Inconsistency Repair in Ontology Matching

Algorithms and Implementation

Shuo Zhang
Oriel College

Supervised by
Dr. Bernardo Cuenca Grau and Dr. Ernesto Jimenez-Ruiz

A master thesis submitted in partial fulfilment of the degree
of
MSc in Computer Science

Department of Computer Science
University of Oxford
September 2014

Acknowledgements

I would like to sincerely thank my project supervisors, Dr. Bernardo Cuenca Grau and Dr. Ernesto Jimenez-Ruiz, for leading me one further step towards the fascinating research field Knowledge Representation and Reasoning. I will always remember the first day we met and the days we worked together. I really appreciate every single meeting, each lasting one to two hours. I really appreciate every single lengthy email thread... Your dedication and enthusiasm towards your research will always be inspiring me throughout the rest of my life.

Besides, as always, I would also like to sincerely thank my parents, for supporting me to study in the UK and to work with brilliant minds at the University of Oxford. These experiences will definitely be invaluable to my whole life.

Abstract

LogMap is a highly scalable ontology matching system with built-in reasoning and mapping repair capabilities created at the Knowledge Representation and Reasoning research group at the Department of Computer Science of the University of Oxford [1]. LogMap accepts as input ontologies in the Web Ontology Language 2 (OWL 2) [2] (or compatible formats). The current reasoning and repair algorithm implemented in LogMap computes a Horn propositional logic projection of the integration of the ontologies and the set of mappings. OWL 2 is based on the description logic SROIQ [3] and hence the algorithm is fast but incomplete. In this project, the Horn propositional logic projection is further enriched to accommodate more elements in the OWL 2 language in order to detect and repair more errors.

Contents

Chapter 1

Introduction

1.1 Dissertation Overview

Chapter 1 gives an introduction to the subject and provides necessary background knowledge for the rest of the dissertation. Chapter 2 gives an explanation of the problem we are dealing with and the context in which our work takes place. Afterwards, Chapter 3 gives a comprehensive description of our method from both a theoretical point of view and a practical point of view. In Chapter 4, our work is evaluated. Testing results and analysis of the results are presented. Finally, Chapter 5 concludes the dissertation. A brief summary of the achievements of the project and some suggestions on future work are given, followed by a brief summary on my experiences throughout the project ending the chapter.

1.2 Background and Preliminaries

1.2.1 Ontologies and Ontology Matching

An ontology, in computer and information science, formally conceptualises a domain of interest by providing a vocabulary describing the domain and the associated meaning of the terms in the vocabulary in a format that is understandable to computers [4, 5]. Ontologies are an important component of the development of the Semantic Web. Also, ontologies have been widely used in many application areas. In particular, they are nowadays developing into a mainstream technology for biomedical information systems [6, 7]. However, different parties may use different ontologies [8]. For complex domains, ontology construction is a very demanding task and hence reuse and integration of independently developed ontologies is desirable in practice [6, 9, 10]. To establish suitable semantic correspondences, or mappings, between the entities in the vocabularies of different ontologies is the first step towards their reuse and integration and this process is called ontology matching or ontology alignment [6, 9, 10]. Ontology matching has been drawing great research interests in both the Semantic Web and bio-informatics research communities in the last decade [11, 9, 10]. This process can be done automatically by computer systems, i.e. ontology matching systems.

However, many ontology matching systems ignore the semantics of the input ontologies, which results in inconsistencies (or logical errors), e.g. unsatisfiabilities, within the union of the input ontologies and the mappings produced by the system [6, 7, 11]. LogMap is a highly scalable ontology matching system with built-in reasoning and mapping repair capabilities created at the Knowledge Representation and Reasoning research group at the Department of Computer Science of the University of Oxford [1]. Its reasoning and repair techniques aimed at minimising logical errors are one of the key features that make LogMap stand out [1, 6, 7]. The Ontology Alignment Evaluation Initiative (OAEI) [12] organises annual international campaigns for the systematic evaluation of ontology matching systems. In both the OAEI 2012 campaign [13] and the OAEI 2013 campaign [14], LogMap was one of the most successful systems in terms of application of reasoning and repair techniques.

1.2.2 Preliminaries of OWL 2

The Web Ontology Language (OWL) and its revision OWL 2 [2] are the most widely used ontology modelling languages [6, 7]. The most common entities in the vocabulary of an OWL 2 ontology are individuals, classes (sets of individuals), object properties (binary relations between individuals) and data properties (binary relations between individuals and data values) [6]. OWL 2 differentiates individuals and data values, unlike in description logic, where they are all called individuals. Hence, classes in OWL 2 correspond to concepts and properties correspond to roles in description logic. An OWL 2 ontology is a finite set of OWL 2 axioms, which corresponds to a finite set of axioms in description logic in a relatively straightforward way. Additionally, OWL 2 axioms can have annotations. They allow additional useful information to be associated with an axiom but have no semantic impact [6]. For a more comprehensive introduction to the OWL 2 language, please refer to Section 2.1 of [6] or [15]. For the rest of this dissertation, we will use notions from description logic and from OWL 2 interchangeably.

1.2.3 Mappings

A mapping is typically formalised as a tuple $\langle id, e_1, e_2, n, \rho\rangle$, with id a unique identifier of the mapping, e_1, e_2 entities in the vocabulary of the relevant ontologies, n a numeric confidence measure between 0 and 1, and ρ a relationship between e_1 and e_2, typically subsumption (e_1 is more specific than e_2), equivalence and disjointness [9, 10].

There are currently a number of different representations of mappings. One of the most commonly adopted choices is to represent mappings as OWL 2 subclass, equivalence and disjointness axioms (identifiers and confidence values of mappings as axiom annotations) and this is the representation adopted by the developers of LogMap. This representation not only captures the meaning of mappings accurately and coherently but also makes the reuses of the wide range of the existing OWL 2 reasoning tools possible [6, 11].

1.2.4 Mapping Repair and Related Work

Unsatisfiable classes reflect modelling errors in behind. In our context, we focus on repairing mappings that leads to unsatisfiable classes in the integrations (or unions) of two ontologies and a set of mappings between them.

Justification-based mapping repair techniques have been recognised as an effective approach towards mapping repair and have been implemented in some ontology matching systems, e.g. Alcomo [16], ContentMap [17] and LogMap-Full [18] [6]. A justification J for an unsatisfiable class C in an ontology O is a subset of O such that C is unsatisfiable with respect to J and C is satisfiable with respect to each strict subset of J. A repair R of C in O is a subset of O such that C is satisfiable with respect to $O \setminus R$. This indicates the fact that any repair must contain at least one axiom in each justification for the unsatisfiable class [6, 7]. In the context of mapping repair, we may consider the integration of the two ontologies and the set of mappings between them as a single ontology and we assume that both of the ontologies are themselves logically consistent. Alcomo and ContentMap compute all justifications for all unsatisfiable classes while LogMap-Full only computes one for each unsatisfiable class in each iteration until the set of mappings is all clean. [6]. There also exist some other systems which implement different mapping repair techniques, such as AML [19], S-MATCH [20], ASMOV [21], CODI [22], KOSIMap [23] and YAM++ [24] [6, 11].

It is manifest that we get 0 logical error if we directly remove all mappings but this is not the intended goal since we aim at keeping as many original mappings as possible. However, mapping repair using full reasoning can be very costly. For example, it takes over 6 weeks to compute all justifications for all unsatisfiable classes in the integration of the well-established large biomedical ontologies FMA and SNOMED and the widely used mapping set UMLS-META, even using the state-of-the-art highly-optimised complete OWL 2 reasoner HermiT [25] [6]. Hence, the key question in mapping repair is how to remove mappings in manner that is both effective and feasible.

Chapter 2

Problem and Context

2.1 Problem Specification

We will refer to the Horn propositional mapping repair facility of LogMap as "LogMap Repair". Although other mapping repair techniques exist in LogMap, this project focuses on this part. The problem that LogMap Repair is coping with is described as follows:

Given two ontologies O_1, O_2 and a set of mappings M between them, compute a set $M' \subseteq M$ such that $O_1 \cup O_2 \cup M'$ contains fewer unsatisfiable classes than $O_1 \cup O_2 \cup M$, assuming that both O_1 and O_2 are themselves logically consistent. That is, LogMap Repair performs an incomplete repair and cannot guarantee that $O_1 \cup O_2 \cup M'$ is unsatisfiability free.

2.2 Context: Original LogMap Repair

LogMap computes an Horn propositional approximation of the integration of the input ontologies and mappings for class satisfiability checks. The projection works as below [6]:

- If class A is a sub class of class B, then the rule $A \rightarrow B$ is added. The subsumption relationships between classes are found by OWL 2 reasoners (i.e., by classifying each ontology) or incompletely in a syntactical way and we call these relationships "taxonomic information".

- If class A and class B are disjoint, then the rule $A \wedge B \rightarrow false$ is added. LogMap only extract explicit disjointness, that is, those that are given directly by OWL 2 disjoint classes axioms as extracting all disjointness is expensive.
- $A_1 \sqcap ... \sqcap A_n$ is a sub class of class B or B is a conjunct of its super class, then the rule $A_1 \wedge ... \wedge A_n \rightarrow B$ is added.

Note that the classes above are atomic classes.

Suppose we have two classes $O_1{:}A$ and $O_1{:}B$ in ontology O_1, two classes $O_2{:}A$ and $O_2{:}B$ in ontology O_2 and a set of mappings between O_1 and O_2. Also, suppose we have the following relevant information from the integration of the input ontologies and mapping set:

r_1**:** $O_1{:}A \sqsubseteq O_1{:}B$ from ontology O_1

r_2**:** $O_2{:}A$ and $O_2{:}B$ are disjoint from ontology O_2

m_1**:** $O_1{:}A \sqsubseteq O_2{:}A$ from the mapping set

m_2**:** $O_1{:}B \sqsubseteq O_2{:}B$ from the mapping set

Then, we can construct the graph given in Figure 2.1.

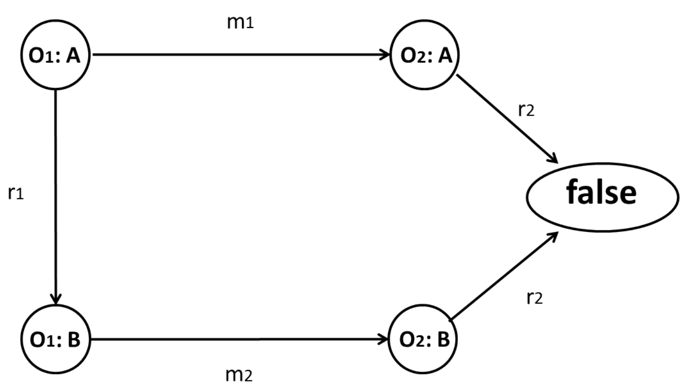

Figure 2.1: Horn Propositional Satisfiability Check in LogMap

In Figure 2.1, each node represents a propositional variable. The directed edges represent Horn propositional rules we can get using the projection method discussed above, given the relevant information. Each edge is labelled with the identifier of its source axiom given before the axiom. Then, it is now clear that the class $O_1{:}A$ is unsatisfiable with respect to the integration because from its node we can go to both the node of $O_2{:}A$ and the node of $O_2{:}B$, which allows us to go further to the $false$ node and this essentially means $O_1{:}A \sqsubseteq \bot$. LogMap implements the linear-time satisfiability testing algorithm for Horn propositional logic, the Dowling-Gallier algorithm [26, 27], which is based on this idea. The Dowling-Gallier algorithm implemented in LogMap is extended so that it also outputs the set of conflictive mappings that may be involved in the detected unsatisfiability (e.g., m_1 and m_2 in Figure 2.1) so as to compute repairs [6].

In this project, we explore different ways to enrich the Horn propositional encoding mechanism of LogMap Repair so that more information of the integration of the input ontologies and mappings is taken into account to detect and repair new errors. Essentially, we are building more sensible paths in the graphical representation of the Horn propositional logic approximation.

Chapter 3

Method

3.1 Theoretical Basis

In this section, we will discuss our extensions from a theoretical point of view. Design and implementation details will be given in the next section.

