regard to the task of matching the CMT ontology on the EKAW ontology by diverse matching systems. Regarding this pair of ontologies, matching systems will typically generate a larger alignment. However, we discuss this limited set of correspondences, because otherwise the interdependencies between the correspondences are hard to compass.

based on a simple pattern of conflicting subsumption and disjointness statements. In the following we use the numbering of correspondences and axioms as used in Table 5.1. Correspondence (1) maps the event or decision of accepting a paper on those papers that have been accepted. This flaw results, taking (2) into account, in the unsatisfiability of $Acceptance_{\#1}$ in the aligned ontology. The relevant line of reasoning is depicted in line (10) to (17). We finally have to conclude that $Acceptance_{\#1}$ is on the one hand a $Decision_{\#1}$ (as stated in $\mathcal{O}_1$) and on the other hand a $Document_{\#1}$, which can be derived from several axioms taking correspondences (1) and (2) into account. However, these concepts are disjoint. Thus, $Acceptance_{\#1}$ is unsatisfiable.

Alignment	
(1) $\langle rejectPaper_{\#1}, reviewerOfPaper_{\#2}, \equiv\rangle$	
Axioms of ontology #1	Axioms of ontology #2
(2) $\exists_{\geq 2} rejectPaper^{-1}.\top \sqsubseteq \bot$	(3) $reviewerOfPaper \equiv hasReviewer^{-1}$ (4) $AssignedPaper \sqsubseteq \exists_{\geq 3} hasReviewer.\top$
Entailments	
(5) $rejectPaper_{\#1} \equiv reviewerOfPaper_{\#2}$	*from (1)*
(6) $rejectPaper^{-1}_{\#1} \equiv hasReviewer_{\#2}$	*from (3) and (5)*
(7) $rejectPaper_{\#1} \equiv hasReviewer^{-1}_{\#2}$	*from (6)*
(8) $AssignedPaper_{\#2} \sqsubseteq \exists_{\geq 3} rejectPaper^{-1}_{\#1}.\top$	*from (4) and (7)*
(9) $AssignedPaper_{\#2} \sqsubseteq \exists_{\geq 2} rejectPaper^{-1}_{\#1}.\top$	*from (8)*
(10) $AssignedPaper_{\#2} \sqsubseteq \bot$	*from (2) and (9)*

Table 5.2: Alignment which introduces a conflict due to contradictory cardinality restrictions.

The second example, presented in Table 5.2, exceeds this simple pattern and is based on the use of cardinality restrictions. On the one hand an $AssignedPaper_{\#2}$ is defined as a paper that has at least three reviewers (4), on the other hand property $rejectPaper_{\#1}$ is defined to be inverse functional (2). These two axioms, together with Correspodence (1), form the core of the unsatisfiability of $AssignedPaper_{\#2}$. As a consequence, $EvaluatedPaper_{\#2}$, $AcceptedPaper_{\#2}$, $RejectedPaper_{\#2}$, which are subconcepts of $AssignedPaper_{\#2}$, are also unsatisfiable.

There are several interesting observations. First of all we notice that MUPS alignments can have different sizes. Of special interest is the MUPS depicted in Table 5.2, because it consists of only one correspondence. This correspondence is an element of each possible diagnosis. Regarding the datasets we used in our experiments, MUPS alignments with only one correspondence occur rarely. Opposed to this, it will turn out that most MUPS alignments consist of a pair of correspondences, as presented in Table 5.1. However, there are also MUPS alignments of higher cardinality. An interesting example with four correspondences is presented

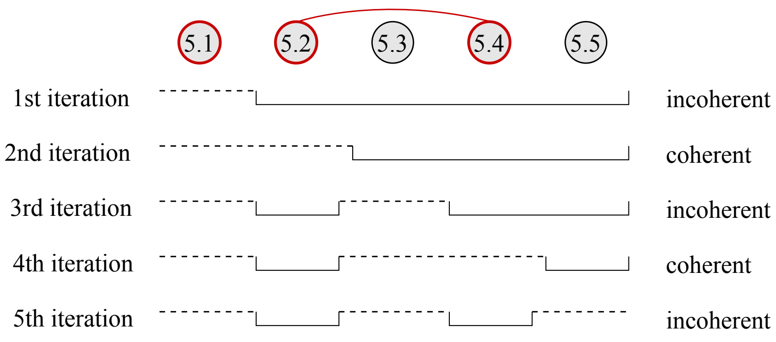

Figure 5.1: Example for applying the algorithm MUPSWALK$_{\mathcal{S}}$ on an alignment with two MUPS alignments. Correspondences 5.1 and 5.5 are described above, $\{5.1\}$ and $\{5.2, 5.4\}$ are MUPS alignments.

in Table A.1 in the appendix.

Figure 5.1 illustrates how MUPSWALK$_{\mathcal{S}}$ finds one of the MUPS alignments in $\mathcal{A}$. Prior to its application we have to find an unsatisfiable concept or property as input parameter. For this purpose we use GETALIGNEDUSATENTITIES$_{\mathcal{S}}$, which returns a set of unsatisfiable entities. We randomly pick one of its elements. According to the examples described above this might be the concept *Acceptance*$_{\#1}$. Now we can run MUPSWALK$_{\mathcal{S}}$ with *Acceptance*$_{\#1}$ as unsatisfiable entity. As indicated by the figure, we reduce $\mathcal{M} = \mathcal{A}$ step by step until we detect for the first time that *Acceptance*$_{\#1}$ becomes satisfiable. This is the case, for the first time, when we remove $\{5.2\}$. We add this correspondence again to $\mathcal{M}$ and continue in the same manner. Finally we detect that $\mathcal{M} = \{5.2, 5.4\}$ is a MUPS for *Acceptance*$_{\#1}$.

The algorithms presented in this section are independent of the chosen alignment semantics $\mathcal{S}$, in the sense that the reasoning is completely conducted in the aligned ontology $\mathcal{A}_{\mathcal{S}}(\mathcal{O}_1, \mathcal{O}_2)$. We can use any state of the art reasoner to decide subsumption and satisfiability tests related to $\mathcal{A}_{\mathcal{S}}(\mathcal{O}_1, \mathcal{O}_2)$. For that reason we added $\mathcal{S}$ as subscript to the methods names. In the following section we introduce a different type of reasoning methods that are specific to a certain semantics.

5.2 Efficient Reasoning Components

The techniques presented in the following differ with regard to several aspects from the previously introduced reasoning techniques. First of all, they are designed for a specific semantics, namely the natural reductionistic alignment semantics $\mathcal{S}_n$. We will, at the end of this section, briefly argue that it is possible to design similar algorithms for other alignment semantics. The overall approach developed in the subsequent sections is thus not restricted to $\mathcal{S}_n$, although we develop the required efficient reasoning components only for this specific semantics.

This difference is based on the fact that the pattern based reasoning approach described in the following does not require any reasoning activities in the aligned

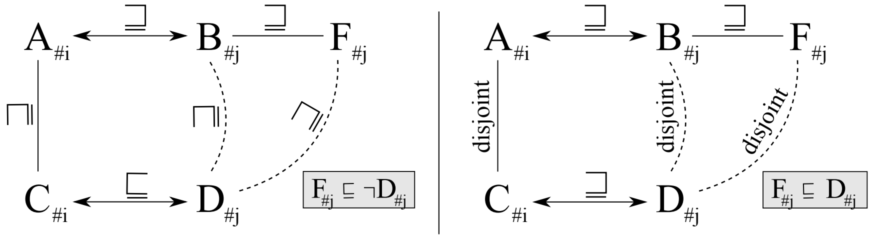

Figure 5.2: Subsumption (on the left) and disjointness propagation pattern (on the right). Arrows represent correspondences, solid or dashed lines represent axioms or entailed statements, respectively.

ontology. Remember that the reasoning methods in the last section were based on reasoning in the aligned ontology, that has to be created (and classified) for each alignment under discussion. Opposed to this, the methods presented in the following require to classify $\mathcal{O}_1$ and $\mathcal{O}_2$ once, while any subsequent *"reasoning task"* is restricted to subsumption tests, in which we ask for information already computed during a preprocessing step.

As a consequence, the required requests can be answered very efficiently, however, the results of these requests are incomplete, because complex and mutual effects between $\mathcal{O}_1$ and $\mathcal{O}_2$ caused by $\mathcal{A}$ cannot be taken into account. Incompleteness regarding incoherence detection refers in this context to the characteristic that an alignment might contain some MUPS that cannot be detected by the efficient reasoning method. Nevertheless, the methods presented in the following are sound in the sense that a set of correspondences – in our case a pair of correspondences – will always be incoherent if the method comes to this decision.

The proposed method is based on the detection of problematic patterns, more precisely, combinations of correspondences and entailed statements that result in the unsatisfiability of a concept or property. In particular, we introduce two patterns, depicted in Figure 5.2. We refer to these patterns as <u>subsumption propagation pattern</u> (on the left) and <u>disjointness propagation pattern</u> (on the right).

First, we focus on the subsumption propagation pattern. Suppose that $\mathcal{A}$ contains correspondences $\langle A_{\#i}, B_{\#j}, \sqsupseteq\rangle$ and $\langle C_{\#i}, D_{\#j}, \sqsubseteq\rangle$ as elements, represented as arrows in Figure 5.2. Further suppose that $\mathcal{O}_i \models A_{\#i} \sqsubseteq C_{\#i}$. Due to the natural semantics $\mathcal{S}_n$ we conclude that in the aligned ontology we have $\mathcal{A}_{\mathcal{S}_n}(\mathcal{O}_i, \mathcal{O}_j) \models B_{\#j} \sqsubseteq D_{\#j}$ and also $\mathcal{A}_{\mathcal{S}_n}(\mathcal{O}_i, \mathcal{O}_j) \models F_{\#j} \sqsubseteq D_{\#j}$ for each subconcept $F_{\#j}$ of $B_{\#j}$ (both indicated by dashed lines). We draw this conclusion about the aligned ontology without reasoning in the aligned ontology. Once we classified both $\mathcal{O}_1$ and $\mathcal{O}_2$, the pattern can be checked by a sequence of lookups in a very efficient way. We do not need to load and classify the aligned ontology.

Given now that $\mathcal{O}_j \models F_{\#j} \sqsubseteq \neg D_{\#j}$ and thus $\mathcal{A}_{\mathcal{S}_n}(\mathcal{O}_i, \mathcal{O}_j) \models F_{\#j} \sqsubseteq \neg D_{\#j}$, it follows that $\mathcal{A}_{\mathcal{S}_n}(\mathcal{O}_i, \mathcal{O}_j) \models F_{\#j} \sqsubseteq \bot$. Thus, $\{\langle A_{\#i}, B_{\#j}, \sqsupseteq\rangle, \langle C_{\#i}, D_{\#j}, \sqsubseteq\rangle\}$ is an incoherent alignment and contains for that reason a MIPS alignment. Notice that in most cases this dual-element set is already a MIPS alignment, since MIPS that consist of only one correspondence occur very rarely.

The pattern is described for the most general case where we have two subsumption correspondences $\langle A_{\#i}, B_{\#j}, \sqsupseteq \rangle$ and $\langle C_{\#i}, D_{\#j}, \sqsubseteq \rangle$, However, it holds also if both correspondences express equivalence, because $\langle A_{\#i}, B_{\#j}, \sqsupseteq \rangle$ is $\mathcal{S}$-entailed by $\{\langle A_{\#i}, B_{\#j}, \equiv \rangle\}$ with respect to $\mathcal{O}_i$ and $\mathcal{O}_j$. The same holds for the disjointness propagation pattern. It works similar as the subsumption propagation pattern. The major difference is based on propagating disjointness from $\mathcal{O}_i$ to $\mathcal{O}_j$ instead of subsumption. This disjointness might then conflict with a subsumption in $\mathcal{O}_j$. Thus, it is inverse to the subsumption propagation pattern. We abstain from a detailed description and refer the reader to Figure 5.2.

Alignment	
(1) $\langle \mathit{writtenBy}_{\#1}, \mathit{writtenBy}_{\#2}, \equiv \rangle$	
(2) $\langle \mathit{Review}_{\#1}, \mathit{Review}_{\#2}, \equiv \rangle$	
Axioms of ontology #1	Axioms of ontology #2
(3) $\exists \mathit{writtenBy}.\top \sqsubseteq \mathit{Review}$	(4) $\mathit{CameraReadyPaper} \sqsubseteq \exists \mathit{writtenBy}.\mathit{Participant}$
	(5) $\mathit{CameraReadyPaper} \sqsubseteq \mathit{Paper}$
	(6) $\mathit{Paper} \sqsubseteq \neg \mathit{Review}$
Entailments	
(7) $\mathit{writtenBy}_{\#1} \equiv \mathit{writtenBy}_{\#2}$	*from (1)*
(8) $\mathit{Review}_{\#1} \equiv \mathit{Review}_{\#2}$	*from (2)*
(9) $\exists \mathit{writtenBy}_{\#2}.\top \sqsubseteq \mathit{Review}_{\#1}$	*from (3) and (7)*
(10) $\exists \mathit{writtenBy}_{\#2}.\top \sqsubseteq \mathit{Review}_{\#2}$	*from (2) and (9)*
(11) $\exists \mathit{writtenBy}_{\#2}.\mathit{Participant}_{\#2} \sqsubseteq \mathit{Review}_{\#2}$	*from (10)*
(12) $\mathit{CameraReadyPaper}_{\#2} \sqsubseteq \mathit{Review}_{\#2}$	*from (4) and (11)*
(13) $\mathit{CameraReadyPaper}_{\#2} \sqsubseteq \neg \mathit{Review}_{\#2}$	*from (5) and (6)*
(14) $\mathit{CameraReadyPaper}_{\#2} \sqsubseteq \bot$	*from (12) and (13)*

Table 5.3: Alignment causing the unsatisfiability of a concept due to the interaction of domain restriction, disjointness and subsumption axioms.

With a minor extension it is possible to apply these patterns also to correspondences between properties. An example is presented in Table 5.3. It describes how a property correspondence $\langle \mathit{writtenBy}_{\#1}, \mathit{writtenBy}_{\#2}, \equiv \rangle$ and a concept correspondence $\langle \mathit{Review}_{\#1}, \mathit{Review}_{\#2}, \equiv \rangle$ result in the unsatisfiability of the concept $\mathit{CameraReadyPaper}_{\#1}$. Note that the reasoning involves several axioms of different types. However, a slight extension of the pattern based method proposed so far allows to detect that this pair of correspondences is incoherent. This extension is depicted in Figure 5.3 in general and at hand of the presented example. The pattern exploits that $\langle \mathit{writtenBy}_{\#1}, \mathit{writtenBy}_{\#2}, \sqsubseteq \rangle$ entails $\mathcal{A}_{\mathcal{S}_n}(\mathcal{O}_1, \mathcal{O}_2) \models \exists \mathit{writtenBy}_{\#1}.\top \sqsubseteq \exists \mathit{writtenBy}_{\#2}.\top$. The concept descriptions $\exists \mathit{writtenBy}_{\#1}.\top$ and $\exists \mathit{writtenBy}_{\#2}.\top$ can then be treated in the same way as the atomic concepts in Figure 5.2. Thus, it is possible to check relevant relations to other concepts. The

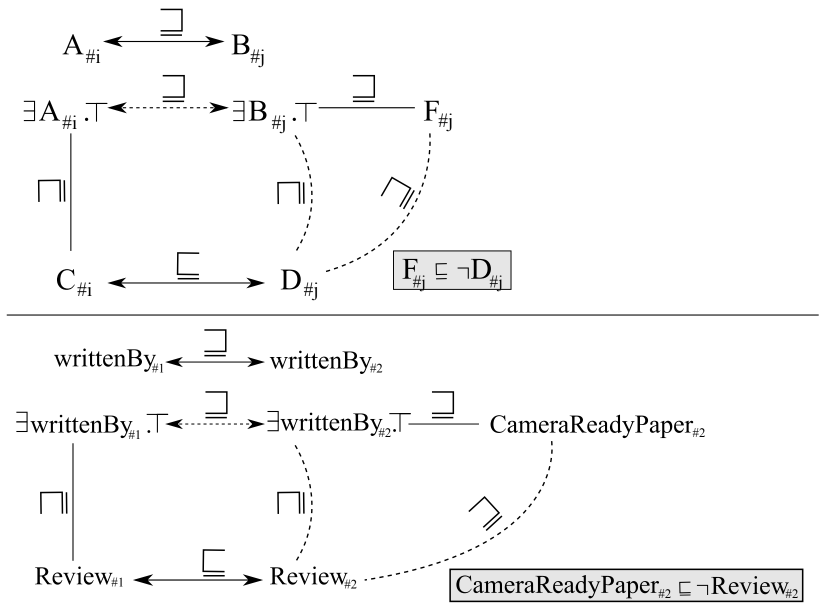

Figure 5.3: Subsumption propagation pattern with property correspondence involved. General and exemplified description.

presented example depicts the case where we focus on the domain of a property. The same approach can also be applied to the range $\exists P^{-1}.\top$ of a property *P*.

All relevant combination are formally listed in Algorithm 3. Notice that the algorithm contains a recursive element in line 13 and 16. However, it is only a concise description of a case distinction required due to the fact that correspondences are reduced to axioms between their domain or range. Algorithm 4 contains the pseudo code for detecting the disjointness propagation pattern. It works similar as Algorithm 3.

These two algorithms are the core of our pattern based method. We have designed them to be applicable for pairs of concept correspondences, pairs of mixed correspondences (concept correspondence & property correspondence), and pairs of property correspondences. However, another algorithm is required that applies these algorithms in all possible permutations to check whether – according to one of these patterns – a pair of correspondences is incoherent. We have, for example, to ensure that we search for the occurrence of the pattern in both directions regarding the order of ontologies, i.e., to call both $\textsc{SubPropConflict}(\ldots, \mathcal{O}_1, \mathcal{O}_2)$ and $\textsc{SubPropConflict}(\ldots, \mathcal{O}_2, \mathcal{O}_1)$. In addition we have to check patterns for $\langle A, B, \sqsubseteq\rangle$ and for $\langle A, B, \sqsupseteq\rangle$ when we analyze an equivalence correspondences $\langle A, B, \equiv\rangle$.

For the sake of completeness we have described the corresponding procedure in Algorithm 5. This algorithm – referred to as $\textsc{IncoherentPair}$ – returns *true* or *unknown* as result, but never the value *false*. This reflects that it is a sound but incomplete method for detecting the incoherence of a pair of correspondences. As

Algorithm 3 Detects the subsumption propagation pattern.

SUBPROPCONFLICT$(A_{\#i}, B_{\#j}, C_{\#i}, D_{\#j}, \mathcal{O}_i, \mathcal{O}_j)$

$\langle \mathbf{C}, \mathbf{P}, \mathbf{R}, \mathbf{I} \rangle \leftarrow$ *signature of* $\mathcal{O}_i \cap \mathcal{O}_j$
if $A_{\#i}, B_{\#j}, C_{\#i}, D_{\#j} \in \mathbf{C}$ **then**
    if $\mathcal{O}_i \models A_{\#i} \sqsubseteq C_{\#i}$ **then**
        for all $F_{\#j} \in \mathbf{C}$ with $\mathcal{O}_j \models F_{\#j} \sqsubseteq B_{\#j}$ **do**
            if $\mathcal{O}_j \models F_{\#j} \sqsubseteq \neg D_{\#j}$ **and** $\mathcal{O}_j \not\models F_{\#j} \sqsubseteq \bot$ **then**
                return true
            end if
        end for
    end if
    return false
else
    if $A_{\#i}, B_{\#j} \in \mathbf{P}$ **then**
        return SUBPROPCONFLICT$(\exists A_{\#i}.\top, \exists B_{\#j}.\top, C_{\#i}, D_{\#j}, \mathcal{O}_i, \mathcal{O}_j) \vee$ SUBPROPCONFLICT$(\exists A^{-1}_{\#i}.\top, \exists B^{-1}_{\#j}.\top, C_{\#i}, D_{\#j}, \mathcal{O}_i, \mathcal{O}_j)$
    end if
    if $C_{\#i}, D_{\#j} \in \mathbf{P}$ **then**
        return SUBPROPCONFLICT$(A_{\#i}, B_{\#j}, \exists C_{\#i}.\top, \exists D_{\#j}.\top, \mathcal{O}_i, \mathcal{O}_j) \vee$ SUBPROPCONFLICT$(A_{\#i}, B_{\#j}, \exists C^{-1}_{\#i}.\top, \exists D^{-1}_{\#j}.\top, \mathcal{O}_i, \mathcal{O}_j)$
    end if
end if

Algorithm 4 Detects the disjointness propagation pattern.

DISPROPCONFLICT$(A_{\#i}, B_{\#j}, C_{\#i}, D_{\#j}, \mathcal{O}_i, \mathcal{O}_j)$

$\langle \mathbf{C}, \mathbf{P}, \mathbf{R}, \mathbf{I} \rangle \leftarrow$ *signature of* $\mathcal{O}_i \cap \mathcal{O}_j$
if $A_{\#i}, B_{\#j}, C_{\#i}, D_{\#j} \in \mathbf{C}$ **then**
    if $\mathcal{O}_i \models A_{\#i} \sqsubseteq \neg C_{\#i}$ **then**
        for all atomic concepts $F_{\#j}$ with $\mathcal{O}_j \models F_{\#j} \sqsubseteq B_{\#j}$ **do**
            if $\mathcal{O}_j \models F_{\#j} \sqsubseteq D_{\#j}$ **and** $\mathcal{O}_j \not\models F_{\#j} \sqsubseteq \bot$ **then**
                return true
            end if
        end for
    end if
    return false
else
    $\triangleright$ *analog to Algorithm 3*
end if

Algorithm 5 Decides whether a pair of correspondences exhibits an incoherence due to a subsumption or disjointness propagation.

INCOHERENTPAIR$(\langle A_{\#i}, B_{\#j}, r\rangle, \langle C_{\#i}, D_{\#j}, r'\rangle, \mathcal{O}_i, \mathcal{O}_j)$

if $r = \sqsubseteq$ **or** $r = \equiv$ **then**
  if $r' = \sqsubseteq$ **or** $r' = \equiv$ **then**
    if DISPROPCONFLICT$(B_{\#j}, A_{\#i}, D_{\#j}, C_{\#i}, \mathcal{O}_j, \mathcal{O}_i)$ **then**
      return *true*
    end if
  end if
  if $r' = \sqsupseteq$ **or** $r' = \equiv$ **then**
    if SUBPROPCONFLICT$(C_{\#i}, D_{\#j}, A_{\#i}, B_{\#j}, \mathcal{O}_i, \mathcal{O}_j)$ **then**
      return *true*
    else if SUBPROPCONFLICT$(B_{\#j}, A_{\#i}, D_{\#j}, C_{\#i}, \mathcal{O}_j, \mathcal{O}_i)$ **then**
      return *true*
    end if
  end if
end if
if $r = \sqsupseteq$ **or** $r = \equiv$ **then**
  if $r' = \sqsubseteq$ **or** $r' = \equiv$ **then**
    if SUBPROPCONFLICT$(A_{\#i}, B_{\#j}, C_{\#i}, D_{\#j}, \mathcal{O}_i, \mathcal{O}_j)$ **then**
      return *true*
    else if SUBPROPCONFLICT$(D_{\#j}, C_{\#i}, B_{\#j}, A_{\#i}, \mathcal{O}_j, \mathcal{O}_i)$ **then**
      return *true*
    end if
  end if
  if $r' = \sqsupseteq$ **or** $r' = \equiv$ **then**
    if DISPROPAGATION$(A_{\#i}, B_{\#j}, C_{\#i}, D_{\#j}, \mathcal{O}_i, \mathcal{O}_j)$ **then**
      return *true*
    end if
  end if
end if
return *unknown*

we will see later, it can be used for different purposes. We can for example iterate over all pairs of correspondences in an alignment to decide whether the complete alignment is incoherent or possibly coherent.

We have already argued that similar algorithms can be designed for different reductionistic alignment semantics. A self-evident example can be found in the semantics of DDL. It can be shown that the subsumption propagation pattern restricted to concept correspondences is a simplified variant of the general DDL propagation rule. For that reason any conflict detected by Algorithm 3 does not only results in an unsatisfiable concept in the natural semantics, but result also in a distributed unsatisfiable concept in DDL. Contrary to this, the disjointness propagation pattern is not valid in DDL. Due to the semantics of DDL a disjointness axiom or entailment cannot be propagated from one ontology to another. We omit the details of this consideration, but point to the fact that an arbitrary alignment semantics can be expected to exhibit patterns resulting in the incoherence of a small set – in our case a pair – of correspondences. The overall approach is thus applicable in general, although we developed it in detail only for the natural alignment semantics $\mathcal{S}_n$.

5.3 Summary

This chapter was not directly concerned with one of the research questions raised at the beginning of the thesis. Instead of that, we presented some reasoning procedures that will be used in the next chapter. There we design algorithms for computing local and global optimal diagnoses. The current chapter was thus concerned with preparatory work required to answer research question R4 in the following chapter.

In Section 5.1 we presented some basic algorithms, which have been adapted from the field of ontology debugging. These algorithms are straight-forward techniques, which use a reasoner as a black box. We use these algorithms to determine, for example, a MUPS alignment by a sequence of reasoning requests. Thus approach suffers from two problems. (1) We perform reasoning in the aligned ontology, and this can be much more expensive than reasoning in each of the aligned ontologies. (2) The reasoner has to start from the scratch for each reasoning request. Both aspects result in an inefficient runtime behaviour.

A major contribution of this thesis is based on the idea to reduce reasoning in the aligned ontology as much as possible. For that reason we introduced a reasoning approach that checks certain patterns that result in incoherence. We also explained that the patterns we proposed are specific to the natural semantics $\mathcal{S}_n$. However, we also argued that it is easy to define similar patterns for other semantics. In the section on future work we discuss alternative solutions to these problems mentioned above.

The pattern based approach is not complete, i.e., there are minimal incoherent sets of correspondences that are not instances of one of the defined patterns. For that reason we cannot completely avoid to reason in the aligned ontology. In the next chapter we will show how to reduce reasoning in the aligned ontology to a minimum. The contribution of the current chapter will thus become apparent to its full degree in the light of the following chapter.

Chapter 6

Computing a Diagnosis

The machine does not isolate man from the great problems [...] but plunges him more deeply into them (Antoine de Saint-Exupéry).

In the following we present the algorithms for computing local and global optimal diagnoses. Due to the different characteristics of these types of diagnosis, the resulting algorithms differ to a large degree. For computing a local optimal diagnosis it is sufficient to design an algorithm that more or less reflects the definition of a local optimal diagnosis in a constructive way. The core element of the algorithm for computing a global optimal diagnosis is a uniform cost search that will finally be extended to an A*-search.

The challenge in designing these algorithms is to reduce the amount of reasoning in the aligned ontology to a minimum. For that purpose we have introduced the efficient reasoning techniques described in the last chapter. However, the resulting diagnosis Δ has to be complete in the sense that $\mathcal{A} \setminus \Delta$ is a coherent alignment. We have seen that a diagnosis Δ can be characterized in terms of a hitting set over all MIPS or MUPS of $\mathcal{A}$. In the following we show, as one of our main contributions, that Δ can be computed without knowing the complete set of all MIPS and MUPS in $\mathcal{A}$.

In Section 6.1 we present two variants of an algorithm for computing a local optimal diagnosis. The first version uses none of the efficient reasoning techniques presented above. All reasoning activities take place in the aligned ontology. We show how to improve this algorithm to use additionally our pattern based reasoning components to speed up the process of computing the local optimal diagnosis.

In Section 6.2 we present a uniform cost search search for finding a global optimal diagnosis. We extend this algorithms to an A*-search that makes use of both complete and efficient reasoning components. Both algorithms start with the complete alignment $\mathcal{A}$ as root of the search tree. A node expansion requires to detect a MUPS $\mathcal{M}$ in the current search state. The resulting states are subsets of the current state where one of the elements of $\mathcal{M}$ has been removed. The final version of the algorithm is based on the use of efficient methods to determine a subset of all MUPS in a preprocessing step as well as to choose the next search state to be expanded.

In Section 6.3 draw conclusions related to the runtimes of the algorithms. In particular, we explain how the algorithms behave in specific settings and discuss how the expected runtime is affected by the fraction of MUPS detectable by our pattern based reasoning components. Finally, we end this chapter with a conclusion in Section 6.4.

The first part of this chapter (Section 6.1) is to a large degree based on work we already published in [MS09b] and presented in detail in an extended technical report [MS09a]. The second part of this chapter (Section 6.2) has not yet been published. To our knowledge it is a novel approach. We refer the reader to the section on related work for a comparison with existing approaches. We developed a predecessor of the algorithm already in [MS07b]. However, the algorithm described there was incomplete in the sense that its results cannot be guaranteed to be coherent. The considerations regarding the runtime of the algorithms, presented in Section 6.3, have partially been published in our technical report [MS09a].

6.1 Computing a Local Optimal Diagnosis

In the following we need to enumerate the correspondences of an alignment to access elements or subsets of the alignment by index or range. Thus, we sometimes treat an alignment $\mathcal{A}$ as an array using a notation $\mathcal{A}[i]$ to refer to the i-th element of $\mathcal{A}$ and $\mathcal{A}[j \dots k]$ to refer to $\{\mathcal{A}[i] \in \mathcal{A} \mid j \leq i \leq k\}$. Further, let the index of an alignment start at 1 such that the last element in an alignment can be accessed via the index $|\mathcal{A}|$. For the sake of convenience we use $\mathcal{A}[\dots k]$ to refer to $\mathcal{A}[1 \dots k]$, similar we use $\mathcal{A}[j \dots]$ to refer to $\mathcal{A}[j \dots |\mathcal{A}|]$.

Algorithm 6 Computes a local optimal diagnosis in a brute-force approach.

BRUTEFORCELOD$_\mathcal{S}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$

if ISCOHERENT$_\mathcal{S}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ **then**
  return $\emptyset$
else
  $\triangleright$ *sort $\mathcal{A}$ descending according to confidence values*
  $\Delta \leftarrow \emptyset$
  for $i \leftarrow 1$ to $|\mathcal{A}|$ **do**
    if not ISCOHERENT$_\mathcal{S}(\mathcal{A}[\dots i] \setminus \Delta, \mathcal{O}_1, \mathcal{O}_2)$ **then**
      $\Delta \leftarrow \Delta \cup \{\mathcal{A}[i]\}$
    end if
  end for
  return Δ
end if

We already argued that the set of accused correspondences forms a special kind of diagnosis referred to as local optimal diagnosis. In Algorithm 6 we describe an iterative procedure that computes such a diagnosis in a straight forward way. First, we check the coherence of $\mathcal{A}$ and return $\emptyset$ as diagnosis for a coherent alignment. Since the empty set $\emptyset$ is the only diagnosis for a coherent alignment, $\emptyset$ is also a local optimal diagnosis. Given that $\mathcal{A}$ is incoherent, we order $\mathcal{A}$ according to its confidence values. Then we start with an empty diagnosis Δ and extend it step by step by iterating over subsets $\mathcal{A}[\ldots i]$ of increasing size. Whenever $\mathcal{A}[\ldots i]$ becomes incoherent, decided by reasoning in the aligned ontology, $\mathcal{A}[i]$ is added to Δ. Finally, we end up with a local optimal diagnosis Δ.

Algorithms 6 is a very simple procedure. Although it is quite easy to see that its return value Δ will always be a diagnosis or a superset of a diagnosis, it is not that easy to prove that it always results in a local optimal diagnosis as stated in Proposition 9.

Proposition 9. $\textsc{BruteForceLOD}_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ *is a local optimal diagnosis for* $\mathcal{A}$ *with respect to* $\mathcal{O}_1$ *and* $\mathcal{O}_2$.

Proof. During the execution of the algorithm, $\mathcal{A}$ can be partitioned into the disjoint union of two alignments. This is the set of correspondences $\mathcal{A}[\ldots i]$ already processed after the i-th iteration took place and the set of remaining correspondences $\mathcal{A}[i+1\ldots]$. Furthermore, we need to refer to the state of Δ depending on the current iteration i. We use the notation Δ_i for that purpose. The transition from iteration i to iteration $i+1$ can now be expressed as follows.

$$\Delta_{i+1} = \begin{cases} \Delta_i & \text{if } \mathcal{A}[\ldots i+1] \setminus \Delta_i \text{ is coherent} \\ \Delta_i \cup \{\mathcal{A}[i+1]\} & \text{if } \mathcal{A}[\ldots i+1] \setminus \Delta_i \text{ is incoherent} \end{cases}$$

It is sufficient to prove the statement that for the ith-iteration Δ_i is a local optimal diagnosis (LOD) for $\mathcal{A}[\ldots i]$. We give a proof by induction.

Base Case: In the first iteration we have $\mathcal{A}[\ldots 1] = \mathcal{A}[1]$ and $\Delta_1 = \emptyset$ if $\{\mathcal{A}[1]\}$ is coherent or $\Delta_1 = \{\mathcal{A}[1]\}$ if $\{\mathcal{A}[1]\}$ is incoherent. It is quite clear that Δ is in both cases the only available diagnosis and thus also a local optimal diagnosis. Thus, the statement holds for $i = 1$.

Inductive Step: Suppose that for some iteration i alignment Δ_i is a LOD for $\mathcal{A}[\ldots i]$. We now have to show that Δ_{i+1} is a LOD for $\mathcal{A}[\ldots i+1]$. Due to the fact that $argmin_{c\in\mathcal{A}[\ldots i]}\alpha(c) > \alpha(\mathcal{A}[i+1])$ we can use Proposition 5 to derive that each $c \in \mathcal{A}[\ldots i]$ is accused by $\mathcal{A}[\ldots i]$ iff c is also accused by $\mathcal{A}[\ldots i+1]$. Given the inductive hypotheses, it follows that $c \in \mathcal{A}[\ldots i]$ is accused by $\mathcal{A}[\ldots i+1]$ iff $c \in \Delta_i$. It remains to be shown that $\mathcal{A}[i+1]$ is accused by $\mathcal{A}[\ldots i+1]$ iff $\mathcal{A}[\ldots i+1] \setminus \Delta_i$ is incoherent. First suppose that $\mathcal{A}[\ldots i+1] \setminus \Delta_i$ is incoherent. Due to our inductive hypotheses we know that $\mathcal{A}[\ldots i] \setminus \Delta_i$ is coherent. Thus, there exists a MIPS $\mathcal{M} \subseteq \mathcal{A}[\ldots i+1] \setminus \Delta_i$ with $\mathcal{A}[i+1] = argmin_{c\in\mathcal{M}}\,\alpha(c)$. Hence, $\mathcal{A}[i+1]$ is accused by $\mathcal{A}[\ldots i+1] \setminus \Delta_i$ and thus $\mathcal{A}[i+1] \in \Delta_{i+1}$. Now suppose that $\mathcal{A}[\ldots i+1] \setminus \Delta_i$ is a coherent alignment. Thus, there exists no such MIPS

$\mathcal{M}$. Hence, $\mathcal{A}[i+1]$ is not accused by $\mathcal{A}[\ldots i+1] \setminus \Delta_i$ and therefore we have $\mathcal{A}[i+1] \notin \Delta_{i+1}$. □

Algorithm 6 does not exploit efficient reasoning techniques. It requires to call IsCoherent$_\mathcal{S}$ $|\mathcal{A}|$-times. In the following we show how to construct a more efficient algorithm that uses the efficient reasoning techniques introduced above.

Algorithm 7 Finds the border between the coherent and the incoherent part of $\mathcal{A}$

FindCoherenceCrack$_\mathcal{S}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$

Require: input alignment $\mathcal{A}$ ordered descending due to its confidences α

```
if IsCoherent_S(A', O1, O2) then
    return NIL
end if
i ← 0
j ← |A|
loop
    if j − i = 1 then
        return j
    end if
    k ← ⌊(i + j)/2⌋
    if IsCoherent_S(A[...k], O1, O2) then
        i ← k
    else
        j ← k
    end if
end loop
```

For that purpose we first need to design a method to solve the following problem: Given an incoherent alignment $\mathcal{A}$ with correspondences ordered descending according to their confidence values α. We want to find an index k such that $\mathcal{A}[\ldots k-1]$ is coherent and $\mathcal{A}[\ldots k]$ is incoherent. It is self-evident to solve the problem with a binary search. The resulting search algorithm requires $log(|\mathcal{A}|)$ calls to IsCoherent$_\mathcal{S}$ until k is found. It splits the alignment $\mathcal{A}$ in two halves of equal size and checks the coherence of $\mathcal{A}[\ldots k]$. In case $\mathcal{A}[\ldots k]$ is incoherent, it applies the same procedure with a split index that divides the alignment into the first $3/4$ part and the last $1/4$ part of the alignment. Otherwise k is set to a value that splits the alignment into the first $1/4$ part and the last $3/4$ part. The procedure – referred to as FindCoherenceCrack$_\mathcal{S}$ in Algorithm 7 – continues like this until the index is uniquely determined. Note that the algorithm returns Nil in case of an alignment that is already coherent.