3.1.1 Existential Property Restrictions

Object Property Existential Restrictions

Object property existential restrictions, e.g. $\exists R.C$, are class expressions (i.e., complex classes) in OWL 2, where R is some object property and C is some class in our context. For convenience, from now, when speaking of classes and properties, we refer to atomic classes and atomic properties. Hence, such restrictions can also be projected as propositional variables as for classes. However, only giving a distinct propositional variable to each distinct restriction and connect the new variables to others just as for classes may not be very useful. Thus, we will also add relevant inference rules as follows:

1. For any class C, if an object property R_1 is a sub property of an object property R_2, then we can have $\exists R_1.C \sqsubseteq \exists R_2.C$.

2. For any object property R, if a class C_1 is a sub class of a class C_2, then we can have $\exists R.C_1 \sqsubseteq \exists R.C_2$.

Suppose we have the following information from the integration of the input ontologies O and O' and mapping set:

r_1**:** $A \sqsubseteq \exists R.C$ from ontology O.

r_2**:** A' and B' are disjoint from ontology O'.

r_3**:** $\exists R'.C' \sqsubseteq B'$ from ontology O'.

m_1**:** $R \sqsubseteq R'$ from the mapping set.

m_2**:** $C \sqsubseteq C'$ from the mapping set.

m_3**:** $A \sqsubseteq A'$ from the mapping set.

where A, C are classes and R is an object property in O; A', B', C' are classes and R' is an object property in O'.

Then, we can construct the graphical representation of the above information given in Figure 3.1. Note that we can go from the node labelled A to the $false$ node, indicating that class A is unsatisfiable.

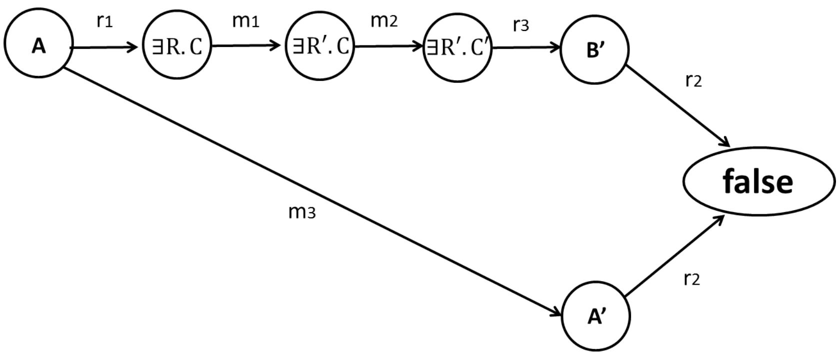

Figure 3.1: Example: Object Property Existential Inference

Data Property Existential Restrictions

Data property existential restrictions, e.g. $\exists R.D_t$, where R is some data property and D_t is some OWL 2 built-in data type in our context, work in a similar way except that the OWL 2 built-in data types are shared between the input ontologies and there is no clear subsumption relationship between them except the top data type. Thus inferences are slightly simpler than object property existential restrictions but follow the same principle:

1. For any data type D_t, if a data property R_1 is a sub property of a data property R_2, then we can have $\exists R_1.D_t \sqsubseteq \exists R_2.D_t$.

Suppose we have the following information from the integration of the input ontologies O and O' and mapping set:

r_1**:** $A \sqsubseteq \exists R.D_t$ from ontology O.

r_2**:** A' and B' are disjoint from ontology O'.

r_3**:** $\exists R'.D_t \sqsubseteq B'$ from ontology O'.

m_1**:** $R \sqsubseteq R'$ from the mapping set.

m_2**:** $A \sqsubseteq A'$ from the mapping set.

with A a class, R a data property in O; A', B' classes and R' a data property in O'; D_t an OWL 2 built-in data type.

Then, we can construct the graphical representation of the above information given in Figure 3.2. Note that we can go from the node labelled A to the $false$ node, indicating that class A is unsatisfiable.

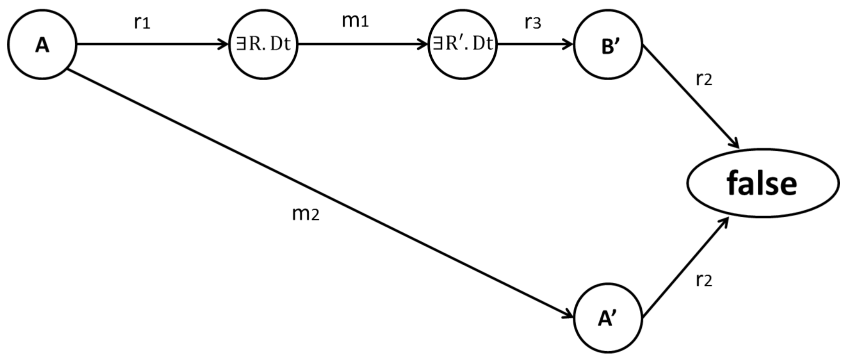

Figure 3.2: Example: Data Property Existential Inference

3.1.2 Universal Property Restrictions

Object Property Universal Restrictions

Following a similar principle with object property existential restrictions, we can have a very similar set of inferences with object property universal restrictions (e.g., $\forall R.C$, where R is some object property and C is some class in our context) but we need to be careful with the direction of the inference resulting from property subsumptions:

1. For any class C, if an object property R_1 is a sub property of an object property R_2, then we can have $\forall R_2.C \sqsubseteq \forall R_1.C$. Note that the inference direction is the reverse of that for existential restrictions.

2. For any object property R, if a class C_1 is a sub class of a class C_2, then we can have $\forall R.C_1 \sqsubseteq \forall R.C_2$.

Suppose we have the following information from the integration of the input ontologies O and O' and mapping set:

r_1: $A \sqsubseteq \forall R.C$ from ontology O.

r_2: A' and B' are disjoint from ontology O'.

r_3: $\forall R'.C' \sqsubseteq B'$ from ontology O'.

m_1: $R' \sqsubseteq R$ from the mapping set.

m_2: $C \sqsubseteq C'$ from the mapping set.

m_3: $A \sqsubseteq A'$ from the mapping set.

where A, C are classes and R is an object property in O; A', B', C' are classes and R' is an object property in O'.

Then, we can construct the graphical representation of the above information given in Figure 3.3. Note that we can go from the node labelled A to the $false$ node, indicating that class A is unsatisfiable.

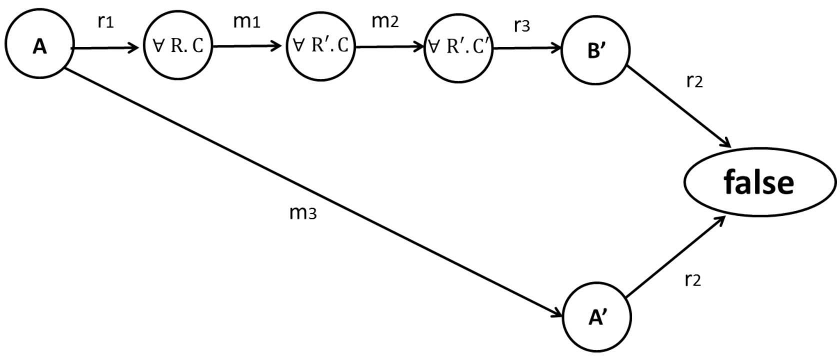

Figure 3.3: Example: Object Property Universal Inference

Data Property Universal Restrictions

The difference between the inferences with data property universal restrictions (e.g., $\forall R.D_t$, where R is some data property and D_t is some OWL 2 built-in data type in our context) and the inferences with object property universal restrictions still originates from the differences between classes and data types. So, we have:

1. For any data type D_t, if a data property R_1 is a sub property of a data property R_2, then we can have $\forall R_2.D_t \sqsubseteq \forall R_1.D_t$.

Suppose we have the following information from the integration of the input ontologies O and O' and mapping set:

r_1**:** $A \sqsubseteq \forall R.D_t$ from ontology O.

r_2**:** A' and B' are disjoint from ontology O'.

r_3**:** $\forall R'.D_t \sqsubseteq B'$ from ontology O'.

m_1**:** $R' \sqsubseteq R$ from the mapping set.

m_2**:** $A \sqsubseteq A'$ from the mapping set.

with A a class, R a data property in O; A', B' classes and R' a data property in O'; D_t an OWL 2 built-in data type.

Then, we can construct the graphical representation of the above information given in Figure 3.4. Note that we can go from the node labelled A to the $false$ node, indicating that class A is unsatisfiable.

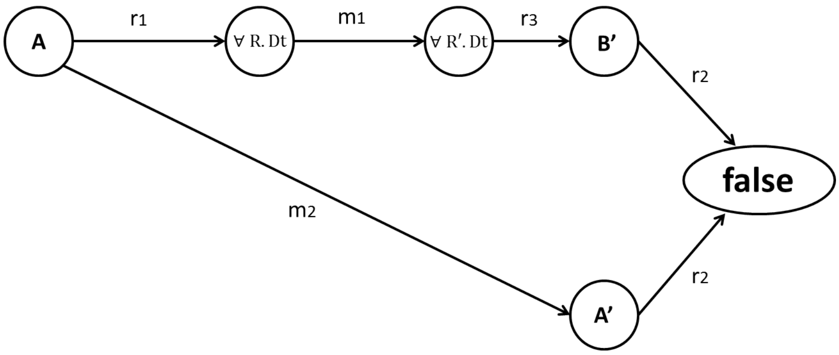

Figure 3.4: Example: Data Property Universal Inference

3.1.3 Property Domains

In OWL 2, the domain of an object property or a data property can be specified by a domain axiom. Logically, a domain axiom for an object property R stating its domain is class C is equivalent to the axiom $\exists R.\top \sqsubseteq C$, where $\top$ is the top concept/class, shared by all ontologies. The same idea applies to data property domains. Hence, with the inference extensions described in previous sections, we can also detect logical errors with property domains caused by problematic mappings. The following gives an example of using an object property domain axiom and the same principle applies to data property domains.

Suppose we have the following information from the integration of the input ontologies O and O' and mapping set:

r_1**:** $A \sqsubseteq \exists R.C$ from ontology O.

r_2**:** $\exists R'.\top \sqsubseteq D'$ from an object property domain axiom in ontology O'.

m_1**:** $R \sqsubseteq R'$ from the mapping set.

where A, C are classes and R is an object property in O; D' is a class and R' is an object property in O'.

Then, we can construct the graphical representation of the above information given in Figure 3.5. Note that we build a new path from the node labelled A to the node labelled D', indicating $A \sqsubseteq D'$. This essentially forces any sub class of an object property existential restriction to comply with the domain of the property involved. The edge with no label in the graph comes from $C \sqsubseteq \top$ by definition of $\top$.

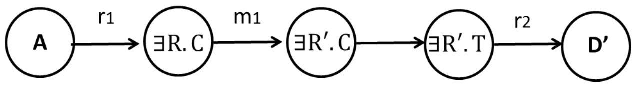

Figure 3.5: Example: Object Property Domain Inference

3.1.4 Property Ranges

In OWL 2, the range of an object property or a data property can be specified by a range axiom. Logically, a range axiom for an object property R stating its range is class C is equivalent to the axiom $\top \sqsubseteq \forall R.C$. The same idea applies to data property ranges. Hence, with the inference extensions described in previous sections, we can also detect logical errors with property ranges caused by problematic mappings.

However, to make use of the range axioms some extra inference rules are needed:

1. For any object property R, if class A and class B are disjoint, then we can have $\exists R.A$ is disjoint with $\forall R.B$ and $\exists R.B$ is disjoint with $\forall R.A$.
2. For any data property R, if data type D_{t1} and data type D_{t2} are disjoint, then we can have $\exists R.D_{t1}$ is disjoint with $\forall R.D_{t2}$ and $\exists R.D_{t2}$ is disjoint with $\forall R.D_{t1}$.

The following gives an example of using an object property range axiom and the same principle applies to data property ranges.

Suppose we have the following information from the integration of the input ontologies O and O' and mapping set:

r_1**:** $A \sqsubseteq \exists R.C$ from ontology O.

r_2**:** $\top \sqsubseteq \forall R'.B'$ from an object property range axiom in ontology O'.

r_3**:** B' and C' are disjoint from ontology O'.

m_1**:** $R \sqsubseteq R'$ from the mapping set.

m_2**:** $C \sqsubseteq C'$ from the mapping set.

where A, C are classes and R is an object property in O; B', C' are classes and R' is an object property in O'.