An example, showing the algorithm in action, is presented in Figure 6.1. In this example the algorithm is applied on an alignment that consists of eight correspondences with two MIPS alignments. We depicted the MIPS alignments to enable a better understanding, however, knowledge about MIPS alignments is not required

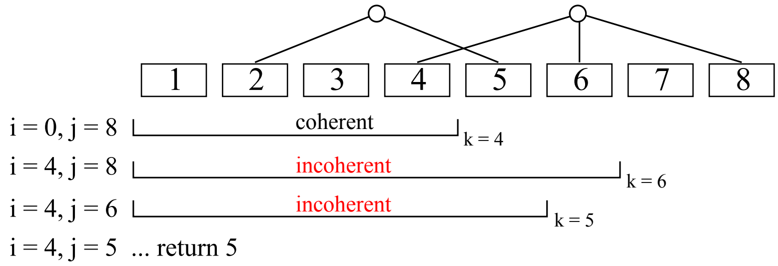

Figure 6.1: Applying the Algorithm FINDCOHERENCECRACK$_{\mathcal{S}}$ on an alignment with two MIPS alignments.

for the algorithm. The procedure solely relies on a series of alignment coherence checks. Regarding the example it requires three calls to ISCOHERENT$_{\mathcal{S}}$ until the index is determined. It is important to understand that a binary search can only be used to detect the *'crack'* between the coherent subset of the alignment and the incoherent superset. It cannot be used to detect a MUPS or a MIPS in an alignment.

Algorithm 8 Computes a local optimal diagnosis efficiently.

EFFICIENTLOD$_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$

$\triangleright$ *sort $\mathcal{A}$ descending according to confidence values*
$\mathcal{A}' \leftarrow \mathcal{A}$
$k' \leftarrow 1$
loop
  for $i \leftarrow k'$ to $|\mathcal{A}'|$ **do**
    for $j \leftarrow 1$ to $i - 1$ **do**
      if INCOHERENTPAIR$(\mathcal{A}'[j], \mathcal{A}'[i], \mathcal{O}_1, \mathcal{O}_2)$ = **true then**
        $\mathcal{A}' \leftarrow \mathcal{A}' \setminus \{\mathcal{A}'[i]\}$
        $i \leftarrow i - 1$ $\triangleright$ *adjust i to continue with next element of $\mathcal{A}'$*
        break $\triangleright$ *exit inner for-loop*
      end if
    end for
  end for
  $k' \leftarrow$ FINDCOHERENCECRACK$_{\mathcal{S}}(\mathcal{A}', \mathcal{O}_1, \mathcal{O}_2)$
  if $k' =$ NIL **then**
    return $\Delta \leftarrow \mathcal{A} \setminus \mathcal{A}'$
  end if
  $\triangleright$ *let k be the counterpart of k' adjusted for $\mathcal{A}$ such that $\mathcal{A}[k] = \mathcal{A}'[k']$*
  $\mathcal{A}' \leftarrow \mathcal{A}'[\ldots k' - 1] \cup \mathcal{A}[k + 1 \ldots]$
end loop

All building blocks are now available to construct an efficient algorithm for computing a local optimal diagnosis (Algorithm 8). First we have to sort the input alignment $\mathcal{A}$, prepare a copy $\mathcal{A}'$ of $\mathcal{A}$, and init an index $k' = 1$. Correspondences are removed from $\mathcal{A}'$ until finally $\Delta \leftarrow \mathcal{A} \setminus \mathcal{A}'$ is returned as local optimal diagnosis. Variables k' and k work as separator between the part of $\mathcal{A}'$ that has already been processed successfully and the part of $\mathcal{A}'$ that has not yet been processed or has not been processed successfully. More precisely, it holds that $\mathcal{A}[\ldots k] \setminus \mathcal{A}'[\ldots k']$ is a local optimal diagnosis for $\mathcal{A}[\ldots k]$ where k is an index such that $\mathcal{A}'[k'] = \mathcal{A}[k]$ (two indices k' and k are required because $\mathcal{A}'$ is – compared to $\mathcal{A}$ – reduced during executing the algorithm).

Within the main loop we have two nested loops (line 5-13). This part of the algorithm is used to check whether one of $\mathcal{A}'[i]_{i \geq k'}$ conflicts with one of $\mathcal{A}'[j]_{j<i}$ according to the pattern based reasoning techniques described above. In case a conflict pattern has been detected, $\mathcal{A}'[i]$ is removed from $\mathcal{A}'$. This approach would directly result in a local optimal diagnosis if

- all MIPS would consist of two elements i.e., would be pairs of correspondences;
- all incoherent pairs of correspondences would be detectable with the help of INCOHERENTPAIR.

Obviously, these assumptions are not correct. Thus, we have to search for an index k' such that $\mathcal{A}[\ldots k] \setminus \mathcal{A}'[\ldots k']$ is a local optimal diagnosis for $\mathcal{A}[\ldots k]$. In other words: We have to find the index where the pattern based reasoning techniques missed the first MIPS. Index k' is determined by the binary search presented above. If such an index cannot be detected, we know that $\mathcal{A} \setminus \mathcal{A}'$ is a local optimal (line 14-16). Otherwise, the value of $\mathcal{A}'$ is readjusted to the union of $\mathcal{A}'[\ldots k' - 1]$, which can be understood as the validated part of $\mathcal{A}'$, and $\mathcal{A}[k+1\ldots]$, which is the remaining part of $\mathcal{A}$ to be processed in the next iteration. $\mathcal{A}'[k']$ is removed from $\mathcal{A}'$ and thus becomes a part of the diagnosis returned finally.

Figure 6.2 illustrates the computation of a local optimal diagnosis for a small example alignment $\mathcal{A}$. Correspondences of $\mathcal{A}$ are depicted as rectangles. Their order is determined by their confidence values. Rightmost we depicted the input alignment $\mathcal{A}$ together with its MIPS alignments. We have to distinguish between two types of MIPS in $\mathcal{A}$: those MIPS that are detectable by INCOHERENTPAIR (correspondences connected by solid lines) and those that require reasoning in the merged ontology (dashed lines). On the leftmost is is shown that the diagnosis (white rectangles) – finally generated by the algorithm – forms a hitting set over all MIPS. The two alignments depicted in the middle illustrate snapshots taken during executing the algorithm. They depict $\mathcal{A}'$ (grey rectangles) and the value of k and k' after line 18 of the first and second iteration of the main loop. Notice that $\mathcal{A}'$- and $\mathcal{A}$-indices are shown inside the rectangles representing the correspondences.

In the first iteration there are three correspondences removed from $\mathcal{A}'$ (white rectangles). Nevertheless, $\mathcal{A}'$ is still an incoherent alignment, in particular it holds that $\mathcal{A}'[\ldots k' - 1]$ is coherent and $\mathcal{A}'[\ldots k']$ is incoherent. For that reason, we accept all removal decisions up to $k' = 8$ and additionally remove $\mathcal{A}'[9]$. The remaining part $\mathcal{A}[k+1\ldots]$ is re-processed in the second iteration. The removal

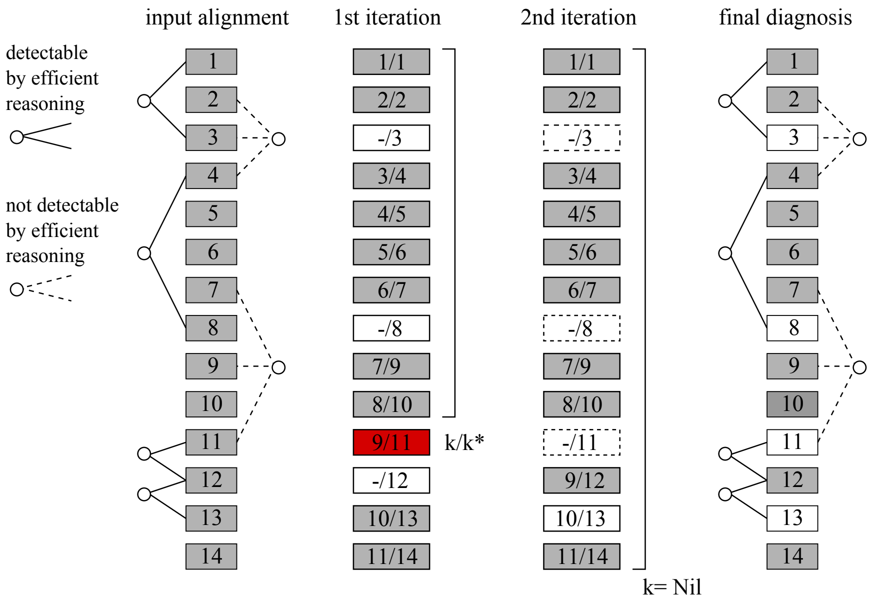

Figure 6.2: Applying the Algorithm EFFICIENTLOD$_{\mathcal{S}}$

decision made in the first iteration for this part of the alignment are withdrawn. Due to the information available by the means of complete reasoning, the accusal-relations have changed and in the second iteration $\mathcal{A}[13]$ (equivalent to $\mathcal{A}'[10]$) is temporarily removed. This time the resulting alignment $\mathcal{A}'$ is coherent. Because of that FINDCOHERENCECRACK$_{\mathcal{S}}$ returns NIL and $\mathcal{A} \setminus \mathcal{A}'$ is returned as local optimal diagnosis.

This example illustrates also an interesting effect pointing to an advantage of the algorithm. Although there exist two MIPS which are not detectable by our pattern based reasoning techniques, only one of these *'hard'* MIPS had to be detected by the binary search to construct a hitting set over all MIPS. The other hard MIPS has already been resolved due to an overlap with another MIPS. It is thus possible to construct a hitting set over all MIPS without knowing them a priori. This might not be a big surprise regarding the local optimal diagnosis. However, in the following we show that the same is possible for a global optimal diagnosis.

Before we continue we have to explicitly give a proof for the following proposition. Remember that we have already shown that BRUTEFORCELOD$_{\mathcal{S}}$ constructs a local optimal diagnosis.

Proposition 10. EFFICIENTLOD$_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ *is a local optimal diagnosis for* $\mathcal{A}$ *with respect to* $\mathcal{O}_1$ *and* $\mathcal{O}_2$.

Proof. Due to Proposition 9 it is sufficient to show that Algorithm 6 and Algorithm 8 have the same result Δ. Suppose now that Δ' is a local optimal diagnosis for a subset of all MIPS, namely those MIPS that are detected by our pattern based reasoning approach, while Δ is the local optimal diagnosis for the complete set of all MIPS. Δ' can be split in a correct part, where the efficient and the brute-force algorithm construct the same partial solution for a subset $\mathcal{A}[\ldots k-1]$, and an incorrect part. The correspondence $\mathcal{A}[k]$ where the correct part ends is exactly the correspondence that is detected by the binary search. Due to the stable ordering, the correct part can be extended over several iterations until we finally end up with a complete and correct local optimal diagnosis Δ. The correctness of this final conclusion is based on Proposition 5 $\square$

6.2 Computing a Global Optimal Diagnosis

In the following we show how to construct a global optimal diagnosis. First we introduce Algorithm 9, which is a uniform cost search that constructs the solution in a naive way. It starts with the complete alignment $\mathcal{A}$ as root of a search tree. The step from a node to one of its successors is equivalent to the removal of a correspondence. The candidates for a removal are determined by the outcome of a reasoning process. In the first version of the algorithm we use the complete reasoning methods proposed in Section 5.1. The improved version of the algorithm is based on two modifications. It makes additionally use of the efficient reasoning methods proposed in Section 5.2 and it applies a heuristic to transform the uniform cost search into an A*-Search [HNR68].[1]

The first version of our algorithm is depicted in the pseudocode of Algorithm 9. The algorithm implicitly creates a search tree. $\mathcal{A}$ is the root of the tree and each node in the tree is a subset of $\mathcal{A}$. To build up the search tree, the algorithm uses a priority queue Q to store these alignments. We use the notation $\text{ADD}(Q, \mathcal{A})$ to add an element to the queue and $\text{POP}(Q)$ to remove the top element from Q. Our priority queue is a minimum priority queue. The ordering of Q is defined by the sum of confidences of those correspondences that have been removed so far. $\text{POP}(Q)$ will thus always remove the alignment with the highest total of confidences from Q. A uniform cost search is an algorithm that takes into account the total costs caused by all previous steps. It always expands the node with the lowest total of all previously accumulated costs. In our case the costs of a step are $\alpha(c)$ where c is the correspondence removed in this step.

During the search process we use $\mathbb{M}$ to store all MUPS detected to far. This avoids that the same time-consuming reasoning process has to be performed several times. For the sake of simplicity $\mathbb{M}$ is referred to as a set of alignments in the pseudocode. In the concrete implementation it is a complex data structure that allows to check efficiently whether an alignment is contained in $\mathbb{M}$. More precisely, it support a function $\text{RETRIEVEMUPS}(\mathcal{A}, \mathbb{M})$ that efficiently retrieves a randomly chosen set $\mathcal{M}$ from $\mathbb{M}$ with $\mathcal{M} \subseteq \mathcal{A}$, in case such an alignment exists. Otherwise it returns NIL.

[1] We assume in the following that the reader is familiar with informed search algorithms. An overview can be found in Chapter 3 of Russell and Norvigs well known textbook on Artificial Intelligence [RN10].

Algorithm 9 Computes a global optimal diagnosis by a uniform cost search.

BRUTEFORCEGOD$_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$

1: $Q \leftarrow \langle\rangle_g$ $\triangleright$ *create an empty priority queue to store subsets of* $\mathcal{A}$
2: $\mathbb{M} \leftarrow \emptyset$ $\triangleright$ *create an empty set to store MUPS to be detected*
3: ADD$(Q, \mathcal{A})$
4: **loop**
5: $\quad \mathcal{A}_{top} \leftarrow$ POP(Q)
6: $\quad \mathcal{M} \leftarrow$ RETRIEVEMUPS$(\mathcal{A}_{top}, \mathbb{M})$
7: $\quad$ **if** $\mathcal{M} =$ NIL **then**
8: $\quad\quad U \leftarrow$ GETSOMEALIGNEDUSATENTITY$(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$
9: $\quad\quad$ **if** $U =$ NIL **then**
10: $\quad\quad\quad$ **return** $\Delta \leftarrow \mathcal{A} \setminus \mathcal{A}_{top}$
11: $\quad\quad$ **end if**
12: $\quad\quad \mathcal{M} \leftarrow$ MUPSWALK$(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2, U)$
13: $\quad\quad$ ADD$(\mathbb{M}, \mathcal{M})$
14: $\quad$ **end if**
15: $\quad$ **for** $c \in \mathcal{M}$ **do**
16: $\quad\quad$ ADD$(Q, \mathcal{A}_{top} \setminus \{c\})$
17: $\quad$ **end for**
18: **end loop**

After initializing these data structures, the input alignment $\mathcal{A}$ is added to Q. The algorithm enters the main loop. The best alignment $\mathcal{A}_{top}$ regarding its sum of confidences is popped from the queue. The algorithm checks whether there exists a previously computed MUPS $\mathcal{M} \in \mathbb{M}$ that is contained in $\mathcal{A}_{top}$. If this is not the case, the algorithm searches for an unsatisfiable entity U in the ontology aligned by $\mathcal{A}_{top}$. If such an entity cannot be found, then the alignment is coherent and $\Delta = \mathcal{A} \setminus \mathcal{A}_{top}$ is returned as global optimal diagnosis. If such an entity can be found, we apply the MUPSWALK algorithm to detect a MUPS $\mathcal{M}$ that explains the unsatisfiability of U. We store this MUPS in $\mathbb{M}$ to make this information available for further iterations. In case the algorithm reaches line 16, there will be a MUPS $\mathcal{M} \subseteq \mathcal{A}$ that (a) has been computed in a previous iteration or (b) has been computed in the current iteration. $\mathcal{M}$ is then used to create a branch in the search tree; $|\mathcal{M}|$ children of $\mathcal{A}_{top}$ are created and added to Q.

Figure 6.3 illustrates the algorithm applied to an alignment $\mathcal{A}$ with six correspondences a, b, c, d, e, f. MIPS of the alignment are depicted on the left side of the illustration. The algorithm expands the nodes in the order indicated by the number in parentheses. As final solution, $\Delta = \{c, f\}$ is returned. Note that a hitting set over all MIPS is constructed without the need for computing all MIPS in $\mathcal{A}$; $\{d, e, f\}$, for example, has never been computed. This is one of the main differences compared to other approaches presented in Chapter 11. Other algorithms first compute all MIPS or MUPS and construct afterwards the diagnosis. In our approach, we compute MUPS on the fly whenever they are required to expand the

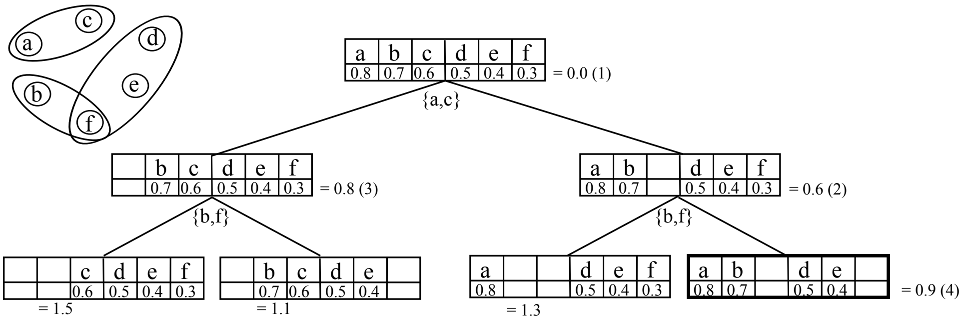

Figure 6.3: Applying the Algorithm BRUTEFORCEGOD$_{\mathcal{S}}$. MIPS of the alignment are depicted on the left side.

search tree.

Figure 6.3 demonstrates also the use of MUPS storage $\mathbb{M}$. Node expansions (1) and (2) are based on a reasoning process that detects $\{a, c\}$ and $\{b, f\}$ as unresolved conflicts. The search-tree is expanded according to this and these MUPS are stored in $\mathbb{M}$. At this time we have $\mathbb{M} = \{\{a, c\}, \{b, f\}\}$. The third expansion is based on a look-up in $\mathbb{M}$; no further reasoning is required to find that $\{b, f\}$ is an unresolved conflict.

The algorithm expands nodes according to the discovery of MUPS, which are not necessarily MIPS, whereas Figure 6.3 illustrates an example where MIPS and MUPS coincide for the sake of simplicity. This issue will be clarified within the proof of Proposition 11, which states that Algorithm 9 constructs a global optimal diagnosis.

Proposition 11. BRUTEFORCEGOD$_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ *is a global optimal diagnosis for* $\mathcal{A}$ *with respect to* $\mathcal{O}_1$ *and* $\mathcal{O}_2$.

Proof. First of all, ignore lines 2, 6-8, and 14 in Algorithm 9. The resulting simplification does not store MUPS that have previously been detected. Anyhow, it is clear that the simplified algorithm will finally find a global optimal diagnosis if and only iff the original algorithm finds such a diagnosis. For that reason it is sufficient to talk about the simplified variant of the algorithm within this proof.

Given the search tree finally constructed by the algorithm, consider the branch for which the final solution is the leaf node. Let this branch consist of n nodes $\mathcal{A}_1, \ldots, \mathcal{A}_n$ with $\mathcal{A} = \mathcal{A}_1$ and $\mathcal{A}_n$ being the coherent alignment such that $\Delta = \mathcal{A}_n \setminus \mathcal{A}_1$ is the final solution. A step from $\mathcal{A}_i$ to $\mathcal{A}_{i+1}$ is motivated by a MUPS $\mathcal{M}_i$ for which we have $\mathcal{A}_{i+1} \setminus \mathcal{A}_i \subseteq \mathcal{M}_i$. This means we construct implicitly a hitting set Δ for *MUPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$. We also know that the uniform cost search ensures that there exists no other Δ' with $\sum_{c \in \Delta'} \alpha(c) < \sum_{c \in \Delta}$. According to Proposition 8 we conclude that Δ is a global optimal diagnosis. □

In the following we introduce two extensions to improve the runtime of Algorithm 9. The improved version of the algorithm is sketched in Algorithm 10. It is an extended version that exploits the efficient reasoning strategies proposed in Section 5.2. Remember that we constructed in a similar way two algorithms – a base algorithm and an improved, more efficient version – for detecting a local optimal diagnosis.

The first extension is a preprocessing step in which $\mathbb{M}$ - the data structure that stores previously found conflicts – is filled with the MUPS detectable by the efficient reasoning strategies presented in Section 5.2. For that purpose the algorithm iterates over all pairs of correspondences in $\mathcal{A}$. Whenever the pattern based reasoning approach implemented in INCOHERENTPAIR decides that the current pair is incoherent, it is stored in $\mathbb{M}$. Then the algorithm continues with the main process as described above.

Algorithm 10 Computes a global optimal diagnosis by an A* search using efficient reasoning techniques.

EFFICIENTGOD$_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$

$Q \leftarrow \langle\rangle_{g+h}$
$\mathbb{M} \leftarrow \emptyset$
for $i \leftarrow 1$ to $|\mathcal{A}| - 1$ **do**
    for $j \leftarrow i$ to $|\mathcal{A}|$ **do**
        if INCOHERENTPAIR$(\mathcal{A}[i], \mathcal{A}[j], \mathcal{O}_1, \mathcal{O}_2)$ = **true then**
            ADD$(\mathbb{M}, \{\mathcal{A}[i], \mathcal{A}[j]\})$
        end if
    end for
end for
$\triangleright$ *continue as in* BRUTEFORCEGOD

Suppose for a moment that all MUPS are detectable during the preprocessing step. No further reasoning is required in the main loop. The algorithm will branch according to what is stored in $\mathbb{M}$. Only in the last iteration of the loop the coherence of $\mathcal{A}_{top}$ has to be proved once by a call to the complete reasoning procedure. However, this is the best case that will not always occur. Contrary to this, suppose now that none of the MUPS can be detected during the preprocessing. Then there is no difference between Algorithm 9 and Algorithm 10 with regard to their main phase. Thus, the effects of this modifications depend highly on the fraction of MUPS detectable by the pattern based reasoning component.

The second improvement is related to the use of a heuristic function to estimate the remaining costs for a search node. Its use is indicated by the subscript $g + h$ in the first line of the algorithm. Given some search node $\mathcal{A}$, the remaining costs are defined as the minimum sum of confidences of all those correspondences that have to be removed from $\mathcal{A}$ to arrive at a coherent alignment. The heuristic we propose is based on an approximation algorithm for the vertex cover problem presented in [Hoc97]. We introduce a weighted variant that constructs a weighted hitting set over the sets stored in $\mathbb{H}$ that are still contained in $\mathcal{A}$. The pseudocode of the algorithm is depicted in Algorithm 11.

Algorithm 11 Estimates the remaining costs required to arrive at a coherent sub alignment given a set of conflict sets.

APPROXHSWEIGHT$(\mathcal{A}, \mathbb{M})$

$h \leftarrow 0$
loop
  $\mathcal{M} \leftarrow$ RETRIEVEMUPS$(\mathcal{A}, \mathbb{M})$
  if $\mathcal{M} =$ NIL **then**
    return h
  end if
  $\mathcal{A} \leftarrow \mathcal{A} \setminus \mathcal{M}$
  $m \leftarrow \infty$
  for $c \in \mathcal{M}$ **do**
    if $m > \alpha(c)$ **then**
      $m \leftarrow \alpha(c)$
    end if
  end for
  $h \leftarrow h + m$
end loop

The algorithms starts with initializing the remaining cost h to 0. Inside the main loop the algorithm uses the helper function RETRIEVEMUPS to efficiently retrieve a randomly chosen set $\mathcal{M}$ with $\mathcal{M} \in \mathbb{M}$ and $\mathcal{M} \subseteq \mathcal{A}$. In case such $\mathcal{M}$ does not exist, the algorithm terminates and returns h. A coherent subset of $\mathcal{A}$ has been found. In case $\mathcal{M}$ exists, all correspondences in $\mathcal{M}$ are removed from $\mathcal{A}$. The minimum confidence $m \leftarrow \alpha(c)$ in $\mathcal{M}$ is determined, and m is added to h. The algorithm continues to reduce $\mathcal{A}$ like this until there exists no $\mathcal{M}$ with $\mathcal{M} \in \mathbb{M}$ and $\mathcal{M} \subseteq \mathcal{A}$. Note that the algorithm works on a local copy of $\mathcal{A}$.

Suppose now that we construct a global optimal diagnosis Δ for $\mathcal{A}$ given that all MUPS are specified in $\mathbb{M}$, i.e., Δ is the smallest hitting set for $\mathbb{M}$ regarding its total of confidence values. Then h can never be larger than the total of confidence values in Δ. The following proposition formally expresses the relation between Δ and h.

Proposition 12. *Let $\mathcal{A}$ be an alignment and let α be a confidence distribution over $\mathcal{A}$. Further, let $\mathbb{M} \subseteq \mathcal{P}(\mathcal{A}) \setminus \emptyset$ be a set of subsets of $\mathcal{A}$. For each hitting Δ over $\mathbb{M}$ we have $\sum_{c \in \Delta} \alpha(c) \geq$ APPROXHSWEIGHT$(\mathcal{A}, \mathbb{M})$.*

Proof. Each hitting-set Δ over $\mathbb{M}$ must have at least one correspondence in common with each $\mathcal{M} \in \mathbb{M}$. Algorithm 11 sums up the minimum confidence value of all $\mathcal{M} \in \mathbb{M}^*$ with $\mathbb{M}^* \subseteq \mathbb{M}$. For that reason we have $\sum_{c \in \Delta} \alpha(c) \geq$ APPROXHSWEIGHT$(\mathcal{A}, \mathbb{M})$ for each hitting set Δ over $\mathbb{M}$. □

Proposition 12 ensures that we can use APPROXHSWEIGHT as heuristic in an A* search to find the global optimal diagnosis, i.e., APPROXHSWEIGHT is a ad-

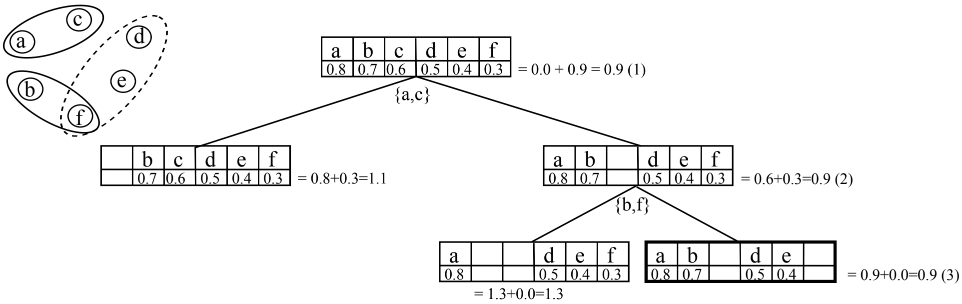

Figure 6.4: Applying the Algorithm EFFICIENTGOD$_{\mathcal{S}}$. MIPS of the alignment are depicted on the left side, the MUPS sketched by a dashed line has not been detected during preprocessing.

missible heuristic. This holds as long as $\mathbb{M} \subseteq$ *MUPS* $(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$, which is obviously the case at any time. Given an input alignment $\mathcal{A}$, an alignment $\mathcal{A}_{top}$ that represents an arbitrary node in the search tree, and a set of MUPS $\mathbb{M}$, we have $g(\mathcal{A}_{top}) = \sum_{c \in \mathcal{A} \setminus \mathcal{A}_{top}} \alpha(c)$ and $h(\mathcal{A}_{top}) = \text{APPROXHSWEIGHT}(\mathcal{A}_{top}, \mathbb{M})$. The efficient variant of the algorithm constructs thus also a global optimal diagnosis.

Proposition 13. EFFICIENTGOD$_{\mathcal{S}}(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ *is a global optimal diagnosis for* $\mathcal{A}$ *with respect to* $\mathcal{O}_1$ *and* $\mathcal{O}_2$ *when using Algorithm 11 as heuristic function.*

Proof. It follows from Proposition 12 that Algorithm 11 is an admissible heuristic to estimate the costs for constructing a global optimal diagnosis. Furthermore, we have already shown the correctness of Proposition 11. For that reason the A* search sketched in the descriptions of Algorithm 10 will – if based on Algorithm 11 as heuristic function – construct a global optimal diagnosis. □

The effectivity of the A*-search depends on the quality of the heuristic function, i.e., the better the heuristic function estimates the remaining costs, the less states will be visited until the final solution is detected. The quality of the heuristic depends mainly on the fraction of MUPS detectable by the pattern based reasoning component during the preprocessing, because $\mathbb{M}$ is the key component of the heuristic. The impact of a good heuristic can have a significant effect on the search-tree, in particular it helps to avoid a (more or less) equally balanced search tree. Instead of that nodes of the search tree that are close to the optimal solution are expanded to a deeper level and nodes that are far from being coherent – in terms of the heuristic – are left unexpanded.

The benefits of the A*-search are of special importance for large diagnostic problems. However, the example already presented above helps to clarify the effects of the proposed improvements. Figure 6.4 shows Algorithm10 applied to this example. Suppose that in the preprocessing two MUPS are detected, namely $\{a, c\}$ and $\{b, f\}$. MUPS $\{d, e, f\}$, depicted with a dashed line, is missed by the pattern based reasoning strategy. Each search node $\mathcal{A}$ is annotated with the score

$g(\mathcal{A}) + h(\mathcal{A})$ which determines the order in which nodes are expanded. Compared to Algorithm 9 there are two differences: (A) After the preprocessing step no further reasoning is required until the final solution is checked for coherence. This is based on the first modification of the base algorithm. (B) Node $\{b, c, d, e, f\}$ is not expanded. The heuristic estimates that at least an additional cost of 0.3 is required to find a coherent solution in one of the subsequent nodes. Instead of that, node $\{a, b, d, e\}$ has the best score. It is picked from Q, checked if its already a coherent alignment, and $\Delta = \{c, f\}$ is returned.

6.3 Runtime Considerations

In the previous sections we first developed basic, *'brute-force'* algorithms for computing a local and a global optimal diagnosis (Algorithm 6 and 9). These algorithms were completely based on reasoning in the aligned ontology. Remember that this type of reasoning should be reduced to a minimum, because it will in most cases – depending on the expressivity of the involved ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ – be the reason for a bad runtime performance of the algorithms. The pattern based reasoning methods developed in Section 5.2 run in polynomial time with respect to the size of the ontologies. Even though the pattern based approach requires to classify both ontologies, this is done only once. Any subsequent reasoning is based on checking entailed subsumption and disjointness in the previously classified ontologies. Opposed to this, reasoning in the aligned ontology depends always on the subset of $\mathcal{A}$ that is currently analyzed. It requires for each $\mathcal{A}' \subseteq \mathcal{A}$ a reclassification or subsumption test in the aligned ontology.

We proposed specific extensions for both algorithms that use the pattern based reasoning approach as far as possible to reduce the amount of reasoning in the aligned ontology. The positive effect of these extensions depends on the fraction of MUPS that is detectable by the pattern based methods. The number of reasoning tasks in the aligned ontology mainly determines the runtimes of the algorithms. This holds, in particular, as long as the other operations run in polynomial time. For that reason we analyze the runtimes of the presented algorithms mainly with respect to the number of calls to these methods in the following.

The brute-force approach to compute a local optimal diagnosis (Algorithm 6) requires $|\mathcal{A}|$ calls to complete reasoning methods, more precisely, $|\mathcal{A}|$ calls to ISCOHERENT. The runtime is not affected by the distribution of MUPS or MIPS over the input alignment $\mathcal{A}$. Contrary to this, the runtime of the efficient variant of the algorithm (Algorithm 8) is determined by the number of MIPS not detected by the pattern based reasoning. A MIPS that has been missed, is afterwards detected by $log(|\mathcal{A}|)$ calls to ISCOHERENT inside the helper algorithm FINDCOHERENCECRACK. Suppose now that we have m calls to FINDCOHERENCECRACK, then the runtime is $m * log(|\mathcal{A}|) + 1$, which is less then $|\mathcal{A}|$ iff $m << |\mathcal{A}|$. The empirical results presented in Chapter 8 will allow us to conclude whether or not this is the case.

The analysis of the algorithms for computing a global optimal diagnosis is more complicated. Here we have to face two problem at once. The reasoning that is involved has in worst case an exponential runtime. The same holds for constructing the hitting set, even if the complete set of all MUPS is known, which is not the case. Thus we have to ensure that we reduce the amount of full-fledged reasoning and build up the search tree in the most efficient way at the same time. The A*-search presented in Algorithm 10 is our proposal to solve this problem.

It is hardly possible to give a useful estimation of an average runtime within this thesis, because it is influenced by too many different factors (expressivity of the ontologies, size of the alignment, degree of coherence, size and overlap of MIPS and MUPS in the alignment). Nevertheless, there exists a special case for which the algorithm never expands an incorrect branch that might lead to a suboptimal diagnosis. This is the case when (a) all MUPS in $MUPS(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ are detectable by the pattern based approach, and (b) none of the MUPS in $MUPS(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$ is overlapping with another MUPS.

For this specific setting, the heuristic estimates the remaining costs perfectly, that means it already constructs implicitly a global optimal diagnosis. Due to this, a node will only be expanded if it is a node on a branch that leads to a global optimal diagnosis. This ensures that independent parts of the diagnostic problem are solved optimally and that the pattern based reasoning can have strong effects as long as it discovers a significant fraction of all MUPS. However, the performance of the approach can finally only be tested in a realistic setting, which is – among other issues - subject to the following part of this thesis.

6.4 Summary

In this chapter we gave (partial) answers to research questions R4 and R6. R4 asks for algorithms to compute specific types of diagnoses. According to previous considerations these types of diagnosis are the local and global optimal diagnosis. R6 is concerned with the runtime performance of these algorithms.

Regarding R4 we gave a complete answer. We formally introduced the notion of a local optimal and global optimal diagnosis in Chapter 4. In Chapter 5 we developed methods appropriate to reason about alignment coherence. In this chapter we have shown how to combine these techniques in algorithms to compute a local (Section 6.1) and global optimal diagnosis (Section 6.2). Moreover, we have formally proofed the correctness of our algorithms.

For each type of diagnosis we developed two variants. We have proposed a variant that we referred to as *'brute-force'* algorithm and a variant that we referred to as *'efficient'* algorithm. We have started to analyze the dependencies between expected runtime behaviour and characteristics of the particular matching problem. This analysis is our first contribution to research question R6. We identified the degree of completeness of the pattern based reasoning method as crucial for the runtime of the efficient algorithms. Degree of completeness refers in this context to the fraction of all MIPS that are detectable by the pattern based method.

Furthermore, we argued that the problem of computing the global optimal diagnosis, and in particular the underlying problem of computing the smallest weighted minimal hitting set, is an NP-complete problem. We emphasize again that this holds if we leave aside all aspects related to reasoning. It is thus an open question if the problem of computing a global optimal diagnosis is dominated by the hitting-set problem or by the reasoning required to create branches in the search-tree.

Part III

Applications

Chapter 7

Preliminary Investigations

Love all, trust a few. Do wrong to none (Shakespeare).

We have argued in the context of research question R1 and R2 about reasons for generating coherent alignments. However, we do not know to which degree state of the art matching systems ensure the coherence of their results. Within this chapter we try to answer this question. In particular, we report about experiments to measure the degree of incoherence of alignments generated by current state of the art ontology matching systems. Based on the results we can conclude in how far our approach is required and goes beyond the techniques implemented on top of those systems that we evaluate.

In Section 7.1 we first describe the ontologies that we use for our experiments. These ontologies origin from the Ontology Alignment Evaluation Initiative (OAEI). The OAEI is an international initiative organizing annual campaigns for evaluating ontology matching systems. We will use these datasets also in the experiments of Chapter 8, 9, and 10, a detailed description is already given in this chapter.

In Section 7.2 we introduce metrics to measure the quality of ontology alignments. The most common metrics are compliance based metrics such as precision and recall, well known from the field of information retrieval. Since these metrics are used in all of the following chapters, we introduce them explicitly. In addition, we propose a metric that allows to quantify the degree of alignment incoherence. Moreover, we argue that this metric allows to compute a strict upper bound for the correctness of an alignment without knowing the reference alignment.

In Section 7.3 we focus on the main topic of this chapter, i.e., we investigate to which degree current state of the art matching systems generate coherent alignments. It is unlikely that a matching system will ever be able to exploit relevant background knowledge and context information in a way that makes its decisions as reasonable as the decisions of a human being. However, matching systems should be able to exploit all information available in the machine-readable semantics of the ontologies to be matched. Having this in mind, we would expect that matching systems take alignment coherence into account. In Section 7.3 we conduct a series of evaluation experiments, illustrating that this assumption is not correct. Finally, we conclude in Section 7.4 and discuss in how far the measured results are relevant for the applications presented in the following chapters.

Since 2007 we have been involved in organizing the OAEI. In particular, we have conducted and reported about the evaluation experiments for the ANATOMY dataset from 2007 to 2010 in the OAEI results paper [EIM$^+$07, CEH$^+$08, EFH$^+$09, EFM$^+$10]. In addition we conducted several evaluation experiments measuring the degree of coherence for the CONFERENCE dataset. The results of these experiments are also published in the OAEI results papers. For the first time we present a more detailed analysis of these results. Based on our work reported in [MS08], we re-introduce the maximum cardinality measure as degree of alignment incoherence in Section 7.2.