Then, we can construct the graphical representation of the above information given in Figure 3.6. It is now manifest that class A is unsatisfiable.

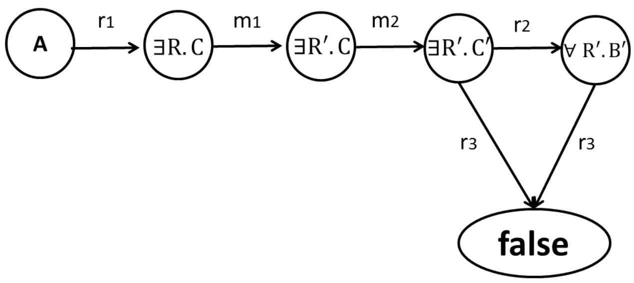

Figure 3.6: Example: Object Property Range Inference

3.2 Design and Implementation

3.2.1 Ontology Processing

Ontology processing refers to the process where we extract information we are interested in from the input ontologies. Rather than taking axioms as they are in the input ontologies, we use the ontology normaliser presented in the paper [28], with minor extensions to put information we are interested in as desired forms, to pre-process each ontology. The normaliser encodes various OWL 2 axioms as axioms of the following form so that it makes it very simple for us to extract information in a uniformed way:

$$A_1 \sqcap A_2 \sqcap ... \sqcap A_m \sqsubseteq B_1 \sqcup B_2 \sqcup ... \sqcup B_n$$

where $m, n \geq 1$ and each A_i is an atomic class or an existential restriction with an atomic class or data type as its filler and each B_j is an atomic class or an existential/universal restriction with an atomic class or data type as its filler.

Domain and range axioms mentioned previously are normalised as their equivalent forms. Recall that a domain axiom for an object property R stating its domain is class C is logically equivalent to the axiom $\exists R.\top \sqsubseteq C$ while a range axiom for an object property R stating its range is class C is logically equivalent to the axiom $\top \sqsubseteq \forall R.C$. The same ideas apply to data properties.

With the above normalisation, we can now easily identify and extract information related to restrictions/properties from ontologies, that is, we take axioms of the following form, which can be easily encoded as Horn propositional rules:

$$A_1 \sqcap A_2 \sqcap ... \sqcap A_n \sqsubseteq B$$

where $n \geq 1$ and each A_i is an atomic class or an existential restriction with an atomic property and an atomic class or data type as its filler and B is an atomic class or an existential/universal restriction with an atomic property and an atomic class or data type as its filler.

3.2.2 Mapping Processing and Further Ontology Processing

Mapping processing refers to the process where we build the paths between (the classes and restrictions of) the two ontologies using the axioms in the given mapping set, as was described in Section 3.1. The relevant mappings to our repair fall into the following two categories:

- **Property Mappings:** mappings that map from an atomic property in one ontology to an atomic property in the other ontology with subsumption or equivalence relationship.
- **Class Mappings:** mappings that map from an atomic class in one ontology to an atomic class in the other ontology with subsumption or equivalence relationship.

For object property restrictions, we process the mappings in two steps, that is, we first build inference paths using property mappings and then build inference paths using class mappings. In both steps, we may create some "dummy restrictions". The following gives an example:

Suppose we **only** have the following information from the integration of the input ontologies O and O' and mapping set:

r_1**:** $A \sqsubseteq \exists R.C$ from ontology O.

r_2**:** $\exists R'_1.C'_2 \sqsubseteq B'$ from ontology O'.

r_3**:** $\exists R'_2.C'_1 \sqsubseteq D'$ from ontology O'.

m_1**:** $R \sqsubseteq R'_1$ from the mapping set.

m_2**:** $R \sqsubseteq R'_2$ from the mapping set.

m_3**:** $C \sqsubseteq C'_1$ from the mapping set.

m_4**:** $C \sqsubseteq C'_2$ from the mapping set.

where A, C are classes and R is an object property in O; B', D', C'_1, C'_2 are classes and R'_1, R'_2 are object properties in O'.

Then, the inference paths we would build are shown in Figure 3.7 and we would do this for every object property restriction we identified in some relevant axiom we took. Note that the square nodes in Figure 3.7 represent "dummy restrictions". There are two types of dummy restrictions:

- Restrictions with mixed entities from two ontologies, such as $\exists R'_1.C$ and $\exists R'_2.C$ in this example: They serve as intermediate steps.
- Restrictions without mixed entities but not existing on the other side, such as $\exists R'_1.C'_1$ and $\exists R'_2.C'_2$ in this example: They are useful when we are taking into account the taxonomic information mentioned in Section 2.2 and the property range information afterwards.

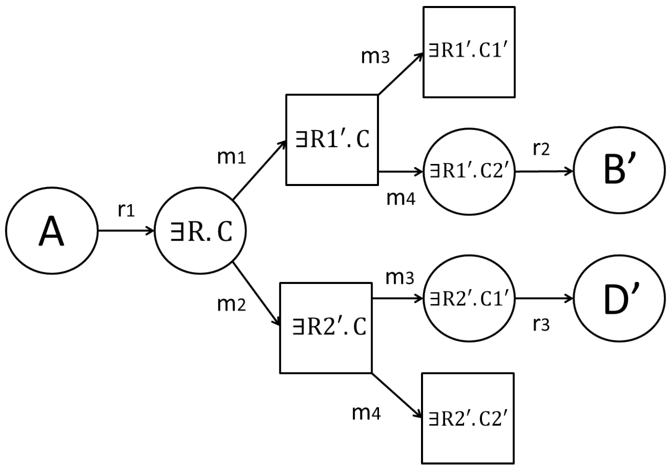

Figure 3.7: Example: Mapping Paths with Object Property Restrictions

After the mappings have been processed, we do a further processing of the ontologies to further extract useful information. To make full of the taxonomic information, which could come from reasoners, and the property range information, this process is done after mappings have been processed, although the pieces of information we are obtaining here would only involve entities in a single ontology. This is because by now we have introduced dummy restrictions to each ontology which are connected with (the classes and restrictions in) the other ontology.

The processing has two steps:

- Use taxonomic information to build inference paths for restrictions just as we did for class mappings.
- Use property range information to build inference paths as described in Section 3.1.4 for each property and range pair.

For data property restrictions, there are 3 differences with respect to the above processes:

- We process mappings in one step, that is, we only need to process property mappings.
- No taxonomic information to use to further process the ontologies (but we still need to further process the ontologies to build data property range inference paths).
- We see different built-in data types except the top data type (i.e., the data type of all data values) as disjoint when building data property range inference paths. For example, $double$ and int are considered disjoint.

3.2.3 Other Aspects

The aspects discussed in the previous two sections are the major additions/changes performed in this project. There are also some other subtle aspects, such as coping with special cases with the top class and the top data type, extending the mapping removal mechanism, extending the propositional projection updating mechanism and coping with side effects that come from using the normaliser.

Chapter 4

Evaluation

Our extensions to LogMap Repair are evaluated with data sets from the OAEI 2013 campaign. Specifically, we use ontologies from the Conference track (describing the domain of conference organisation) [29] and mapping sets produced by the participating ontology matching systems. This track has a reasonable number of ontologies and the involved ontologies are rich in semantics while relatively small in size. For convenience, we will refer to our extended version of LogMap Repair as "LogMap Repair +". In the first section of this chapter, we will evaluate the error detection capability of LogMap Repair +. Then, in the second section, we will evaluate the repair outcomes of LogMap Repair +. In both cases, we will compare LogMap Repair + against LogMap Repair.

4.1 Detecting New Errors

Table 4.1 shows the number of unsatisfiable classes in the relevant integrations of ontologies and mappings detected by each tool. This table contains only a selection of typical cases showing that more errors are actually detected by LogMap Repair +. Please refer to Appendix A for the complete table.

Table 4.1: Error Detection Tests (partial table extracted from full results given in Appendix A)

Test Case (matcher - ontology 1 - ontology 2)	**Pellet**	**LogMap Repair +**	**LogMap Repair**
	Unsat.	**Unsat.**	**Unsat.**
CIDER_ CL-confof-edas	**59**	**59**	**22**
CroMatcher-cmt-confof	**41**	**41**	**11**
CroMatcher-confof-edas	54	52	30
CroMatcher-confof-sigkdd	61	57	23
HerTUDA-conference-confof	**8**	**8**	**0**
HerTUDA-confof-edas	29	21	0
HerTUDA-confof-ekaw	14	12	0
HotMatch-cmt-confof	10	9	0
HotMatch-conference-confof	**8**	**8**	**0**
HotMatch-confof-edas	29	21	0
HotMatch-confof-ekaw	14	12	0
IAMA-confof-edas	29	21	0
LogMapLite-confof-edas	**29**	**29**	**0**
MaasMatch-cmt-confof	**54**	**54**	**20**
MaasMatch-cmt-iasted	149	137	37
MaasMatch-conference-confof	74	61	28
MaasMatch-confof-ekaw	70	68	29
MaasMatch-confof-iasted	164	134	77
MaasMatch-edas-sigkdd	**86**	**86**	**59**
MaasMatch-ekaw-iasted	198	91	43
MaasMatch-ekaw-sigkdd	90	81	45
ODGOMS-conference-confof	**8**	**8**	**0**
ODGOMS-confof-edas	29	21	0
ODGOMS1_ 2-confof-edas	29	21	0
OntoK2-confof-edas	29	21	0
SYNTHESIS-confof-edas	29	21	0
WikiMatch-cmt-confof	10	9	0
WikiMatch-conference-confof	**8**	**8**	**0**
WikiMatch-confof-edas	29	21	0
XMapSiG1_ 3-confof-edas	29	21	0

As we can see clearly in Table 4.1, LogMap Repair + is capable of detecting many more errors than LogMap Repair. In a few cases (in bold), such as “CIDER_ CL-confof-edas”, “CroMatcher-cmt-confof”and “HerTUDA-conference-confof”, LogMap Repair + is even able to detect all unsatisfiable classes that the complete OWL 2 reasoner Pellet [30] can detect. In fact, in 136 test cases out of the 221 test cases where there is error involved, LogMap Repair +

is able to detect strictly more error(s) than LogMap Repair and in no case does LogMap Repair + detect fewer error(s) than LogMap Repair.

These results have shown the achievement of the objective of detecting more errors. The next section will show that LogMap Repair + is also able to repair more errors.

4.2 Repair Outcomes

Table 4.2 shows a selection of very positive repair results for LogMap Repair +. In our tables for repair tests, an equivalence mapping is counted as 2 mappings while a subsumption mapping is counted as 1 when computing repair sizes, where the size of a repair is the number of mappings being removed due to the repair. Additionally, all the numbers of unsatisfiable classes before and after repair were obtained using Pellet. Although the enhanced repair capability of LogMap Repair + does not appear very strong in testing results for every participating system, we can identify some very positive results for the majority of them and some of the other participating systems have already implemented mapping repair techniques, such as AML, LogMap and YAM++. As a general conclusion, our results are very positive. The full table of repair testing results is given in Appendix B.

Table 4.2: Repair Tests (partial table extracted from full results given in Appendix B)

Test Case (matcher - ontology 1 - ontology 2)	**Before Repair**		**LogMap Repair**		**LogMap Repair +**	
	Num. of Mappings	**Unsat.**	**Repair Size**	**Unsat.**	**Repair Size**	**Unsat.**
CIDER_ CL-conference-confof	46	40	7	24	10	7
CIDER_ CL-confof-edas	58	59	8	47	12	5
CIDER_ CL-confof-sigkdd	32	14	1	14	5	0
CIDER_ CL-edas-iasted	44	170	6	159	10	4
CIDER_ CL-edas-sigkdd	56	41	4	34	6	0
CroMatcher-cmt-confof	54	41	9	37	14	5
CroMatcher-confof-ekaw	74	45	27	14	30	0
CroMatcher-edas-sigkdd	98	63	20	39	22	1
CroMatcher-iasted-sigkdd	94	174	6	42	8	0
HerTUDA-confof-edas	36	29	0	29	2	3
HerTUDA-confof-ekaw	28	14	0	14	2	0
HotMatch-cmt-confof	18	10	0	10	2	1
HotMatch-confof-edas	36	29	0	29	2	3
HotMatch-confof-ekaw	28	14	0	14	2	0
IAMA-confof-edas	36	29	2	29	4	3
LogMapLite-confof-edas	38	29	2	29	6	0
MaasMatch-cmt-edas	150	56	50	24	57	9
MaasMatch-edas-ekaw	202	121	96	19	99	28
MaasMatch-ekaw-sigkdd	136	90	29	81	58	17
ODGOMS-confof-edas	36	29	0	29	2	3
ODGOMS1_ 2-confof-edas	36	29	0	29	2	3
OntoK2-confof-edas	38	29	4	29	6	3
RIMOM2013-confof-edas	158	60	45	35	50	9
RIMOM2013-iasted-sigkdd	240	15	1	15	3	0
SYNTHESIS-confof-edas	38	29	4	29	6	3
WikiMatch-confof-edas	38	29	0	29	2	3
XMapGen-confof-edas	50	31	8	29	10	3
XMapGen1_ 4-confof-edas	64	47	17	29	19	3
XMapSiG1_ 3-confof-edas	40	29	4	29	6	3
XMapSiG1_ 4-confof-edas	38	29	4	29	6	3

However, there also exist a small number of special results, which seem to be strange but all can be explained.

Table 4.3 shows some cases where both LogMap Repair and LogMap Repair + are attempting to repair unsatisfiable classes but failed or nearly failed eventually. These reflect an intrinsic limitation of using an incomplete projection. Essentially, we have no picture of the world outside our projection. Altough we can guarantee that a class is satisfiable in our projection, it can still be unsatisfiable given the whole OWL 2 expressiveness. In this sense, "knowing more"might only mean some extra work in vain. In all the cases in Table 4.3, LogMap Repair + is detecting more unsatisfiable classes than LogMap Repair, which is essentially the reason why LogMap Repair + is removing more mappings while achieving the same results as LogMap Repair. However, on the other hand, we can be lucky in some cases when we unintentionally remove some mappings that lead to unsatisfiabilities outside our projection. This accounts for some rare cases like those in Table 4.4. Finally, there are cases where both LogMap Repair + and LogMap Repair are removing mappings while there is no error at all, such as those in Table 4.5. This is because both LogMap Repair + and LogMap Repair are removing obviously invalid mappings (e.g., a mapping that maps from a class to a property) and unnecessary mappings (e.g., a mapping that maps from $\top$ to $\top$) and they are not taken into account for the unsatisfiability test for the original integration of ontologies and mappings before repair.

Table 4.3: Special Repair Test Results 1

Test Case (matcher - ontology 1 - ontology 2)	**Before Repair**		**LogMap Repair**		**LogMap Repair +**	
	Num. of Mappings	**Unsat.**	**Repair Size**	**Unsat.**	**Repair Size**	**Unsat.**
HerTUDA-cmt-sigkdd	20	3	0	3	2	3
CroMatcher-cmt-iasted	78	146	9	145	21	145
XMapGen-cmt-sigkdd	22	1	2	1	4	1
XMapSiG1_ 3-cmt-sigkdd	26	3	2	3	4	3

Table 4.4: Special Repair Test Results 2

Test Case (matcher - ontology 1 - ontology 2)	**Before Repair**		**LogMap Repair**		**LogMap Repair +**	
	Num. of Mappings	**Unsat.**	**Repair Size**	**Unsat.**	**Repair Size**	**Unsat.**
CroMatcher-confof-sigkdd	60	61	18	53	20	55
MapSSS-conference-confof	26	4	3	0	4	2

Table 4.5: Special Repair Test Results 3

Test Case (matcher - ontology 1 - ontology 2)	**Before Repair**		**LogMap Repair**		**LogMap Repair +**	
	Num. of Mappings	**Unsat.**	**Repair Size**	**Unsat.**	**Repair Size**	**Unsat.**
IAMA-conference-edas	22	0	2	0	2	0
MapSSS-conference-edas	24	0	2	0	2	0
StringsAuto-cmt-confof	12	0	2	0	2	0
XMapGen-conference-confof	20	0	2	0	2	0
XMapGen1_4-cmt-iasted	24	0	2	0	2	0
XMapSiG1_3-confof-ekaw	28	0	2	0	2	0
XMapSiG1_4-cmt-conference	10	0	2	0	2	0

Chapter 5

Conclusion

5.1 Achievements

In this project, LogMap Repair has been improved significantly in both error detection and repair capabilities. Specifically, its Horn propositional projection has been extended to accommodate reasoning with existential property restrictions, universal property restrictions, property domains and property ranges. Error detection testing results are very positive and, in a few cases, LogMap Repair + is even able to detect all unsatisfiable classes. Repair testing results are also very positive, although in a very small number of cases LogMap Repair + computes repairs of larger sizes but failed to repair more unsatisfiable classes than LogMap Repair, as was discussed in Section 4.2. As a research output, our results are very likely to be included in a paper to be published in the Ninth International Workshop on Ontology Matching [31].

5.2 Future Work

The following is a few suggested directions for future work.

As is shown in appendix B LogMap Repair + is working well with outputs of some ontology matching systems while not with some others. We can investigate the causes of these differences and try to figure out how to combine our extensions (not just limited to those presented in this dissertation) with different ontology matching techniques, that is, we can apply different extensions with different ontology matching systems so that the matching process and the repair process can complement each other.

Although full reasoning is expensive and does not scale well, it can guarantee a clean output. However, we may combine the current incomplete repair with full reasoning by first applying an incomplete repair and then applying full reasoning repair techniques if there is only a very small number of unsatisfiable classes left by the incomplete repair.

During this project, priority was given to the capability of detecting and repairing more errors. However, adding new reasoning elements can damage the scalability of the system. Also, the problem of increased unnecessary repair discussed in Section 4.2 can become more severe when LogMap Repair + is given larger and/or more complex ontologies as inputs. Hence, some scalability analysis will be useful to help with keeping a good balance between the error repair capability and the scalability of the system.

5.3 Summary

LogMap is the first practical and professional system that I have ever worked on. I was deeply impressed by how complex principles in theory can become when put into practice in such a system and really enjoyed it when I successfully applied theories into practice eventually. I have gained a more practical picture of the field of Knowledge Representation and Reasoning (KRR) than in the course. At the very beginning of this project, I thought programming with LogMap should be simple but then I realised that I was completely wrong and programming in a practical and professional context of KRR can be extremely tricky. I should thank again Dr. Ernesto Jimenez-Ruiz for giving me more than sufficient practical guidance throughout the whole project.

As a summary, through doing this project, I have gained a much clear picture of practical research ongoing in the field of KRR and enjoyed it a lot.

Bibliography

[1] *logmap-matcher*. URL: `http://code.google.com/p/logmap-matcher/`.

[2] *OWL 2 Web Ontology Language Document Overview (Second Edition)*. URL: `http://www.w3.org/TR/owl2-overview/`.

[3] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. "The Even More Irresistible SROIQ". In: *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*. 2006, pp. 57–67. URL: `http://www.aaai.org/Library/KR/2006/kr06-009.php`.

[4] Thomas R Gruber. "A translation approach to portable ontology specifications". In: *Knowledge acquisition* 5.2 (1993), pp. 199–220.

[5] Nicola Guarino, Daniel Oberle, and Steffen Staab. "What is an Ontology?" In: *Handbook on ontologies*. Springer, 2009, pp. 1–17.

[6] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, and Ian Horrocks. *LogMap: Scalable, Logic-based and Interactive Ontology Matching*. Department of Computer Science, University of Oxford.

[7] Ernesto Jiménez-Ruiz et al. "Large-scale Interactive Ontology Matching: Algorithms and Implementation". In: *ECAI 2012: 20th European Conference on Artificial Intelligence*. 2012.

[8] *Ontology Matching*. 2014. URL: `http://www.ontologymatching.org/`.

[9] Pavel Shvaiko and Jérôme Euzenat. "Ontology matching: state of the art and future challenges". In: *Knowledge and Data Engineering, IEEE Transactions on Knowledge and Data Engineering* 25.1 (2013), pp. 158–176.

[10] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching, Second Edition*. Springer, 2013. ISBN: 978-3-642-38720-3.

[11] Ernesto Jiménez-Ruiz et al. "Evaluating Mapping Repair Systems with Large Biomedical Ontologies." In: *Description Logics*. 2013, pp. 246–257.

[12] *Ontology Alignment Evaluation Initiative*. URL: `http://oaei.ontologymatching.org/`.

[13] *Ontology Alignment Evaluation Initiative 2012 Campaign*. URL: `http://oaei.ontologymatching.org/2012/`.

[14] *Ontology Alignment Evaluation Initiative 2013 Campaign*. URL: `http://oaei.ontologymatching.org/2013/`.

[15] B. Cuenca Grau et al. "OWL 2: The next step for OWL". In: *J. Web Sem.* 6.4 (2008), pp. 309–322.

[16] Christian Meilicke. "Alignment Incoherence in Ontology Matching". PhD thesis. University of Mannheim, 2011.

[17] Ernesto Jiménez-Ruiz et al. "Ontology integration using mappings: Towards getting the right logical consequences". In: *The Semantic Web: Research and Applications*. Springer, 2009, pp. 173–187.

[18] E. Jiménez-Ruiz, B. Cuenca Grau, and I. Horrocks. "On the Feasibility of Using OWL 2 DL Reasoners for Ontology Matching Problems". In: *OWL Reasoner Evaluation Workshop*. 2012.

[19] Daniel Faria et al. "The AgreementMakerLight Ontology Matching System". In: *OTM Conferences*. 2013, pp. 527–541.

[20] Fausto Giunchiglia, Mikalai Yatskevich, and Pavel Shvaiko. "Semantic matching: Algorithms and implementation". In: *Journal on Data Semantics IX*. Springer, 2007, pp. 1–38.

[21] Yves R Jean-Mary, E Patrick Shironoshita, and Mansur R Kabuka. "Ontology matching with semantic verification". In: *Web Semantics: Science, Services and Agents on the World Wide Web* 7.3 (2009), pp. 235–251.

[22] Jakob Huber et al. "CODI: Combinatorial optimization for data integration–results for OAEI 2011". In: *Ontology Matching* (2011), p. 134.

[23] Quentin Reul and Jeff Z Pan. "KOSIMap: Use of description logic reasoning to align heterogeneous ontologies". In: *23rd International Workshop on Description Logics DL2010*. Citeseer. 2010, p. 489.

[24] Duy Hoa Ngo and Zohra Bellahsene. "YAM++ A combination of graph matching and machine learning approach to ontology alignment task". In: *J. Web Sem.* 16 (2012).

[25] Boris Motik, Rob Shearer, and Ian Horrocks. "Hypertableau Reasoning for Description Logics". In: *J. Artif. Intell. Res. (JAIR)* 36 (2009), pp. 165–228.

[26] William F Dowling and Jean H Gallier. "Linear-time algorithms for testing the satisfiability of propositional Horn formulae". In: *The Journal of Logic Programming* 1.3 (1984), pp. 267–284.

[27] Giorgio Gallo and Giampaolo Urbani. "Algorithms for testing the satisfiability of propositional formulae". In: *The Journal of Logic Programming* 7.1 (1989), pp. 45–61.

[28] David Carral et al. "Is Your Ontology as Hard as You Think? Rewriting Ontologies into Simpler DLs". In: (2014).

[29] *Conference track*. URL: `http://oaei.ontologymatching.org/2013/conference/index.html`.

[30] Evren Sirin et al. "Pellet: A practical OWL-DL reasoner." In: *J. Web Sem.* 5.2 (2007), pp. 51–53. DOI: `10.1016/j.websem.2007.03.004`. URL: `http://dx.doi.org/10.1016/j.websem.2007.03.004`.

[31] *The Ninth International Workshop on Ontology Matching*. URL: `http://om2014.ontologymatching.org/`.