7.1 Datasets

The datasets we use in our experiments origin from the OAEI, which is a coordinated international initiative that organizes the evaluation of ontology matching systems [EMS$^+$11]. The goal of the OAEI is to compare matching systems on the same basis in order to enable conclusions about strength and weaknesses of different approaches. The ambition is that from such evaluations tool developers can learn and improve their systems. The test sets used in the OAEI are hosted by different groups of researchers. Each of the test sets corresponds to a specific track in the OAEI. A track refers to both a set of matching tasks using the ontologies of the test set and the evaluation of the matching results with a set of general or track-specific methods. Over the last years between three and eight different tracks have been offered by the organizers. Some of them have been organized continuously with (nearly) the same dataset for several years.

The process of participation has been more or less the same for the last years. The required datasets are available at the webpage of organizers and can be downloaded for test purpose as well as for generating the final results.[1] The matching process that generates the results is conducted by the participants themselves. Final results, which are the automatically generated alignments, are then sent to the organizers, who evaluate these alignments. The alignments generated as results have to be in the format of the Alignment API [Euz04]. In 2010 we have been involved in the development of an partially automated evaluation process as reported in [TMES10].

In the following we describe the datasets used within our experiments. These are the datasets of the OAEI BENCHMARK, ANATOMY and CONFERENCE track.

[1] The datasets for the 2010 campaign are available via `http://oaei.ontologymatching.org/2010/`

BENCHMARK The benchmark track uses a synthetic test set that is built around one reference ontology and many variations of it. The reference ontology contains 33 named classes, 24 object properties, 40 data properties, and 76 individuals. Participants have to match the reference ontology with variations of it organized in three groups.

- #101-#104 Some simple tests comparing the reference ontology with itself and with its restriction to some OWL dialects.
- #201-#266 Systematic tests are obtained by discarding features from the reference ontology. The aim is to investigate how an algorithm behaves when a particular type of information is missing. This comprises for example scrambled or removed names and comments, a partial suppressed concept hierarchy, and discarded property restrictions.
- #301-#304 Reference alignments to four real-life ontologies of bibliographic references have been constructed manually.

The benchmark track had strong impact on the development of matching systems and the highest rate of participation compared to the other tracks.

ANATOMY The ontologies of the anatomy track are a small part of the NCI Thesaurus describing the human anatomy, published by the National Cancer Institute (NCI)[2], and the Adult Mouse Anatomical Dictionary[3], which has been developed as part of the Mouse Gene Expression Database project. Both resources are part of the Open Biomedical Ontologies (OBO). The ontologies are typical examples of large, carefully designed ontologies described in technical terms. The HUMAN ontology contains 3304 concepts and the MOUSE ontology 2744 concepts. The complex and laborious task of generating the reference alignment has been conducted by a combination of computational methods and an extensive manual evaluation with the help of domain experts (see [BHR+05] for more details).

CONFERENCE The conference dataset consists of a collection of 15 ontologies that describe the domain of conference organization. This dataset has been developed by a group of researchers from the University of Economics, Prague. Its origin is described in [SSB+05]. Since 2005 is has continuously been refined, extended and used as test data of the OAEI conference/consensus track. For a subset of seven ontologies reference alignments between all pairs have been created

The structure of the datasets is depicted in Figure 7.1. Edges between ontologies depict reference alignments. For the ontologies of the conference track we show only those ontologies for which reference alignments exist.

[2] http://www.cancer.gov/cancerinfo/terminologyresources/

[3] http://www.informatics.jax.org/searches/AMA_form.shtml

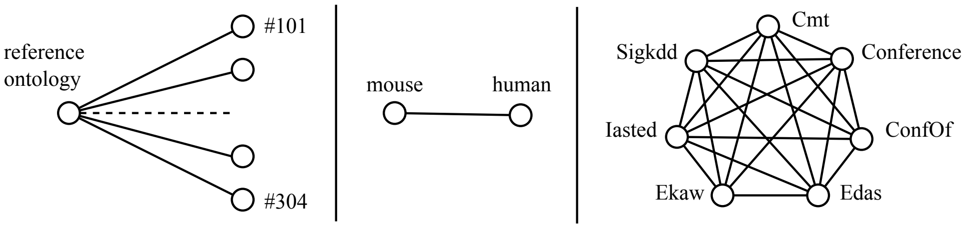

Figure 7.1: Structure of the BENCHMARK, ANATOMY and CONFERENCE dataset.

There are several reasons for the choice of these datasets. The first and most important reason is related to the existence of reference alignments. We already argued that ensuring the coherence of an alignment can be expected to increase its quality in terms of compliance against a reference alignment. To prove this claim datasets that comprise a reference alignment are required. The second reason is a pragmatic reason. All of the datasets have been used for several years and attracted a high number of participants. Since the OAEI organizers make all submitted alignments available (on request), there exists a rich test set for our approach. In addition we can compare our results against the top results that have been achieved over the last years. Note that this is not the case for the other OAEI tracks, with the exception of the DIRECTORY track. The third reason is related to the expressivity of the ontologies. Our method is without further extensions not applicable to alignments between ontologies that are mere concept hierarchies. The datasets of the DIRECTORY and the LIBRARY track are examples. For these datasets each combination of correspondences results in a coherent alignment and our approach has no effects.

Name	Type	#Concepts	# Dataprop.	# Objectprop.	DL
Ekaw	Insider	77	0	33	$\mathcal{SHIN}$
Conference	Insider	60	18	46	$\mathcal{ALCHIF}(\mathcal{D})$
Sigkdd	Web	49	11	17	$\mathcal{ALEI}(\mathcal{D})$
Iasted	Web	140	3	38	$\mathcal{ALCIN}(\mathcal{D})$
ConfOf	Tool	38	23	13	$\mathcal{SIN}(\mathcal{D})$
Cmt	Tool	36	10	49	$\mathcal{ALCIN}(\mathcal{D})$
Edas	Tool	104	20	30	$\mathcal{ALCOIN}(\mathcal{D})$

Table 7.1: Ontologies from the CONFERENCE dataset, also known as ONTOFARM collection

Contrary to this the CONFERENCE dataset comprises more complex modeling styles using negation and property restrictions. Details for the ontologies that come with a reference alignment are presented in Table 7.1. For that reason it will be in the main focus of our experiments. The dataset of the ANATOMY track has recently been refined and contains now also a small amount of disjointness statement [EFM+10]. We will in particular make use of it to investigate in how far our approach can be applied to large alignments. The dataset of the benchmark track lacks any disjointness. For that reason we will in our experiments use a variant where we added some disjointness statements in the reference ontology. The additional axioms have been created manually. Details can be found in Appendix B.

7.2 Metrics

The most prominent metrics used in the evaluation of ontology matching are precision and recall. They are based on comparing an alignment against a reference alignment. For that reason they are called compliance based measures. Contrary to this, we introduce a metric to measure the degree of alignment incoherence that does not require a reference alignment. As an answer to research question R2 we will show that both metrics are related and that it is possible to draw conclusions about the precision of an alignment based on its degree of incoherence.

Precision and recall are derived from the field of Information Retrieval. A definition for ontology matching has been given in [ES07].

Definition 30 (Precision and Recall)**.** *Given an alignment $\mathcal{A}$ and a reference alignment $\mathcal{R}$ between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. The precision p of $\mathcal{A}$ with respect to $\mathcal{R}$ defined as*

$$p(\mathcal{A}, \mathcal{R}) = \frac{|\mathcal{A} \cap \mathcal{R}|}{|\mathcal{A}|}.$$

The recall r of $\mathcal{A}$ with respect to $\mathcal{R}$ is defined as

$$r(\mathcal{A}, \mathcal{R}) = \frac{|\mathcal{A} \cap \mathcal{R}|}{|\mathcal{R}|}.$$

Precision counts how many detected correspondences are actually correct. Recall counts how many correct correspondences have actually been detected. It is thus a measure for the completeness of an alignment. A perfect alignment has a precision and recall of 1.0 and is thus identical to the reference alignment R.

Sometimes we want to specify the quality of an alignment by a single score. The arithmetic mean between precision and recall is an inappropriate choice for this purpose. Suppose we have an alignment that comprises correspondences between all pairs of entities. Its recall is thus 1.0 and its precision is close to 0.0. As arithmetic mean we would have a score > 0.5, while a score close to 0 makes more sense. A solution is the weighted harmonic mean of precision and recall, known as f-measure.

Definition 31 (f-measure)**.** *Given an alignment $\mathcal{A}$ and a reference alignment $\mathcal{R}$ between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. Let $p(\mathcal{A}, \mathcal{R})$ be the precision of $\mathcal{A}$ with respect to $\mathcal{R}$, and let $r(\mathcal{A}, \mathcal{R})$ be the recall of $\mathcal{A}$ with respect to $\mathcal{R}$. Then*

$$f(\mathcal{A}, \mathcal{R}) = \frac{2 \cdot p(\mathcal{A}, \mathcal{R}) \cdot r(\mathcal{A}, \mathcal{R})}{p(\mathcal{A}, \mathcal{R}) + r(\mathcal{A}, \mathcal{R})}$$

is the f-measure of $\mathcal{A}$ with respect to $\mathcal{R}$.

In this thesis we use the classic notion of precision and recall and do not apply extensions as relaxed [Ehr05] or semantic precision and recall [Euz07, Fle10]. This approach is sufficient with respect to our experimental settings, because we only analyze equivalence correspondences. Furthermore, in none of the ontologies used in our experiments there are equivalent concepts or properties. Semantic precision and recall is under these restrictions congruent with the classic notion of these concepts, because the deductive closure of an alignment $\mathcal{A}$ – if restricted to equivalence – is identical to $\mathcal{A}$.

Whenever we compute precision, recall, and f-measure for several alignments $\mathcal{A}_1, \ldots, \mathcal{A}_n$ we can aggregate the results in different ways. We have the choice between micro- or macro-average. The micro-average for precision is defined as $p(\bigcup_i^n \mathcal{A}_i, \bigcup_i^n \mathcal{R}_i)$ while the macro-average is defined as $\sum_i^n p(\mathcal{A}_i, \mathcal{R}_i)/n$. A similar definition can be given for recall. The micro/macro-average of the f-measure is computed by using the micro/macro-average of precision and recall in Definition 31. Given alignments of similar size, the differences between both ways to aggregate are limited. If the alignments differ in size (examples can be found in the CONFERENCE dataset), the values for macro- and micro-average can also differ. Whenever we present aggregated values, we present scores of the micro-average aggregation. This avoids that small matching tasks can have an influence on the average score that is too strong compared to the small number of correspondences that are generated. However, for all of our datasets we observed only minor differences between both ways to aggregate.

While precision and recall score in the range $[0, 1]$, alignment incoherence – with regard to Definition 21 – is a boolean criterion. It does not distinguish between different degrees of incoherence. We present now a measure we first defined in [MS08]. This measure allows to distinguish between different degrees of incoherence.

Definition 32 (Degree of incoherence)**.** *The cardinality-based degree of incoherence c of an alignment $\mathcal{A}$ between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ is defined by*

$$c(\mathcal{O}_1, \mathcal{O}_2, \mathcal{A}) = \frac{|\Delta|}{|\mathcal{A}|}$$

where Δ is a diagnosis with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ and there exists no $\Delta' \subseteq \mathcal{A}$ with $|\Delta'| < |\Delta|$ such that Δ' is a diagnosis with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$.

In this definition, Δ refers to the diagnosis with the smallest number of correspondences. We already got to know a very similar diagnosis, namely the global optimal diagnosis Δ_G defined as the diagnosis with the smallest total of confidence values $\sum_{c \in \Delta_G} \alpha(c)$ where α referred to a confidence allocation over $\mathcal{A}$. Suppose now that we set $\alpha(c) = 1$ for each $c \in \mathcal{A}$ or to some other constant value $\epsilon > 0$. Then the global optimal diagnosis Δ_G will also be the diagnosis with the smallest number of correspondence. Thus, we can easily modify the confidence allocation α

to compute the cardinality-based degree of incoherence by applying the algorithms we developed in Section 6.2 to compute a global optimal solution.

Even though the incorrectness of a correspondence does not necessarily result in an incoherent alignment, we argued that an incoherent alignment contains at least one incorrect correspondence. It follows from the contraposition that a set of correct correspondences – and thus also a reference alignment – is always coherent. We conclude the following.

Proposition 14 (Incoherence and Precision)**.** *Let $\mathcal{R}$ be a reference alignment between $\mathcal{O}_1$ and $\mathcal{O}_2$. Further let $\mathcal{A}$ be incoherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$. Then we have $p(\mathcal{A}, \mathcal{R}) < 1$.*

Proof. Given the coherence of $\mathcal{R}$, it can be concluded that every subset of $\mathcal{R}$ is coherent, too. Since $\mathcal{A}$ is incoherent, it is thus not equal to nor a subset of $\mathcal{R}$. We conclude that $\mathcal{A} \cap \mathcal{R} \subset \mathcal{A}$. It follows $p(\mathcal{A}, \mathcal{R}) < 1$. □

Proposition 14 is of limited benefit. Especially automatically generated alignments do not have a precision of 1 in most cases. Nevertheless, the proposition can be generalized by exploiting the definition of the cardinality-based degree of incoherence given above. This generalization allows us to compute a non-trivial upper bound for precision without any knowledge of $\mathcal{R}$.

Proposition 15 (Upper Bound for Precision)**.** *Let $\mathcal{R}$ be a reference alignment between $\mathcal{O}_1$ and $\mathcal{O}_2$. Then we have $p(\mathcal{A}, \mathcal{R}) \leq 1 - c(\mathcal{O}_1, \mathcal{O}_2, \mathcal{A})$.*

Proof. Accordant to Definition 32 let $\mathcal{A}' = \mathcal{A} \setminus \Delta$ be the coherent subset of $\mathcal{A}$ with maximum cardinality. Further let $\mathcal{A}^* = \mathcal{A} \cap \mathcal{R}$ consist of all correct correspondences in $\mathcal{A}$. Since $\mathcal{A}^*$ is a subset of $\mathcal{R}$ and $\mathcal{R}$ is coherent, we conclude that $\mathcal{A}^*$ is also coherent. It follows that $|\mathcal{A}^*| \leq |\mathcal{A}'|$, because otherwise $\mathcal{A}'$ would not be the coherent subset of maximum cardinality contrary to Definition 32. In summary, the following inequation holds.

$$p(\mathcal{A}, \mathcal{R}) = \frac{|\mathcal{A} \cap \mathcal{R}|}{|\mathcal{A}|} = \frac{|\mathcal{A}^*|}{|\mathcal{A}|} \leq \frac{|\mathcal{A}'|}{|\mathcal{A}|} = \frac{|\mathcal{A} \setminus \Delta|}{|\mathcal{A}|} = 1 - \frac{|\Delta|}{|\mathcal{A}|} = 1 - c(\mathcal{O}_1, \mathcal{O}_2, \mathcal{A})$$

□

We have defined a metric for measuring the degree of incoherence. This allows us to measure to which degree state of the art matching systems generate coherent alignments. We report on the result of our experimental analysis in the following section. In addition, we have gained a deeper insight in the relation between precision and coherence. Remember that we already touched the issue in Section 3.1.

7.3 Experimental Results

In the following we first apply the incoherence metric to the submissions of the CONFERENCE track of the OAEI 2010. Remember that the dataset is highly expressive and therefore well suited to understand whether matching systems avoid alignment incoherence. At the end of this section we also present measurements for the anatomy track, to see in how far coherence is relevant for matching less expressive ontologies.

With respect to the CONFERENCE track we analyze the matching results for nearly all pairs of ontologies. However, we exclude the ontologies LINKLINGS and CONFIOUS, because Pellet [SPG$^+$07] – the OWL reasoner we use to perform the reasoning tasks – cannot classify some of the ontologies that result from merging these ontologies with one of the other conference ontologies. For that reason we have 13 conference ontologies resulting in 78 matching tasks. In 2010 seven matching systems participated at the CONFERENCE track. Thus, we analyze a total of 624 alignments. A short description of these matching systems (OAEI 2010) and OAEI participants from 2008 and 2009 can be found in Appendix C.

The organizers of the track have used in 2009 [EFH$^+$09] and 2010 [EFM$^+$10] a specific approach to enable an evaluation that takes possible misconfigurations into account. This approach is based on searching and applying a-posteriori a threshold, that results in an optimal f-measure for the system. With respect to our experimental results, we distinguish thus between the original and the thresholded alignments. In doing so, we avoid the objection that the strong effects of your approach are related to a misconfiguration of matching systems.

The main results for the thresholded alignments are depicted in Figure 7.2. An overview on the frequency of different degrees of incoherence is given by dividing the set of alignments generated by a matching system into quartiles. Given a list of alignments ordered ascending regarding its degree of incoherence, the first quartile is the subset of alignments that covers the first 25% in the list, the second quartile continues with the next 25% and ends with the median, and so on.

The results clearly show that alignment incoherence is barely used during the matching process by most of the systems. The system with the highest degree of incoherence is AROMA. 25% of all generated alignments have a degree of incoherence between 0.26 and 0.45. This means that for one out of four alignments at least 26% of all correspondences have to be removed to reach a coherent subset. Due to Proposition 15 we also know that for logical reasons at least 26% of the correspondences of these alignments are incorrect. The results for most of the other matching systems are similar. With the exception of CODI and FALCON the median – the border between 2nd and 3rd quartile – can be found in the range from 0.05 to 0.18.

Table 7.2 shows the average degree of incoherence $c(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$, the average size of the alignments, the average precision of the alignment, and the number of incoherent alignments. We present both the values for the original alignments and their thresholded counterparts. To ease the comparison between degree of incoherence and precision, we show the values for $1-$ precision in the table. Note also that precision values have been computed only for the subset of seven ontologies for which reference alignments are available.

Even though we applied a-posteriori an optimal threshold, the size of the alignments varies due to the fact that some systems generate their top f-measure with

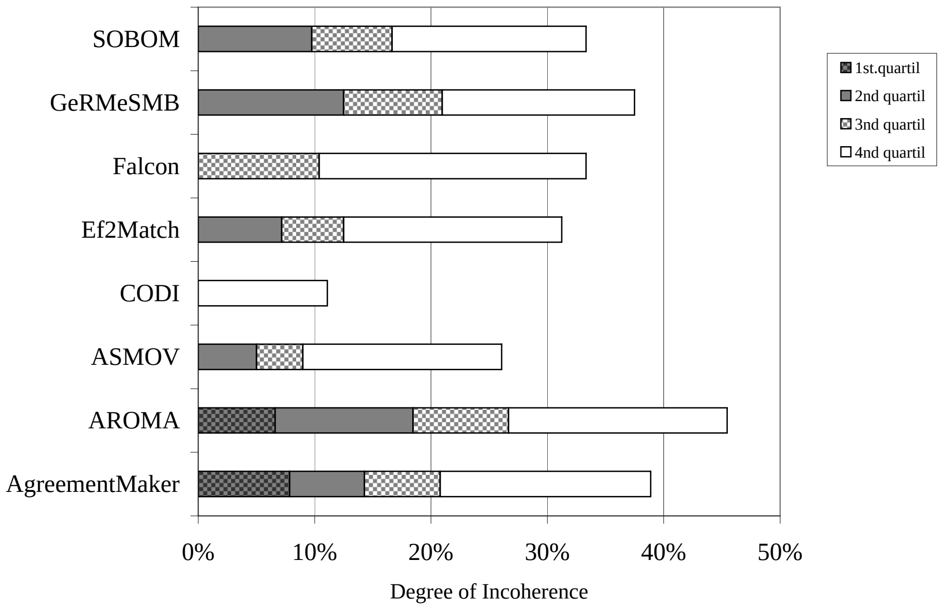

Figure 7.2: Alignment coherence for the conference dataset regarding OAEI 2010 submissions (with optimal a-posteriori threshold applied).

highly precise alignments (e.g., CODI, FALCON) while some of them achieve this score with a higher recall value (e.g., ASMOV, AGREEMENTMAKER).

The results show that none of the systems could generate coherent alignments for all of the 78 matching tasks. A positive exception is CODI that fails only in one case.[4] However, the alignments generated by CODI are by far the smallest. The degree of incoherence is negatively correlated with the size of the alignments. Note that the use of the denominator $|\mathcal{A}|$ in Definition 32 is not a sufficient normalization factor, because the numerator Δ is a smallest hitting set over a subset of $\mathcal{A}^2$ (power set if $\mathcal{A}$). An appropriate normalization is thus hardly possible. For that reason we restrict a comparative analysis to systems that generate alignments of similar size.

The results indicate that the verification component of ASMOV has some positive effects on the degree of incoherence. For the thresholded submissions we observe an average size of 18.2 correspondences for ASMOV. This is comparable to the alignments of AGREEMENTMAKER, AROMA, and GERMESMB. ASMOVs degree of incoherence is with 0.056 significantly lower than the values of these systems. Anyhow, the system is still far away from generating coherent alignments. Overall, we observe that most systems – with the exception of CODI – clearly fail to generate coherent alignments for the CONFERENCE dataset.

According to Proposition 15 we know that $1 - c(\mathcal{O}_1, \mathcal{O}_2, \mathcal{A})$ is an upper bound

[4]Note that some of the ideas presented in this thesis have been taken up by the developers of CODI, who have been collaborating with the author of this thesis. Details can be found in [NMS10, NN10].

Matcher	$c(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$	Size	$1 - p(\mathcal{A}, \mathcal{R})$	# Inc. Alignments
AGRMAKER	0.151	17.4	0.507	68
with threshold	0.148	17.1	0.498	67
AROMA	0.178	16.7	0.65	61
with threshold	0.175	16.4	0.647	60
ASMOV	0.102	26.5	0.652	64
with threshold	0.056	18.2	0.485	45
CODI	0.001	6.9	0.151	1
EF2MATCH	0.085	16.1	0.512	52
with threshold	0.072	12.8	0.401	44
FALCON	0.121	13.4	0.416	59
with threshold	0.048	8.9	0.236	29
GERME	0.133	19.5	0.672	55
with threshold	0.126	18.2	0.649	56
SOBOM	0.202	28.6	0.711	66
with threshold	0.107	11.7	0.422	54

Table 7.2: Average degree of incoherence referred to as $c(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$, average size of alignments, average precision for subset with reference alignments, and number of incoherent alignment for the OAEI CONFERENCE submissions.

for the precision $p(\mathcal{A}, \mathcal{R})$ of an alignment $\mathcal{A}$, i.e., that $1-p(\mathcal{A}, \mathcal{R}) \geq c(\mathcal{O}_1, \mathcal{O}_2, \mathcal{A})$). Our experimental results show that actual precision scores are significantly lower than the upper bound. However, at the same time we observe a high correlation and, even more, a relatively constant factor of ≈ 4 between the degree of incoherence and $1 - p(\mathcal{A}, \mathcal{R})$. This means that the incoherence of an alignment is a good criterion to assess the precision of an alignment, especially when the matched ontologies are expressive and contain disjointness axioms as it is the case for the CONFERENCE dataset.

We conducted the same analysis to the 2010 submissions of the ANATOMY track analyzing the submissions for subtask #1. In this subtask the matcher has to be run in its standard setting. While for the CONFERENCE track the average precision score is ≈0.57, the average precision for the ANATOMY track is ≈0.92. Room for improvement is thus only limited. Moreover, the ontologies contain only a small number of disjointness statements between top-level concepts.

These observations are reflected in Table 7.3. The average degree of incoherence is significantly lower than the values we measured for the CONFERENCE dataset. The effects of repairing such an alignment with the use of a diagnostic approach will thus only be limited. Nevertheless, we measured that none of the systems, with the exception of CODI and GERME, manages to generate a coherent alignment.

Of additional interest is a fact not depicted in the table. With only one ex-

Matcher	$c(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$	Alignment Size	# Inc. Alignments
AgrMaker	0.002	1436	1
AROMA	0.004	1158	1
ASMOV	0.012	1468	1
BLOOMS	0.004	1164	1
CODI	0	1023	0
Ef2Match	0.002	1243	1
GeRMe	0	528	0
NBJLM	0.008	1327	1
TaxoMap	0.004	1223	1

Table 7.3: Degree of incoherence referred to as $c(\mathcal{A}, \mathcal{O}_1, \mathcal{O}_2)$, size of alignment, and number of incoherent alignments for the OAEI ANATOMY submissions.

ception all MIPS contained in these alignments can be detected by the efficient reasoning methods described in Section 5.2. This shows that none of the participating systems apply a filtering or verification technique to eliminate these obviously incorrect combinations of correspondences. This is surprising because such an approach does not require the use of sophisticated reasoning. This observation conflicts with the system description of ASMOV [JMSK09]. A reasonable explanation is currently missing. Maybe the verification component has been turned of for matching the ANATOMY ontologies.

7.4 Conclusions

Within this chapter we first introduced in Section 7.1 the datasets used in the experiments of this and the following chapters. Further, we introduced the well known metrics of precision and recall and defined a way to measure the degree of alignment incoherence in Section 7.2. We proved that the degree of incoherence is an upper bound for the precision of an alignment. Proposition 15 can be seen as a precise answer to research question R2. It is thus one of the two main contributions of this chapter.

Knowing that the degree of incoherence is an upper bound for the correctness of an alignment, however, is not sufficient to motivate the benefits of our approach. In particular, we analyzed to which degree alignments generated by state of the art matching systems are incoherent. The ability to reason with ontologies – and also the ability to detect incoherences – is a crucial feature that distinguishes ontologies from other ways to structure data. Thus, we expected that most or at least some ontology matching systems focus especially on these aspects and generate coherent results.

We conducted several experiments to verify our assumptions in Section 7.3 and were surprised to see that the opposite is the case. We observed some cases where

40% of the correspondences of an alignment had to be removed for logical reasons. None of the matching systems, with the exception of CODI, could ensure the coherence for a significant number of alignments. Thus, the diagnostical methods developed in Part II will have strong effects if applied to repair the incoherent alignments generated by most of todays matching systems.

We have also seen that these effects depend largely on the expressivity of the ontologies to be aligned. In particular, we have analyzed the alignments for the ANATOMY dataset. The ontologies of this dataset are in $\mathcal{EL}$ and $\mathcal{EL}$++. While we measured only a very low degree of incoherence (in most cases less than 1%), only two of nine systems managed to generate coherent alignments for this matching task.

The aim of our experiments was not directly related to the research questions presented in the introduction. We conducted them to show that a specific objection to our approach is not valid. This is the objection that the presented approach might be theoretically interesting, but is not required in a practical context. Our results clearly show that the opposite is the case. We know now that the removal of a diagnosis can have strong effects on the precision of an alignment. This follows from analyzing the experimental results in the light of Proposition 15. However, we do not know if the local or global optimal diagnosis will indeed help us to identify those correspondences that are incorrect. This means that we still have to answer research question R5 by analyzing concrete effects on precision and recall. The answer will be given in the following two chapters.

Chapter 8

Alignment Debugging

A successful person is one who can lay a firm foundation with the bricks that others throw at him or her (David Brinkley).

In the previous section we have seen with surprise that many ontology matching systems do not take alignment coherence into account. As a result, alignments generated by these systems are incoherent. We can thus expect that the application of our approach will have a significant impact. In particular, we can distinguish between two effects of debugging an incoherent alignment. First, an alignment, that has been incoherent, becomes coherent. This sounds trivial, however, we already argued that an incoherent alignment will raise problems in different application scenarios. Debugging an incoherent alignment is thus a necessary prerequisite to enable its use. Second, the removal of the diagnosis Δ will change the characteristic of the alignment in terms of precision and recall. Effects related to this second aspect, which is the main concern of research question R5, can only be clarified by an experimental study.

We start our experiments in Section 8.1 with an analysis of results for the CONFERENCE dataset. We use the official OAEI submissions of 2008, 2009, and 2010 as input alignments and compute both a local and global diagnosis. In Section 8.2 we analyze the impact of your approach on the submissions to the BENCHMARK and ANATOMY track. In the first two sections we focus on precision and recall, while in Section 8.3 we compare runtimes and try to understand the effects of the pattern based reasoning methods. Finally, we summarize our results in Section 8.4.

A fragment of the experimental results of Section 8.1 has been presented in [MS09b]. However, the results presented there covered only a small fraction of the tests we conducted in the context of this thesis. Moreover, there we only measured the effects of computing a local optimal diagnosis. The results for computing the global optimal diagnosis are novel. The same holds for the measurement and analysis of runtimes. These results have not been presented in any other publication. Results we published previously were based on the use of

DDL [MST06, MTS07, MST09], while the results presented in [MS07a, MS07b] were based on incomplete reasoning methods mixed with one-to-one extraction methods.

8.1 Results for the Conference Dataset

In this section we focus on the CONFERENCE dataset. We have already argued that it is the dataset that comprises ontologies with a high expressivity in terms of different DL constructs and axiom types. For that reason we expect that our approach has a significant impact. The input alignments to our algorithms are the alignments generated by the matching systems participating in 2008, 2009, and 2010.

Since all our algorithms for computing a diagnosis require that an alignment is annotated with a confidence allocation α, we analyze only those systems generating different confidence values in the range $[0, 1]$. It might still occur that some correspondences in an alignment are annotated with the same confidence value. In this case the outcome of the algorithms might not be fully determined and a random choice is made. However, since we process a large amount of alignments (21 matching tasks $\times$ 12 participations of matching systems = 252 alignments) and present average results, the effects of a random choice will be balanced.

We start our report with analyzing the results for computing a local optimal diagnosis. Aggregated results are depicted in Table 8.1. Note that in our experiments we compare an automatically generated alignment $\mathcal{A}$ against $\mathcal{A}' = \mathcal{A} \setminus \Delta$ where Δ is a diagnosis. In the first column you can see the matching systems that generated the alignments we used in our experiments. The subscript informs about the year of participation. Most systems have evolved over the years. For that reason some systems have been participating in two or three years with different results.

The second and third column informs about the total of correspondences generated by the matching system – summed up over all 21 alignments for each row – and the total of correspondences in the computed diagnoses. The last rows shows the average scores. In average 18.25% of all correspondences are removed if we compute for each of the alignments the local optimal diagnosis and retract it. That means our approach of debugging an alignment be removing a local optimal diagnosis has a notable effect. In the following three columns precision, f-measure, and recall for the input alignments are shown, while in the subsequent columns the same scores for the repaired alignments $\mathcal{A}' = \mathcal{A} \setminus \Delta$ are shown. In the final column we show the difference between the f-measure of the input alignment and the repaired alignment. A positive value indicates that our approach could increase the f-measure of the repaired alignment.

The average f-measure increases in 8 of 12 cases. In average we measure an increase by 1.4%. This result is based in a notable increase in precision from 42.2% to 46.6%. However, we loose at the same time a similar amount of recall, which decreases from 55.6% to 51.3%. Overall there is a small win in f-measure. Analysing

Matcher	$\lvert\mathcal{A}\rvert$	$\lvert\Delta\rvert$	$p(\mathcal{A},\mathcal{R})$	$f(\mathcal{A},\mathcal{R})$	$r(\mathcal{A},\mathcal{R})$	$p(\mathcal{A}',\mathcal{R})$	$f(\mathcal{A}',\mathcal{R})$	$r(\mathcal{A}',\mathcal{R})$	+/-
AGRMAKER$_{10}$	402	60	0.493	0.559	0.647	0.545	0.57	0.598	0.011
ASMOV$_{10}$	633	64	0.348	0.469	0.719	0.382	0.496	0.706	0.027
Ef2Match$_{10}$	394	45	0.487	0.549	0.627	0.513	0.54	0.569	-0.009
Falcon$_{10}$	300	39	0.583	0.578	0.572	0.66	0.59	0.533	0.012
GeRMeSMB$_{10}$	470	64	0.328	0.397	0.503	0.343	0.39	0.451	-0.007
SOBOM$_{10}$	657	173	0.288	0.396	0.631	0.362	0.439	0.556	0.043
AgrMaker$_{09}$	443	98	0.395	0.472	0.588	0.493	0.521	0.552	0.048
AgrMakerE$_{09}$	634	159	0.286	0.385	0.588	0.356	0.418	0.507	0.033
Aroma$_{09}$	423	80	0.363	0.42	0.5	0.433	0.446	0.461	0.026
ASMOV$_{09}$	337	7	0.364	0.392	0.425	0.376	0.398	0.422	0.005
ASMOV$_{08}$	474	67	0.314	0.381	0.484	0.332	0.374	0.428	-0.007
Lily$_{08}$	394	58	0.395	0.442	0.5	0.427	0.437	0.448	-0.005
Average	442.2	82.3	0.422	0.463	0.556	0.467	0.477	0.513	0.014

Table 8.1: Results of debugging the alignments of the CONFERENCE track by computing a local optimal diagnosis.

the results for the specific matching systems, we observe that the approach seems to work for some systems better than for others. For the submissions of AGREEMENTMAKER and SOBOM in 2009 we observe a win of $\approx$ 5% in f-measure, while some systems loose approximately 1% f-measure.

We conducted the same experiments for the global optimal diagnosis. Results are presented in Table 8.2. The average global optimal diagnoses is smaller compared to the average local optimal diagnosis. The approach removes less correspondences than the top down approach of the local optimal diagnosis. With respect to precision and recall, we observe the same pattern: The diagnostic approach results in a trade of between precision and recall. However, this time we loose less recall. As a result there is a difference of 1.8% (1.4% for the local optimal diagnosis) between the average f-measure of the input alignment and the average f-measure of the repaired alignment. Analyzing the aggregated results for the specific matching systems, we observe less differences. Contrary to the results of the local optimal diagnosis, we measure for each of the matching systems an increased f-measure.

Up to now we have analyzed the aggregated results, which are computed from a set of 21 matching tasks. Figure 8.1 shows the effects of debugging the alignments of the FALCON system [HQ08] in detail. It shows the values for computing the global optimal diagnosis. We will later also visualize and discuss the results of applying the local optimal diagnosis on the same set of input alignments (see Figure 8.2). Both figures display for each matching task the differences between the input alignment and its repaired subset as a bar in the diagram. The first bar in Figure 8.1, for example, depicts a matching task where precision has been raised by 7.1% (white area of the bar), and recall has been decreased by 6.3% (gray area of the bar). The resulting f-measure has been increased by 0.4%, which is indicated

Matcher	$\|\mathcal{A}\|$	$\|\Delta\|$	$p(\mathcal{A},\mathcal{R})$	$f(\mathcal{A},\mathcal{R})$	$r(\mathcal{A},\mathcal{R})$	$p(\mathcal{A}',\mathcal{R})$	$f(\mathcal{A}',\mathcal{R})$	$r(\mathcal{A}',\mathcal{R})$	+/-
AgrMaker_{10}	402	60	0.493	0.559	0.647	0.55	0.58	0.614	0.021
ASMOV_{10}	633	64	0.348	0.469	0.719	0.381	0.496	0.709	0.027
Ef2Match_{10}	394	45	0.487	0.549	0.627	0.53	0.565	0.605	0.016
Falcon_{10}	300	39	0.583	0.578	0.572	0.659	0.607	0.562	0.029
GeRMe_{10}	470	64	0.328	0.397	0.503	0.352	0.402	0.467	0.005
SOBOM_{10}	657	173	0.282	0.384	0.603	0.337	0.412	0.531	0.028
AgrMaker_{09}	443	98	0.404	0.478	0.585	0.484	0.513	0.546	0.035
AgrMaker^{E}_{09}	634	159	0.282	0.381	0.585	0.316	0.384	0.49	0.003
Aroma_{09}	423	80	0.352	0.409	0.487	0.411	0.435	0.461	0.026
ASMOV_{09}	337	7	0.374	0.392	0.412	0.382	0.396	0.412	0.004
ASMOV_{08}	474	67	0.312	0.379	0.484	0.344	0.393	0.458	0.013
Lily_{08}	394	58	0.406	0.457	0.523	0.443	0.464	0.487	0.007
Average	463.4	76.1	0.388	0.453	0.562	0.432	0.471	0.528	0.018

Table 8.2: Results of debugging the alignments of the CONFERENCE track by computing a global optimal diagnosis.

by the small red area in the middle. This illustration shows that the aggregated small win in f-measure for FALCON is based on very different results for the particular matching tasks. In some cases the diagnostic approach has no effects, while there are other cases in which our method could increase precision by more than 20%. A similar characteristic can be observed for the other matching systems. We have added the same figures for all other participants of OAEI 2010 in Appendix D.

In Figure 8.2 we depict results for the same setting computing the local optimal diagnosis. The results for some of the matching tasks do not change, however, there are also some tasks where we observe a higher precision and a lower recall. Even more, for a few cases the diagnostic approach eliminates only correct and non of the incorrect correspondences. Overall, there is a higher variance compared to the global optimal diagnoses. Similar results can be again observed for the other matching systems.