Appendix A

Full Table of Error Detection Tests

Note: '-'indicates the reasoner could not finish the task.

Test Case (matcher - ontology 1 - ontology 2)	**Pellet**	**LogMap Repair +**	**LogMap Repair**
	Unsat.	**Unsat.**	**Unsat.**
AML-cmt-conference	0	0	0
AML-cmt-confof	0	0	0
AML-cmt-edas	0	0	0
AML-cmt-ekaw	0	0	0
AML-cmt-iasted	0	0	0
AML-cmt-sigkdd	0	0	0
AML-conference-confof	0	0	0
AML-conference-edas	0	0	0
AML-conference-ekaw	0	0	0
AML-conference-iasted	0	0	0
AML-conference-sigkdd	0	0	0
AML-confof-edas	0	0	0
AML-confof-ekaw	0	0	0
AML-confof-iasted	0	0	0
AML-confof-sigkdd	0	0	0
AML-edas-ekaw	0	0	0
AML-edas-iasted	0	0	0
AML-edas-sigkdd	0	0	0
AML-ekaw-iasted	0	0	0
AML-ekaw-sigkdd	0	0	0
AML-iasted-sigkdd	0	0	0
AMLback-cmt-conference	0	0	0
AMLback-cmt-confof	0	0	0

AMLback-cmt-edas	0	0	0
AMLback-cmt-ekaw	0	0	0
AMLback-cmt-iasted	0	0	0
AMLback-cmt-sigkdd	0	0	0
AMLback-conference-confof	0	0	0
AMLback-conference-edas	0	0	0
AMLback-conference-ekaw	0	0	0
AMLback-conference-iasted	0	0	0
AMLback-conference-sigkdd	0	0	0
AMLback-confof-edas	0	0	0
AMLback-confof-ekaw	0	0	0
AMLback-confof-iasted	0	0	0
AMLback-confof-sigkdd	0	0	0
AMLback-edas-ekaw	0	0	0
AMLback-edas-iasted	0	0	0
AMLback-edas-sigkdd	0	0	0
AMLback-ekaw-iasted	0	0	0
AMLback-ekaw-sigkdd	0	0	0
AMLback-iasted-sigkdd	0	0	0
CIDER_ CL-cmt-conference	33	9	9
CIDER_ CL-cmt-confof	24	16	13
CIDER_ CL-cmt-edas	41	30	26
CIDER_ CL-cmt-ekaw	11	10	9
CIDER_ CL-cmt-iasted	1	1	0
CIDER_ CL-cmt-sigkdd	5	4	2
CIDER_ CL-conference-confof	40	34	34
CIDER_ CL-conference-edas	59	43	42
CIDER_ CL-conference-ekaw	82	80	80
CIDER_ CL-conference-iasted	1	1	1
CIDER_ CL-conference-sigkdd	3	0	0
CIDER_ CL-confof-edas	59	59	22
CIDER_ CL-confof-ekaw	53	44	44
CIDER_ CL-confof-iasted	50	14	14
CIDER_ CL-confof-sigkdd	14	9	2
CIDER_ CL-edas-ekaw	83	82	70
CIDER_ CL-edas-iasted	170	32	26
CIDER_ CL-edas-sigkdd	41	36	20
CIDER_ CL-ekaw-iasted	188	44	44
CIDER_ CL-ekaw-sigkdd	22	14	14
CIDER_ CL-iasted-sigkdd	6	3	0
CroMatcher-cmt-confof	41	41	11
CroMatcher-cmt-edas	38	19	16

CroMatcher-cmt-ekaw	9	8	4
CroMatcher-cmt-iasted	146	46	26
CroMatcher-cmt-sigkdd	50	23	20
CroMatcher-confof-edas	54	52	30
CroMatcher-confof-ekaw	45	43	34
CroMatcher-confof-iasted	155	12	6
CroMatcher-confof-sigkdd	61	57	23
CroMatcher-edas-ekaw	67	61	46
CroMatcher-edas-iasted	182	107	100
CroMatcher-edas-sigkdd	63	59	48
CroMatcher-ekaw-iasted	189	94	94
CroMatcher-ekaw-sigkdd	42	30	27
CroMatcher-iasted-sigkdd	174	36	13
HerTUDA-cmt-conference	0	0	0
HerTUDA-cmt-confof	24	11	0
HerTUDA-cmt-edas	0	0	0
HerTUDA-cmt-ekaw	2	1	0
HerTUDA-cmt-iasted	0	0	0
HerTUDA-cmt-sigkdd	3	2	0
HerTUDA-conference-confof	8	8	0
HerTUDA-conference-edas	0	0	0
HerTUDA-conference-ekaw	0	0	0
HerTUDA-conference-iasted	0	0	0
HerTUDA-conference-sigkdd	0	0	0
HerTUDA-confof-edas	29	21	0
HerTUDA-confof-ekaw	14	12	0
HerTUDA-confof-iasted	0	0	0
HerTUDA-confof-sigkdd	41	9	9
HerTUDA-edas-ekaw	8	3	0
HerTUDA-edas-iasted	4	2	0
HerTUDA-edas-sigkdd	22	15	11
HerTUDA-ekaw-iasted	0	0	0
HerTUDA-ekaw-sigkdd	0	0	0
HerTUDA-iasted-sigkdd	5	3	0
HotMatch-cmt-conference	0	0	0
HotMatch-cmt-confof	10	9	0
HotMatch-cmt-edas	0	0	0
HotMatch-cmt-ekaw	2	1	0
HotMatch-cmt-iasted	0	0	0
HotMatch-cmt-sigkdd	3	2	0
HotMatch-conference-confof	8	8	0
HotMatch-conference-edas	0	0	0

HotMatch-conference-ekaw	0	0	0
HotMatch-conference-iasted	0	0	0
HotMatch-conference-sigkdd	0	0	0
HotMatch-confof-edas	29	21	0
HotMatch-confof-ekaw	14	12	0
HotMatch-confof-iasted	0	0	0
HotMatch-confof-sigkdd	41	9	9
HotMatch-edas-ekaw	8	3	0
HotMatch-edas-iasted	4	2	0
HotMatch-edas-sigkdd	22	15	11
HotMatch-ekaw-iasted	0	0	0
HotMatch-ekaw-sigkdd	0	0	0
HotMatch-iasted-sigkdd	7	3	0
IAMA-cmt-conference	1	0	0
IAMA-cmt-confof	1	0	0
IAMA-cmt-edas	0	0	0
IAMA-cmt-ekaw	2	1	0
IAMA-cmt-iasted	0	0	0
IAMA-cmt-sigkdd	3	2	0
IAMA-conference-confof	0	0	0
IAMA-conference-edas	0	0	0
IAMA-conference-ekaw	0	0	0
IAMA-conference-iasted	0	0	0
IAMA-conference-sigkdd	0	0	0
IAMA-confof-edas	29	21	0
IAMA-confof-ekaw	0	0	0
IAMA-confof-iasted	0	0	0
IAMA-confof-sigkdd	0	0	0
IAMA-edas-ekaw	0	0	0
IAMA-edas-iasted	4	2	0
IAMA-edas-sigkdd	0	0	0
IAMA-ekaw-iasted	0	0	0
IAMA-ekaw-sigkdd	0	0	0
IAMA-iasted-sigkdd	5	3	0
LogMap-cmt-conference	0	0	0
LogMap-cmt-confof	0	0	0
LogMap-cmt-edas	0	0	0
LogMap-cmt-ekaw	0	0	0
LogMap-cmt-iasted	0	0	0
LogMap-cmt-sigkdd	0	0	0
LogMap-conference-confof	0	0	0
LogMap-conference-edas	0	0	0

LogMap-conference-ekaw	0	0	0
LogMap-conference-iasted	0	0	0
LogMap-conference-sigkdd	0	0	0
LogMap-confof-edas	0	0	0
LogMap-confof-ekaw	0	0	0
LogMap-confof-iasted	0	0	0
LogMap-confof-sigkdd	0	0	0
LogMap-edas-ekaw	0	0	0
LogMap-edas-iasted	0	0	0
LogMap-edas-sigkdd	0	0	0
LogMap-ekaw-iasted	0	0	0
LogMap-ekaw-sigkdd	0	0	0
LogMap-iasted-sigkdd	0	0	0
LogMapLite-cmt-conference	0	0	0
LogMapLite-cmt-confof	21	11	0
LogMapLite-cmt-edas	4	0	0
LogMapLite-cmt-ekaw	5	4	0
LogMapLite-cmt-iasted	0	0	0
LogMapLite-cmt-sigkdd	3	0	0
LogMapLite-conference-confof	0	0	0
LogMapLite-conference-edas	0	0	0
LogMapLite-conference-ekaw	0	0	0
LogMapLite-conference-iasted	0	0	0
LogMapLite-conference-sigkdd	0	0	0
LogMapLite-confof-edas	29	29	0
LogMapLite-confof-ekaw	13	13	13
LogMapLite-confof-iasted	0	0	0
LogMapLite-confof-sigkdd	0	0	0
LogMapLite-edas-ekaw	8	3	0
LogMapLite-edas-iasted	0	0	0
LogMapLite-edas-sigkdd	0	0	0
LogMapLite-ekaw-iasted	0	0	0
LogMapLite-ekaw-sigkdd	0	0	0
LogMapLite-iasted-sigkdd	5	3	0
MaasMatch-cmt-conference	41	11	7
MaasMatch-cmt-confof	54	54	20
MaasMatch-cmt-edas	56	41	41
MaasMatch-cmt-ekaw	33	33	14
MaasMatch-cmt-iasted	149	137	37
MaasMatch-cmt-sigkdd	53	19	8
MaasMatch-conference-confof	74	61	28
MaasMatch-conference-edas	93	79	70

MaasMatch-conference-ekaw	89	83	79
MaasMatch-conference-iasted	183	13	2
MaasMatch-conference-sigkdd	60	3	2
MaasMatch-confof-edas	68	68	52
MaasMatch-confof-ekaw	70	68	29
MaasMatch-confof-iasted	164	134	77
MaasMatch-confof-sigkdd	65	59	44
MaasMatch-edas-ekaw	121	119	117
MaasMatch-edas-iasted	226	210	187
MaasMatch-edas-sigkdd	86	86	59
MaasMatch-ekaw-iasted	198	91	43
MaasMatch-ekaw-sigkdd	90	81	45
MaasMatch-iasted-sigkdd	175	31	10
MapSSS-cmt-conference	2	2	2
MapSSS-cmt-confof	0	0	0
MapSSS-cmt-edas	2	2	2
MapSSS-cmt-ekaw	0	0	0
MapSSS-cmt-iasted	0	0	0
MapSSS-cmt-sigkdd	0	0	0
MapSSS-conference-confof	4	2	2
MapSSS-conference-edas	0	0	0
MapSSS-conference-ekaw	0	0	0
MapSSS-conference-iasted	0	0	0
MapSSS-conference-sigkdd	0	0	0
MapSSS-confof-edas	0	0	0
MapSSS-confof-ekaw	0	0	0
MapSSS-confof-iasted	0	0	0
MapSSS-confof-sigkdd	41	9	9
MapSSS-edas-ekaw	0	0	0
MapSSS-edas-iasted	0	0	0
MapSSS-edas-sigkdd	17	11	11
MapSSS-ekaw-iasted	0	0	0
MapSSS-ekaw-sigkdd	0	0	0
MapSSS-iasted-sigkdd	0	0	0
ODGOMS-cmt-conference	1	0	0
ODGOMS-cmt-confof	6	2	0
ODGOMS-cmt-edas	0	0	0
ODGOMS-cmt-ekaw	2	1	0
ODGOMS-cmt-iasted	0	0	0
ODGOMS-cmt-sigkdd	3	2	0
ODGOMS-conference-confof	8	8	0
ODGOMS-conference-edas	0	0	0

ODGOMS-conference-ekaw	7	0	0
ODGOMS-conference-iasted	0	0	0
ODGOMS-conference-sigkdd	0	0	0
ODGOMS-confof-edas	29	21	0
ODGOMS-confof-ekaw	0	0	0
ODGOMS-confof-iasted	0	0	0
ODGOMS-confof-sigkdd	0	0	0
ODGOMS-edas-ekaw	0	0	0
ODGOMS-edas-iasted	4	2	0
ODGOMS-edas-sigkdd	0	0	0
ODGOMS-ekaw-iasted	0	0	0
ODGOMS-ekaw-sigkdd	0	0	0
ODGOMS-iasted-sigkdd	5	3	0
ODGOMS1_2-cmt-conference	3	2	2
ODGOMS1_2-cmt-confof	6	2	0
ODGOMS1_2-cmt-edas	0	0	0
ODGOMS1_2-cmt-ekaw	2	1	0
ODGOMS1_2-cmt-iasted	0	0	0
ODGOMS1_2-cmt-sigkdd	3	2	0
ODGOMS1_2-conference-confof	12	10	2
ODGOMS1_2-conference-edas	6	5	5
ODGOMS1_2-conference-ekaw	12	4	4
ODGOMS1_2-conference-iasted	0	0	0
ODGOMS1_2-conference-sigkdd	0	0	0
ODGOMS1_2-confof-edas	29	21	0
ODGOMS1_2-confof-ekaw	7	6	6
ODGOMS1_2-confof-iasted	38	0	0
ODGOMS1_2-confof-sigkdd	0	0	0
ODGOMS1_2-edas-ekaw	0	0	0
ODGOMS1_2-edas-iasted	137	7	5
ODGOMS1_2-edas-sigkdd	0	0	0
ODGOMS1_2-ekaw-iasted	0	0	0
ODGOMS1_2-ekaw-sigkdd	0	0	0
ODGOMS1_2-iasted-sigkdd	6	3	0
OntoK2-cmt-conference	0	0	0
OntoK2-cmt-confof	6	2	0
OntoK2-cmt-edas	0	0	0
OntoK2-cmt-ekaw	2	1	0
OntoK2-cmt-iasted	0	0	0
OntoK2-cmt-sigkdd	3	2	0
OntoK2-conference-confof	0	0	0
OntoK2-conference-edas	0	0	0

OntoK2-conference-ekaw	12	3	0
OntoK2-conference-iasted	0	0	0
OntoK2-conference-sigkdd	0	0	0
OntoK2-confof-edas	29	21	0
OntoK2-confof-ekaw	7	5	0
OntoK2-confof-iasted	0	0	0
OntoK2-confof-sigkdd	0	0	0
OntoK2-edas-ekaw	0	0	0
OntoK2-edas-iasted	0	0	0
OntoK2-edas-sigkdd	5	4	0
OntoK2-ekaw-iasted	0	0	0
OntoK2-ekaw-sigkdd	0	0	0
OntoK2-iasted-sigkdd	5	3	0
RIMOM2013-cmt-conference	63	45	45
RIMOM2013-cmt-confof	32	28	28
RIMOM2013-cmt-edas	41	26	26
RIMOM2013-cmt-ekaw	64	33	33
RIMOM2013-cmt-iasted	145	41	41
RIMOM2013-cmt-sigkdd	41	19	17
RIMOM2013-conference-confof	43	11	11
RIMOM2013-conference-edas	66	38	38
RIMOM2013-conference-ekaw	60	57	57
RIMOM2013-conference-iasted	162	9	9
RIMOM2013-conference-sigkdd	48	10	10
RIMOM2013-confof-edas	60	60	57
RIMOM2013-confof-ekaw	36	36	36
RIMOM2013-confof-iasted	17	9	9
RIMOM2013-confof-sigkdd	0	0	0
RIMOM2013-edas-ekaw	93	93	93
RIMOM2013-edas-iasted	155	38	38
RIMOM2013-edas-sigkdd	21	20	20
RIMOM2013-ekaw-iasted	184	42	42
RIMOM2013-ekaw-sigkdd	33	29	29
RIMOM2013-iasted-sigkdd	15	3	3
ServOMap_ v104-cmt-conference	0	0	0
ServOMap_ v104-cmt-confof	0	0	0
ServOMap_ v104-cmt-edas	6	6	6
ServOMap_ v104-cmt-ekaw	0	0	0
ServOMap_ v104-cmt-iasted	0	0	0
ServOMap_ v104-cmt-sigkdd	0	0	0
ServOMap_ v104-conference-confof	0	0	0
ServOMap_ v104-conference-edas	6	3	3

ServOMap_ v104-conference-ekaw	4	2	2
ServOMap_ v104-conference-iasted	0	0	0
ServOMap_ v104-conference-sigkdd	0	0	0
ServOMap_ v104-confof-edas	4	4	4
ServOMap_ v104-confof-ekaw	0	0	0
ServOMap_ v104-confof-iasted	0	0	0
ServOMap_ v104-confof-sigkdd	0	0	0
ServOMap_ v104-edas-ekaw	72	72	72
ServOMap_ v104-edas-iasted	0	0	0
ServOMap_ v104-edas-sigkdd	0	0	0
ServOMap_ v104-ekaw-iasted	0	0	0
ServOMap_ v104-ekaw-sigkdd	0	0	0
ServOMap_ v104-iasted-sigkdd	0	0	0
StringsAuto-cmt-conference	2	2	2
StringsAuto-cmt-confof	0	0	0
StringsAuto-cmt-edas	4	4	4
StringsAuto-cmt-ekaw	0	0	0
StringsAuto-cmt-iasted	0	0	0
StringsAuto-cmt-sigkdd	0	0	0
StringsAuto-conference-confof	4	2	2
StringsAuto-conference-edas	6	5	5
StringsAuto-conference-ekaw	2	2	2
StringsAuto-conference-iasted	0	0	0
StringsAuto-conference-sigkdd	0	0	0
StringsAuto-confof-edas	0	0	0
StringsAuto-confof-ekaw	7	6	6
StringsAuto-confof-iasted	0	0	0
StringsAuto-confof-sigkdd	41	9	9
StringsAuto-edas-ekaw	0	0	0
StringsAuto-edas-iasted	0	0	0
StringsAuto-edas-sigkdd	27	20	20
StringsAuto-ekaw-iasted	0	0	0
StringsAuto-ekaw-sigkdd	18	12	12
StringsAuto-iasted-sigkdd	0	0	0
SYNTHESIS-cmt-conference	0	0	0
SYNTHESIS-cmt-confof	6	2	0
SYNTHESIS-cmt-edas	0	0	0
SYNTHESIS-cmt-ekaw	2	1	0
SYNTHESIS-cmt-iasted	0	0	0
SYNTHESIS-cmt-sigkdd	3	2	0
SYNTHESIS-conference-confof	0	0	0
SYNTHESIS-conference-edas	0	0	0

SYNTHESIS-conference-ekaw	7	6	6
SYNTHESIS-conference-iasted	0	0	0
SYNTHESIS-conference-sigkdd	0	0	0
SYNTHESIS-confof-edas	29	21	0
SYNTHESIS-confof-ekaw	0	0	0
SYNTHESIS-confof-iasted	0	0	0
SYNTHESIS-confof-sigkdd	0	0	0
SYNTHESIS-edas-ekaw	0	0	0
SYNTHESIS-edas-iasted	5	2	2
SYNTHESIS-edas-sigkdd	5	4	0
SYNTHESIS-ekaw-iasted	0	0	0
SYNTHESIS-ekaw-sigkdd	0	0	0
SYNTHESIS-iasted-sigkdd	5	3	0
WeSeEMatch-cmt-conference	0	0	0
WeSeEMatch-cmt-confof	0	0	0
WeSeEMatch-cmt-edas	0	0	0
WeSeEMatch-cmt-ekaw	0	0	0
WeSeEMatch-cmt-iasted	0	0	0
WeSeEMatch-cmt-sigkdd	0	0	0
WeSeEMatch-conference-confof	0	0	0
WeSeEMatch-conference-edas	0	0	0
WeSeEMatch-conference-ekaw	7	0	0
WeSeEMatch-conference-iasted	0	0	0
WeSeEMatch-conference-sigkdd	0	0	0
WeSeEMatch-confof-edas	0	0	0
WeSeEMatch-confof-ekaw	0	0	0
WeSeEMatch-confof-iasted	0	0	0
WeSeEMatch-confof-sigkdd	0	0	0
WeSeEMatch-edas-ekaw	0	0	0
WeSeEMatch-edas-iasted	0	0	0
WeSeEMatch-edas-sigkdd	0	0	0
WeSeEMatch-ekaw-iasted	0	0	0
WeSeEMatch-ekaw-sigkdd	0	0	0
WeSeEMatch-iasted-sigkdd	0	0	0
WikiMatch-cmt-conference	1	0	0
WikiMatch-cmt-confof	10	9	0
WikiMatch-cmt-edas	0	0	0
WikiMatch-cmt-ekaw	2	1	0
WikiMatch-cmt-iasted	0	0	0
WikiMatch-cmt-sigkdd	3	2	0
WikiMatch-conference-confof	8	8	0
WikiMatch-conference-edas	4	4	0

WikiMatch-conference-ekaw	0	0	0
WikiMatch-conference-iasted	0	0	0
WikiMatch-conference-sigkdd	0	0	0
WikiMatch-confof-edas	29	21	0
WikiMatch-confof-ekaw	0	0	0
WikiMatch-confof-iasted	0	0	0
WikiMatch-confof-sigkdd	41	9	9
WikiMatch-edas-ekaw	8	3	0
WikiMatch-edas-iasted	0	0	0
WikiMatch-edas-sigkdd	22	15	11
WikiMatch-ekaw-iasted	0	0	0
WikiMatch-ekaw-sigkdd	0	0	0
WikiMatch-iasted-sigkdd	5	3	0
XMapGen-cmt-conference	0	0	0
XMapGen-cmt-confof	6	2	0
XMapGen-cmt-edas	8	8	8
XMapGen-cmt-ekaw	70	9	8
XMapGen-cmt-iasted	0	0	0
XMapGen-cmt-sigkdd	1	1	0
XMapGen-conference-confof	0	0	0
XMapGen-conference-edas	7	5	5
XMapGen-conference-ekaw	7	6	6
XMapGen-conference-iasted	0	0	0
XMapGen-conference-sigkdd	0	0	0
XMapGen-confof-edas	31	23	6
XMapGen-confof-ekaw	0	0	0
XMapGen-confof-iasted	0	0	0
XMapGen-confof-sigkdd	0	0	0
XMapGen-edas-ekaw	157	0	0
XMapGen-edas-iasted	0	0	0
XMapGen-edas-sigkdd	3	3	0
XMapGen-ekaw-iasted	0	0	0
XMapGen-ekaw-sigkdd	0	0	0
XMapGen-iasted-sigkdd	6	3	0
XMapGen1_4-cmt-conference	44	38	38
XMapGen1_4-cmt-confof	17	12	9
XMapGen1_4-cmt-edas	34	22	22
XMapGen1_4-cmt-ekaw	86	33	32
XMapGen1_4-cmt-iasted	0	0	0
XMapGen1_4-cmt-sigkdd	23	10	8
XMapGen1_4-conference-confof	9	8	8
XMapGen1_4-conference-edas	14	11	11