Overall, the results of the global approach are slightly better than the results of the local approach. However, the global approach reduces the number of negative (and sometimes positive) outliers. Given a correspondences c with high confidence that conflicts with many other correspondences $c_1, c_2, ...$ with low confidence, the local approach will always trust in the correctness of c and will remove $c_1, c_2, ...$, while in a global approach the confidences of c and $c_1, c_2, ...$ are compared. Due to this, the global approach generates slightly better and well balanced results compared to the local approach. This consideration has been supported by our experimental results.

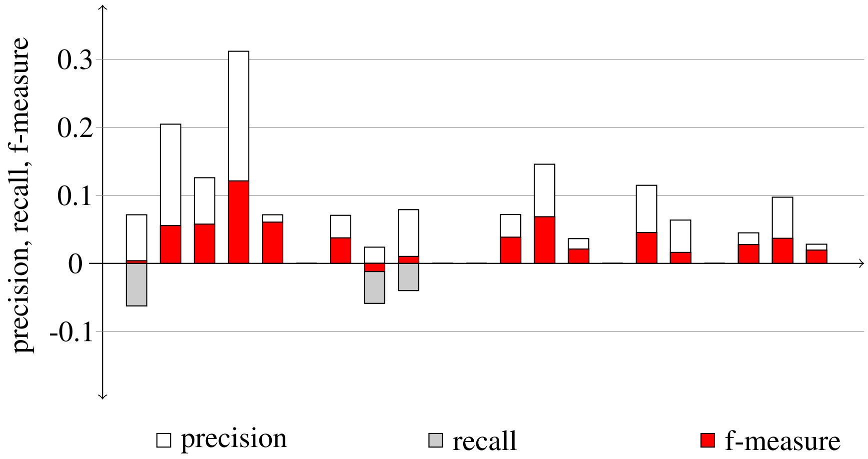

Figure 8.1: Effects of debugging alignments of the FALCON system computing global optimal diagnoses.

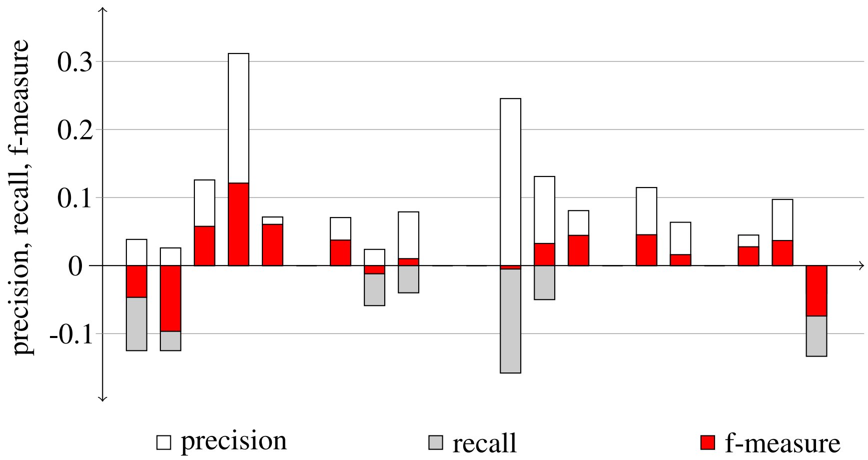

Figure 8.2: Effects of debugging alignments of the FALCON system computing local optimal diagnoses.

Matcher	input alignment		local		global	
	$\|\mathcal{A}\|$	$p(\mathcal{A},\mathcal{R})$	$\|\Delta \cap \mathcal{R}\|$	$\|\Delta \setminus \mathcal{R}\|$	$\|\Delta \cap \mathcal{R}\|$	$\|\Delta \setminus \mathcal{R}\|$
AGRMAKER	1436	0.903	0	3	0	3
ASMOV	1469	0.799	3	20	5	12
BLOOMS	1164	0.954	1	4	1	4
EF2MATCH	1243	0.955	1	2	1	2
GERMESMB	528	0.884	0	0	0	0
NBJLM	1327	0.920	12	11	1	10
TAXOMAP	1223	0.924	1	4	1	4

Table 8.3: Debugging alignments of the ANATOMY dataset.

8.2 Results for Benchmark and Anatomy Dataset

The result for applying our approach on the ANATOMY dataset are presented in Table 8.3. Each row in this table shows the results for a specific matching system. The alignments generated by the matching systems vary to a large degree. We present the size of the alignments in number of correspondences together with their precision in the second and third column. We have computed for each input alignment a global and local optimal diagnosis Δ and counted the number of correspondences in $|\Delta \cap \mathcal{R}|$, i.e., those correspondences that have been eliminated but are contained in the reference alignment, and the number of correspondences in $|\Delta \setminus \mathcal{R}|$, i.e., those correspondences that have been eliminated correctly.

We present the results in absolute numbers instead of presenting the precision/recall scores before and after applying our methods. Due to very limited use of negation in the ANATOMY dataset, the overall effects in terms of an increased precision are very small. The anatomy datasets contains only a small number of disjointness axioms: 17 disjointness axioms in the human ontology that comprises 3304 concepts, and no disjointness axioms in the mouse ontology. However, we can still analyze some interesting characteristics of our approach. The precision of the matching systems varies between 0.8 and 0.95. If we randomly pick ten correspondence from an alignment with a precision of 90% nine of them will be correct and only one will be incorrect. Thus, the circumstances are in general very hard for any approach that filters out incorrect correspondences.

Contrary to this, the results of our approach are much better. Based on the local approach we remove 62 correspondences among which 44 are incorrect; based on the global optimal approach we remove 43 correspondences among which 35 are incorrect. That means that the precision of the debugging is surprisingly high given that the precision of the input alignments is already very high.

Comparing the local optimal diagnoses against the global optimal diagnoses, we notice differences only for the two systems ASMOV and NBJLM. In both cases the global optimal diagnosis is *'less aggressive'* in terms of removed correspondences. While the global optimal approach yields worse result for ASMOV,

Matcher	input alignment		global	
	$\lvert\mathcal{A}\rvert$	$p(\mathcal{A},\mathcal{R})$	$\lvert\Delta \cap \mathcal{R}\rvert$	$\lvert\Delta \setminus \mathcal{R}\rvert$
AGRMAKER	152	0.922	4	10
AROMA	160	0.8	0	4
ASMOV	221	0.869	1	2
EF2MATCH	189	0.912	0	2
FALCON	197	0.886	0	2
GERMESMB	108	0.88	2	0
MAPPSO	121	0.656	1	5
RIMOM	187	0.944	1	4
SOBOM	220	0.847	1	9
TAXOMAP	103	0.711	8	22
Total	1658	0.842	18	60

Table 8.4: Debugging alignments of the BENCHMARK dataset.

the impact on the alignment generated by NBJLM is very good. These differences fit with the insights that we gained from the comparison of Figure 8.1 and Figure 8.2 in the previous section. The global optimal diagnosis generates in average slightly better results than the local optimal diagnosis and avoids negative and sometimes also positive outliers.

We applied the same experiments to the BENCHMARK dataset. In particular, we used the slightly modified variant described in Appendix B. When trying to apply our approach to the test cases of the #1xx and #2xx series we observed problems. While we could classify each of the ontologies on their own with the Pellet reasoner, we had problems with reasoning in the aligned ontology for most of the testcases.[1] In addition to these problems, many matching systems generate very good and sometimes nearly perfect results for the testcases of the #1xx and #2xx series. The top system had an average of 99% precision and 89% recall for the #2xx series (many systems have a perfect score for the #1xx series). Obviously, there is not much room for improvement given a method that aims at improving the precision of an alignment. A detailed analysis of the BENCHMARK results over the last five years can be found in [EMS+11].

Thus, we used the #3xx series, which consist of four matching tasks. For each matching system participating in 2010, we aggregated the results over these matching tasks. Table 8.2 presents the results of our experiments. This time we suppress the results for the local optimal diagnosis and present only the results for the global approach, since we observed only minor differences in few cases.

[1] In particular, Pellet threw an exception which had its root in the tableau algorithm. This might have been caused by the fact, that the BENCHMARK ontologies are not in the OWL DL nor in the OWL 2 profile according to `http://www.mygrid.org.uk/OWL/Validator` and `http://owl.cs.manchester.ac.uk/validator/`. A description of our implementation as well as a reference to the version of Pellet we used for our experiments can be found in Appendix E.

The results are similar to the results for the anatomy track (see Table 8.4). We remove only a small number of correspondences, however, most of them are incorrect correspondences. This holds for each matching system with the exception of GERMESMB. Overall, we removed 78 correspondences. 60 of them have been incorrect, which results in a debugging precision of 77%. This is - given that the average precision of the alignments is already very high - a very good result. In addition to the number presented in the table, we counted that in total 227 incorrect correspondences have been generated by the matching systems for all of the test cases. Thus, our approach has a debugging recall of 26%. This means we can remove a fourth of all incorrect correspondences by an automized approach that is completely based on logical reasoning in the context of a well defined optimization approach.

Overall we conclude that the good results measured for the CONFERENCE dataset can partially be repeated for other datasets. These datasets describe different domains and differ with respect to their size. The positive impact on the quality of the alignment depends on the expressivity of the ontologies in terms of disjointness axioms. This holds especially for the recall of the debugging, while we measure a high precision of the debugging throughout all our experiments.

8.3 Runtime Efficiency

In this section we analyze the runtime efficiency of our approach. First, we are interested in a general impression of the runtime performance of our approach, e.g., we want to know how much time the algorithms require for different kind of debugging problems. Furthermore, we compare the efficient variant against the brute-force variant of our algorithms. This helps us to understand the benefit of the pattern based reasoning methods in the context of a diagnostic approach.[2] All experiments have been conducted on an Intel Core 2 Duo CPU with 2.26 GHz and 2 GB RAM.

We start with an analysis of the two algorithms for computing a local optimal diagnosis, i.e., we analyze Algorithm 6 (brute-force implementation) and Algorithm 8 (efficient implementation). Results are presented in Table 8.5. In each row we aggregate the results of a specific matcher for one of the three relevant datasets. In particular, we use testcases #301 to #304 of the BENCHMARK dataset, the testcases of the CONFERENCE dataset equipped with a reference alignment, and the two ontologies of the ANATOMY dataset.

For the BENCHMARK and CONFERENCE subsets we present the average values aggregated over all testcases. Runtimes are always presented in seconds in the second and third column. The fourth column compares both runtimes by presenting the factor of the speed-up. It requires, for example, in average 10.5 seconds (16.6

[2]Throughout the previous sections we used the efficient variant of our algorithms to compute the results. Remember that the results generated by the efficient and the brute-force variants are the same. They differ only with respect to their runtime behaviour.

seconds) to process the alignments of AGREEMENTMAKER for one of the four BENCHMARK test cases computing a local optimal diagnosis with the brute-force (efficient) approach. According to the fourth column the efficient algorithm is 1.6 times faster than the brute-force approach.

To better understand under which circumstances Algorithm 8 performs better, we added columns presenting the size of the input alignment $\mathcal{A}$ and the size of the diagnosis Δ. The column captioned with '$k \neq$ NIL' refers to the number of correspondences that have additionally been detected due to complete reasoning techniques. In particular, it displays how often $k =$ NIL is evaluated as false in line 15 of Algorithm 8. This number refers to the conflicts that require full-fledged reasoning to be detected. Finally, we present the fraction of those correspondences that have been detected by efficient reasoning techniques. It is computed by dividing the numbers of the two previous columns.

The grey-colored rows show the results aggregated over all matching systems. With respect to the efficient algorithm the BENCHMARK test cases require in average 4.3 seconds (between 15 and 55 concepts), the CONFERENCE test cases require in average 37.8 seconds (between 36 and 140 concepts), and the anatomy test case requires 376.3 seconds, i.e., $\approx$ 6 min (2744 and 3304 concepts). We could successfully compute the diagnosis for most ontology pairs. However, we encountered problems with Pellet for the IASTED-SIGKDD pair of the CONFERENCE dataset for some of the alignments. We excluded this testcase.

The brute-force approach is approximately 3 times slower for both BENCHMARK and CONFERENCE dataset. This clearly supports the claim that the pattern based reasoning approach can effectively be used to speed up the process. This becomes even more obvious for the ANATOMY test case, where the runtime for the brute-force algorithm explodes. We stopped each debugging process that was still running after one hour. This was the case for all incoherent alignments of the ANATOMY dataset.

The result for GERMESMB requires an additional explanation. Here we have an example for a coherent alignment. The brute-force algorithm is significantly faster than the efficient algorithm. In case of the efficient algorithm (Algorithm 8) we apply the pattern based reasoning techniques first on all pairs of correspondences to pre-compute possible conflicts without checking whether the alignment is already coherent, while for the brute-force algorithm we first check whether the alignment is incoherent. This results for a coherent alignment in a faster runtime.

The left-most columns informs about the completeness of the pattern based reasoning. For the CONFERENCE set it is in average required to start the binary search of the FINDCOHERENCECRACK one time per matching task. Compared to the average size of the diagnosis this means that 75% of all conflicts have been detected sucessfully by the pattern based reasoning methods. We observe a similar result for the BENCHMARK test set with a reduduced absolute number of conflicts. The results for the ANATOMY ontologies differ. Here we observe one case where complete reasoning is required, i.e., the pattern based reasoning approach is nearly complete. Note that in this case (ASMOV) we measured a runtime twice as high

Testcase		Runtime Comparison			Alignment Size & Deleted Correspondences			
	Matcher	Alg.8	Alg.6	Coeff.	$\|\mathcal{A}\|$	$\|\Delta\|$	'$k \neq$ NIL'	Frac.
BENCHMARK (#301-#304)	AGRMAKER	10.5	16.6	1.6	38	2	0.8	63%
	AROMA	4.1	16.4	4	40	1	0.5	50%
	ASMOV	4.7	19.9	4.3	55.3	0.8	0.3	67%
	EF2MATCH	3.5	5.8	1.6	47.3	0.5	0	100%
	FALCON	4.4	5.6	1.3	49.3	0.5	0.3	50%
	GERMESMB	2.5	12.5	5.1	27	0.5	0	100%
	MAPPSO	2.8	15.2	5.5	30.3	1.3	0	100%
	RIMOM	3.5	15.8	4.5	46.8	1	0	100%
	SOBOM	5.3	13.4	2.5	55	2.5	0.5	80%
	TAXOMAP	1.8	6.6	3.6	25.8	7.5	0	100%
	∅	4.3	12.8	3	41.5	1.8	0.2	87%
CONFERENCE	AGRMAKER	27.8	80.5	2.9	19.1	3.1	0.8	74%
	AROMA	21.2	131	6.2	19.6	4.4	0.5	90%
	ASMOV	51.4	192.3	3.7	30.1	3.2	1.6	50%
	EF2MATCH	28.3	88.6	3.1	18.8	2.6	1	64%
	FALCON	30.2	50.8	1.7	14.3	2.5	1.1	55%
	GERMESMB	34	76.2	2.2	22.4	3.2	1	71%
	SOBOM	71.9	157.7	2.2	31.5	9.9	1.3	87%
	∅	37.8	111	2.9	22.2	4.2	1	75%
ANATOMY	AGRMAKER	515.9	>1h	-	1436	3	0	100%
	AROMA	330.1	>1h	-	1156	10	0	100%
	ASMOV	688.7	>1h	-	1469	23	1	96%
	BLOOMS	321.1	>1h	-	1164	5	0	100%
	EF2MATCH	352.3	>1h	-	1243	3	0	100%
	GERMESMB	72	16.5	0.2	528	0	0	100%
	NBJLM	393.3	>1h	-	1327	23	0	100%
	TAXOMAP	336.8	>1h	-	1223	5	0	100%
	∅	376.3	-	-	1193.3	9	0.1	99%

Table 8.5: Runtimes of efficient and brute-force algorithm to compute a local optimal diagnosis.

as for the other incoherent alignments.

The results for computing the global optimal are presented in Table 8.6. It is structured as Table 8.5. There is only one difference namely the column entitled MUPSWALK. It shows the how often Algorithm 10 is forced to call the method MUPSWALK, i.e., how often reasoning in the aligned ontology is required. The values shown in the right-most column are computed based on this value.

Overall, the results are similar to the result we measured for computing the local optimal diagnosis, especially when we compare the efficient against the brute-force approach. The efficient approach is between $\approx$ 1 and 16 times faster then the brute-force approach for BENCHMARK and CONFERENCE test cases. For the ANATOMY ontologies we observe again that the brute-force algorithms does not terminate for the incoherent alignments. Even more, applying both variants of the algorithms on the alignment generated by ASMOV results in an exception thrown by the underlying reasoner. The size of the ANATOMY matching task seems to mark the borders for successfully applying the approach.

A surprising result can be observed when we compare the runtimes of Algorithm 8 (Table 8.5) and Algorithm 10 (Table 8.6), i.e., the efficient algorithms for computing local and global optimal diagnosis. The runtimes of both algorithms vary only slightly. Contrary to this, we expected that computing the global optimal diagnosis is much harder. There are two possible explanations. First, the size of the diagnoses is relatively small. The largest optimal diagnosis occurred for one of the CONFERENCE tasks and contained 17 correspondences. A search tree of a maximal depth of 17 or less can still be handled without problem. We conducted additional experiments and have noticed that the approach runs into problems for a search tree of depth >40, i.e., we encounter problems when the global optimal diagnosis contains 40 or more correspondences.

Second, the global approach expands a node, i.e., an alignment $\mathcal{A}$, by searching for some kind of conflict using the MUPSWALK algorithm. This algorithm requires to check $|\mathcal{A}|$ times the unsatisfiability of a specific class. The local approach makes use of the algorithm FINDCOHERENCECRACK. This algorithm requires to check $log_2(|\mathcal{A}|)$ times whether there exists some unsatisfiable concept. While the local approach requires significantly less reasoning tasks, it is much harder to check whether there exists some unsatisfiable concept compared to checking whether a specific concept is unsatisfiable.

8.4 Conclusions

According to results gathered from experiments conducted with different datasets, we can transform an incoherent alignment into a coherent alignment without loosing relevant parts of the information encoded in the alignment. The debugged alignment can then be used in different scenarios that require the coherence of the alignment. This is by no means a trivial result. It was not clear that our approach would terminate in acceptable time. Nor did we previously know that our

Testcase	Matcher	Runtime Comparison Alg.10	Alg.9	Coeff.	Alignment Size & Deleted Correspondences $\lvert\mathcal{A}\rvert$	$\lvert\Delta\rvert$	MUPSWALK	Frac.
BENCHMARK (#301-#304)	AGRMAKER	10.5	16.6	1.6	38	2	0.8	63%
	AGRMAKER	13	37.7	2.9	38	2	0.8	63%
	AROMA	5.8	8.6	1.5	40	1	0.5	50%
	ASMOV	4.5	6.9	1.5	55.3	0.8	0.3	67%
	EF2MATCH	3.1	2.8	0.9	47.3	0.5	0	100%
	FALCON	5.3	5.4	1	49.3	0.5	0.3	50%
	GERMESMB	2	4.7	2.4	27	0.5	0	100%
	MAPPSO	2.6	19.5	7.4	30.3	1.3	0	100%
	RIMOM	3.2	9.3	2.9	46.8	1	0	100%
	SOBOM	7.1	27.5	3.9	55	2.5	0.5	80%
	TAXOMAP	1.6	25.2	16.1	25.8	6.5	0	100%
	⌀	4.8	14.8	3.1	41.5	1.7	0.2	86%
CONFERENCE	AGRMAKER	18.6	38.3	2.1	19.1	2.9	1	63%
	AROMA	34.5	82.2	2.4	20.2	3.9	0.9	77%
	ASMOV	33.1	49.7	1.5	30.1	3	2	33%
	EF2MATCH	18.9	42.9	2.3	18.8	2.1	0.9	58%
	FALCON	14.4	19.5	1.3	14.3	1.9	0.9	51%
	GERMESMB	29.3	47.5	1.6	22.4	3	1.7	45%
	SOBOM	59.1	221.4	3.7	31.5	8.1	3	63%
	⌀	29.7	71.6	2.4	22.3	3.6	1.5	58%
ANATOMY	AGRMAKER	503.8	>1h	-	1436	3	0	100%
	AROMA	351.5	>1h	-	1156	5	0	100%
	ASMOV	-	>1h	-	1469	-	-	-
	BLOOMS	330.1	>1h	-	1164	5	0	100%
	EF2MATCH	356.7	>1h	-	1243	3	0	100%
	GERMESMB	69.3	17.6	-	528	0	0	100%
	NBJLM	425	>1h		1327	11	0	100%
	⌀	282.9	-	-	984.6	3.4	0	100%

Table 8.6: Runtimes of efficient and brute-force algorithm to compute a global optimal diagnosis.

algorithms would not remove a large fraction of correct correspondences from the alignment.

With respect to research question R5 we conclude that both methods (local and global) increase the quality of the alignment in terms of its f-measure in average. We observe a win in precision and a smaller loss in recall as general tendency. The results of applying local and global optimal diagnosis differ sometimes slightly and in few cases significantly. The global approach produces less outliers. Incorrect decisions that might result in series of removals are avoided. As a consequence, the results of applying the global optimal diagnosis are slightly better in average. Overall, we have successfully applied our method in a post-processing step to increase the quality of an alignment. Note, in particular, that our approach requires no configuration. It is completely independent of any specific parameter setting.

The positive impact of a diagnostic debugging depends critically on the expressivity in terms of negation used in the axioms of the ontologies that have to be aligned. This holds for the number of incorrect correspondences that can be detected by the method (= recall of debugging). The precision of the debugging is relatively high throughout our experiments. It seems not to be affected negatively by missing disjointness statement. We measured this result across different datasets.

We also reported about measuring runtimes of our algorithms, i.e., we were concerned with research question R6. One of our main insights is that the pattern based reasoning method reduces the runtime of the algorithms significantly. In particular, the efficient variant of our algorithms speeds up the debugging process for large matching tasks significantly while we find a moderate increase by a factor ≈ 3 for smaller, more expressive ontologies. Moreover, the efficient variants of our algorithms scale well with respect to size of ontologies and alignments. Even for large debugging tasks – in a setting where the brute-force approach fails or does not terminate within an acceptable time limit – we can successfully compute a local and in some cases also a global optimal diagnosis.

Chapter 9

Alignment Extraction

I chose and my world was shaken. So what? The choice may have been mistaken; the choosing was not. You have to move on (Stephen Sondheim).

Any matching system will generate during the matching process a set of hypotheses, which have to be accepted or rejected in a final step. Based on these decisions the final alignment $\mathcal{A}$ is generated by the matching system. Whenever we compute a diagnosis Δ, again, we treat $\mathcal{A}$ as a set of hypotheses and extract a subset $\mathcal{A} \setminus \Delta$ as final outcome. Thus, it is self-evident to combine the final step of the matching system with our diagnostic approach. We refer to this final step under the notion of alignment extraction. In the previous section, we analyzed the results of a subsequent approach. First the matching systems performs its own extraction, then we extract a coherent alignment from the result of the first extraction. In this chapter we investigate the effects of a combined approach in two different scenarios.

First of all, we start with a motivating example in Section 9.1. This example will help us to understand the difference between a subsequent and a combined approach in detail. In Section 9.2 we present some experiments where we used a rudimentary matching system to compare the subsequent against the combined approach in different settings. A special case of the extraction problem is the problem of extracting an alignment from a set of alignments generated by different matching systems. This setting is subject of Section 9.3. Finally, we summarize the results of our experiments in Section 9.4.

Some of the issues analyzed in Section 9.1 and Section 9.2 have already been touched in [MS07a]. However, there we used only a restricted dataset in a different setting. In 2010 we developed a matching system, which uses Markov Logic Networks to compute the global optimal diagnosis [NMS10]. This system is an example that shows how to integrate logical reasoning in a similar way as proposed

	Review$_{\#1}$	*ExternalReviewer*$_{\#1}$	*Document*$_{\#1}$
Reviewer$_{\#2}$	0.75	0.5	~~0.0~~
PaperReview$_{\#2}$	0.55	0.5	~~0.09~~
Document$_{\#2}$	~~0.13~~	~~0.06~~	1.0

Table 9.1: Example of a 3×3 similarity matrix.

here [EFM+10, NN10]. Experimental results of applying the approach in the context of merging several alignments (Section 9.3) have not yet been published.

9.1 Motivating Example

In the following we refer to a matching hypothesis as a correspondence that might possibly be part of the final alignment. In many systems matching hypotheses are stored during the matching process as a cell in a similarity matrix (for example PRIOR+ [MPS10] or LILY [WX08b]). A similarity matrix is a data structure that stores for each pair of matchable entities, i.e., for each correspondence $\langle X_{\#1}, Y_{\#2}, \equiv\rangle$, a similarity score $\alpha(X_{\#1}, Y_{\#2})$. Our approach is applicable to any set of matching hypotheses independently from their internal generation or storage, however, a similarity matrix helps us to illustrate a realistic example.

Table 9.1 shows a similarity matrix or at least a part from a complete similarity matrix. Actually, it is a similarity matrix that might be the intermediate result of matching the ontologies depicted in Figure 1.2 in the introduction of this thesis. The first row and the first column display the concepts of $\mathcal{O}_1$ and $\mathcal{O}_2$, while the inner cells show the value of a similarity distribution α. These values have been computed with the Levenshtein distance. In particular, $\alpha(X_{\#1}, Y_{\#2})$ is defined as $1 - l(x, y)$ where l is the Levenshtein distance and x,y are the local names of $X_{\#1}$ and $Y_{\#2}$. Let us assume in the following that correspondences $\langle \mathit{Review}_{\#1}, \mathit{PaperReview}_{\#2}, \equiv\rangle$ and $\langle \mathit{Document}_{\#1}, \mathit{Document}_{\#2}, \equiv\rangle$ are correct (grey cells), while the other correspondences are incorrect.

Given this similarity matrix, the set of matching hypotheses needs to be reduced to arrive at the final alignment. According to [ES07], a typical approach is the application of a threshold. In our example we apply threshold 0.5 and remove all correspondences c with $\alpha(c) < 0.5$. We have marked the cells of the removed correspondences by striking out the similarity score. However, we still have different alternatives to match some of the concepts. Matching the same concept $X_{\#1}$ on two concepts $Y_{\#2}$ and $Z_{\#2}$ allows us to conclude that $Y_{\#2}$ and $Z_{\#2}$ are equivalent even though we might not be able to entail the equivalence in $\mathcal{O}_2$. For that reason it makes sense to require that the final alignment $\mathcal{A}$ is a 1:1 alignment. The property of being a 1:1 alignment can be defined as follows.

Definition 33. *Given an alignment $\mathcal{A}$ that contains only equivalence correspondences. $\mathcal{A}$ is a 1:1 alignment iff for each $\langle a, b, \equiv\rangle \in \mathcal{A}$ there exists no $\langle c, d, \equiv\rangle \in \mathcal{A}$ with $a = c \wedge b \neq d$ or $a \neq c \wedge b = d$.*

A standard extraction method used by a matching system is a sequence of (a) first applying a threshold followed by (b) the extraction of a 1:1 alignment. Obviously, there are several 1:1 alignments contained in the thresholded similarity matrix. The most promising alignment is the 1:1 alignment with highest total of similarity values. The Hungarian Method can be used to compute this alignment [Kuh55]. For our example the optimal 1:1 alignment within the thresholded set of matching hypotheses consists of the following correspondences.

$$\langle \mathit{Review}_{\#1}, \mathit{Reviewer}_{\#2}, \equiv \rangle$$
$$\left\langle \mathit{ExternalReviewer}_{\#1}, \mathit{PaperReview}_{\#2}, \equiv \right\rangle$$
$$\langle \mathit{Document}_{\#1}, \mathit{Document}_{\#2}, \equiv \rangle$$

As a third step (c) we apply our approach of computing a diagnosis to find a coherent subset within the set of these correspondences. The debugging step requires to take into account the axioms of the ontologies to be aligned. We refer the reader back to the introduction where we informally described the relations between the concepts of these ontologies. We skip over a formal description and conclude that there are two MIPS $\mathcal{M}_1$ and $\mathcal{M}_2$ in this alignment.

$$\begin{aligned}
\mathcal{M}_1 &= \{\langle \mathit{Document}_{\#1}, \mathit{Document}_{\#2}, \equiv \rangle, \langle \mathit{Review}_{\#1}, \mathit{Reviewer}_{\#2}, \equiv \rangle\} \\
\mathcal{M}_2 &= \{\langle \mathit{Document}_{\#1}, \mathit{Document}_{\#2}, \equiv \rangle, \\
&\quad \left\langle \mathit{ExternalReviewer}_{\#1}, \mathit{PaperReview}_{\#2}, \equiv \right\rangle\}
\end{aligned}$$

The removal of the global optimal diagnosis Δ_{global} results in the following set of correspondences.

$$\begin{aligned}
\mathcal{A} \setminus \Delta_{global} &= \{\langle \mathit{Review}_{\#1}, \mathit{Reviewer}_{\#2}, \equiv \rangle, \\
&\quad \left\langle \mathit{ExternalReviewer}_{\#1}, \mathit{PaperReview}_{\#2}, \equiv \right\rangle\}
\end{aligned}$$

This is obviously a bad solution to the extraction problem. Applying the diagnostic approach results in an alignment that contains no correct correspondence at all.

What happens if we compute the global optimal diagnosis and the optimal extraction of the 1:1 alignment at the same time? With a minor modification we can force our algorithm to generate a global optimal diagnosis that is at the same time a 1:1 alignment. It is easy to add this extension as part of the pattern based reasoning during the preprocessing step. In particular, a pair of correspondences that violates the 1:1 constraint can be formalized as an additional pattern, similar to the propagation patterns we have introduced in Section 5.2. The main part of Algorithm 10 requires no further modifications.

This approach - applied to the thresholded matrix as input alignment – results in a different extraction compared to the sequential approach. In particular, we have

$$\mathcal{A} \setminus \Delta_{global \wedge 1:1} = \{ \langle \mathit{ExternalReviewer}_{\#1}, \mathit{Reviewer}_{\#2}, \equiv \rangle , \\ \langle \mathit{Review}_{\#1}, \mathit{PaperReview}_{\#2}, \equiv \rangle , \langle \mathit{Document}_{\#1}, \mathit{Document}_{\#2}, \equiv \rangle \}.$$

This result is better than the result of the standard approach (no diagnosis) and also better than the sequential approach. Table 9.1 gives an overview on the results. In particular, we increase not only the precision but also the recall of the final alignment compared to the outcome of the standard approach that uses no diagnostic elements. A diagnostic approach can thus be used as a component that is more than just a pure logical filter, which eliminates some probably incorrect correspondences.

Extraction method	$p(\mathcal{A}^*, \mathcal{R})$	$r(\mathcal{A}^*, \mathcal{R})$	$f(\mathcal{A}^*, \mathcal{R})$
$\text{threshold}_{0.5} \rightarrow$ 1:1	0.33	0.5	0.4
$\text{threshold}_{0.5} \rightarrow$ 1:1 $\rightarrow \Delta_{global}$	0.0	0.0	0.0
$\text{threshold}_{0.5} \rightarrow \Delta_{global \wedge 1:1}$	0.66	1.0	0.8

Table 9.2: Varying results of different extraction methods.

Integrating the diagnostic approach into the extraction method of a matching system will not always have the strong positive impact indicated by our example. Nevertheless, we have seen that a combined approach can increase both precision and recall. For that reason we have to analyze whether the positive impact of the combined approach is stronger than the impact of repairing an incoherent alignment in a subsequent step. The experimental results presented in the next section show in how far our considerations are valid.

9.2 Extracting from a Similarity Matrix

The approach of the previous section is the basis for the experiments reported in this section. We have developed a simple matcher that allows us to have full control on the extraction process. This matcher compares the class and property names of the ontologies to be aligned by using the Levenshtein distance. On top of the matcher we compare the different extraction methods presented in the previous section. We use the following naming convention to refer to these methods.

t $\rightarrow$ **1:1** - A standard approach of extracting an alignment from a similarity matrix. First a threshold t is applied and then a 1:1 alignment is extracted from the remaining correspondences. In particular, we extract an 1:1 alignment that is optimal with respect to its sum of confidences.

t $\rightarrow$ **1:1** $\rightarrow \Delta$ - The standard approach is extended by a subsequent repairing step in which we compute a global optimal diagnosis. Thus, the 1:1 extraction is independent of the subsequent diagnostic approach.

t $\rightarrow \Delta_{1:1}$ - After applying a threshold, the optimal 1:1 extraction method is combined with the diagnostic approach to compute a global optimal diagnosis in a single step.

We restrict our experiments to the CONFERENCE dataset. Note that our minimal matching system generates acceptable results for the CONFERENCE dataset, but cannot compete with the other systems on the other tracks. Moreover, we focus only on the global optimal diagnosis. The results of the previous chapter have shown that the global optimal diagnosis is, with respect to the quality of the repaired/extracted alignment, the better choice than the local optimal diagnosis,

The results of our experiments are presented in Table 9.2. In the first row we listed the threshold t that we applied prior to any other extraction method. Then there are three blocks that contain the results for each of the extraction methods. Each block comprises three columns headed with letters p, f, and r that refer to precision, f-measure, and recall. In the two rightmost columns we present the difference between the first method (t $\rightarrow$ 1:1) and the second method (t $\rightarrow$ 1:1 $\rightarrow$ Δ), and the difference between the second method (t $\rightarrow$ 1:1 $\rightarrow$ Δ) and the third method (t $\rightarrow \Delta_{1:1}$) in terms of f-measure. In the last row we show the average scores over all thresholds.

First of all, the results of repairing our simple matching system conforms with the results we measured in the previous chapter. Repairing the alignments of our simple matching system yields similar results like repairing the alignments of an OAEI participant. In average we can increase the f-measure of the 1:1 alignment by 0.018. In the worst case (highest threshold) we win 0.009 in f-measure and in the best case (lowest threshold) we gain 0.035. There are only a few exceptions from a general trend: the lower the threshold the higher the win in f-measure.

Our main interest with respect to these results, however, is related to the differences between the subsequent and the combined approach. The relevant differences in terms of f-measure are depicted in the last column. Contrary to our expectations, there are only minor differences that do not imply a general tendency. In some cases we do not observe any differences, sometimes results are slightly worse and sometimes slightly better. We cannot conclude that the f-measure can be improved with the combined approach.

However, a tendency can be observed when we directly compare precision and recall of both approaches. The combined approach slightly increases recall but decreases the precision of the alignments at the same time. This is also illustrated in Figure 9.1. It depicts the precision/recall value pairs of both approaches in a precision/recall graph. We can see that the red curve ($\Delta_{1:1}$) sits above the black curve (1:1 $\rightarrow$ Δ) for many different thresholds. However, there is also an offset to the left, which illustrates the lower precision.

Another interesting aspect is related to the relation between threshold and recall. Recall values for high thresholds can be increased only to a very limited degree (from 0.447 to 0.52) by decreasing the threshold. A top-score of 0.52 for recall is reached at a threshold of 0.7 for the combined $\Delta_{1:1}$-approach. Note that

	t → 1:1			t → 1:1 → Δ			t → $\Delta_{1:1}$			+/- f-measure	
t	p	f	r	p	f	r	p	f	r	repairing	rep vs. ext
0.625	0.471	0.483	0.497	0.549	0.518	0.49	0.526	0.511	0.497	0.035	-0.007
0.65	0.513	0.513	0.513	0.585	0.543	0.507	0.567	0.537	0.51	0.03	-0.006
0.675	0.551	0.53	0.51	0.622	0.559	0.507	0.601	0.55	0.507	0.029	-0.009
0.7	0.592	0.551	0.516	0.665	0.579	0.513	0.654	0.579	0.52	0.028	0
0.725	0.623	0.557	0.503	0.699	0.583	0.5	0.688	0.581	0.503	0.026	-0.002
0.75	0.679	0.569	0.49	0.738	0.587	0.487	0.737	0.591	0.493	0.017	0.004
0.775	0.686	0.574	0.493	0.743	0.591	0.49	0.744	0.593	0.493	0.016	0.003
0.8	0.702	0.58	0.493	0.761	0.596	0.49	0.763	0.599	0.493	0.017	0.003
0.825	0.745	0.589	0.487	0.787	0.599	0.484	0.789	0.605	0.49	0.01	0.006
0.85	0.759	0.591	0.484	0.804	0.604	0.484	0.801	0.606	0.487	0.013	0.002
0.875	0.778	0.594	0.48	0.812	0.604	0.48	0.808	0.602	0.48	0.01	-0.001
0.9	0.786	0.596	0.48	0.821	0.606	0.48	0.817	0.605	0.48	0.01	-0.001
0.925	0.79	0.587	0.467	0.822	0.596	0.467	0.817	0.595	0.467	0.009	-0.001
0.95	0.79	0.587	0.467	0.822	0.596	0.467	0.817	0.595	0.467	0.009	-0.001
∅	0.676	0.564	0.492	0.731	0.583	0.489	0.723	0.582	0.492	0.018	-0.001

Table 9.3: Extracting from a similarity matrix. The column entitled *'repairing'* refers to the difference in f-measure between $t \rightarrow$ 1:1 and t → 1:1 → Δ, the column entitled *'rep vs. ext'* refers to the difference between the sequential approach of first extracting an 1:1 alignment that is repaired afterwards and the approach of combining 1:1 extraction and resolving incoherence in one step (i.e., it compares t → 1:1 → Δ against t → $\Delta_{1:1}$).

it is not possible for our simple matching system to exceed a certain degree of recall, without loosing a significant degree of precision. Before applying one of our extraction methods, directly after applying a threshold of 0.625, we have a precision of 0.26 and a recall of 0.529. This is also the upper bound for any extraction method that is applied to the set of hypotheses. A recall of 0.52 is thus a good result and shows that some of the effects described in the example of the previous section occur also for real matching problems. However, their impact is only limited.