XMapGen1_4-conference-ekaw	25	14	14
XMapGen1_4-conference-iasted	0	0	0
XMapGen1_4-conference-sigkdd	0	0	0
XMapGen1_4-confof-edas	47	45	31
XMapGen1_4-confof-ekaw	83	52	52
XMapGen1_4-confof-iasted	0	0	0
XMapGen1_4-confof-sigkdd	-	10	10
XMapGen1_4-edas-ekaw	-	54	54
XMapGen1_4-edas-iasted	6	2	2
XMapGen1_4-edas-sigkdd	7	6	2
XMapGen1_4-ekaw-iasted	2	2	2
XMapGen1_4-ekaw-sigkdd	20	15	15
XMapGen1_4-iasted-sigkdd	181	36	21
XMapSiG1_3-cmt-conference	0	0	0
XMapSiG1_3-cmt-confof	1	0	0
XMapSiG1_3-cmt-edas	0	0	0
XMapSiG1_3-cmt-ekaw	70	9	8
XMapSiG1_3-cmt-iasted	0	0	0
XMapSiG1_3-cmt-sigkdd	3	2	0
XMapSiG1_3-conference-confof	0	0	0
XMapSiG1_3-conference-edas	2	2	2
XMapSiG1_3-conference-ekaw	20	12	12
XMapSiG1_3-conference-iasted	0	0	0
XMapSiG1_3-conference-sigkdd	0	0	0
XMapSiG1_3-confof-edas	29	21	0
XMapSiG1_3-confof-ekaw	0	0	0
XMapSiG1_3-confof-iasted	0	0	0
XMapSiG1_3-confof-sigkdd	0	0	0
XMapSiG1_3-edas-ekaw	151	0	0
XMapSiG1_3-edas-iasted	0	0	0
XMapSiG1_3-edas-sigkdd	0	0	0
XMapSiG1_3-ekaw-iasted	0	0	0
XMapSiG1_3-ekaw-sigkdd	0	0	0
XMapSiG1_3-iasted-sigkdd	6	3	0
XMapSiG1_4-cmt-conference	0	0	0
XMapSiG1_4-cmt-confof	1	0	0
XMapSiG1_4-cmt-edas	0	0	0
XMapSiG1_4-cmt-ekaw	2	1	0
XMapSiG1_4-cmt-iasted	0	0	0
XMapSiG1_4-cmt-sigkdd	1	1	0
XMapSiG1_4-conference-confof	0	0	0
XMapSiG1_4-conference-edas	0	0	0

XMapSiG1_4-conference-ekaw	0	0	0
XMapSiG1_4-conference-iasted	0	0	0
XMapSiG1_4-conference-sigkdd	0	0	0
XMapSiG1_4-confof-edas	29	21	0
XMapSiG1_4-confof-ekaw	0	0	0
XMapSiG1_4-confof-iasted	0	0	0
XMapSiG1_4-confof-sigkdd	0	0	0
XMapSiG1_4-edas-ekaw	0	0	0
XMapSiG1_4-edas-iasted	0	0	0
XMapSiG1_4-edas-sigkdd	0	0	0
XMapSiG1_4-ekaw-iasted	0	0	0
XMapSiG1_4-ekaw-sigkdd	0	0	0
XMapSiG1_4-iasted-sigkdd	5	3	0
YAM++-cmt-conference	0	0	0
YAM++-cmt-confof	0	0	0
YAM++-cmt-edas	0	0	0
YAM++-cmt-ekaw	0	0	0
YAM++-cmt-iasted	0	0	0
YAM++-cmt-sigkdd	3	2	0
YAM++-conference-confof	0	0	0
YAM++-conference-edas	0	0	0
YAM++-conference-ekaw	0	0	0
YAM++-conference-iasted	0	0	0
YAM++-conference-sigkdd	0	0	0
YAM++-confof-edas	0	0	0
YAM++-confof-ekaw	0	0	0
YAM++-confof-iasted	0	0	0
YAM++-confof-sigkdd	0	0	0
YAM++-edas-ekaw	0	0	0
YAM++-edas-iasted	0	0	0
YAM++-edas-sigkdd	0	0	0
YAM++-ekaw-iasted	0	0	0
YAM++-ekaw-sigkdd	0	0	0
YAM++-iasted-sigkdd	0	0	0

Appendix B

Full Table of Repair Tests

Note: '-'indicates the reasoner could not finish the task.

Test Case (matcher - ontology 1 - ontology 2)	**Before Repair**		**LogMap Repair**		**LogMap Repair +**	
	Num. of Mappings	**Unsat.**	**Repair Size**	**Unsat.**	**Repair Size**	**Unsat.**
AML-cmt-conference	20	0	0	0	0	0
AML-cmt-confof	12	0	0	0	0	0
AML-cmt-edas	18	0	0	0	0	0
AML-cmt-ekaw	12	0	0	0	0	0
AML-cmt-iasted	10	0	0	0	0	0
AML-cmt-sigkdd	20	0	0	0	0	0
AML-conference-confof	24	0	0	0	0	0
AML-conference-edas	22	0	0	0	0	0
AML-conference-ekaw	30	0	0	0	0	0
AML-conference-iasted	12	0	0	0	0	0
AML-conference-sigkdd	22	0	0	0	0	0
AML-confof-edas	22	0	0	0	0	0
AML-confof-ekaw	32	0	0	0	0	0
AML-confof-iasted	10	0	0	0	0	0
AML-confof-sigkdd	8	0	0	0	0	0
AML-edas-ekaw	28	0	0	0	0	0
AML-edas-iasted	16	0	0	0	0	0
AML-edas-sigkdd	14	0	0	0	0	0
AML-ekaw-iasted	14	0	0	0	0	0
AML-ekaw-sigkdd	16	0	0	0	0	0
AML-iasted-sigkdd	30	0	0	0	0	0
AMLback-cmt-conference	24	0	0	0	0	0
AMLback-cmt-confof	14	0	0	0	0	0

AMLback-cmt-edas	18	0	0	0	0	0
AMLback-cmt-ekaw	12	0	0	0	0	0
AMLback-cmt-iasted	10	0	0	0	0	0
AMLback-cmt-sigkdd	20	0	0	0	0	0
AMLback-conference-confof	24	0	0	0	0	0
AMLback-conference-edas	22	0	0	0	0	0
AMLback-conference-ekaw	34	0	0	0	0	0
AMLback-conference-iasted	12	0	0	0	0	0
AMLback-conference-sigkdd	22	0	0	0	0	0
AMLback-confof-edas	22	0	0	0	0	0
AMLback-confof-ekaw	34	0	0	0	0	0
AMLback-confof-iasted	10	0	0	0	0	0
AMLback-confof-sigkdd	8	0	0	0	0	0
AMLback-edas-ekaw	28	0	0	0	0	0
AMLback-edas-iasted	16	0	0	0	0	0
AMLback-edas-sigkdd	18	0	0	0	0	0
AMLback-ekaw-iasted	14	0	0	0	0	0
AMLback-ekaw-sigkdd	16	0	0	0	0	0
AMLback-iasted-sigkdd	30	0	0	0	0	0
CIDER_CL-cmt-conference	64	33	4	7	4	7
CIDER_CL-cmt-confof	30	24	2	15	4	13
CIDER_CL-cmt-edas	52	41	6	4	8	0
CIDER_CL-cmt-ekaw	30	11	4	5	6	5
CIDER_CL-cmt-iasted	26	1	0	1	1	0
CIDER_CL-cmt-sigkdd	40	5	1	3	3	3
CIDER_CL-conference-confof	46	40	7	24	10	7
CIDER_CL-conference-edas	70	59	11	12	10	7
CIDER_CL-conference-ekaw	66	82	9	9	13	7
CIDER_CL-conference-iasted	50	1	1	0	1	0
CIDER_CL-conference-sigkdd	50	3	0	3	0	3
CIDER_CL-confof-edas	58	59	8	47	12	5
CIDER_CL-confof-ekaw	36	53	6	0	6	0
CIDER_CL-confof-iasted	22	50	2	4	1	4
CIDER_CL-confof-sigkdd	32	14	1	14	5	0
CIDER_CL-edas-ekaw	56	83	19	10	25	7
CIDER_CL-edas-iasted	44	170	6	159	10	4
CIDER_CL-edas-sigkdd	56	41	4	34	6	0
CIDER_CL-ekaw-iasted	56	188	10	0	10	0
CIDER_CL-ekaw-sigkdd	34	22	4	8	6	0
CIDER_CL-iasted-sigkdd	42	6	0	6	2	0
CroMatcher-cmt-confof	54	41	9	37	14	5
CroMatcher-cmt-edas	86	38	13	26	15	26

CroMatcher-cmt-ekaw	70	9	8	5	12	2
CroMatcher-cmt-iasted	78	146	9	145	21	145
CroMatcher-cmt-sigkdd	66	50	5	50	10	49
CroMatcher-confof-edas	96	54	22	39	48	5
CroMatcher-confof-ekaw	74	45	27	14	30	0
CroMatcher-confof-iasted	82	155	18	154	22	154
CroMatcher-confof-sigkdd	60	61	18	53	20	55
CroMatcher-edas-ekaw	120	67	21	33	34	22
CroMatcher-edas-iasted	112	182	36	160	47	157
CroMatcher-edas-sigkdd	98	63	20	39	22	1
CroMatcher-ekaw-iasted	110	189	15	179	21	179
CroMatcher-ekaw-sigkdd	64	42	16	13	20	13
CroMatcher-iasted-sigkdd	94	174	6	42	8	0
HerTUDA-cmt-conference	16	0	0	0	0	0
HerTUDA-cmt-confof	20	24	2	6	4	1
HerTUDA-cmt-edas	20	0	0	0	0	0
HerTUDA-cmt-ekaw	12	2	0	2	2	0
HerTUDA-cmt-iasted	10	0	0	0	0	0
HerTUDA-cmt-sigkdd	20	3	0	3	2	3
HerTUDA-conference-confof	24	8	0	8	2	0
HerTUDA-conference-edas	26	0	0	0	0	0
HerTUDA-conference-ekaw	28	0	0	0	0	0
HerTUDA-conference-iasted	12	0	0	0	0	0
HerTUDA-conference-sigkdd	22	0	0	0	0	0
HerTUDA-confof-edas	36	29	0	29	2	3
HerTUDA-confof-ekaw	28	14	0	14	2	0
HerTUDA-confof-iasted	10	0	0	0	0	0
HerTUDA-confof-sigkdd	10	41	2	0	2	0
HerTUDA-edas-ekaw	28	8	0	8	2	0
HerTUDA-edas-iasted	20	4	0	4	2	0
HerTUDA-edas-sigkdd	18	22	2	5	4	0
HerTUDA-ekaw-iasted	12	0	0	0	0	0
HerTUDA-ekaw-sigkdd	14	0	0	0	0	0
HerTUDA-iasted-sigkdd	28	5	0	5	2	0
HotMatch-cmt-conference	16	0	0	0	0	0
HotMatch-cmt-confof	18	10	0	10	2	1
HotMatch-cmt-edas	24	0	0	0	0	0
HotMatch-cmt-ekaw	14	2	0	2	2	0
HotMatch-cmt-iasted	10	0	0	0	0	0
HotMatch-cmt-sigkdd	22	3	0	3	2	0
HotMatch-conference-confof	24	8	0	8	2	0
HotMatch-conference-edas	26	0	0	0	0	0

HotMatch-conference-ekaw	28	0	0	0	0	0
HotMatch-conference-iasted	14	0	0	0	0	0
HotMatch-conference-sigkdd	22	0	0	0	0	0
HotMatch-confof-edas	36	29	0	29	2	3
HotMatch-confof-ekaw	28	14	0	14	2	0
HotMatch-confof-iasted	10	0	0	0	0	0
HotMatch-confof-sigkdd	10	41	2	0	2	0
HotMatch-edas-ekaw	30	8	0	8	2	0
HotMatch-edas-iasted	22	4	0	4	2	0
HotMatch-edas-sigkdd	22	22	2	5	4	0
HotMatch-ekaw-iasted	12	0	0	0	0	0
HotMatch-ekaw-sigkdd	14	0	0	0	0	0
HotMatch-iasted-sigkdd	40	7	0	7	2	0
IAMA-cmt-conference	14	1	0	1	0	1
IAMA-cmt-confof	12	1	0	1	0	1
IAMA-cmt-edas	16	0	0	0	0	0
IAMA-cmt-ekaw	12	2	0	2	2	0
IAMA-cmt-iasted	10	0	0	0	0	0
IAMA-cmt-sigkdd	22	3	0	3	2	3
IAMA-conference-confof	16	0	0	0	0	0
IAMA-conference-edas	22	0	2	0	2	0
IAMA-conference-ekaw	28	0	0	0	0	0
IAMA-conference-iasted	12	0	0	0	0	0
IAMA-conference-sigkdd	18	0	0	0	0	0
IAMA-confof-edas	36	29	2	29	4	3
IAMA-confof-ekaw	22	0	0	0	0	0
IAMA-confof-iasted	8	0	0	0	0	0
IAMA-confof-sigkdd	8	0	0	0	0	0
IAMA-edas-ekaw	28	0	0	0	0	0
IAMA-edas-iasted	20	4	0	4	2	0
IAMA-edas-sigkdd	14	0	0	0	0	0
IAMA-ekaw-iasted	12	0	0	0	0	0
IAMA-ekaw-sigkdd	14	0	0	0	0	0
IAMA-iasted-sigkdd	28	5	0	5	2	0
LogMap-cmt-conference	24	0	0	0	0	0
LogMap-cmt-confof	12	0	0	0	0	0
LogMap-cmt-edas	18	0	0	0	0	0
LogMap-cmt-ekaw	14	0	0	0	0	0
LogMap-cmt-iasted	10	0	0	0	0	0
LogMap-cmt-sigkdd	22	0	0	0	0	0
LogMap-conference-confof	28	0	0	0	0	0
LogMap-conference-edas	26	0	0	0	0	0