We can conclude that there are only minor differences between the sequential and the combined approach. Both approaches increase the quality of the alignment in terms of its f-measure to a similar degree. This differs from our expectations. With respect to recall we can observe a tendency. The combined approach results in an increased recall and a decreased precision. However, the results comply with our expectations only to a limited degree. The following two reasons have to be taken into account.

1. The simple string-based similarity measure, which is the basis of our matcher, cannot exceed a certain upper bound for recall. For that purpose an approach

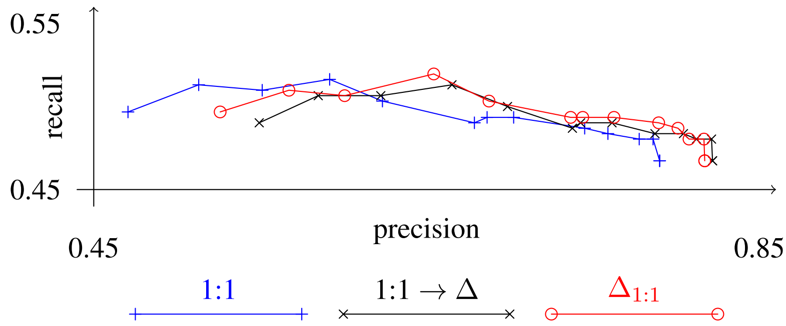

Figure 9.1: Precision and recall when extracting from a similarity matrix.

that aims at an increased recall is of limited benefit.

2. Our matcher generates a confidence distribution α that is based on only one source of evidence. As a result we have many correspondences with the same confidence value and a confidence distribution that is, overall, not reliable.

Both reasons are related to the matching system we used for test purposes. In the next section we investigate a different setting, in which a combined approach might be more useful.

9.3 Extracting from a Merged Alignment

The results of the previous section have only partially and to a limited degree supported our theoretical considerations. One of the main problems has been caused by the limited recall of our experimental matching system. In this section we discuss another scenario that does not suffer from this problem. It is the scenario of merging the results of different matching systems.

Suppose n matching systems have been applied to generate an alignment for the same matching problem. As a result we have the alignments $\mathcal{A}_1, \ldots, \mathcal{A}_n$ and their confidence allocations $\alpha_1, \ldots, \alpha_n$. How do we choose a subset from $\bigcup_{i=1\ldots n} \mathcal{A}_i$ that is a good choice in terms of precision and recall? This problems is normally dealt with in the context of machine learning approaches for combining different similarity measures [Ich08] or in the context of argumentation frameworks [TMQV08]. The latter approach is discussed in detail in the chapter on related work (Section 11.2).

To apply our method we can treat the union of the alignments $\mathcal{A}_1, \ldots, \mathcal{A}_n$ as a single set of matching hypotheses $\mathcal{A} = \bigcup_{i=1\ldots n} \mathcal{A}_i$ that is annotated with a unique confidence distribution α. There are several ways to compute α from $\alpha_1, \ldots, \alpha_n$. We use the following simple method. We first normalize each α_i by spanning its values to the range $(0, 1]$ with $\alpha_i(c) = 0$ iff $c \notin \mathcal{A}_i$. The normalization is required to ensure that each matching system contributes to the final confidence distribution

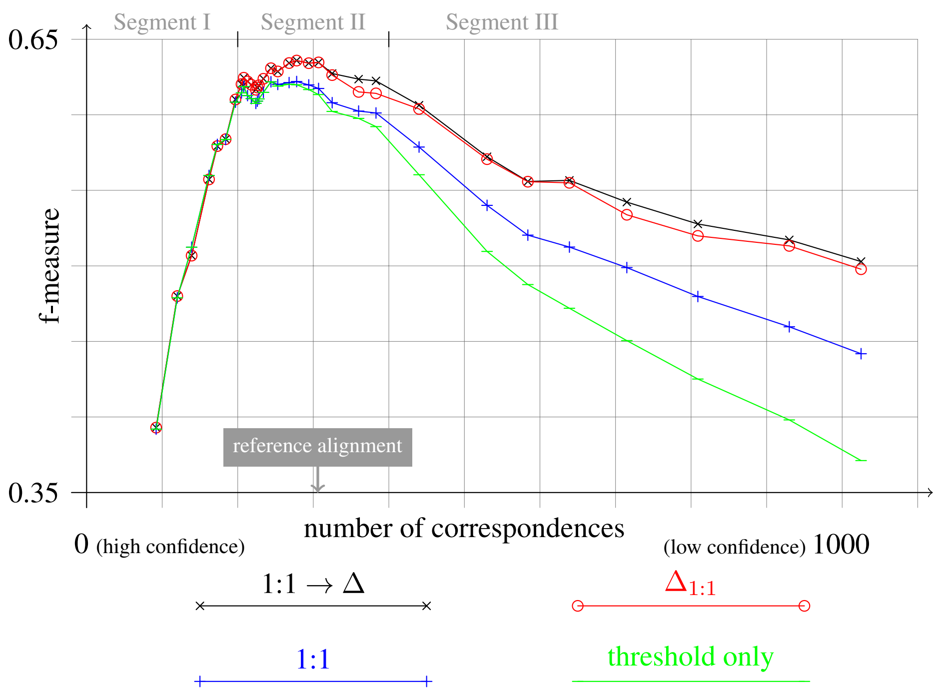

Figure 9.2: Results of extracting from a merged alignment for different thresholds.

in the same way. Then we introduce α as the sum of all specific confidence values, i.e., $\alpha(c) = \sum_{i=1,\dots n} \alpha_i(c)$ for each $c \in \bigcup_{i=1\dots n} \mathcal{A}_i$.

As a result we have an alignment $\mathcal{A}$ with high recall and low precision. The confidence values of this alignment are computed by the use of very different sources of evidence. It is relatively improbable that two correspondences have the same confidence value. Furthermore, it can be expected that $\mathcal{A}$ is not a 1:1 alignment, even though most of $\mathcal{A}_1, \dots, \mathcal{A}_n$ might have been 1:1 alignments. The merged alignment can also be expected to be highly incoherent. These characteristics are well suited for applying a diagnostic approach to extract the final alignment.

In the following we report on our experiments with the CONFERENCE dataset. For this dataset we took the alignments submitted to the OAEI 2010 and computed their union as described above. We did not include the alignments generated by CODI [NN10], because it uses techniques we developed in accordance to some of the ideas presented in this thesis. We omitted CODI to avoid any kind of unintended interference. Aside from CODI, seven systems participated, namely AGREEMENTMAKER [CSC$^+$10], AROMA [Dav08], ASMOV [JMSK09], EFF2MAT [CK10], FALCON [HQ08], GEROME [QGKL08], and SOBOM [XWCZ10]. Thus, we had a rich set of input alignments.

The raw results of our experiments are presented in detail in Table D in Appendix D. We present a condensed graphical presentation of the main results in

Figure 9.2 and later on in Figure 9.3. Figure 9.2 shows the f-measure for all of our three extraction methods as well as the f-measure of the thresholded input alignment $\mathcal{A}_t$. We applied different thresholds to $\mathcal{A}$ before we conducted one of the three extraction methods. This resulted in a large variance with respect to the size of the input alignments (between 64 and 1025 correspondences in 21 alignments). In the same way as in the previous section we distinguished between a simple threshold (green line, $-$ marker), the optimal one-to-one extraction (1:1, blue line, $+$ marker), the optimal one-to-one extraction followed by computing a global optimal diagnosis (1:1 $\rightarrow \Delta$, black line, $\times$ marker), and the combined approach ($\Delta_{1:1}$, red line, $\circ$ marker).

We have divided the diagram in three segments regarding the size of the input alignment and the resulting characteristics. All of the different approaches start with a low f-measure in Segment I. The alignments are highly precise and have limited recall. Then, in a range from approximately 200 to 400 correspondences, they reach the top values for f-measure (Segment II). Note that the union of the reference alignments consists of 306 correspondences. With an increasing size the quality of the alignments decreases slowly in Segment III.

This general tendency had to be expected. In Segment I there are only minor differences between the four curves. Obviously, there is only a limited number of coherence conflicts contained in the small alignments. In Segment II, there is small though significant difference between our approaches and the baselines. With both the sequential and the combined approach we reach an f-measure of 0.636, the top f-measure for the optimal 1:1 extraction and the thresholded input alignment is 0.622. A similar offset can be found in most parts of Segment II.

In Segment III we see that the offset becomes more and more significant. Both the sequential and the combined approach start with $+1.5\%$ advantage over the optimal 1:1 extraction and reach up to $+5\%$ when 1000 correspondences are reached. The optimal 1:1 extraction behaves similar compared to the thresholded input alignment. It follows that our reasoning based approach is also well suited to increase the quality of alignments that aim at high recall values. Moreover, the approach is robust against misconfigurations that result in too large alignments.

These results support the claim that the reasoning-based approach can increase the quality of an alignment. However, they do not support our hypotheses about the effects of a combined approach, i.e., that it is more effective to combine standard extraction methods and the method for resolving incoherence in one step instead of applying two subsequent steps. There are only minor differences between the red (marked by a dot) and the black line (marked by x). Moreover, the combined approach is sometimes slightly worse compared to the sequential approach as long as we focus on the f-measure. We conclude that the effectiveness of the general approach has been proved in a different application scenario, however, we have not shown that a combined approach has an advantage over the sequential approach.

It can be argued that the technique to combine the alignments is not well chosen and that it is not hard to exceed our baselines. Note that we have developed and analyzed a method similar to our baselines in [EMS09]. There we found that this simple method is nearly as good as machine learning approaches, which require parts of the reference alignment to be available as training examples. Moreover, we will see in the following that even our baseline – the aggregation method combined with an optimal 1:1 extraction – clearly outperforms most matching systems.

We have additionally added Figure 9.3 to enable a better understanding of the results. The triangular view on the results represent both precision and recall in the same graph, i.e., each precision/recall pair of values is depicted as a point in the graph.[1] The distance to the left end of the x-axis determines the precision of the alignment, i.e., a point on the arc of the circle on the right side has a precision of 1.0. Analogous, the distance to the right end of the x-axis determines the recall of the alignment. It follows that the point that refers to the reference alignment $\mathcal{R}$ can be found on at the top where both arcs meet. In addition to some helper lines we have also added curves for all combinations of precision/recall values that result in an f-measure of 0.5 and 0.6, respectively.

In this figure we depict the results shown in Figure 9.2 from a different perspective. We included the precision/recall scores for all of the 2010 submissions, i.e., the alignments that we used as input to our aggregation. In particular, we present the results of both the originally submitted alignments as well as the alignments after applying an optimal a posteriori threshold (indicated by a t-subscript). Note that nearly all of the points (p, r) that refer to an input alignment (original or thresholded) are *'surrounded by the curves'* that refer to the simple baselines. This means that for each matcher generated (p, r) even the simple base lines result in precision-recall pairs (p', r') such that $p' > p$ with $r' = r$ or $r' > r$ with $p' = p$. There are only two exceptions, both of them in an area that is below an f-measure of 0.6.

We observe again that our reasoning based approach outperforms the optimal 1:1 extraction for many constellations. Especially, in the area that corresponds to Segment II we see that the red/black curve is constantly above the green and blue curves. Note also that this is the area, that results in an f-measure ≥ 0.6. This top score can be reached with different precision/recall characteristics. This is an important insight, because it illustrates that the good results in term of f-measure visualized in Figure 9.2 are not just based on a restrictive filtering, but that both the combined and the sequential approach adapt to the input alignments and can also be used to generate alignments with relatively high recall.

We also observe some differences between the $\Delta_{1:1}$ and the 1:1 $\rightarrow \Delta$ extraction. For some thresholds the combined approach favors recall over precision. However, there are also thresholds in which the opposite is the case. Overall, the results are again inconclusive and do not support our hypothesis about the positive impact of a combined approach.

[1] The use of the triangular representation was inspired by its regular usage in the OAEI BENCHMARK results presentation. The draft of the latex-code underlying the figure has been generated by the Alignment API [DESdS11].

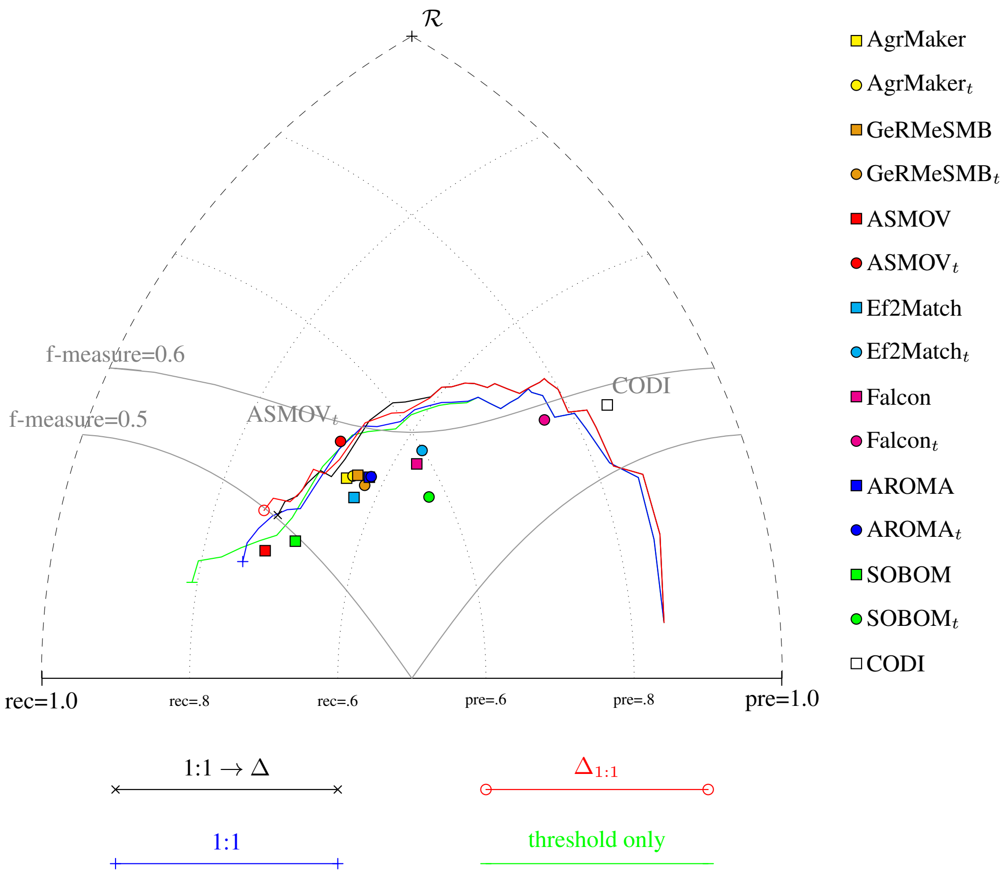

Figure 9.3: Precision/recall triangular.

9.4 Conclusions

In this chapter we have analyzed the effects of applying our reasoning-based approach in a different application scenario. This is the scenario of extracting an alignment from a set of matching hypotheses. We were thus again concerned with research question R5, but analyzed it in the context of a different scenario. From a logical point of view the problem of extracting an alignment does not differ from the problem of repairing an alignment. However, an automatically generated alignment is already extracted from a set of hypotheses. Many matching systems are, for example, pre-configured to generate 1:1 alignments. The output of a matching system is thus already a reduced set of matching hypotheses.

In this chapter we analyzed in how far it is possible to combine our approach with a standard extraction method. In particular, we were interested in differences between a sequential approach and a combined approach. Due to our theoretical considerations we expected that the combined approach results in better alignments, in particular, we expected a higher recall. We conducted two sets of experiments to prove our expectations. First, we analyzed the effects of a combined approach on top of a very simple and limited matching system (Section 9.2). Second, we discussed the scenario of merging alignments generated by different matching systems (Section 9.3).

Our results have to be evaluated from different perspectives. First of all, we could not support our claim that a combined approach performs better than a sequential approach. The results were inconclusive in both of the scenarios we analyzed. We did not observe a general tendency in favor of the combined approach, especially when we focus on the f-measure. At least for the experiments reported in Section 9.2 the results with highest recall have been generated by the combined extraction. However, these results are not sufficient to support our hypotheses. We can only conclude that some of the effects explained at hand of the example in Section 9.1 occur in concrete matching tasks. However, these effects are not strong enough to increase the overall quality of the alignments in average, nor is the impact of these effects always positive.

With respect to our original research question our experiments helped us to gain additional insights in the behaviour of our algorithms. First, we have seen that the algorithm for computing a global optimal diagnosis works well in different settings. This holds for both the sequential and combined approach. We have measured good results when using a simple string similarity α as well as for the case where we computed α as an aggregation of confidence values. We have also seen that the behaviour of the algorithm depends on the size of the input alignment. Applied on a highly precise set of matching hypotheses, the algorithm has no effects or a rather restricted impact on the quality of the alignments. The more hypotheses we add, the stronger are the effects compared to a standard approach of extracting an alignment from the set of hypothesis.

Our approach can also increase the quality of a very good alignment that has been generated with an optimal threshold. This has in particular become clear in the context of merging alignments generated by different matching systems. Without using any additional information about the quality of the input alignments, we were able to increase the f-measure above the f-measure of the best system. To our knowledge their exists no approach that generates results as good as our results in a similar setting. Note that our approach does not require training examples nor does it depend on any kind of knowledge related to the quality of the input alignments.

Chapter 10

Manual Alignment Revision

The real problem is not whether machines think but whether men do (Burrhus Frederic Skinner).

In the previous chapters we focused on fully automated application scenarios. We measured the degree of alignment incoherence, debugged incoherent alignments, and extracted subsets from incoherent sets of matching hypothesis. In this chapter we focus on the role of a human in the alignment process. In particular, we analyze in how far our approach is capable of supporting a user in selecting a subset of correct correspondences from an alignment. We refer to this process in the following as alignment revision.

All of the evaluations conducted by the OAEI over the last years have shown that there exists no perfect matching system [EMS$^+$11]. Whenever a complete and correct alignment is required, it is thus inevitable that the alignment is revised by a human person. The algorithms presented so far guarantee alignment coherence and can increase the precision of the alignment, however, it is by far not guaranteed that the alignment contains only correct correspondences after applying these algorithms. Moreover, conflicts are sometimes resolved in an awkward way, i.e., our algorithms eliminate correct correspondence and accept incorrect correspondences.

The same will not (or only rarely) happen when we replace the decision component in our algorithms by a human user. The user can profit from reasoning support in two respects. First, the consequences of a decision can be displayed to the user in order to point to dependencies between correspondences that might not become explicit otherwise. Suppose for example that a user wants to accept two correspondences that conflict with each other. Given this information the user has to reconsider his decision. Second, the user can save time whenever he is sure about one of its decisions. Suppose that a user accepts a correspondence that conflicts with a set of other correspondences. The conflicting correspondences can be eliminated without any additional effort. In the following we present some examples and experiments that illustrate the benefits of such an approach.

In Section 10.1 we introduce some definitions to describe the revision process in a formal way. In Section 10.2 we present a tool that we developed to support the revision process. By presenting an example, we highlight the benefit of making dependencies between correspondences explicit. In Section 10.3 we measure in how far our approach can be used to reduce the manual effort of the revision process. Finally, we present a conclusion in Section 10.4 and get back to the research questions presented in the introduction.

We have published some parts of the following three sections in [MST08, MST09, MSSZ09]. The theoretical framework that we present in Section 10.1 has been developed in [MST08]. Our reasoning-based approach was first based on DDL; in this thesis we present its application on the natural DL-based semantics. The examples and screenshots presented in Section 10.2 have been partially taken from our demonstration of a web-based tool for alignment revision [MSSZ09]. The experiments presented in Section 10.3 make use of an evaluation approach that we first applied in [MST08, MST09].

10.1 Preliminaries

Given an incoherent alignment $\mathcal{A}$, a revision process can be modeled as a series of decisions. The expert has to choose for each $c \in \mathcal{A}$ between one of $correct$ and $incorrect$. Further, we suppose that by default each correspondence is evaluated implicitly as $unknown$ as long as no positive or negative decision is available. Thus, for each point in time the current status of the revision process can be modeled as a function e that assigns to each correspondence of an alignment a value from the set $\{correct, incorrect, unknown\}$. The following definition formally introduces the notion of an evaluation as a function of this type.

Definition 34 (Evaluation)**.** *An evaluation* $e : \mathcal{A} \rightarrow \{correct, incorrect, unknown\}$ *is defined by*

$$e(c) \mapsto \begin{cases} correct & \text{if } c \text{ is accepted} \\ incorrect & \text{if } c \text{ is rejected} \\ unknown & \text{otherwise} \end{cases} \quad \text{for all } c \in \mathcal{A}$$

Furthermore, let $e(\mathcal{A}, v) \subseteq \mathcal{A}$ *be defined as* $e(\mathcal{A}, v) = \{c \in \mathcal{A} | e(c) = v\}$ *for all* $v \in \{correct, incorrect, unknown\}$.

In an unsupported revision process each correspondence has to be analyzed in one step of an iterative process. With each decision the domain expert moves forward from an evaluation e to an evaluation e' such that e' assigns $incorrect$ or $correct$ to a correspondence previously set to $unknown$. At the end of the process we have $e(\mathcal{A}, unknown) = \emptyset$. In the following we define a (direct) successor evaluation e' of e as an evaluation that follows (directly) on e within a revision process.

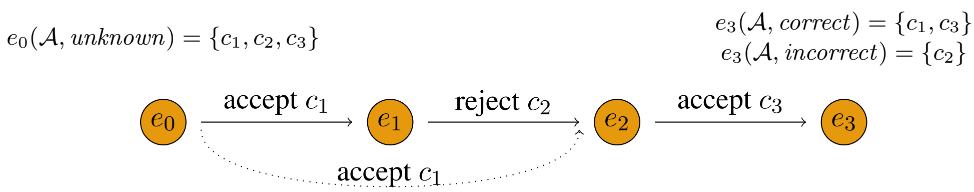

Figure 10.1: Revision process as sequence of evaluations.

Definition 35 (Successor evaluation)**.** *Given an evaluation* e*, an evaluation* e' *is called a successor of* e *iff* $e(\mathcal{A}, \mathit{correct}) \subseteq e'(\mathcal{A}, \mathit{correct})$*,* $e(\mathcal{A}, \mathit{incorrect}) \subseteq e'(\mathcal{A}, \mathit{incorrect})$ *and* $e(\mathcal{A}, \mathit{unknown}) \supset e'(\mathcal{A}, \mathit{unknown})$*. A successor* e' *of* e *is a direct successor iff* $|e(\mathcal{A}, \mathit{unknown})| - 1 = |e'(\mathcal{A}, \mathit{unknown})|$*.*

In Figure 10.1 we illustrate a revision process for an alignment with three correspondences c_1, c_2, and c_3. The standard workflow for revising the alignment is depicted by solid arrows. Obviously, three decision are required by the user until every correspondence has been evaluated, i.e., the process consists of a sequence of four evaluations. The dotted arrow illustrates what happens, if we support the user in his task. Suppose that $\{c_1, c_2\}$ is a MIPS in $\mathcal{A}$ with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$. Then the acceptance of c_1 allows us to automatically reject c_2, because every $\mathcal{A}' \subseteq \mathcal{A}$ with $c_1, c_2 \in \mathcal{A}'$ is incoherent. In this example one direct successor evaluation can be skipped.

There are two ways to exploit the additional information resulting from our approach. The user can be informed about the consequences of his decision. During our experiments we detected correspondences $\langle C, D, \equiv \rangle$ that seem to be correct at first sight. Especially, when the decision is based on a comparing labels of C and D. However, a closer look at the axioms describing C and D reveals that the first impression is incorrect. Such an analysis is time-consuming and will probably only be conducted if there are already some doubts in the correctness of $\langle C, D, \equiv \rangle$. The information that another acceptable correspondence conflicts with the currently accepted correspondence, can be a reason to reconsider the decision or to analyze the disputable correspondence in detail. A reasoning based-support can thus increase the correctness of the revision process.

In case the user is sure about his decision or the consequences of his decision are in line with his considerations, all conflicting correspondences can be removed from the alignment. This aspect can be used to increase the efficiency of the revision process. This is especially the case, if the main intention of the user is to determine a coherent alignment in an efficient way. We can quantify the manual effort that has been saved by counting the number of successor evaluations that can be skipped.

Both aspects – increasing correctness vs. increasing efficiency – are in conflict with each other. The benefit that comes with the use of a reasoning based support tool depends thus on the aspect that is more important to the user. We expect that a mixed strategy is the most appropriate approach in most concrete revision scenarios. However, such an approach makes it hard to quantify the benefit of a reasoning based support. Thus, we focus in the following two sections mainly on the extremities of the continuum.

10.2 Tool Support

In this section we describe the tool we developed to support the user in the revision of an alignment. We have explained above in how far such a tool can be used to increase the precision of an alignment. This aspect is object to our considerations in the following. Since it is hard to quantify this aspect, we focus on an example and report on our experiences in a concrete application scenario.

We have seen in Section 8.3 that a reasoning-based approach can in some cases result in relatively long runtimes, even though it is combined with the efficient methods we propose. Contrary to this, a user interface should give direct feedback to the actions of the user. For that reason we do not use complete reasoning techniques, but solely rely on the pattern based reasoning approach in this section. Given an alignment $\mathcal{A}$, we check for each pair of correspondences $\langle a, b \rangle \in \mathcal{A}$ whether $\{a, b\}$ is a MUPS according to the pattern based reasoning approach of Algorithm 5. If this is the case, we say that a conflicts with b and vice versa. Remember that we apply the same preprocessing step at the beginning of Algorithm 10 to precompute a significant fraction of all MUPS.

Our tool presents a list of correspondences together with their confidence value as provided by the matching system that generated the correspondence. The user can accept or reject each of these correspondences. This choice effects other correspondences and can trigger a sequence of changes in the status of the other correspondences. We use the following symbols to describe the status of the correspondences. This status is completely determined (1) by the current evaluation e and (2) by conflicts between correspondences. Given a correspondence c, we distinguish between five different states.

- ✔ The correspondence has been manually accepted, i.e., $e(c) = correct$.
- ✔ The correspondence has not yet been checked by the human expert, i.e., $e(c) = unknown$, and it does not conflict with any accepted correspondence nor does it conflict with a not yet evaluated correspondence.
- The correspondence conflicts with a correspondence that has not yet been evaluated, and is itself in the status of being not yet evaluated, i.e., $e(c) = unknown$.
- The correspondence has been dismissed automatically as it conflicts with a manually accepted correspondence. As a consequence we have $e(c) = incorrect$.

- 🗑 The correspondence has been manually dismissed by the user, i.e., $e(c) = incorrect$.

Ideally, a revision is carried out until all correspondences are labeled with ✓, 🗑 or 🗑. This indicates that the correctness or incorrectness of all correspondences has been confirmed directly or indirectly. In the case of large alignments, this will not always be possible. The user will focus on resolving all conflicts to finally come up with a coherent subset of the alignment. In this case, correspondences labeled with ✓ or ✓ are assumed to be correct and correspondences labeled with 🗑 or 🗑 are assumed to be incorrect. In this scenario, the work is done when none of the correspondences is labeled with ⚡ finally.

Figure 10.2 shows a typical example of our revision tool in use by depicting a sequence of steps enumerated from I to V. The alignment depicted in this example has been generated by the ASMOV matching system on the EDAS and MYREVIEW ontology. Both are part of the CONFERENCE dataset. ASMOV generates 25 correspondences, 8 of which are involved in a conflict. In the following we focus only on these correspondences. They are presented to the user at the beginning of the evaluation process labeled with a ⚡ sign (see Figure 10.2-I).

The user can accept or reject a correspondence by clicking on the + or - symbol beside the correspondence. In our example the user decides first to accept $\langle isMemberOf, isMemberOf, \equiv\rangle$. His choice is guided by the fact that it is the first correspondence where both entities are described with the same label. As a result of accepting this correspondence two other correspondences change their status. $\langle Conference, Conference, \equiv\rangle$ is rejected automatically and, as a consequence, $\langle hasMember, hasMember, \equiv\rangle$ is no longer involved in a conflict (Figure 10.2-II).

This raises a doubt. Why has $\langle Conference, Conference, \equiv\rangle$ been rejected as a consequence of the first decision? The user clicks on the magnifying glass to view context information that might help. Our tool shows a minimal amount of context information. For a property correspondence domain and range of the involved properties are displayed (see Figure 10.2-III). For a concept correspondence superclasses are depicted. This information is sufficient to understand that $\langle isMemberOf, isMemberOf, \equiv\rangle$ is incorrect. In one ontology *isMemberOf* describes the property of being a conference member, while in the other ontology *isMemberOf* refers to the property of being a member of a committee. This conflicts with $\langle Conference, Conference, \equiv\rangle$ and a disjointness axiom in the second ontology, which states that conferences and committees are disjoint.

The user changes his mind and rejects this correspondence. Instead of that, he accepts $\langle Conference, Conference, \equiv\rangle$. The result is depicted in Figure 10.2-IV. Again, the decision can be propagated. This time $\langle hasMember, hasMember, \equiv\rangle$ is rejected. This is based on the fact that in both ontologies *hasMember* is defined to be the inverse of *isMemberOf*. Without explicitly looking into the ontologies the user already had this assumption. He is supported in his believe and continues with the revision process.

Figure 10.2: User interface in action. After an interaction with the tool all conflicts are resolved and the correct subset of the alignment has been determined. The tool can be tested via `http://web.informatik.uni-mannheim.de/alcomo/revision/asmov/`. Click on the link `edas-myreview` in the menu to the left to revise this specific example on your own.

The next correspondence is $\langle Document, Document, \equiv\rangle$. The user accepts it and – as a consequence – all conflicts are removed (see Figure 10.2-V). It has been involved in a conflict with the first and the last correspondence in the list. Since these correspondences have also been involved in a conflict with the remaining correspondences $\langle Person, Person, \equiv\rangle$ and $\langle Review, Review, \equiv\rangle$, both are now no longer involved in a conflict. The user accepts all consequences and the revision process is finished.

This example illustrates two interesting aspects of our approach.

- Visualizing a conflict by propagating a decision can be a useful warning for the user. The decision has to be verified by a more detailed analysis.
- Resolving a conflict by accepting a correspondence can indirectly resolve other conflicts. These indirect consequences can also be a warning or an affirmation for the decisions of the user.

Both aspects support our claim that our approach increases the precision of the revision process. In particular, it is highly probably that without reasoning-based support both $\langle hasMember, hasMember, \equiv\rangle$ and $\langle isMemberOf, isMemberOf, \equiv\rangle$ would have been part of the final alignment.

From 2006 on the evaluation procedure of the OAEI CONFERENCE track comprises many different types of evaluation methods. Nevertheless, the standard approach for measuring precision and recall was missing before 2008 due to the absence of reference alignments. Instead of that, the submitted alignments had been evaluated manually. Since an enormous number of correspondences had to be processed, not every correspondence had been annotated correctly. Thus, we used the annotated corpus together with some automatically generated alignments as input to our tool, revised this corpus, and generated precise reference alignments over a subset of seven ontologies. This approach resulted finally in the reference alignments of the widely used CONFERENCE dataset.

Our tool turned out to be very useful in this setting. It would have required an extensive and time consuming inspection of the involved ontologies to gain this information without support. We experienced that the most efficient way to use the tool is indeed to label promising correspondences at first. This results very often in automatically eliminating doubtful correspondences without the need for further investigation or points sometimes to an incorrect decision. Overall, we significantly reduced the input corpus of correspondences previously annotated as correct and could thus create a basis for highly precise reference alignments.

A rich set of examples can be found at `http://web.informatik.uni-mannheim.de/alcomo/revision/matcher/`. The string `matcher` has to be replaced by one of `asmov`, `dssim`, or `lily`. These examples are based on the alignments that have been generated by the OAEI 2008 participants of the CONFERENCE track.

10.3 Efficiency

In the following we quantify in how far the approach can increase the efficiency of the manual revision in terms of time required to revise an alignment. Note that our tool contains some elements to support the user, which are not specific to our approach (for example, displaying context information about matched entities). In general, a tool that supports the revision process must contain different elements to present the required information in an appropriate way [Fal09]. However, within this thesis we are only interested in the benefits of the logical support. For that reason we abstract from other aspects by simulating a perfect user, i.e., a user that never fails in his decisions. This allows to ignore concrete implementation details and allows to abstract from interdependences between correctness and efficiency.

In particular, our experiments are based on the following setting. Given a confidence allocation α, an ordered alignment $\mathcal{A} = \{c_1, \ldots, c_n\}$ with $\alpha(c_i) \geq \alpha(c_{i+1})$, and a reference alignment $\mathcal{R}$. We present correspondences of $\mathcal{A}$ to a fictitious user in the sequence $c_1, \ldots c_n$. Let e_i denotes the evaluation function after our user made his i-th decision. If $c_i \in R$ we suppose that the user accepts c_i, otherwise the user rejects c_i. After each step i we compute the set of all correspondences in $e_i(\mathcal{A}, \mathit{unknown})$ that conflict with $e_i(\mathcal{A}, \mathit{correct})$. Then we remove all correspondences in $\mathcal{A}$ from the set of correspondences that have not yet been analyzed and mark them as $\mathit{incorrect}$. The total of these correspondences - summed up over all steps of the process - is equivalent to the number of direct successor evaluations that have been skipped. By comparing this number against the number of all correspondences, we can compute how much effort has been saved.

Our experiments are again conducted on the subset of the CONFERENCE dataset where a reference alignment is available. As input alignments we use the alignments generated by the matching systems participating in OAEI 2010. We do not revise each of the alignments on its own, but revise the union of the alignments for each of the testcases. We use again the approach that we already applied in Section 9.3 to compute the confidence allocation of the resulting merged alignment. Thus, the user has to revise a set of input hypotheses that aims at high recall but contains at the same time a high number of incorrect correspondences.

The user will probably try to construct a one-to-one alignment. Suppose he accepts a certain correspondence c. It follows that an alternative correspondences, which is not consistent with the one-to-one constraint, can be eliminated as consequence. Coherence constraints and one-to-one constraints can thus be taken into account at the same time. Note that we already followed this approach in Section 9.3 in a fully automated setting. In the following we compare the combined approach ($\Delta \wedge$ 1:1) and the simple one-to-one approach (1:1), which exploits only one-to-one constraints, against a manual revision where each of the correspondences has to be evaluated on its own.

The result of our experiments are depicted in Table 10.1. The first data row in this table, for example, shows that for testcase CMT-CONFERENCE the user has to evaluate an input alignment $\mathcal{A}$ of 78 correspondences. Reference alignment $\mathcal{R}$

Conference Testcase	$\vert\mathcal{A}\vert$	$\vert\mathcal{R}\vert$	$\vert e(\mathcal{A}, correct)\vert$	Saving$_{1:1}$	Saving$_{\Delta\wedge 1:1}$
CMT-CONFERENCE	78	16	12	10.3%	44.9%
CMT-CONFOF	47	16	10	17%	29.8%
CMT-EDAS	59	13	11	16.9%	45.8%
CMT-EKAW	65	11	8	13.8%	46.2%
CMT-IASTED	47	4	4	12.8%	17%
CMT-SIGKDD	47	12	12	10.6%	17%
CONFERENCE-CONFOF	69	15	14	10.1%	23.2%
CONFERENCE-EDAS	112	17	15	12.5%	46.4%
CONFERENCE-EKAW	97	25	21	13.4%	24.7%
CONFERENCE-IASTED	80	14	10	8.8%	8.8%
CONFERENCE-SIGKDD	49	15	13	10.2%	10.2%
CONFOF-EDAS	73	19	17	11%	50.7%
CONFOF-EKAW	61	20	19	11.5%	39.3%
CONFOF-IASTED	63	9	7	0%	14.3%
CONFOF-SIGKDD	43	7	6	7%	30.2%
EDAS-EKAW	108	23	21	13%	50.9%
EDAS-IASTED	100	19	14	13%	29%
EDAS-SIGKDD	62	15	14	14.5%	35.5%
EKAW-IASTED	90	10	8	4.4%	13.3%
EKAW-SIGKDD	48	11	9	6.3%	25%
IASTED-SIGKDD	89	15	14	10.1%	11.2%
∅	70.8	14.6	12.3	10.9%	30.2%

Table 10.1: Savings in revision effort due to reasoning support.

has only 16 correspondences, 12 of these correspondences are contained in $\mathcal{A}$ and are evaluated as correct (e denotes the final evaluation function).This means that 66 correspondences were rejected directly or indirectly. If we support the user with the simple one-to-one propagation rule 10.3% of his decisions are implicit decisions propagated by our tool. If the user revises the alignment with a reasoning-based approach 44.9% of his decision are propagated.

Average results are presented in the last row of the table. The simple one-to-one constraint allows already to save 10.9% effort in terms of evaluated correspondences. Using incoherence as additional source of propagating user decisions, allows to save 30.2% in average. This means that in average only $\approx$ 70% of all correspondences have to be analyzed based on our approach, instead of $\approx$ 90% (one-to-one only) or 100%.

Our results show also that there is a high variance with respect to the savings. There are some cases in which the reasoning-based approach cannot save more effort than the one-to-one constraint. This holds for CONFERENCE-IASTED and CONFERENCE-SIGKDD. In these cases the involved ontologies contain no or only a very limited number of disjointness axioms. The results show also that it is sufficient that at least one of the ontologies contains a rich disjointness modeling. Note that we can reduce the required effort in some cases to the half. This means that for such cases the efficiency of a user is twice as high in terms of evaluation speed if he uses a tool that implements our approach.

10.4 Conclusions

Within the previous sections we have been concerned with research question R7: *How can alignment incoherence be exploited to support a user in revising an automatically generated alignment?* Our answer is, in short, that it can be used to increase both precision and efficiency of the revision process. Precision refers in this context to the fraction of correct evaluation decisions, while efficiency refers to the time required to revise an alignment. However, there will always be a trade-off between these two aspects and the short answer requires further explanations.

If the approach is used in a context where highly precise alignments are required, the effects of propagating a decision – or at least some of the effects - are displayed to the user. This information can then additionally confirm the decision or motivate a doubt in its correctness. Above we have presented an example to illustrate that there are correspondences that will probably be accepted if such a doubt is not raised in the first place. A final falsification will often require a further analysis of the concepts or properties under discussion, however, such an analysis is time consuming and thus conducted only if there is an initial suspicion. There might also be situations that require to construct in short time a coherent alignment from a comprehensive set of correspondences. In this case the consequences of a decision are not presented to the user. Instead of this, they are propagated directly as a reduction of the hypotheses that have to be evaluated. We reported above about some examples for which the set of correspondences has been reduced to 50% of its original size. However, these scenarios are the two ends of a continuum and we expect that most revision tasks are positioned somewhere in between.

We do not argue that our approach is the ultimate solution to the revision of ontology alignments. This is by far not the case. We know that a tool for this purpose has to offer a rich functionality to present and visualize relevant aspects of the ontology and the alignment under discussion. Examples can be found in the user interfaces of systems described in [FS07] and [CSC$^+$10]. See also related work presented in Section 11.3. Nevertheless, we have shown that our approach can increase precision and efficiency significantly by exploiting interdependencies between alignment and aligned ontologies.

Part IV

Related Work & Conclusion

Chapter 11

Related Work

There are two ways of spreading light: to be the candle or the mirror that reflects it (Edith Wharton).

We divide work related to our approach in four different areas. We present work that deals with the automated debugging of incoherent alignments, we show in how far our approach is related to argumentation frameworks, we discuss approaches that deal with the manual revision of alignments, and finally we present matching systems that implement some components or techniques similar or relevant to the approach developed in this thesis.

In Section 11.1 we review the algorithms developed by Qi et al. [QHH$^+$08, QJH09]. Similar to our approach, they are inspired by algorithms designed for ontology debugging. In addition to a theoretical analysis, we also report about experiments to compare runtimes of our algorithms against the algorithms developed by Qi et al.

In Section 11.2 we discuss in how far our approach can be understood within the framework of argumentation theory [Dun95]. For this purpose we first briefly introduce the relevant notions in an informal way. Then we explain differences and commonalities between a diagnostic approach and an approach grounded on argumentation theory. Finally, we review some approaches that use argumentation theory to resolve conflicts in alignments.

In Section 11.3 we come back to the topic of manual alignment revision. We discuss alternative approaches and related work. Hereby, we focus on the work of Jimenez et al. [JRGHB09], in which the authors implement comparable methods in a tool that supports a human expert in the revision process. In addition, we discuss work that is concerned with the order in which correspondences are presented to the user [Nik10, NRG11].

In Section 11.4 we review the uptake of reasoning and debugging techniques in ontology matching systems. Note that it is impossible to give a complete overview of the field. A large number of systems use structural methods, which are often described as semantic methods. Moreover, the available system descriptions make it often hard to understand what is actually meant with terms as *'verification'*, *'validation'*, or *'debugging component'*. We present some relevant systems and try to illustrate which kind of reasoning-based methods are used in current ontology matching systems to reduce the degree of alignment incoherence.

11.1 Algorithms for Alignment Debugging

The work of Qi et al. is closely related to the approach described in this thesis. In [QHH$^+$08, QJH09] the authors propose several algorithms to debug (or repair) an incoherent alignment. Before analyzing these algorithms in detail, we focus on their theoretical foundation. While our approach is based on the framework of computing a diagnosis [Rei87], Qi et al. describe their algorithms in the framework of belief revision [Han99]. An important postulate in the theory of belief revision is the principle of minimal change, i.e., to minimize the changes whenever new beliefs are integrated in existing beliefs.

We find a counterpart of this principle in our work. Our algorithms are designed to remove as less confidence weighted correspondences as possible. In particular, a diagnosis is defined as a minimal hitting set, i.e., removing a proper subset of a diagnosis does not result in a coherent alignment. Moreover, the global optimal diagnosis is defined as the smallest diagnosis with respect to the total of confidences. We have also shown that the algorithms for computing a global optimal diagnosis can also be used to construct a smallest diagnosis Δ with respect to the number of correspondences contained in Δ.

Very similar to our approach, Qi builds on work concerned with ontology debugging (see for example [SC03, SH05] and [KPHS07]). While our approach is specific to ontology alignment, Qi et al. have developed in [QHH$^+$08] an approach applicable to a more general scenario. This scenario is based on the distinction between a trusted ontology ($\mathcal{O}_1$ and $\mathcal{O}_2$ in the case of alignment debugging) and an untrusted ontology ($\mathcal{A}$ interpreted as set of axioms). It can, for example, also be applied to the scenario of ontology learning. The algorithms presented in [QJH09] are designed especially for the alignment debugging scenario. We discuss the algorithms from both publications, by dividing them in three groups discussed in Section 11.1.1, Section 11.1.2, and Section 11.1.3. In Section 11.1.4 we compare runtimes of our algorithms against runtimes of the most promising algorithms presented in the following sections.

11.1.1 Complete MIPS Computation

The algorithms of this group compute first all MIPS. This information is used in the following step to resolve alignment incoherence.

ALG-1 (presented by Qi et al. as Algorithm 1 in [QHH$^+$08]) The algorithm computes first all MIPS $\mathcal{M}_1, \ldots, \mathcal{M}_n$ based on the algorithm presented in [KPHS07]. In the next step a score is assigned to each correspondence $c \in \mathcal{M}_i$. The score is computed as number of $\mathcal{M}_i$ in which c occurs. Each $\mathcal{M}_i$ is reduced to the subset $\mathcal{M}'_i \subseteq \mathcal{M}_i$ of correspondences with highest score. Finally, the hitting-set-tree algorithm [SHCvH07] is used to compute a hitting-set over the set of reduced MIPS $\mathcal{M}'_1, \ldots, \mathcal{M}'_n$.

ALG-2 (developed by Qi et al. as Algorithm 2 in [QHH+08]) This algorithm is a variant of ALG-1. The difference is the reduction step, which is not based on the score, but on the confidence value. Those correspondences with highest confidence are chosen. Finally, again a hitting-set over the set of reduced MIPS $\mathcal{M}'_1, \ldots, \mathcal{M}'_n$ is computed.

The computation of all MIPS is a very expensive process in terms of runtime. This is illustrated by Qi et al. in Table 1 and Figure 1 in [QHH+08]. It seems that these algorithms have been designed based on the implicit assumption that a minimal hitting set can only be computed by first computing all MIPS (or MUPS). This assumption is incorrect. In none of our algorithms we compute all MIPS or MUPS, however, for all of our algorithms we guarantee that the solution is a minimal hitting set over all MIPS.

A second problem is related to the principle of minimal change. The step of reducing each of the MIPS to a smaller subset, which can be found in both algorithms, is motivated by the reduction of the complexity of the input to the following step, which is the computation of the hitting set. However, the hitting-set for the reduced MIPS sets is not necessarily a hitting set for the unreduced MIPS set. This is also pointed out by the authors themselves and contradicts the principal of minimal change.

11.1.2 Iterative MUPS Computation

These algorithms are based on an approach that iterates over the set of unsatisfiable concepts and removes, for each unsatisfiable concept, one of the correspondences involved in causing the unsatisfiability. Contrary to the first two algorithms, it is not required to compute all MIPS or MUPS in advance. For that reason the algorithms do not suffer from the severe runtime problems of ALG-1 and ALG-2.

WBONE (reimplemented by Qi et al. as *Weightbased-One* in [QJH09], based on an Algorithm we presented in [MST06]). The algorithm is an iterative algorithm, which picks an unsatisfiable concept and computes a randomly chosen MUPS for this concept. From this MUPS the element with lowest confidence is removed. The algorithm continues like this until the whole alignment is coherent.

WBALL (developed by Qi et al. as Algorithm 3 in [QHH+08], referred to as *Weightbased-All* in [QJH09]). This algorithm picks an unsatisfiable concept and computes all MUPS for this concept. Computing all MUPS at once speeds up the process compared to the previous algorithm. For each MUPS the element with lowest confidence is removed. The algorithm continues like this until the whole alignment in coherent.

Both WBALL and WBONE follow a principle similar to the principle underlying the design of the local optimal diagnosis. Even though there is no reduction step, as used by the algorithms of the previous section, the solutions generated by these algorithms are nevertheless not minimal. We gave an example in the third paragraph of Section 4.2.1 to explain why the algorithm we developed in [MST06], which is WBONE on top of DDL, does not always construct a minimal hitting set.

11.1.3 Iterative Coherence Checks

The third group of algorithms is capable of solving the problem of the previous algorithms. These algorithms are based on a sketch of an algorithm we presented first in [MVS08], which finally resulted in our brute-force algorithm for computing a local optimal diagnosis. However, the approach of Qi et al. differs with respect to the treatment of confidence values. In particular, their algorithms might generate different results if and only if an alignment contains correspondences annotated with the same confidence value. If this is not the case, the result of both algorithms is a local optimal diagnosis.

LINEAR The algorithms of this group require that $\mathcal{A}$ is ordered descending by confidences. If $\mathcal{A}$ is incoherent, the algorithm determines the highest value h such that the set of correspondence $\mathcal{A}' = \{c \in \mathcal{A} \mid \alpha(c) > h\}$ is coherent and $\mathcal{A}'' = \{c \in \mathcal{A} \mid \alpha(c) \geq h\}$ is incoherent. All correspondences in $\mathcal{A}'$ are accepted and a subset from $\mathcal{A}'' \setminus \mathcal{A}'$ is removed such that $\mathcal{A}''$ is coherent afterwards. The algorithm continues like this – decreasing the value of h in each iteration – until the whole alignment is coherent. The following two variants differ with respect to the selection of correspondences that have to be removed.

L-RELEVANCE This variant uses an algorithm similar to WBONE to solve the problem of selecting from $\mathcal{A}'' \setminus \mathcal{A}'$ in each iteration the correspondences that have to be removed. However, there is a significant difference. Qi et al. expand systematically the set of axioms to avoid reasoning in a large ontology. They first start with those axioms in which the unsatisfiable concept occurs and then extend this set recursively until it is strong enough to cause the unsatisfiability.

L-SCORE This variant uses ALG-1 to solve the problem of selecting from $\mathcal{A}'' \setminus \mathcal{A}'$ in each iteration correspondences that have to be removed.

Given an alignment $\mathcal{A}$ generated by some specific matching system, it is often the case to have several correspondences with the same confidence value in $\mathcal{A}$. Within this thesis we did not cope with this problem. Note that the local optimal diagnosis is not uniquely defined, if this is the case. We implicitly assume that there is a way to specify a complete order for all correspondences in $\mathcal{A}$. If for two correspondences c and c' we have $\alpha(c) = \alpha(c')$, we assumed that there will also be another source of evidence β with $\beta(c) \neq \beta(c')$. However, we did not include different sources of evidence in our experiments. We have decided to chose a fixed random order as a substitute for some specific β. Another option would have been to experiment with several concrete sources of evidence $\beta_1, \ldots, \beta_n$. However, these experiments would put a focus on the quality of these confidence measures, an object of research that is not in line with the research questions of this thesis. Note also that, contrary to the local optimal solution, the global optimal diagnosis is often uniquely determined even though many correspondences are annotated with the same confidence value.

Again, both algorithms do not compute a minimal hitting set. Suppose that we have $\alpha(c) = \alpha(c')$ for all correspondences $c, c' \in \mathcal{A}$. In this case the algorithm L-SCORE behaves like ALG-1 and LINEAR-R behaves like WBONE extended with the reasoning optimization based on the recursive definition of relevance. Note that our algorithms for computing a local optimal diagnosis are in this scenario completely determined by a fixed random order as long as no additional confidence distribution β is available. For that reason, our algorithms ensure that a minimal hitting set is constructed.

The main benefit of the relevance-based variant is its positive effects on the runtime of the linear algorithm. It is thus an alternative to the use of pattern based reasoning techniques integrated in Algorithm 8. According to the results presented in the following section, positive effects of the pattern based reasoning are stronger. A further speed-up might be possible by combining both methods.

Qi et al. point to another advantage of the relevance-based approach. They argue as follows. *'The motivation behind the algorithm is that when choosing between two correspondences to remove, we always remove the one which is more relevant to an unsatisfiable concept and thus is more likely to be problematic.'* [QJH09] However, the length of a chain of axioms that leads from an unsatisfiable concept to a correspondence c is in not correlated with the probability that c is incorrect. We presented examples in Table 5.1, 5.2 , 5.3, and A.1 that illustrate this.

11.1.4 Runtime Comparison

With respect to scalability issues, Qi et al. have focused on a specific alignment from the 2008 submissions to the CONFERENCE track generated by matching system LILY. It is the alignment between the CMT ontology and the EKAW ontology. The authors have added 100 dummy correspondences and between 1000 and 5000 dummy axioms to both ontologies in order to test their algorithms on problems of different size. For the 4000er version they also varied the number of additional correspondences from 100 to 400 dummy correspondences. We use the same datasets

Datasets		On top of KAON (implemented by Qi et al.)					On top of Pellet		
Axioms	Corr.	WBONE	WBALL	LINEAR	L-RELEVANCE	L-SCORE	Alg. 6	Alg. 8	Alg. 10
1000	100	101	42	129	69	68	90	6	115
2000	100	395	91	304	143	140	146	9	157
3000	100	733	134	504	202	220	255	12	205
4000	100	872	162	799	277	298	514	18	275
4000	200	1990	158	1493	282	281	-	27	-
4000	400	4125	162	3215	279	268	-	67	-
5000	100	1478	206	1112	347	326	2248	31	377

Table 11.1: Runtimes in seconds for algorithms implemented/designed by Qi et al. compared against our algorithms (brute-force local optimal diagnosis in Alg. 6, efficient local optimal diagnosis in Alg. 8, efficient global optimal diagnosis in Alg. 10).

for the experiments reported in the following.[1]

Qi et al. measured the runtimes of five different algorithms. The results of their experiments are shown in left part of Table 11.1. The Algorithm entitled LINEAR is the equivalent to our brute-force approach of computing a local optimal diagnosis (Alg. 6). This algorithm has been re-implemented by the authors to compare their algorithms against our algorithm. The other four algorithms are described above. On the right side of Table 11.1 we present results that we measured in experiments running our algorithms for computing local (brute-force and efficient) and global optimal diagnoses (only efficient)

Note that the results cannot be compared directly, because Qi et al. have used the KAON reasoner[2], while we use Pellet [SPG$^+$07]. Moreover, they have conducted their experiments on a different machine (Intel CPU Xeon(TM) 3.2 GHz with allotted 2 GB heap space). Fortunately, Qi et al. have also implemented the brute-force approach for computing a local optimal diagnosis (referred to as LINEAR). This allows to estimate a factor to model the differences between Pellet and KAON specific for this dataset/setting that is our basis for a rough comparison. Our results show that the algorithm on top of Pellet is between 1.5 to 2.0 times faster for this dataset (with the exception of testcase 5000-100).

Taking this into account, the efficient variant of our algorithm for computing a local optimal diagnosis clearly outperforms all other algorithms. The computations of the global optimal diagnosis requires more time. However, we measured runtimes that are still similar to the runtimes of the fastest algorithms developed by Qi et al. A argued above, these algorithms do not guarantee to generate a diagnosis nor do they obey to a global optimality criterion. Our experiments show also

[1] We would like to thank Qiu Ji for providing the dataset and the detailed runtimes of the experiments reported in [QJH09].

[2] `http://kaon2.semanticweb.org/`

that Pellet runs out of memory, when we apply it to some of the testcases. these testcases are marked by a '-'. Currently we miss an explanation for these problems.

11.2 Argumentation Frameworks

In Definition 27 we introduced the notion of an accused correspondence. Remember that a correspondence c is accused only if none of the other correspondences in $\mathcal{M}$ are accused. In this section we show that similar ideas can be found in the classical argument system of Dung [Dun95]. Moreover, argumentation frameworks have also been used to model and resolve disagreement in alignments generated by different matching systems [dSQVI10]. We first explain the relation between our approach and Dungs framework. Then we briefly discuss existing extensions specific to the field of ontology matching.

Dung defines an argumentation framework as a pair $\langle AR, \textit{attacks} \rangle$ where AR is a set of arguments and *attacks* is a binary relation on AR, i.e., *attacks* $\subseteq AR \times AR$. With respect to ontology matching an argument is a correspondence or a tuple that contains a correspondence [dSQVI10]. Within the context of alignment incoherence, it suggests itself that the attack relation can be used to express a coherence conflict between two correspondences. However, we will see below that argumentation frameworks have been used in a different way for ontology matching in the past.

Dung defines a set of arguments $S \subseteq AR$ to be conflict-free, if there exists no $x, y \in S$ such that x attacks y. The arguments in a conflict-free set S do not attack each other, however, they can be attacked from arguments in $AR \setminus S$. For that reason Dung introduces the notion of an admissible set of arguments. An argument set S is called admissible if (1) S is a conflict-free set and (2) if for each $x \in AR$ and $y \in S$ with x attacks y there exists an $z \in S$ such that z attacks x. In other words: S is admissible if there are no attacks *'inside'* of S and all arguments from $AR \setminus S$ are counter-attacked by some element in S. An admissible set S is called preferred extension, if S is maximal with respect to set inclusion, i.e., if each proper superset of S is not admissible.

Let us define now the attack relation as follows. Given an incoherent alignment $\mathcal{A}$, correspondence c attacks c' if and only if there exists a MIPS $\mathcal{M}$ of $\mathcal{A}$ with $\mathcal{M} = \{c, c'\}$. Based on this definition we have introduced an attack relation that is symmetric, i.e., c *attacks* $c' \Leftrightarrow c'$ *attacks* c. It follows that a conflict-free set is always admissible. Moreover, we conclude that a conflict-free set, which is maximal w.r.t. set inclusion, is a preferred extension. We defined a diagnosis Δ as a minimal subset of an alignment $\mathcal{A}$ such that $\mathcal{A} \setminus \Delta$ is coherent, i.e., $\mathcal{A} \setminus \Delta$ is a maximal conflict-free set and thus a preferred extension. In particular, both local and global optimal diagnosis are specific types of preferred extensions. We did not base our approach on the theory of argumentation, because not all MIPS contain exactly two elements. See again the examples depicted in Table A.1 and Table 5.2. Given such a situation, it requires a more complicated way to model the attack relation.

The application of argumentation theory has already been proposed for ontology matching in several publications. However, the focus here is on composite ontology matching [dSQVI10], i.e., combining alignments generated by different matching techniques or matching systems. In Chapter 9 we have analyzed a similar scenario. Trojahn et al. [dSQV08] propose, an extension of Dungs argumentation framework applicable to this scenario. In this framework matching systems (or techniques) are modeled as audiences, which have preferences regarding their trust in (other) matching techniques. Moreover, confidence values are used to model the strength of an argument. One important difference to our approach is the definition of the attack relation. Instead of exploiting logical conflicts, an argument c_{neg} that attacks c is generated iff a matching system does not output c as part of the generated alignment. The attack relation is thus not based on logical conflicts. One important extension is related to the use of confidence values α. Given that c attacks c', an attack succeeds if $\alpha(c) > \alpha(c')$, leaving aside differences between audiences and their preference ordering.

Trojahn and Euzenat combine the notion of alignment coherence and argumentation frameworks for the first time in [TE10]. However, instead of using alignment coherence to extend and strengthen the attack relation, they use alignment coherence in a subsequent step to extract a coherent subset from the result of the argumentation process. The experiments conducted in [TE10], indicate that this combination cannot improve the results significantly. As alternative approach we have suggested above to model the attack relation via alignment incoherence, an approach that would probably be in line with the generic approach presented in [AB09]. If applied to one matcher, leaving aside issues related to audiences and their preference ordering, the local optimal diagnosis is in this simplified setting the set difference of a preferred extension.

11.3 Manual Revision

There are several systems that support the manual revision or refinement of alignments. Examples are the COGZ system [FS07, Fal09] and matching systems like TAXOMAP [HSNR10] or AGREEMENTMAKER [CSC+10], which have a built-in user interface to view, modify or revise an alignment. However, to our knowledge there exists only one system that uses diagnostic methods, similar to the approach proposed in this work, to support manual alignment revision. This is the system CONTENTMAP described in [JRGHB08, JRGHB09].

While we restrict our reasoning-based approach to the resolution of alignment incoherence, the authors of CONTENTMAP suggest a more general approach. They divide consequences of an alignment into intended and unintended consequences. The user can define which consequences are intended and unintended. This allows more flexibility, however, the typical example for an unintended consequence is the existence of a concept that becomes unsatisfiable, i.e., the incoherence of the alignment.

CONTENTMAP can be used to display intended and unintended consequences that follow from an alignment. The existence of unintended consequences allows to conclude that there are errors in the alignment. An unintended (or intended) consequence can be chosen and its justifications are computed and displayed. A justification for an unintended consequence corresponds to a MIPS in our terminology. Moreover, the user can analyze and chose between all possible repair plans. A repair plan is a minimal subset Δ of $\mathcal{A}$ such that none of the unintended consequences follow from $\mathcal{A} \setminus \Delta$ and all intended consequences follow from $\mathcal{A} \setminus \Delta$. If the set of intended consequences is empty and the unintended consequences are defined to be unsatisfiable concepts and properties, *'repair plan'* and *'diagnosis'* are equivalent notions.

The basic idea of the approach, which has been developed independently of our work, is in the same line with the ideas we proposed in [MST08, MSSZ09, MST09]. Some extensions - namely the distinction between intended and unintended consequences - introduce some new interesting elements. A major problem of the approach is related to the computation of <u>all</u> repair plans. In many cases the number of repair plans - in our terminology diagnoses - will be very high. For the CONFERENCE dataset we observed that many automatically generated alignments have more than 1000 different diagnoses. Note that the number of diagnoses in an alignment $\mathcal{A}$ grows exponentially with respect to number of MIPS in $\mathcal{A}$. Thus only a limited number of MIPS result in a high number of diagnoses. Thus, matching relatively small ontologies can already result in an enormous amount of different diagnoses. For a user it is very hard or even impossible to analyze all different diagnoses. Contrary to this it is possible to construct a single diagnosis in the sequential approach that we described in Chapter 10 with minimal cognitive load for the user.

The benefit of our sequential approach depends on the order, in which correspondences are presented to the user. We have already analyzed this issue already in [MST08]. In this thesis we have presented results for only one order, namely the order determined by α. The effects of different orders have been investigated in detail by Nikitina [Nik10] for the more general problem of revising ontologies [NRG11]. A specific order referred to as MINSETRANK has been proposed in [Nik10]. It results in an average reduction of 83% in terms of effort. The approach exploits additionally entailment, i.e., to automatically accept correspondences that can be entailed from the set of correspondences accepted so far. This was also a topic of our experiments in [MST08].

However, within this thesis we put a focus on the role of alignment incoherence. In this context the strongest effects can be expected if the most probable correspondences are evaluated first, because only the acceptance of a correspondence can result in consequences.

11.4 Ontology Matching Systems

Some matching systems use, in addition to many other techniques, specific methods to avoid logical problems in the generated alignment. In the following we describe some of these methods in detail. To find matching systems that use relevant methods, we analyzed the systems participating in the OAEI from 2008 to 2010. The results of our survey can be found in Appendix C.

According to the system descriptions that we analyzed, ASMOV [JMSK09] and LILY [WX08a] use techniques to reduce alignment incoherence. Both systems use a blacklist of combinations of correspondences (often pairs of correspondences) that result in logical problems. The main idea of this approach is very similar to our pattern based reasoning techniques.

Matching system LILY uses the following patterns. Note that we use in the following listing notions as *'inconsistency'* or *'incoherence'* in the same way as they are used by the authors of these systems.

(I) Inconsistent Alignment Causing is-a Circles An alignment $\mathcal{A}$ is inconsistent if it causes an is-a circle in the aligned ontology, i.e., there exists a sequence of concepts $C_1, \ldots, C_n$ such that $\mathcal{A}_{\mathcal{S}}(\mathcal{O}_1, \mathcal{O}_2) \models C_i \sqsubseteq C_{i+1 \bmod n}$ while the same entailment does not hold in $\mathcal{O}_1$ or $\mathcal{O}_2$. Is-a circles result in the equivalence of all concepts involved in the circle.

(II) Inconsistent Mappings Violating Axioms Two correspondences $\langle A, A', \equiv \rangle$ and $\langle B, B', \equiv \rangle$ are inconsistent if $\mathcal{O}_1 \models A \equiv B \wedge \mathcal{O}_2 \not\models A' \equiv B'$. The same holds if we replace equivalence by disjointness.

LILY does not automatically resolve these kind of conflicts. The system informs a user and asks him remove one of the involved correspondences. Note, first of all, that this notion of inconsistency is not at all related to the definition of inconsistency, nor it is related to the definition of incoherence (Definition 22 and 12). It describes specific cases in which an alignment results in new consequences that cannot be derived without the alignment. It is assumed that these consequences are incorrect, because otherwise they would hold already in one of $\mathcal{O}_1$ or $\mathcal{O}_2$.

Matching system ASMOV avoids instances of the following patterns. We will see that these patterns partially overlap with the patterns used by LILY.

(III) Multiple-entity correspondences This pattern occurs iff two different entities are matched on the same entity. Thus, the pattern describes a violation of a one-to-one constraint.

(IV) Criss Cross Correspondences Correspondences $\langle A, B', \equiv \rangle$ and $\langle B, A', \equiv \rangle$ are an instance of this pattern, if $\mathcal{O}_1 \models A \sqsubseteq B$ and $\mathcal{O}_1 \models A' \sqsubseteq B'$. It is a special case of Pattern I implemented by LILY.

(V) Disjointness-subsumption contradiction Correspondences $\langle A, A', \equiv \rangle$ and $\langle B, B', \equiv \rangle$ are an instance of this pattern if $\mathcal{O}_1 \models A \sqsubseteq B$ and $\mathcal{O}_1 \models A \sqsubseteq$

$\neg B$. This pattern is a special case of our subsumption propagation pattern presented in Section 5.2.

(VI) Subsumption and equivalence incompleteness This pattern is similar to Pattern II. Instead of equivalence and disjointness it is concerned with equivalence and subsumption.

(VII) Domain and Range Incompleteness This pattern can occur for a pair that consists of concept correspondence $\langle C, C', \equiv \rangle$ and object property correspondence $\langle P, P', \equiv \rangle$. It occurs if $\mathcal{O}_1 \models C \sqsubseteq \exists P.\top \wedge \mathcal{O}_2 \not\models C' \sqsubseteq \exists P'.\top$. The same pattern is used for the range of the involved properties.

The authors of ASMOV refer to their approach as semantic verification or validation. Detected conflicts are eliminated by removing one of the conflicting correspondences, probably by the use of an iterative algorithm. Among the rich set of patterns implemented by ASMOV, we find one pattern that points to a special case of incoherence (Pattern V). All of the other patterns are specific cases of preventing entailments in the signature of $\mathcal{O}_1$ or $\mathcal{O}_2$ that cannot be derived from each of $\mathcal{O}_1$ or $\mathcal{O}_2$ solely, but are caused by the alignment. This comprises new equivalences (I, II, III, IV, VI) and new subsumptions (II, VI, VII) induced by the alignment.

The underlying principle is that an equivalence or subsumption axiom in the signature of $\mathcal{O}$, which cannot be derived from $\mathcal{O}$, is incorrect. Thus, it is assumed that $\mathcal{O}$ is complete with respect to any subsumption that is stated or can be derived. This is not at all the case. Given an incompletely modeled ontology, the filtering used by LILY and ASMOV is much too strict and will remove many correct correspondences. Our approach relies on the existence of disjointness axioms, which are also often missing. However, there is a crucial difference. In case of missing disjointness our approach removes less correspondences, i.e., there are no negative effects on the quality of the generated alignment. We can only exploit the information that is available, but do not assume that the information encoded in the ontologies is complete.

A general approach that focuses especially on preserving the structure of the aligned ontology by prohibiting new entailments is implemented in the S-Match system [GYM07]. The S-Match system, *'S-Match'* is a shortcut for *'Semantic Matching'*, first computes a complex representation of a concept. This representation is a propositional description of the concept that takes its position in the hierarchy into account. Given an tentative alignment (a matrix of relations) between the concepts of a hierarchy, the final step of matching two concepts can be formalized as the proof of a propositional formula. The correspondence is accepted if the formula representing the correspondence can be proved on top of the axioms describing the ontologies.

The approach makes use of a SAT-solver in the final step. However, it is slightly misleading to say that the correctness of the correspondences is proved. Instead of this, the approach ensures that the finally generated correspondence can be derived from all axioms and assumptions collected and prepared in the previous steps. In doing so the system ensures (indirectly) that no new axioms in the signature of $\mathcal{O}_1$ or $\mathcal{O}_2$ can be entailed from the resulting alignment. Otherwise, the final proof would fail. The approach suffers from the same problems that we mentioned before. Due to the use of a propositional encoding, it is also not clear in how far the system can exploit all of the knowledge encoded in more expressive ontologies.

Reasoning with ontology alignments has also been a topic in many other publications. Most of the approaches described are based on a weak or modified variant of the principle that an alignment should not introduce new terminological knowledge in one of the aligned ontologies. Instead of using the principle as a filtering mechanism, a weak form of the principle is used as soft constraint. Examples can be found in systems as PROMPT [NM03], ANCHOR-PROMPT [NM01], and OMEN [MNJ05]. OMEN defines a set of so-called metarules as soft constraints to describe interdependencies between correspondences as Bayesian Network. The rules that guide the process are specific rules that are partially based on the principles described above. A model-theoretic justification is missing.

Two novel matching systems have taken up this approach by combining soft and hard constraints. In [ABEZS09] the authors propose the use of Markov Networks to model hard one-to-one constraints and soft constraints inspired by the OMEN approach. We have recently proposed the matching system CODI [NMS10, NN10]. The system extracts a final alignment from the set of matching hypotheses by solving a MAP inference problem in a Markov Logic Network. The inference problems consists of a set of hard constraints, based on the patterns described in Section 5.2, and a set of soft constraints, which formalize Pattern VII and VIII . CODI cannot guarantee the coherence of the alignment, however, the generated alignments have a limited degree of incoherence (see again Section 7.3). Moreover, the distinction between hard and soft constraints is in line with the distinction between incoherence and the entailment of new consequences.

We conclude that most matching systems do not interpret correspondences on top of a well-founded semantics. Semantic rules, which guide the matching process, are often presented ad-hoc. They are not coupled to a generic model-theoretic approach. We have reported about the consequences in Section 7.3. However, we have also seen that it is possible to combine existing matching techniques with the algorithms presented in this thesis.

11.5 Summary

In the previous sections we have presented research from different related areas. Some of our ideas have already influenced other researchers, while other relevant approaches have been developed independently. Our comparative analysis illustrated also a specific characteristic of our approach. Our algorithms are on the one hand based on a well-defined semantics and embedded in a clear and reasonable theoretical framework. On the other we applied them successfully and efficiently to a large set of different ontologies and alignments in the context of different application scenarios. The combination of both aspects is a major contribution of this work.

We also identified several interesting ideas and possible extensions. Qi et al. proposed relevance-based reasoning techniques that can possibly be used to speed-up our algorithms [QJH09]. The distinction between intended and unintended consequences is an interesting generalization of our approach [JRGHB09]. In case of missing disjointness axioms, we can prohibit certain types of entailments on top of a general framework. We find a counterpart in the pattern based constraints implemented on top of several matching systems. However, within the matcher community a clear distinction between incoherence and unintended consequence is missing. Only few matching systems focus on the semantic aspects of the generated alignment. These systems make use of a bundle of loosely coupled methods or constraints. Again, a well-defined semantics is missing.

One reason can be seen in the insufficient availability of relevant test datasets. The OAEI has coined the field of alignment evaluation over the last years. However, expressive ontologies that contain disjointness axioms and exhibit a rich modeling style do occur only in the OAEI CONFERENCE track. Thus, aspects that are more relevant for logically less expressive taxonomies (e.g., similarity computation and aggregation) have come to the fore in the development of competitive matching systems.

Chapter 12

Conclusions

Seven holy paths to hell, and your trip begins (Iron Maiden).

We started this thesis in the introduction with a listing of seven research questions. In Section 12.1 we revisit these questions and the answers we gave to them. Most of the previous chapters were concerned with one of these questions. Thus, we first give a summary of these chapters by answering R1 to R7, before we briefly discuss contributions of the other chapters and their role within this thesis.

In Section 12.2 we are concerned with future work. We focus on extensions and improvements of our algorithms and suggest additional experiments required to answer questions left open. In particular, we think about ways to apply our approach to less expressive ontologies, we discuss a more powerful user interface for revision support, we suggest *'white-box reasoning'* to speed up the algorithms, we talk about the integration of our approach into a matching system, and we propose an extension of our approach to match a whole network of ontologies in one step.

Finally, we end this thesis we some closing remarks in Section 12.3.

12.1 Summary

Our research questions can be divided in three groups assigned to the parts *'Motivation & Foundations'*, *'Methods'*, and *'Applications'*. The first two questions (R1 and R2) are concerned with the impact of alignment incoherence (R1) in an application scenario and with the relation between alignment incoherence and precision and recall (R2).

With respect to R1, we clearly showed that alignment incoherence results in severe problems for three important scenarios, in which an alignment is used as key component of an application (Section 3.2). These scenarios have been described as Terminological Reasoning, Data Transformation, and Query Processing. Instead of discussing the issue from a theoretical point of view, we presented an example to illustrate that the problems resulting from the use of an incoherent alignment cannot be avoided.

In Section 3.1 we discussed R2 by introducing an example of radical translation. We argued that ontology matching can be seen as a special case of radical translation. With the help of our example we illustrated and motivated the basic assumption underlying our approach. It says that an incoherent alignment contains at least one incorrect correspondence. While it is not possible to prove our assumption, we rebutted several counterarguments. Later, in Section 7.2, we used the assumption to show that the degree of incoherence results in a non-trivial upper bound for the precision of an alignment.

Research questions R3 and R4 are concerned with the design of algorithms that remove a subset Δ from an incoherent alignment $\mathcal{A}$ such that $\mathcal{A} \setminus \Delta$ is coherent. We have introduced a diagnosis as a minimal subset that fulfills this criteria. R3 is concerned with the definition of different types of diagnosis and R4 with the design of algorithms for computing certain types of alignment diagnosis.

We identified two types of diagnoses as a reasonable choice to resolve conflicts in an alignment. We referred to them as local optimal and global optimal diagnosis. We characterized the local optimal diagnosis by a recursive definition. The underlying idea is that any conflict is locally resolved in an optimal way, and the effects of this decision are propagated in an appropriate way such that a minimal hitting set is constructed. The global optimal diagnosis is defined as a diagnosis which is minimal with respect to its total of confidences. This diagnosis is motivated by the principle of minimal change. It removes as less correspondences – weighted by their confidence – as possible.

As answer to R4 we have designed two algorithms for computing local and global diagnoses. In both cases we first developed a straight-forward algorithm. Then we improved both algorithms by integrating efficient reasoning components. As a result we finally presented in Algorithm 8 and Algorithm 10 the pseudocode of our algorithms. Note that EFFICIENTGOD is an algorithm that computes a smallest weighted hitting set, which is known to be a NP-complete problem. At the same time it reduces reasoning in the aligned ontology to a minimum. We implemented all algorithms in an alignment debugging system that makes use of the Pellet reasoner to perform the required reasoning tasks.[1] This system has been used for the experiments reported about in the following.

In the third part of this thesis we were concerned with research questions R5, R6, and R7. We tried to clarify the impact of our diagnostic methods on precision and recall (R5), we measured the runtimes of our algorithms (R6) and analyzed the results, and we investigated in how far our approach can be used to support a user in the revision of an incoherent alignment (R7). Research questions R2 and R5 are closely related at first sight, however, there is also a clear distinction. Given an

[1] Appendix E informs about some implementation details and availability of our system.

incoherent alignment $\mathcal{A}$, R2 is concerned with conclusions that can be drawn about precision and recall of $\mathcal{A}$. Contrary to this, R5 is concerned with a comparison of $\mathcal{A}$ and $\mathcal{A} \setminus \Delta$ with respect to precision and recall. Hereby, Δ is a local or global diagnosis. R5 can only be answered by an experimental study.

Experiments were conducted for the scenario of *'Alignment Debugging'* and for the scenario of *'Alignment Extraction'*. Results show clearly that we can use our methods to debug an incoherent alignment without loosing quality in terms of precision and recall. Moreover, both local and global methods increase the quality of the alignment in terms of its f-measure. We observed a win in precision and a smaller loss in recall as general tendency. Results of local and global optimal diagnoses differ only slightly in most cases. However, the global approach produces less negative outliers. Incorrect decisions that result in a series of removals are avoided. As a consequence, results for the global optimal diagnosis are slightly better in average. We have also seen that the strength of impact depends critically on the expressivity of the ontologies to be aligned.

In the context of extracting an alignment from a rich set of matching hypotheses, we focused on the global optimal diagnosis only. In one of our experiments we used our approach to select an alignment from the set of correspondences generated by different matching systems. In particular, our experiments were based on the alignments submitted to the CONFERENCE track of OAEI 2010. In this setting, our approach constructs an alignment that is better than the best input alignment, without knowing which of the input alignments is the best alignment. Moreover, our diagnostic method had positive effects for nearly all threshold configurations.

Within this setting an additional research question – more specific than R5 – emerged. We analyzed whether a combined approach (extract best coherent 1:1 alignment) works better than a sequential approach (extract best 1:1 alignment first, then extract best coherent subalignment). We could not give an affirmative answer. Our results were ambiguous. Significant differences with respect to the average f-measure have not been observed. At least, we found a minor tendency that the combined approach favors recall over precision.

We measured runtimes of our algorithms in Section 8.3. We observed that the pattern based reasoning components reduce the runtime of our algorithms significantly. With respect to their effects, we have to distinguish between more expressive ontologies (BENCHMARK and CONFERENCE) and less expressive ontologies (ANATOMY). For the first group the use of the pattern based reasoning components results in a runtime that is approximately three times faster in average. This holds for local and global optimal algorithms. For less expressive ontologies the speed-up is even more significant. This is related to the fact that efficient reasoning components are nearly complete in such a setting.

We also compared our runtimes against the runtimes presented by Qi et al. in [QJH09]. Remember that these algorithms fail to construct a minimal hitting set, nor do they construct a solution that is optimal from some global point of view. Our algorithm for computing a local optimal diagnosis (efficient variant) is between 2.5 and 5 times faster than the fastest algorithm of Qi et al. Even more, the efficient variant of our global optimal algorithm is only two or three times slower than the fastest algorithm of Qi et al., which constructs a solution that shares characteristics similar to a local optimal diagnosis but fails to be a minimal hitting set.[2]

Finally, we analyzed in how far our approach can support a user in the revision of an alignment. We described a tool that we developed for this purpose. It informs a user about consequences of decisions that result in a logical conflict. We walked through a sequence of screenshots to illustrate the usage of our tool. Our example illustrated that a user can change his mind by taking cognizance of the unintended consequences of his decision. Our approach can thus increase the precision of the resulting alignment. If the user trusts in his decisions, our approach can decrease the effort of the user by reducing the number of correspondences that have to be evaluated manually. In our experiments we found some cases where the number of correspondences to be evaluated was cut down to the half.

Most chapters of this thesis were dedicated to a specific research question. In the following, we briefly discuss the contributions of those chapters that were not directly concerned with one of our research questions. In Chapter 2 we described syntax and semantics of ontologies and alignments. This chapter introduced a formal framework required to describe alignment incoherence in an appropriate way. In Section 5 we presented the reasoning techniques used as building blocks of the algorithms developed in the subsequent chapter. Chapter 7 was related to the motivation of your work. We measured the degree of alignment incoherence for current state of the art ontology matching systems.

We have presented related work in Chapter 11. Some of the approaches mentioned there are partially based or motivated by our work on the topic. We reported about an uptake of our approach by Qi et al. In [QHH+08] Qi et al. distinguish between a trusted TBox and a TBox that has to be revised. Qi et al. present alignment debugging, referring to our work, as a typical use case for their general approach, where the alignment represents the TBox to be revised. We also reported about the Protégé-Plugin, presented in [JRGHB08, JRGHB09], that supports a user in manually debugging an incoherent alignment. The authors point to our DDL-based approach as a predecessor of their work. Our focus on the manual effort that can be saved in the revision process has also influenced work by Nikitina et al. [Nik10, NRG11] on the topic of reasoning-supported interactive revision of knowledge bases.

[2]The coefficients presented in this paragraph are rough estimations based on the assumption that Pellet is, for the specific dataset chosen in these experiments, 1.5 to 2 times faster than KAON. Note that without this adjustment we would come to the conclusion that our algorithms are even more efficient. This conclusion is valid for a comparison of the overall system that consists of both our algorithms and Pellet. Note also that the experiments are restricted to a very specific pair of ontologies from the CONFERENCE dataset chosen by Qi et al.

12.2 Future Work

In the following we present some work that remains to be done. This is on the one hand related to possible improvements and extensions of our approach. On the other hand we have not always given a satisfactorily answer to our research questions or further questions have emerged.

Applicability on less expressive datasets

We have argued at several places that our approach has two benefits. (1) An incoherent alignment results in severe problems if it is used in one of the scenarios described in Section 3.2. (2) Coherent alignments are more precise than incoherent alignments. The second aspect can be exploited as a side effect of alignment debugging or within the process of revising an alignment. However, even a highly incorrect alignment will not be incoherent if the aligned ontologies do not contain axioms that express negation, e.g., disjointness axioms. This means that our approach is – without further extensions – not applicable to many ontologies. A possible solution is to enrich such ontologies with missing disjointness axioms. We have shown that this is possible by learning disjointness in [MST06].

However, our approach was based on supervised machine learning techniques and requires an sufficient set of training examples. An alternative approach called semantic clarification is proposed by Schlobach in [Sch05]. The method is based on the assumption that sibling classes are always disjoint unless adding a corresponding axiom does not introduce a conflict. This is a very strong assumption. However, it might make sense to include an option to our system, which allows to enrich the ontologies under discussion based on this approach.

Black Box vs. White Box Reasoning

We distinguished between two types of reasoning techniques that are used in our algorithms: techniques that use reasoning in the aligned ontology and a pattern based reasoning strategy, which requires to reason in each of the aligned ontologies separately. Another distinction, well-known from the field of ontology debugging, is the distinction between black-box and white-box approach. The latter is sometimes also referred to as glass-box approach [SHCvH07].

A black-box approach uses a reasoner as an oracle. The debugging algorithm asks a question and the reasoner returns an answer. Its is not important what happens internally in the reasoner. An example for a white-box approach is the top-down algorithm for calculating MUPS proposed in [SHCvH07]. Instead of asking a black-box many questions, one question is asked, and the reasoning (i.e., the tableau in case of a tableau-reasoner) that happens internally is traced and analyzed. This allows to conclude which axioms are involved in the entailment of an unsatisfiability, i.e., which axioms are contained in a MUPS.

Stuckenschmidt has evaluated several approaches for debugging incoherent ontologies in [Stu08]. He confirms the common knowledge and concludes that black-box approaches suffer from their computational complexity. Moving to a white-box approach might thus speed up our algorithms significantly. This is especially relevant for the components that use full-fledged black-box reasoning in the aligned ontology.

User Interface for Revision Support

We presented a prototypical web-based tool to illustrate the benefits of our approach in revising incoherent alignments. Our tool has already proofed its usability in a specific revision scenario, however, we also noticed several problems and drawbacks.

- Our tool is not well-suited for the evaluation of large alignments ($>$ 100 correspondences). The user interface is not designed to display consequences in an aggregated way, but only shows consequences related to correspondences currently displayed on the screen.
- Sometimes it is required to explore details related to the context of a concept or property that appears in a correspondence. Currently our tool displays only subclasses and domain/range restrictions.
- The conflicts contained in an alignment have to be computed in a preprocessing step. Currently, this step is not fully automated.

Even though these and other problems have not yet been solved, we believe that the main approach is very useful for many revision tasks. Some of these problems can be solved by redeveloping our tool as a Protégé Plugin [KFNM04]. Another benefit of offering the functionality of our tool in the Protégé framework is a wide community who is familiar with the use of Protégé.

Integration in a Matching System

In Section 9.2 we experimented with a simple matching system that uses our algorithms not just within a sequential approach in a post-processing step but in an integrated way. Contrary to our expectations, the results of the integrated approach could not exceed the results of the sequential approach. However, we still believe in the benefits of an integrated solution. Especially, when we have a rich set of matching hypotheses, a reasonable confidence distribution α, and expressive ontologies.

Our work in supporting the development of CODI [NN10] was a first step towards a better understanding of integrating our approach into a complex matching system. CODI uses both soft and hard constraints that guide the matching process. It defines a matching problem as an optimization problem that takes care of both types of constraints. If we ignore the soft constraints the correspondences filtered out by CODI nearly coincide with a global optimal diagnosis. CODI has generated the best alignments in terms of f-measure for the CONFERENCE track of OAEI 2010. Further improvements have to aim at generating good alignments with higher recall. Alignments generated by CODI have a very low degree of incoherence, however, CODI cannot guarantee coherence of the generated alignments in general. We already analyzed how to extend CODI with complete reasoning strategies.

Incoherence and Entailment

Within this thesis we focused on the role of alignment incoherence in the context of ontology matching. We dealt with relations between correspondences that can be expressed in a propositional form as $c_1 \wedge \ldots \wedge c_n \rightarrow \neg c'$. What about formulae of type $c_1 \wedge \ldots \wedge c_n \rightarrow c'$? This formula expresses an entailment between correspondences. As long as we focus on equivalence correspondences – as it is currently state of the art in ontology matching – such entailments will occur rarely in a standard ontology matching scenario.

This differs if we analyze a scenario where we pairwise match several ontologies $\mathcal{O}_1, \ldots, \mathcal{O}_n$ at the same time. In this case we might have correspondence $\langle X_{\#1}, Y_{\#2}, \equiv\rangle$ between $\mathcal{O}_1$ and $\mathcal{O}_2$ and $\langle Y_{\#2}, Z_{\#3}, \equiv\rangle$ between $\mathcal{O}_2$ and $\mathcal{O}_3$. Due to the transitivity of $\equiv$ we can conclude that $\langle X_{\#1}, Z_{\#3}, \equiv\rangle$. Thus, we have an instance for the second type of formulae. But what happens if there is at the same time evidence for a correspondence that allows to entail that $\langle X_{\#1}, Z_{\#3}, \equiv\rangle$ cannot be the case? The resulting optimization problem becomes more complex and interdependencies between correspondences and ontologies are propagated in a network of aligned ontologies.

We have already started to work on the this topic. First results indicate that the f-measure of the generated alignments is significantly higher compared to the alignments that result from debugging each alignment on its own. Moreover, the approach increases both precision and recall of the alignment. Incorrect correspondences are eliminated and new, non-trivial correspondences are derived.

12.3 Closing Remarks

Shvaiko and Euzenat discussed *'reasoning with alignments'* as one of ten major challenges for ontology matching. They referred to our work [MTS07] as a first example. Orsio and Tanca [OT10] describe one of our publications [MTS07] as a *'seminal work on mapping revision'*. Pathak et al. [PJC09] have argued about the need to extend our older work from DDL to *'a more general setting that will enable resolution of inconsistencies between ontology modules'*. Note that we realized this proposal within this thesis. We conclude that our approach has been reviewed positively in several survey papers.

We have also discussed the influence of our approach on the work presented by Qi et al. [QJH09], Jimenez-Ruiz et al. [JRGHB09], and Nikitina [Nik10]. The approach described in the cited papers is, very similar to the approach of this thesis, based on a well-defined alignment semantics. It seems that our work had a significant impact on work that is, first of all, grounded on a well defined theoretical framework.

Contrary to this, the adaption of our approach by developers of ontology matching systems seems to be limited. We started to evaluate alignment coherence as part of the OAEI CONFERENCE track in 2008 [CEH+08]. From 2008 to 2010 we could not observe a significant change in the degree of alignment incoherence. There seems to be a gap between approaches that are based on a well-defined alignment semantics and the pragmatic considerations that guide the design of ontology matching systems. This situation is reinforced by the fact that many tracks at OAEI use inexpressive ontologies, in most cases without disjointness axioms. There is some evidence that system developers focus on good results in some of these tracks (see [EMS+11]). The results achieved in the CONFERENCE track play a subordinate role, this holds especially for the evaluation concerned with alignment coherence.

While the incoherence of an ontology is treated as a clear symptom of a modeling error [SC03], alignment incoherence still plays a subordinate role in the field of ontology matching. In this thesis we emphasized again the importance of alignment coherence. Moreover, we presented experimental results to show that alignment coherence is a powerful means to exploit the semantics of the aligned ontologies. Moreover, we published for the first time a comprehensive and detailed presentation of stable and efficient algorithms to debug incoherent alignments. We made these algorithms available in a system, briefly described in Appendix E. We hope that the precise presentation of our approach, its extensive analysis in a set of experiments, and the availability of a tool for repairing incoherent alignments helps to foster the further uptake of our approach.

Part V

Appendix

Appendix A

Further Alignment Incoherence Examples

Table A.1 illustrates an incoherent alignment that is an example for a MIPS with four elements. The existence of this example shows that it is not sufficient to look at all subsets of an alignment $\mathcal{A}$ with one, two, or three elements to find all MIPS in $\mathcal{A}$. Note that we did not detect MIPS of cardinality higher than four in our experiments, however, it is obvious that such MIPS might occur for other datasets.

The example shows also that it is sometimes hard to decide which involved correspondence is incorrect. Analyzing the ontologies reveals that probably correspondence $\langle \mathit{Proceedings}_{\#1}, \mathit{Proceedings}_{\#2}, \equiv \rangle$ or $\langle \mathit{contains}_{\#1}, \mathit{includes}_{\#2}, \equiv \rangle$ is an incorrect correspondence. $\mathit{Proceedings}_{\#1}$ are defined to $\mathit{contain}_{\#1}$ only documents that are instances of $\mathit{AcceptedPaper}_{\#1}$ or $\mathit{InvitedTalkAbstract}_{\#1}$, while $\mathit{Proceedings}_{\#2}$ $\mathit{include}_{\#2}$ always a $\mathit{PCMembersList}_{\#2}$. This results in a conflict together with the other correspondences listed.

Alignment	
(1) $\langle Paper_{\#1}, Paper_{\#2}, \equiv\rangle$	
(2) $\langle Abstract_{\#1}, PaperAbstract_{\#2}, \equiv\rangle$	
(3) $\langle Proceedings_{\#1}, Proceedings_{\#2}, \equiv\rangle$	
(4) $\langle contains_{\#1}, includes_{\#2}, \equiv\rangle$	
Axioms of ontology #1	**Axioms of ontology #2**
(5) $AcceptedPaper \sqsubseteq \ldots \sqsubseteq Paper$	(8) $Paper \sqsubseteq Doc$
(6) $InvitedTalkAbstract \sqsubseteq Abstract$	(9) $PaperAbstract \sqsubseteq Doc$
(7) $Proceedings \sqsubseteq \forall contains.$ $(AcceptedPaper \sqcup InvitedTalkAbstract)$	(10) $Proceedings \sqsubseteq \exists includes.$ $PCMembersList$
	(11) $PCMembersList \sqsubseteq OutputDoc$
	(12) $OutputDoc \sqsubseteq \neg Doc$
Entailments	
(13) $Paper_{\#1} \equiv Paper_{\#2}$	*from (1)*
(14) $Abstract_{\#1} \equiv PaperAbstract_{\#2}$	*from (2)*
(15) $Proceedings_{\#1} \equiv Proceedings_{\#2}$	*from (3)*
(16) $contains_{\#1} \equiv includes_{\#2}$	*from (4)*
(17) $AcceptedPaper_{\#1} \sqsubseteq Doc_{\#2}$	*from (5), (8) and (13)*
(18) $InvitedTalkAbstract_{\#1} \sqsubseteq Doc_{\#2}$	*from (6), (9) and (14)*
(19) $Proceedings_{\#1} \sqsubseteq \forall contains_{\#1}.Doc_{\#2}$	*from (7), (17) and (18)*
(20) $Proceedings_{\#2} \sqsubseteq \forall includes_{\#2}.Doc_{\#2}$	*from (15), (16) and (19)*
(21) $Proceedings_{\#2} \sqsubseteq \exists includes_{\#2}.OutputDoc_{\#2}$	*from (10) and (11)*
(22) $Proceedings_{\#2} \sqsubseteq \bot$	*from (20) and (21)*

Table A.1: Alignment with four elements that is incoherent due to a complex interaction between many involved axioms.

Appendix B

Modifications of the Benchmark Dataset

The ontologies of the BENCHMARK dataset do not contain disjointness axioms. For that reason we have added the following disjointness axioms to the reference ontology of the dataset. The reference ontology is the ontology that has to be matched on a variant of itself (#1xx and #2xx series) or on another ontology (#3xx series) for each of the testcases.

```
Person
Disjoint with: Organization List Journal PageRange Date
Conference Reference Address

MastersThesis
Disjoint with: PhdThesis

Report
Disjoint with: Misc MotionPicture Part Book Academic

Misc
Disjoint with: Report MotionPicture Part Book Academic

MotionPicture
Disjoint with: Report Misc Part Book Informal Academic

List
Disjoint with: Person Organization Journal PageRange
Date Conference Reference Address

Book
Disjoint with: MotionPicture Misc Report Part Informal
Academic

Journal
Disjoint with: Person Organization List Date PageRange
Conference Reference Address
```

```
Part
Disjoint with: MotionPicture Report Misc Book Informal
Academic

PageRange
Disjoint with: Person Organization List Journal Date
Conference Reference Address

Conference
Disjoint with: Person Organization List Journal PageRange
Date Reference Address

Reference
Disjoint with: Person Organization List Journal PageRange
Date Conference Address

PhdThesis
Disjoint with: MastersThesis

Organization
Disjoint with: Person List Journal PageRange Date Conference
Reference Address

Date
Disjoint with: Person Organization List Journal PageRange
Conference Reference Address

Informal
Disjoint with: MotionPicture Part Book

Academic
Disjoint with: Report Misc MotionPicture Part Book

Address
Disjoint with: Person Organization List Journal Date PageRange
Conference Reference
```

These axioms have been added following the rule of thumb to define sibling concepts as disjoint as long as there are no reasons against this procedure. There are only a few exceptions from the general rule.

Appendix C

Listing of Matching Systems

In the following we list the systems participating in OAEI 2008 to 2010 in alphabetical order. Within the system descriptions we try to focus on aspects related to the reduction of incoherences.

AFLOOD (ANCHORFLOOD) is a matching system for matching large ontologies efficiently [SA09]. It first determines a set of anchor correspondences. Based on this seed, AFLOOD collects blocks of neighboring concepts. The concepts and properties within these blocks are compared and possibly aligned. This process is repeated where each newly found correspondence is used as seed. This strategy reduces the time required to generate an alignment significantly. AFLOOD is not known to comprise a component that ensures the coherence of the generated alignment. The system participated in the OAEI in 2008 and 2009.

AGREEMENTMAKER offers a user interface built on an extensible architecture. This architecture allows to configure the matching process to a high degree. For that purpose AGREEMENTMAKER uses internally different methods and similarity measures that can be combined in different ways. A component dedicated to the detection and avoidance of incoherences is not mentioned. AGREEMENTMAKER participated in 2009 and 2010 [CSC+10] with good results in the CONFERENCE and ANATOMY track.

AROMA divides the matching process in three successive steps [Dav08]. The final step aims at cleaning and enhancing the resulting alignment. This is done, for example, by removing redundant correspondences. While cycles in the alignment graph are suppressed in this step, the suppression of incoherence is not mentioned. AROMA participated in the OAEI from 2008 to 2010.

ASMOV is an abbreviation for *'Automated Semantic Mapping of Ontologies with Validation'*. The term *'validation'* refers to a technique that avoids erroneous combinations of correspondences. In Section 11.4, we argue that this strategy is closely related to the pattern based reasoning method defined in Section 5.2. In [JMSK09] the system is described in detail. ASMOV participated in the OAEI consecutively from 2007 to 2010. It was one of the top performers in the BENCHMARK track and participated also in many other tracks with good results.

BLOOMS is a method mainly intended for the application to biomedical ontologies [PSCC10]. It has a sequential architecture composed of three techniques. While the first two techniques compute lexical similarities, the third is based on the propagation of previously calculated similarities throughout the ontology graph. BLOOMS participated in the OAEI in 2010 for the first time. BLOOMS has no specific component dedicated to the avoidance of incoherence.

CODI is an abbreviation for *'Combinatorial Optimization for Data Integration'*. The system is based on the syntax and semantics of Markov logic [DLK$^+$08]. It transforms the alignment problem to a maximum-a-posteriori optimization problem. The author of this thesis has been involved in the development of the system [NMS10]. Coherence and consistency are taken into account by a set of hard constraints inspired by the patterns presented in Section 5.2. CODI participated in 2010 for the first time [NN10] and generated good results for the CONFERENCE track.

DSSIM The authors of the system focus on uncertainty related to the matching process. In order to represent and reason with this uncertainty they propose the Dempster-Shafer theory. In this context the authors developed an fuzzy voting model approach [NVVM08] to resolve conflicting beliefs given the belief in the correctness of a correspondence. However, a conflict is in this context not related to a logical contradiction. DSSIM participated in the OAEI consecutively from 2006 to 2009.

EFF2MAT is a new matching system that participated in 2010 for the first time. Similar to the AFLOOD system, it focuses on the efficiency of the matching process and makes also use of anchors to speed it up. In addition, it uses hash methods to enable a fast comparison of relevant entities. Methods related to the coherence of the generated alignment are not mentioned in the system description in [CK10].

FALCON (more precisely Falcon-AO) is the experienced matching system that still participates in the OAEI. It has been developed since 2005. An overview of the main components is given in [HQ08]. It comprises linguistic matching methods, structural methods, and methods to combine different similarity scores. A partition-based method for block matching can be applied to match large-scale ontologies. FALCON obtained top results in the BENCHMARK track in the early years of the OAEI. A component for generating coherent alignments is to our knowledge not implemented in the FALCON system.

GEROME (or GEROMESUITE) is a a generic model management tool [QGKL08]. The tool is thus well suited for matching ontologies described in heterogeneous modeling languages. It provides several well known matching strategies, however, a component dedicated to alignment coherence is not included. GEROME has participated in 2008, 2009 and in 2010 as GEROME-SMB, an extension of the system.

LILY participated from 2007 to 2009 at the OAEI. While in 2007 a component related to alignment coherence was not mentioned, the system has been extended in 2008 by a component for mapping debugging [WX08b, WX08a]. This component can be used to detect some types of mapping errors referred to as redundancy and inconsistency. While some conflicts have to be resolved by a human user, some of them are resolved automatically without user interaction. We discuss the approach in detail in Section 11.4.

MAPPSO is an ontology alignment approach based on discrete particle swarm optimization [Boc10]. The core element of the underlying optimization problem is the objective function which supplies a fitness value for each candidate alignment. For each correspondence the quality score is calculated based on an aggregation of scores from a configurable set of base matchers. These base matchers are string-based, use wordnet, and analyze structural properties. A component dedicated to alignment coherence is not known. MAPPSO participated in the OAEI from 2008 to 2010.

NBJLM uses a matching strategy that considers literal similarity and structural similarity, simultaneously. The computation of the literal similarity involves the use of WordNet. To the best of our knowledge, the computation of the structural similarity does not include alignment coherence. NBJLM participated in the OAEI only in 2010 at the ANATOMY track [WWL10].

RIMOM participated in the OAEI from 2006 to 2010 in many tracks. It has continuously been extended and improved over the years. In 2010 it used a combination of a name based strategy (edit distance between labels), a vector based strategy (cosine similarity between vectors), and a strategy taking instances of concepts into account [WZH+10]. A reasoning component dedicated to the avoidance of alignment incoherence is not known.

SOBOM The unique feature of SOBOM is to combine sub-ontology extraction with ontology matching. As one of its components it uses a strategy called Semantic Inductive Similarity Flooding. It is used the propagate the similarity between previously detected anchor alignments along the structure defined by the axioms in the ontologies. SOBOM participated in the OAEI in 2009 and 2010 [XWCZ10].

TAXOMAP is an alignment tool which aims to discover different types of correspondences expressing equivalence, subsumption or proximity relations. It takes into account labels and sub-class descriptions [HSNR10]. As part of the TAXOMAP framework it is possible for a domain expert to define rules that are used to refine (filter or extend) the set of automatically generated correspondences. This approach might also allow to specify rules that filter out certain incoherent combinations of correspondences. However, an application like this is not mentioned by the authors of the system. TAXOMAP participated in the OAEI from 2007 to 2010.

We conclude that only few matching systems take alignment coherence into account. Leaving aside CODI, which uses some of the ideas presented in this thesis, we identified ASMOV and LILY as systems that try to ensure the coherence of the generated alignment. However, a precise definition of alignment coherence is not given in the context of a model theoretic semantics. As a consequence the analysis of the *'completeness'* of the approach is missing.

Appendix D

Additional Experimental Results

In Section 8.1 we showed Figure 8.2 and 8.1 to illustrate the differences between local optimal diagnosis and global optimal diagnosis in detail for each matching task. In the following we use the same kind of illustration for all matching systems participating in the CONFERENCE OAEI 2010 track. Each figure shows 21 bars for the testcases that result from matching all combinations from the set CMT, CONFERENCE, CONFOF, EDAS, EKAW, IASTED, SIGKDD in the sequence CMT-CONFERENCE, CMT-CONFOF, ..., IASTED-SIGKDD.

To avoid misunderstandings while analyzing these figures, we would like to point to the fact that the f-measure is the harmonic and not the arithmetic mean of precision and recall. This has, at first sight, some uncommon effects. If we compare two precision, recall, f-measure triples $\langle p, r, f \rangle$ and $\langle p', r', f' \rangle$ it might occur that $r = r'$ and $|p - p'| < |f - f'|$. This occurs in some of the figures depicted below. We find this effect, for example, for the 5th testcase of SOBOM in Figure D.10.

Table D shows the results for extracting the merged alignments of the CONFERENCE dataset. We applied different thresholds to see how the algorithms work with input alignments of different sizes. We have noted the total number of correspondences in all alignments after applying a certain threshold in the first column. For each threshold we computed precision (p), recall (r), and f-measure (f) for each of the four methods. Further explanations on the experimental setting are given in Section 9.3.

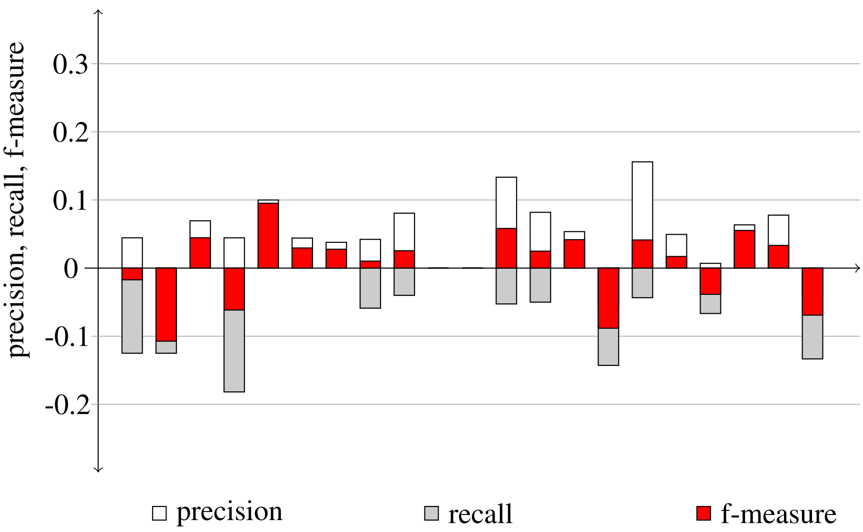

Figure D.1: Effects of debugging alignments of the AGREEMENTMAKER system computing local optimal diagnoses.

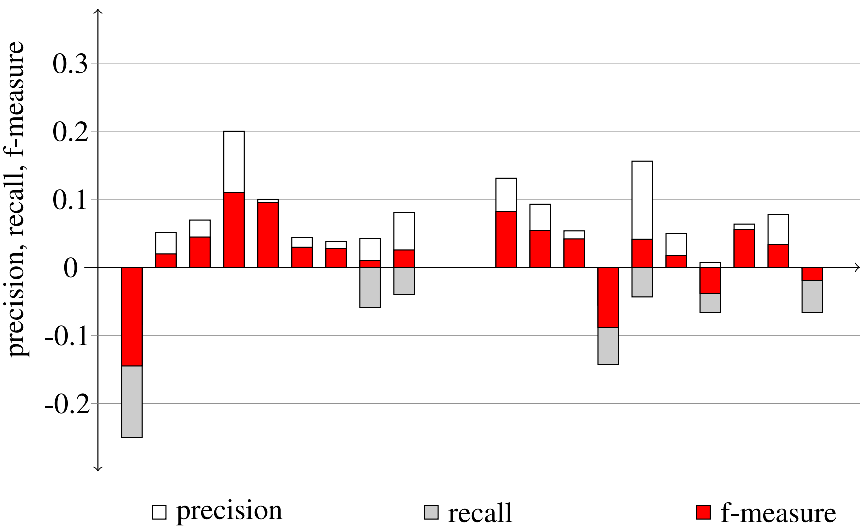

Figure D.2: Effects of debugging alignments of the AGREEMENTMAKER system computing global optimal diagnoses.

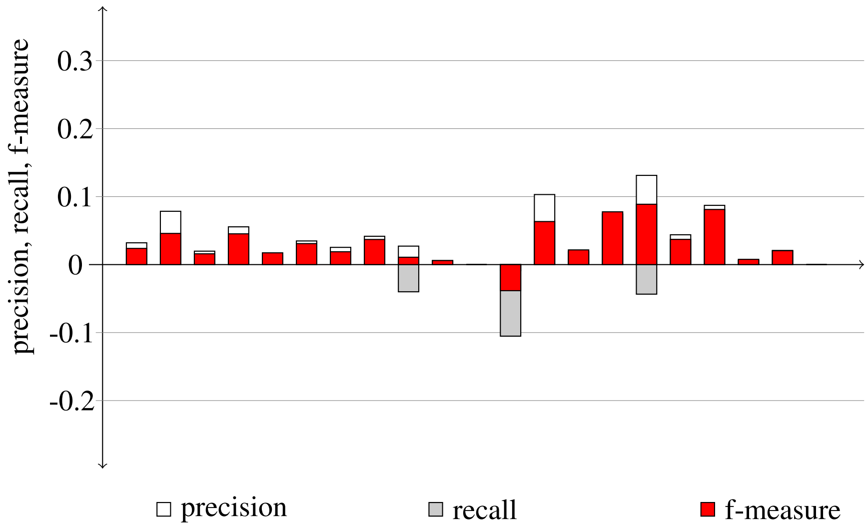

Figure D.3: Effects of debugging alignments of the ASMOV system computing local optimal diagnoses.

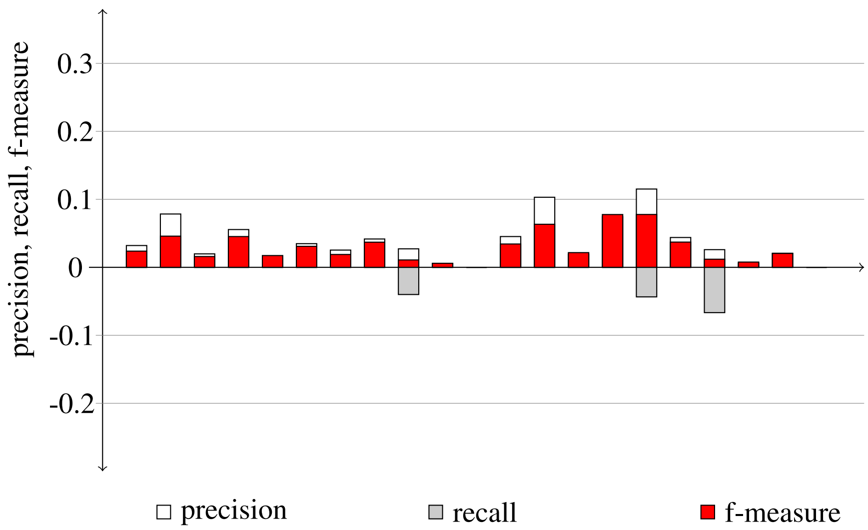

Figure D.4: Effects of debugging alignments of the ASMOV system computing global optimal diagnoses.

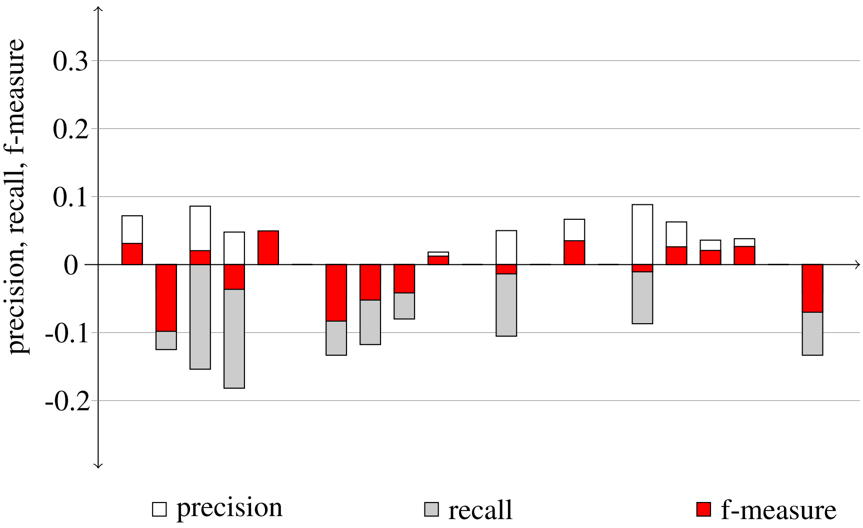

Figure D.5: Effects of debugging alignments of the EFF2MATCH system computing local optimal diagnoses.

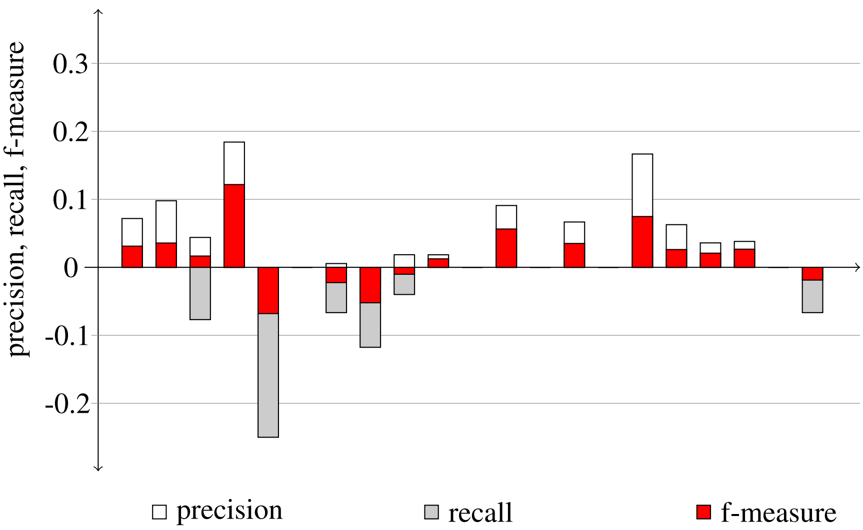

Figure D.6: Effects of debugging alignments of the EFF2MATCH system computing global optimal diagnoses.

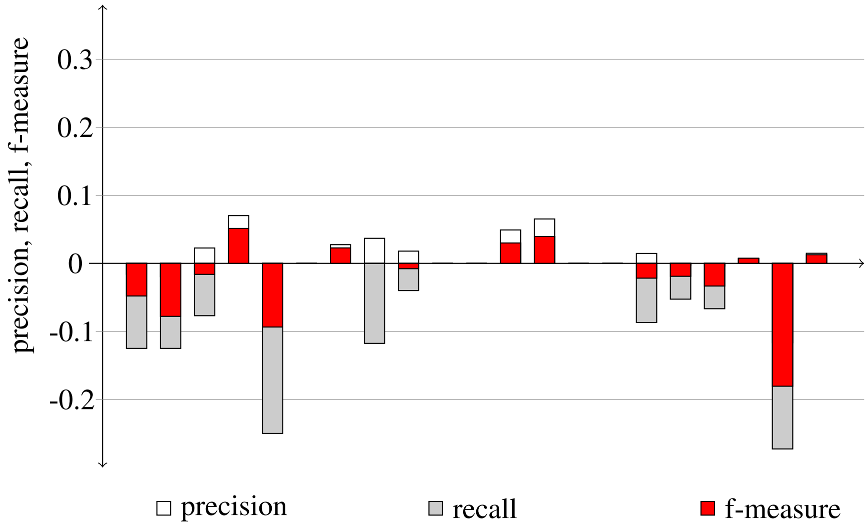

Figure D.7: Effects of debugging alignments of the GERMESMB system computing local optimal diagnoses.

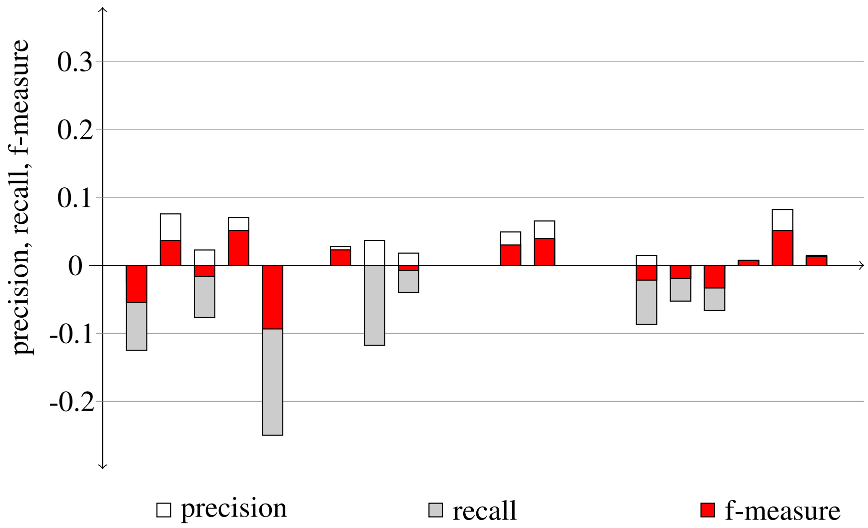

Figure D.8: Effects of debugging alignments of the GERMESMB system computing global optimal diagnoses.

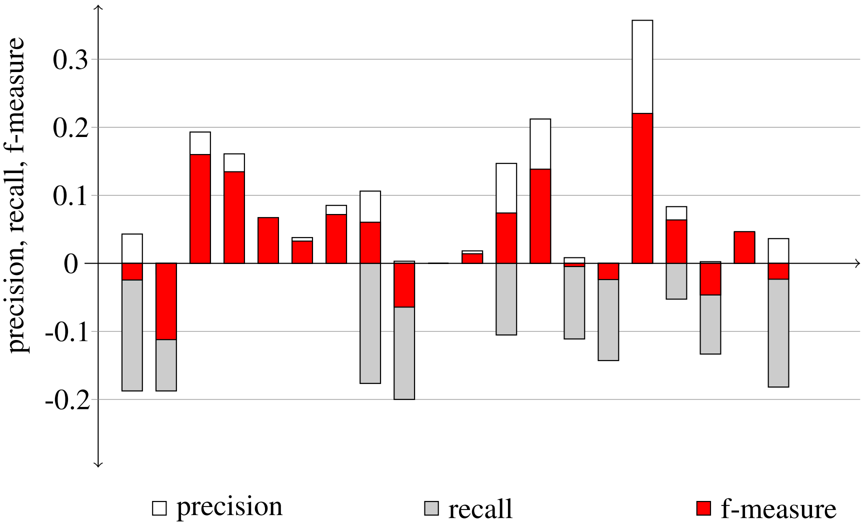

Figure D.9: Effects of debugging alignments of the SOBOM system computing local optimal diagnoses.

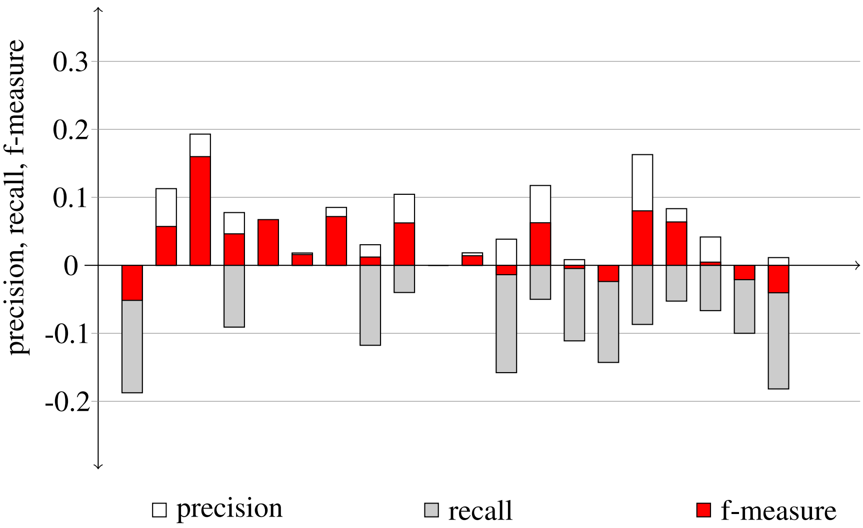

Figure D.10: Effects of debugging alignments of the SOBOM system computing global optimal diagnoses.

	threshold only			1:1			1:1 $\rightarrow \Delta_{global}$			$\Delta_{global \wedge 1:1}$		
#	p	f	r	p	f	r	p	f	r	p	f	r
64	0.844	0.292	0.176	0.844	0.292	0.176	0.844	0.292	0.176	0.844	0.292	0.176
92	0.848	0.392	0.255	0.848	0.392	0.255	0.857	0.393	0.255	0.857	0.393	0.255
120	0.85	0.479	0.333	0.85	0.479	0.333	0.857	0.48	0.333	0.857	0.48	0.333
139	0.82	0.512	0.373	0.82	0.512	0.373	0.824	0.507	0.366	0.824	0.507	0.366
162	0.809	0.56	0.428	0.809	0.56	0.428	0.822	0.557	0.422	0.822	0.557	0.422
173	0.803	0.58	0.454	0.803	0.58	0.454	0.82	0.579	0.448	0.82	0.579	0.448
184	0.777	0.584	0.467	0.777	0.584	0.467	0.797	0.584	0.461	0.797	0.584	0.461
197	0.777	0.608	0.5	0.777	0.608	0.5	0.799	0.61	0.493	0.799	0.61	0.493
205	0.766	0.614	0.513	0.766	0.614	0.513	0.792	0.62	0.51	0.792	0.62	0.51
208	0.764	0.619	0.52	0.764	0.619	0.52	0.79	0.625	0.516	0.79	0.625	0.516
213	0.746	0.613	0.52	0.746	0.613	0.52	0.782	0.622	0.516	0.782	0.622	0.516
218	0.734	0.611	0.523	0.734	0.611	0.523	0.768	0.62	0.52	0.768	0.62	0.52
224	0.719	0.608	0.526	0.719	0.608	0.526	0.751	0.617	0.523	0.751	0.617	0.523
226	0.717	0.609	0.529	0.717	0.609	0.529	0.749	0.618	0.526	0.749	0.618	0.526
228	0.715	0.61	0.533	0.715	0.61	0.533	0.747	0.62	0.529	0.747	0.62	0.529
234	0.709	0.615	0.542	0.709	0.615	0.542	0.74	0.624	0.539	0.74	0.624	0.539
244	0.701	0.622	0.559	0.701	0.622	0.559	0.73	0.631	0.556	0.73	0.631	0.556
253	0.684	0.619	0.565	0.687	0.62	0.565	0.718	0.629	0.559	0.718	0.629	0.559
268	0.664	0.62	0.582	0.667	0.621	0.582	0.707	0.634	0.575	0.707	0.634	0.575
278	0.651	0.62	0.592	0.656	0.622	0.592	0.696	0.636	0.585	0.696	0.636	0.585
294	0.629	0.617	0.605	0.636	0.62	0.605	0.679	0.634	0.595	0.679	0.634	0.595
307	0.612	0.613	0.614	0.62	0.617	0.614	0.668	0.635	0.605	0.668	0.635	0.605
325	0.585	0.602	0.621	0.596	0.608	0.621	0.648	0.627	0.608	0.646	0.626	0.608
360	0.553	0.598	0.65	0.566	0.602	0.644	0.617	0.624	0.631	0.606	0.615	0.624
383	0.533	0.592	0.667	0.552	0.601	0.66	0.602	0.622	0.644	0.593	0.614	0.637
440	0.475	0.56	0.683	0.507	0.579	0.673	0.563	0.606	0.657	0.556	0.604	0.66
530	0.402	0.51	0.696	0.447	0.54	0.683	0.505	0.572	0.66	0.499	0.571	0.667
584	0.372	0.488	0.709	0.418	0.52	0.69	0.477	0.556	0.667	0.472	0.556	0.676
639	0.349	0.472	0.729	0.402	0.512	0.706	0.47	0.557	0.683	0.463	0.555	0.693
715	0.322	0.451	0.752	0.381	0.499	0.722	0.445	0.542	0.693	0.433	0.534	0.696
809	0.293	0.425	0.775	0.356	0.48	0.735	0.424	0.528	0.699	0.411	0.52	0.706
930	0.265	0.398	0.804	0.332	0.46	0.745	0.406	0.517	0.712	0.396	0.513	0.729
1025	0.241	0.371	0.807	0.314	0.442	0.745	0.388	0.503	0.716	0.376	0.498	0.735

Table D.1: Additional experimental results: Extracting from merged alignments of the CONFERENCE dataset.

Appendix E

System Availability

We developed a system called ALCOMO for conducting our experiments. The system is freely available under MIT License. The abbreviation ALCOMO stands for Applying Logical Constraints on Matching Ontologies. ALCOMO uses internally the Pellet Reasoner in version 2.2.2 and the OWL API in version 3.1.0. A minimal documentation and a usage example of the system is available at the systems homepage `http://web.informatik.uni-mannheim.de/alcomo/`.

Bibliography

[AB09] Leila Amgoud and Philippe Besnard. Bridging the gap between abstract argumentation systems and logic. In Lluis Godo and Andrea Pugliese, editors, Proceedings of the 3rd International Conference on Scalable UncertaintyManagement, pages 12–27. Springer, 2009.

[ABEZS09] Sivan Albagli, Rachel Ben-Eliyahu-Zohary, and Solomon E. Shimony. Markov network based ontology matching. In Craig Boutilier, editor, Proceedings of the 21st International Joint Conference on Artificial Intelligence, pages 1884–1889. AAAI Publications, 2009.

[AvH09] Grigoris Antaniou and Frank van Harmelen. Web Ontology Language: OWL. In Steffen Staab and Rudi Studer, editors, Handbook on Ontologies, pages 67–92. Springer, 2009.

[BCM+03] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Pater Patel-Schneider. The Description Logic Handbook: theory, implementation, and applications. Cambridge University Press, 2003.

[BGvH+04] Paolo Bouquet, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. Contextualizing ontologies. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 1(4):325–343, 2004.

[BHR+05] Oliver Bodenreider, Terry F. Hayamizu, Martin Ringwald, Sherri De Coronado, and Songmao Zhang. Of mice and men: Aligning mouse and human anatomies. In Proceedings of the American Medical Informatics Association (AIMA), Annual Symposium, pages 61–65, 2005.

[BLHL01] Tim Berners-Lee, Jim Hendler, and Ora Lassila. The semantic web. Scientific American, 284(5):34–43, 2001.

[BN03] Franz Baader and Werner Nutt. Basic description logics. In Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Pater Patel-Schneider, editors, The Description Logic Handbook, pages 43–95. Cambridge University Press, 2003.

[Boc10] Jürgen Bock. MapPSO results for OAEI 2010. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel Cruz, editors, Proceedings of the 5th International Workshop on Ontology Matching (OM-2010) collocated with the 9th International Semantic Web Conference (ISWC-2010), pages 180–186. CEUR-WS, 2010.

[BS03] Alex Borgida and Luciano Serafini. Distributed description logics: Assimilating information from peer sources. Journal on Data Semantics, I:153–184, 2003.

[CEH+08] Caterina Caracciolo, Jérôme Euzenat, Laura Hollink, Ryutaro Ichise, Antoine Isaac, Véronique Malaisé, Christian Meilicke, Juan Pane, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, and Vojtech Svátek. Results of the ontology alignment evaluation initiative 2008. In Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) collocated with the 7th International Semantic Web Conference (ISWC-2008), pages 73–119. CEUR-WS, 2008.

[CK10] Watson Wei Khong Chua and Jung-Jae Kim. Eff2Match results for OAEI 2010. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel Cruz, editors, Proceedings of the 5th International Workshop on Ontology Matching (OM-2010) collocated with the 9th International Semantic Web Conference (ISWC-2010), pages 150–157. CEUR-WS, 2010.

[Cri98] Mark Crimmins. Philosophy of language. In E. Craig, editor, Routledge Encyclopedia of Philosophy. Routledge, 1998.

[CSC+10] Isabel F. Cruz, Cosmin Stroe, Michele Caci, Federico Caimi, Matteo Palmonari, Flavio, Palandri Antonelli, and Ulas C. Keles. Using agreementmaker to align ontologies for OAEI 2010. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel Cruz, editors, Proceedings of the 5th International Workshop on Ontology Matching (OM-2010) collocated with the 9th International Semantic Web Conference (ISWC-2010), pages 118–125. CEUR-WS, 2010.

[CSM+10] Gianluca Correndo, Manuel Salvadores, Ian Millard, Hugh Glaser, and Nigel Shadbolt. SPARQL query rewriting for implementing data integration over linked data. In Florian Daniel, Lois M. L. Delcambre, Farshad Fotouhi, Irene Garrigós, Giovanna Guerrini, Jose-Norberto Mazón, Marco Mesiti, Sascha Müller-Feuerstein, Juan Trujillo, Traian Marius Truta, Bernhard Volz, Emmanuel

Waller, Li Xiong, and Esteban Zimányi, editors, Proceedings of the International Workshop on Data Semantics, pages 1–11. ACM, 2010.

[Dav74] Donald Davidson. On the very idea of a conceptual scheme. Proceedings and Addresses of the American Philosophical Association, 47:5–20, 1974.

[Dav08] Jérôme David. AROMA results for OAEI 2008. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Heiner Stuckenschmidt, editors, Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) collocated with the 7th International Semantic Web Conference (ISWC-2008), pages 128–131. CEUR-WS, 2008.

[DESdS11] Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment API 4.0. Semantic Web: Interoperability, Usability, Applicability, 2(1), 2011.

[DLK+08] Pedro Domingos, Daniel Lowd, Stanley Kok, Hoifung Poon, Matthew Richardson, and Parag Singla. Just add weights: Markov logic for the semantic web. In Paulo Cesar G. da Costa, Claudia d'Amato, Nicola Fanizzi, Kathryn B. Laskey, Kenneth J. Laskey, Thomas Lukasiewicz, Matthias Nickles, and Michael Pool, editors, Uncertainty Reasoning for the Semantic Web I, ISWC International Workshops, URSW 2005-2007, Revised Selected and Invited Papers, pages 1–25. Springer, 2008.

[Doy90] Sir Arthur Conan Doyle. The Sign of the Four. Spencer Blackett, 1890.

[dSQV08] Cássia Trojahn dos Santos, Paulo Quaresma, and Renata Vieira. An argumentation framework based on confidence degrees to combine ontology mapping approaches. International Journal of Metadata, Semantics and Ontologies, 3(2):142–150, 2008.

[dSQVI10] Cássia Trojahn dos Santos, Paulo Quaresma, Renata Vieira, and Antoine Isaac. Comparing argumentation frameworks for composite ontology matching. In Peter McBurney, Iyad Rahwan, Simon Parsons, and Nicolas Maudet, editors, Argumentation in Multi-Agent Systems, 6th International Workshop, ArgMAS 2009, pages 305–320. Springer, 2010.

[Dun95] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and nperson games. Artificial Intelligence, 77(2):321–357, 1995.

[EFH+09] Jérôme Euzenat, Alfio Ferrara, Laura Hollink, Antoine Isaac, Cliff Joslyn, Véronique Malaisé, Christian Meilicke, Andriy Nikolov, Juan Pane, Marta Sabou, François Scharffe, Pavel Shvaiko, Vassilis Spiliopoulos, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, Cássia Trojahn dos Santos, George A. Vouros, and Shenghui Wang. Results of the ontology alignment evaluation initiative 2009. In Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009), pages 73–126. CEUR-WS, 2009.

[EFM+10] Jérôme Euzenat, Alfio Ferrara, Christian Meilicke, Andriy Nikolov, Juan Pane, François Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, and Cássia Trojahn. Results of the ontology alignment evaluation initiative 2010. In Proceedings of the 5th International Workshop on Ontology Matching (OM-2010) collocated with the 9th International Semantic Web Conference (ISWC-2010), pages 85–115. CEUR-WS, 2010.

[Ehr05] Marc Ehrig. Relaxed precision and recall for ontology matching. In Benjamin Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt, editors, Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies. CEUR-WS, 2005.

[EIM+07] Jérôme Euzenat, Antoine Isaac, Christian Meilicke, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb, Vojtech Svátek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2007. In Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) collocated with the 7th International Semantic Web Conference (ISWC-2007), pages 96–132. CEUR-WS, 2007.

[EMS09] Kai Eckert, Christian Meilicke, and Heiner Stuckenschmidt. Improving ontology matching using meta-level learning. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Paslaru Bontas Simperl, editors, The Semantic Web: Research and Applications: Proceedings of the 6th European Semantic Web Conference, pages 158–172. Springer, 2009.

[EMS+11] Jérôme Euzenat, Christian Meilicke, Heiner Stuckenschmidt, Pavel Shvaiko, and Cássia Trojahn. Ontology alignment evaluation initiative: six years of experience. Journal on Data Semantics, XV:158–192, 2011.

[EPS08] Jérôme Euzenat, Axel Polleres, and François Scharffe. Processing ontology alignments with SPARQL. In Fatos Xhafa and Leonard Barolli, editors, Second International Conference on Complex, Intelligent and Software Intensive Systems (CISIS-2008), pages 913–917. IEEE Computer Society, 2008.

[ES07] Jérôme Euzenat and Pavel Shvaiko. Ontology Matching. Springer, 2007.

[Euz04] Jérôme Euzenat. An API for ontology alignment. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, The Semantic Web - ISWC 2004: Third International Semantic Web Conference, pages 698–712. Springer, 2004.

[Euz07] Jérôme Euzenat. Semantic precision and recall for ontology alignment evaluation. In Manuela M. Veloso, editor, Proceedings of the 20th International Joint Conference on Artificial Intelligence, pages 348–353. AAAI Publications, 2007.

[Fal09] Sean M. Falconer. Cognitive support for semi-automatic ontology mapping. PhD thesis, University of Victoria, 2009.

[FHP+06] Giorgos Flouris, Zhisheng Huang, Jeff Z. Pan, Dimitris Plexousakis, and Holger Wache. Inconsistencies, negations and changes in ontologies. In Proceedings of the 21th National Conference on Artificial Intelligence, pages 1295–1300. AAAI Publications, 2006.

[Fle10] Daniel Fleischhacker. A practical implementation of semantic precision and recall. In Leonard Barolli, Fatos Xhafa, Salvatore Vitabile, and Hui-Huang Hsu, editors, Third International Workshop on Ontology Alignment and Visualization at the fourth International Conference on Complex, Intelligent and Software Intensive Systems, pages 986–991. IEEE Computer Society, 2010.

[Fre84] Gottlob Frege. Grundlagen der Arithmetik: Eine logisch mathematische Untersuchung ber den Begriff der Zahl. Wilhelm Koebner, 1884.

[FS07] Sean M. Falconer and Margaret-Anne Storey. A cognitive support framework for ontology mapping. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, The Semantic Web, Proceedings of the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), pages 114–127. Springer, 2007.

[Gae92] Peter Gaerdenfors. Belief revision: An introduction. In Belief Revision. Cambridge University Press, 1992.

[GJ79] Michael R. Garey and David S. Johnson. Computers and intractability: A guide to the theory of NP-completeness. W. H. Freeman and Company, 1979.

[GYM07] Fausto Giunchiglia, Mikalai Yatskevich, and Fiona McNeill. Structure preserving semantic matching. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Bin He, editors, Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), pages 13–24. CEUR-WS, 2007.

[Han99] Sven Ove Hansson. A Textbook of Belief Dynamics, Theory Change and Database Updating. Kluwer Academic Publishers, 1999.

[HNR68] Peter Hart, Nils Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4:100–107, 1968.

[Hoc97] Dorit S. Hochbaum. Approximaton algorithms for NP-Hard Problems. PWS Publishing Company, 1997.

[HPS04] Ian Horrocks and Peter Patel-Schneider. Reducing owl entailment to description logic satisfiability. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 1(4):345–357, 2004.

[HQ07] Peter Haase and Guilin Qi. An analysis of approaches to resolving inconsistencies in DL-based ontologies. In Giorgos Flouris and Mathieu d'Aquin, editors, Proceedings of the International Workshop on Ontology Dynamics, pages 97–109, 2007.

[HQ08] Wei Hu and Yuzhong Qu. Falcon-AO: A practical ontology matching system. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 6(3):237–239, 2008.

[HSNR10] Faycal Hamdi, Brigitte Safar, Nobal B. Niraula, and Chantal Reynaud. Taxomap alignment and refinement modules: Results for OAEI 2010. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel Cruz, editors, Proceedings of the 5th International Workshop on Ontology Matching (OM-2010) collocated with the 9th International Semantic Web Conference (ISWC-2010), pages 212–220. CEUR-WS, 2010.

[HvHtT05] Zhisheng Huang, Frank van Harmelen, and Annette ten Teije. Reasoning with inconsistent ontologies. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, Proceedings of the International 19th Joint Conference on Artificial Intelligence, pages 454–459. AAAI Publications, 2005.

[Ich08] Ryutaro Ichise. Machine learning approach for ontology mapping using multiple concept similarity measures. In Roger Y. Lee, editor, Proceedings of the Seventh IEEE/ACIS International Conference on Computer and Information Science, pages 340–346. IEEE Computer Society, 2008.

[JMSK09] Yves R. Jean-Mary, E. Patrick Shironoshitaa, and Mansur R. Kabuka. Ontology matching with semantic verification. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 7(3):235–251, 2009.

[JRGHB08] Ernesto Jimenez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks, and Rafael Berlanga. Ontology integration using mappings: Towards getting the right logical consequences (technical report). Technical report, Universitat Jaume, University of Oxford, 2008.

[JRGHB09] Ernesto Jimenez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks, and Rafael Berlanga. Ontology integration using mappings: Towards getting the right logical consequences. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Paslaru Bontas Simperl, editors, The Semantic Web: Research and Applications, 6th European Semantic Web Conference, pages 173–187. Springer, 2009.

[Kal06] Aditya Anand Kalyanpur. Debugging and Repair of OWL Ontologies. PhD thesis, University of Maryland, 2006.

[KFNM04] Holger Knublauch, Ray W. Fergerson, Natalya Fridman Noy, and Mark A. Musen. The protégé owl plugin: An open development environment for semantic web applications. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, The Semantic Web - ISWC 2004: Third International Semantic Web Conference, pages 229–243. Springer, 2004.

[KPHS07] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin. Finding all justifications of owl dl entailments. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, The Semantic Web, Proceedings of the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), pages 267–280. Springer, 2007.

[KS03] Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. The Knowledge Engineering Review, 18:1–31, 2003.

[Kuh55] H.W. Kuhn. The hungarian method for the assignment problem. Naval Research Logistics, 2:83–97, 1955.

[MH03] Ralf Möller and Volker Haarslev. Description logics for the semantic web: Racer as a basis for building agent systems. Künstliche Intelligenz, 17(1):10–16, 2003.

[MNJ05] Prasenjit Mitra, Natasha F. Noy, and Anuj Rattan Jaiswal. Omen: A probabilistic ontology mapping tool. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, pages 537–547. Springer, 2005.

[MPS10] Ming Mao, Yefei Peng, and Michael Spring. An adaptive ontology mapping approach with neural network based constraint satisfaction. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 8(1):14–25, 2010.

[MS07a] Christian Meilicke and Heiner Stuckenschmidt. Analyzing mapping extraction approaches. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Bin He, editors, Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), pages 25–36. CEUR-WS, 2007.

[MS07b] Christian Meilicke and Heiner Stuckenschmidt. Applying logical constraints to ontology matching. In Joachim Hertzberg, Michael Beetz, and Roman Englert, editors, KI 2007: Advances in Artificial Intelligence, 30th Annual German Conference on AI, pages 99–113. Springer, 2007.

[MS08] Christian Meilicke and Heiner Stuckenschmidt. Incoherence as a basis for measuring the quality of ontology mappings. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Heiner Stuckenschmidt, editors, Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) collocated with the 7th International Semantic Web Conference (ISWC-2008), pages 1–12. CEUR-WS, 2008.

[MS09a] Christian Meilicke and Heiner Stuckenschmidt. An efficient method for computing a local optimal alignment diagnosis. Technical report, University Mannheim, Computer Science Institute, 2009.

[MS09b] Christian Meilicke and Heiner Stuckenschmidt. An efficient method for computing alignment diagnoses. In Axel Polleres and Terrance Swift, editors, Web Reasoning and Rule Systems, Third International Conference, RR 2009, pages 182–196. Springer, 2009.

[MSSZ09] Christian Meilicke, Heiner Stuckenschmidt, and Ondrej Sváb-Zamazal. A reasoning-based support tool for ontology mapping evaluation. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Paslaru Bontas Simperl, editors, The Semantic Web: Research and Applications, 6th European Semantic Web Conference, ESWC 2009, pages 878–882. Springer, 2009.

[MST06] Christian Meilicke, Heiner Stuckenschmidt, and Andrei Tamilin. Improving automatically created mappings using logical reasoning. In Pavel Shvaiko, Jérôme Euzenat, Natalya Fridman Noy, Heiner Stuckenschmidt, V. Richard Benjamins, and Michael Uschold, editors, Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) collocated with the 5th International Semantic Web Conference (ISWC-2006). CEUR-WS, 2006.

[MST08] Christian Meilicke, Heiner Stuckenschmidt, and Andrei Tamilin. Supporting manual mapping revision using logical reasoning. In Dieter Fox and Carla P. Gomes, editors, Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, pages 1213–1218. AAAI Press, 2008.

[MST09] Christian Meilicke, Heiner Stuckenschmidt, and Andrei Tamilin. Reasoning support for mapping revision. Journal of Logic and Computation, 19:807–829, 2009.

[MTS07] Christian Meilicke, Andrei Tamilin, and Heiner Stuckenschmidt. Repairing ontology mappings. In Proceedings of the Twenty-Second Conference on Artificial Intelligence, pages 1408–1413. AAAI Press, 2007.

[MVS08] Christian Meilicke, Johanna Völker, and Heiner Stuckenschmidt. Learning disjointness for debugging mappings between lightweight ontologies. In Aldo Gangemi and Jérôme Euzenat, editors, Knowledge Engineering: Practice and Patterns, 16th International Conference, EKAW 2008, pages 93–108. Springer, 2008.

[Nik10] Nadejda Nikitina. Semi-automatic revision of formalized knowledge. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, ECAI 2010: 19th European Conference on Artificial Intelligence, pages 1097–1098. IOS Press, 2010.

[NM01] Natalya Fridman Noy and Mark A. Musen. Anchor-prompt: Using non-local context for semantic matching. In Asunción Gómez-Pérez, Michael Gruninger, Heiner Stuckenschmidt, and Michael Uschold, editors, Proceedings of the IJCAI Workhsop on Ontologies and Information Sharing, pages 63–70, 2001.

[NM03] Natascha Noy and Mark Musen. The prompt suite: interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies, 59(6):983–1024, 2003.

[NMS10] Mathias Niepert, Christian Meilicke, and Heiner Stuckenschmidt. A probabilistic-logical framework for ontology matching. In Maria Fox and David Poole, editors, Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, pages 1413–1418. AAAI Press, 2010.

[NN10] Jan Noessner and Mathias Niepert. CODI: Combinatorial optimization for data integration results for OAEI 2010. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel Cruz, editors, Proceedings of the 5th International Workshop on Ontology Matching (OM-2010) collocated with the 9th International Semantic Web Conference (ISWC-2010), pages 142–149. CEUR-WS, 2010.

[NRG11] Nadeschda Nikitina, Sebastian Rudolph, and Birte Glimm. Reasoning-supported interactive revision of knowledge bases. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence, page to be published. AAAI Publications, 2011.

[NS05] Natasha Noy and Heiner Stuckenschmidt. Ontology alignment: An annotated bibliography. In Yannis Kalfoglou, W. Marco Schorlemmer, Amit P. Sheth, Steffen Staab, and Michael Uschold, editors, Semantic Interoperability and Integration. IBFI, Schloss Dagstuhl, Germany, 2005.

[NVVM08] Miklos Nagy, Maria Vargas-Vera, and Enrico Motta. Managing conflicting beliefs with fuzzy trust on the semantic web. In Alexander F. Gelbukh and Eduardo F. Morales, editors, MICAI 2008: Advances in Artificial Intelligence, 7th Mexican International Conference on Artificial Intelligence, pages 827–837. Springer, 2008.

[OT10] Giorgio Orsi and Letizia Tanca. Preserving semantics in automatically created ontology alignments. In Roberto De Virgilio, Fausto Giunchiglia, and Letizia Tanca, editors, Semantic Web Information Management: A Model-Based Perspective, pages 203–223. Springer, 2010.

[PJC09] Jyotishman Pathak, Thomas M. Johnson, and Christopher G. Chute. Survey of modular ontology techniques and their applications in the biomedical domain. Integrated Computer-Aided Engineering, 16(3):225–242, 2009.

[PSCC10] Catia Pesquita, Cosmin Stroe, Isabel F. Cruz, and Francisco M. Couto. BLOOMS on AgreementMaker: results for OAEI 2010. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel Cruz, editors, Proceedings of the 5th International Workshop on Ontology Matching (OM-2010) collocated with the 9th International Semantic Web Conference (ISWC-2010), pages 142–149. CEUR-WS, 2010.

[QGKL08] Christoph Quix, Sandra Geisler, David Kensche, and Xiang Li. Results of GeRoMeSuite for OAEI 2008. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Heiner Stuckenschmidt, editors, Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) collocated with the 7th International Semantic Web Conference (ISWC-2008), pages 160–166. CEUR-WS, 2008.

[QHH$^+$08] Guilin Qi, Peter Haase, Zhisheng Huang, Qiu Ji, Jeff Z. Pan, and Johanna Völker. A kernel revision operator for terminologies - algorithms and evaluation. In Amit P. Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy W. Finin, and Krishnaprasad Thirunarayan, editors, The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, pages 419–434. Springer, 2008.

[QJH09] Guilin Qi, Qiu Ji, and Peter Haase. A conflict-based operator for mapping revision. In Abraham Bernstein, David R. Karger, Tom Heath, Lee Feigenbaum, Diana Maynard, Enrico Motta, and Krishnaprasad Thirunarayan, editors, The Semantic Web - ISWC 2009, 8th International Semantic Web Conference, ISWC 2009, pages 521–536. Springer, 2009.

[Qui73] Willard Van Orman Quine. Word and object. Massachusetts Institute of Technology, 1973.

[Rei87] Raymond Reiter. A theory of diagnosis from first principles. Artificial Intelligence, 32:57–95, 1987.

[RN10] Stuart J. Russell and Peter Norvig. Artificial Intelligence: A Modern Approach. Pearson, 3rd edition, 2010.

[SA09] Md. Hanif Seddiqui and Masaki Aono. Anchor-flood: Results for OAEI 2009. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Natalya Fridman Noy, and Arnon Rosenthal, editors, Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009), pages 127–134. CEUR-WS, 2009.

[SB05] François Scharffe and Jos Bruijn. A language to specify mappings between ontologies. In Richard Chbeir, Albert Dipanda, and Kokou Yétongnon, editors, Proceedings of the 1st International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2005, pages 267–271. Dicolor Press, 2005.

[SBL06] Nigel Shadbolt and Tim Berners-Lee. The semantic web revisited. IEEE Intelligent Systems, 21(4):96–101, 2006.

[SBT05] Luciano Serafini, Alexander Borgida, and Andrei Tamilin. Aspects of distributed and modular ontology reasoning. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, pages 570–575, 2005.

[SC03] Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In Georg Gottlob and Toby Walsh, editors, IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, pages 355–362. Morgan Kaufmann, 2003.

[Sch05] Stefan Schlobach. Debugging and semantic clarification by pinpointing. In Asunción Gómez-Pérez and Jérôme Euzenat, editors, The Semantic Web: Research and Applications, Second European Semantic Web Conference, ESWC 2005. Springer, 2005.

[SE05] Pavel Shvaiko and Jérôme Euzenat. A survey of schema-based matching approaches. Journal on Data Semantics, IV:146–171, 2005.

[SE08] Pavel Shvaiko and Jérôme Euzenat. Ten challenges for ontology matching. In Robert Meersman and Zahir Tari, editors, On the Move to Meaningful Internet Systems: OTM 2008, OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Part II, pages 1164–1182. Springer, 2008.

[SF08] François Scharffe and Dieter Fensel. Correspondence patterns for ontology alignment. In Aldo Gangemi and Jérôme Euzenat, editors, Knowledge Engineering: Practice and Patterns, 16th International Conference, EKAW 2008, pages 83–92. Springer, 2008.

[SH05] Stefan Schlobach and Zhisheng Huang. Inconsistent ontology diagnosis. Technical report, Vrije Universiteit Amsterdam, 2005.

[SHCvH07] Stefan Schlobach, Zhisheng Huang, Ronald Cornet, and Frank van Harmelen. Debugging incoherent terminologies. Journal of Automated Reasoning, 39(3):317349, 2007.

[SPG+07] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: a practical OWL-DL reasoner. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 5(2):51–53, 2007.

[SSB+05] Ondrej Sváb, Vojtech Svatek, Petr Berka, Dusan Rak, and Petr Tomasek. Ontofarm: Towards an experimental collection of parallel ontologies. In The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, Poster Track of ISWC, 2005.

[Stu08] Heiner Stuckenschmidt. Debugging description logic ontologies - a reality check. In Raul Garcia-Castro, Asunción Gómez-Pérez, Charles J. Petrie, Emanuele Della Valle, Ulrich Küster, Michal Zaremba, and M. Omair Shafiq, editors, Proceedings of the 6th International Workshop on Evaluation of Ontology-based Tools and the Semantic Web Service Challenge (EON-SWSC-2008). CEUR-WS, 2008.

[TE10] Cássia Trojahn and Jérôme Euzenat. Consistency-driven argumentation for alignment agreement. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel Cruz, editors, Proceedings of the 5th International Workshop on Ontology Matching (OM-2010) collocated with the 9th International Semantic Web Conference (ISWC-2010), pages 37–48. CEUR-WS, 2010.

[TMES10] Cássia Trojahn, Christian Meilicke, Jérôme Euzenat, and Heiner Stuckenschmidt. Automating OAEI campaigns (first report). In Asunción Gómez-Pérez, Fabio Ciravegna, Frank van Harmelen, and Jeff Hefflin, editors, Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST2010), volume 666. CEUR-WS, 2010.

[TMQV08] Cássia Trojahn, Marcia Moraes, Paulo Quaresma, and Renata Vieira. A cooperative approach for composite ontology mapping. Journal on Data Semantics, X:237–263, 2008.

[vH08] Willem Robert van Hage. Evaluating Ontology-Alignment Techniques. PhD thesis, Vrije Universiteit Amsterdam, 2008.

[vHKS08] Willem Robert van Hage, Hap Kolb, and Guus Schreiber. Relevance-based evaluation of alignment approaches: the OAEI 2007 food task revisited. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Heiner Stuckenschmidt, editors, Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) collocated with the 7th International Semantic Web Conference (ISWC-2008), pages 234–238. CEUR-WS, 2008.

[WWL10] Song Wang, Gang Wang, and Xiaoguang Liu. Results of NBJLM for OAEI 2010. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel Cruz, editors, Proceedings of the 5th International Workshop on Ontology Matching (OM-2010) collocated with the 9th International Semantic Web Conference (ISWC-2010), pages 187–194. CEUR-WS, 2010.

[WX08a] Peng Wang and Baowen Xu. Debugging ontology mapping: A static method. Computing and Informatics, 27:21–36, 2008.

[WX08b] Peng Wang and Baowen Xu. Lily: Ontology alignment results for OAEI 2008. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Heiner Stuckenschmidt, editors, Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) collocated with the 7th International Semantic Web Conference (ISWC-2008), pages 167–175. CEUR-WS, 2008.

[WZH$^+$10] Zhichun Wang, Xiao Zhang, Lei Hou, Yue Zhao, Juanzi Li, Yu Qi, and Jie Tang. RiMOM results for OAEI 2010. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel Cruz, editors, Proceedings of the 5th International Workshop on Ontology Matching (OM-2010) collocated with the 9th International Semantic Web Conference (ISWC-2010), pages 195–202. CEUR-WS, 2010.

[XWCZ10] Peigang Xu, Yadong Wang, Liang Cheng, and Tianyi Zang. Alignment results of sobom for OAEI 2010. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel Cruz, editors, Proceedings of the 5th International Workshop on Ontology Matching (OM-2010) collocated with the 9th International Semantic Web Conference (ISWC-2010), pages 203–211. CEUR-WS, 2010.