LogMap-conference-ekaw	40	0	0	0	0	0
LogMap-conference-iasted	18	0	0	0	0	0
LogMap-conference-sigkdd	28	0	0	0	0	0
LogMap-confof-edas	26	0	0	0	0	0
LogMap-confof-ekaw	30	0	0	0	0	0
LogMap-confof-iasted	8	0	0	0	0	0
LogMap-confof-sigkdd	10	0	0	0	0	0
LogMap-edas-ekaw	32	0	0	0	0	0
LogMap-edas-iasted	16	0	0	0	0	0
LogMap-edas-sigkdd	16	0	0	0	0	0
LogMap-ekaw-iasted	20	0	0	0	0	0
LogMap-ekaw-sigkdd	16	0	0	0	0	0
LogMap-iasted-sigkdd	36	0	0	0	0	0
LogMapLite-cmt-conference	18	0	0	0	0	0
LogMapLite-cmt-confof	18	21	2	6	4	1
LogMapLite-cmt-edas	22	4	0	4	0	4
LogMapLite-cmt-ekaw	18	5	0	5	4	2
LogMapLite-cmt-iasted	10	0	0	0	0	0
LogMapLite-cmt-sigkdd	18	3	0	3	0	3
LogMapLite-conference-confof	20	0	0	0	0	0
LogMapLite-conference-edas	24	0	0	0	0	0
LogMapLite-conference-ekaw	26	0	0	0	0	0
LogMapLite-conference-iasted	10	0	0	0	0	0
LogMapLite-conference-sigkdd	20	0	0	0	0	0
LogMapLite-confof-edas	38	29	2	29	6	0
LogMapLite-confof-ekaw	26	13	1	0	3	0
LogMapLite-confof-iasted	8	0	0	0	0	0
LogMapLite-confof-sigkdd	8	0	0	0	0	0
LogMapLite-edas-ekaw	34	8	0	8	2	0
LogMapLite-edas-iasted	18	0	0	0	0	0
LogMapLite-edas-sigkdd	16	0	0	0	0	0
LogMapLite-ekaw-iasted	20	0	4	0	4	0
LogMapLite-ekaw-sigkdd	16	0	0	0	0	0
LogMapLite-iasted-sigkdd	30	5	0	5	2	0
MaasMatch-cmt-conference	174	41	33	32	40	30
MaasMatch-cmt-confof	128	54	46	44	81	9
MaasMatch-cmt-edas	150	56	50	24	57	9
MaasMatch-cmt-ekaw	116	33	18	21	31	11
MaasMatch-cmt-iasted	140	149	20	148	66	142
MaasMatch-cmt-sigkdd	114	53	25	50	33	50
MaasMatch-conference-confof	148	74	41	49	44	46
MaasMatch-conference-edas	216	93	64	23	67	15

MaasMatch-conference-ekaw	180	89	39	43	49	39
MaasMatch-conference-iasted	198	183	18	183	21	-
MaasMatch-conference-sigkdd	144	60	11	1	13	1
MaasMatch-confof-edas	148	68	57	41	71	6
MaasMatch-confof-ekaw	130	70	52	49	78	26
MaasMatch-confof-iasted	148	164	49	164	71	152
MaasMatch-confof-sigkdd	122	65	40	45	38	45
MaasMatch-edas-ekaw	202	121	96	19	99	28
MaasMatch-edas-iasted	288	226	116	193	190	0
MaasMatch-edas-sigkdd	154	86	44	59	77	12
MaasMatch-ekaw-iasted	198	198	22	198	43	192
MaasMatch-ekaw-sigkdd	136	90	29	81	58	17
MaasMatch-iasted-sigkdd	144	175	13	-	19	0
MapSSS-cmt-conference	22	2	4	0	4	0
MapSSS-cmt-confof	12	0	2	0	2	0
MapSSS-cmt-edas	22	2	4	0	4	0
MapSSS-cmt-ekaw	12	0	2	0	2	0
MapSSS-cmt-iasted	12	0	2	0	2	0
MapSSS-cmt-sigkdd	20	0	2	0	2	0
MapSSS-conference-confof	26	4	3	0	4	2
MapSSS-conference-edas	24	0	2	0	2	0
MapSSS-conference-ekaw	26	0	2	0	2	0
MapSSS-conference-iasted	12	0	2	0	2	0
MapSSS-conference-sigkdd	24	0	2	0	2	0
MapSSS-confof-edas	22	0	2	0	2	0
MapSSS-confof-ekaw	28	0	2	0	2	0
MapSSS-confof-iasted	12	0	2	0	2	0
MapSSS-confof-sigkdd	12	41	3	0	4	0
MapSSS-edas-ekaw	28	0	2	0	2	0
MapSSS-edas-iasted	20	0	2	0	2	0
MapSSS-edas-sigkdd	18	17	4	0	3	0
MapSSS-ekaw-iasted	14	0	2	0	2	0
MapSSS-ekaw-sigkdd	16	0	2	0	2	0
MapSSS-iasted-sigkdd	32	0	2	0	2	0
ODGOMS-cmt-conference	16	1	0	1	0	1
ODGOMS-cmt-confof	16	6	0	6	2	1
ODGOMS-cmt-edas	20	0	0	0	0	0
ODGOMS-cmt-ekaw	12	2	0	2	2	0
ODGOMS-cmt-iasted	10	0	0	0	0	0
ODGOMS-cmt-sigkdd	22	3	0	3	2	3
ODGOMS-conference-confof	24	8	0	8	2	0
ODGOMS-conference-edas	28	0	0	0	0	0

ODGOMS-conference-ekaw	32	7	0	7	0	7
ODGOMS-conference-iasted	12	0	0	0	0	0
ODGOMS-conference-sigkdd	18	0	0	0	0	0
ODGOMS-confof-edas	36	29	0	29	2	3
ODGOMS-confof-ekaw	22	0	0	0	0	0
ODGOMS-confof-iasted	10	0	0	0	0	0
ODGOMS-confof-sigkdd	8	0	0	0	0	0
ODGOMS-edas-ekaw	32	0	0	0	0	0
ODGOMS-edas-iasted	20	4	0	4	2	0
ODGOMS-edas-sigkdd	14	0	0	0	0	0
ODGOMS-ekaw-iasted	12	0	0	0	0	0
ODGOMS-ekaw-sigkdd	14	0	0	0	0	0
ODGOMS-iasted-sigkdd	28	5	0	5	2	0
ODGOMS1_2-cmt-conference	26	3	2	1	2	1
ODGOMS1_2-cmt-confof	16	6	0	6	2	1
ODGOMS1_2-cmt-edas	20	0	0	0	0	0
ODGOMS1_2-cmt-ekaw	16	2	0	2	2	0
ODGOMS1_2-cmt-iasted	10	0	0	0	0	0
ODGOMS1_2-cmt-sigkdd	26	3	0	3	2	3
ODGOMS1_2-conference-confof	32	12	1	8	3	0
ODGOMS1_2-conference-edas	34	6	2	0	2	0
ODGOMS1_2-conference-ekaw	46	12	4	7	4	7
ODGOMS1_2-conference-iasted	14	0	0	0	0	0
ODGOMS1_2-conference-sigkdd	20	0	0	0	0	0
ODGOMS1_2-confof-edas	36	29	0	29	2	3
ODGOMS1_2-confof-ekaw	40	7	6	0	6	0
ODGOMS1_2-confof-iasted	14	38	0	38	0	38
ODGOMS1_2-confof-sigkdd	8	0	0	0	0	0
ODGOMS1_2-edas-ekaw	34	0	0	0	0	0
ODGOMS1_2-edas-iasted	26	137	2	4	4	0
ODGOMS1_2-edas-sigkdd	14	0	0	0	0	0
ODGOMS1_2-ekaw-iasted	16	0	0	0	0	0
ODGOMS1_2-ekaw-sigkdd	16	0	0	0	0	0
ODGOMS1_2-iasted-sigkdd	30	6	0	6	2	0
OntoK2-cmt-conference	12	0	0	0	0	0
OntoK2-cmt-confof	16	6	0	6	2	1
OntoK2-cmt-edas	20	0	0	0	0	0
OntoK2-cmt-ekaw	12	2	0	2	2	0
OntoK2-cmt-iasted	10	0	0	0	0	0
OntoK2-cmt-sigkdd	22	3	0	3	2	3
OntoK2-conference-confof	18	0	0	0	0	0
OntoK2-conference-edas	16	0	0	0	0	0

OntoK2-conference-ekaw	28	12	0	12	2	7
OntoK2-conference-iasted	12	0	0	0	0	0
OntoK2-conference-sigkdd	18	0	0	0	0	0
OntoK2-confof-edas	38	29	4	29	6	3
OntoK2-confof-ekaw	22	7	0	7	2	0
OntoK2-confof-iasted	8	0	0	0	0	0
OntoK2-confof-sigkdd	10	0	0	0	0	0
OntoK2-edas-ekaw	24	0	0	0	0	0
OntoK2-edas-iasted	16	0	0	0	0	0
OntoK2-edas-sigkdd	18	5	0	5	2	0
OntoK2-ekaw-iasted	14	0	0	0	0	0
OntoK2-ekaw-sigkdd	14	0	0	0	0	0
OntoK2-iasted-sigkdd	28	5	0	5	2	0
RIMOM2013-cmt-conference	86	63	22	30	44	0
RIMOM2013-cmt-confof	124	32	53	2	53	2
RIMOM2013-cmt-edas	94	41	52	0	42	0
RIMOM2013-cmt-ekaw	114	64	50	7	73	0
RIMOM2013-cmt-iasted	100	145	45	0	31	0
RIMOM2013-cmt-sigkdd	66	41	16	3	15	35
RIMOM2013-conference-confof	144	43	17	0	18	0
RIMOM2013-conference-edas	74	66	5	0	5	0
RIMOM2013-conference-ekaw	138	60	28	1	30	2
RIMOM2013-conference-iasted	72	162	22	1	22	1
RIMOM2013-conference-sigkdd	74	48	13	0	13	0
RIMOM2013-confof-edas	158	60	45	35	50	9
RIMOM2013-confof-ekaw	178	36	43	0	44	0
RIMOM2013-confof-iasted	68	17	21	0	25	0
RIMOM2013-confof-sigkdd	66	0	0	0	0	0
RIMOM2013-edas-ekaw	152	93	78	0	75	0
RIMOM2013-edas-iasted	188	155	58	16	110	0
RIMOM2013-edas-sigkdd	92	21	33	0	24	0
RIMOM2013-ekaw-iasted	76	184	20	0	19	0
RIMOM2013-ekaw-sigkdd	82	33	29	0	36	0
RIMOM2013-iasted-sigkdd	240	15	1	15	3	0
ServOMap_ v104-cmt-conference	24	0	0	0	0	0
ServOMap_ v104-cmt-confof	10	0	0	0	0	0
ServOMap_ v104-cmt-edas	20	6	1	0	1	0
ServOMap_ v104-cmt-ekaw	12	0	0	0	0	0
ServOMap_ v104-cmt-iasted	10	0	0	0	0	0
ServOMap_ v104-cmt-sigkdd	16	0	0	0	0	0
ServOMap_ v104-conference-confof	16	0	0	0	0	0
ServOMap_ v104-conference-edas	24	6	1	6	4	0

ServOMap_ v104-conference-ekaw	38	4	1	0	1	0
ServOMap_ v104-conference-iasted	22	0	0	0	0	0
ServOMap_ v104-conference-sigkdd	26	0	0	0	0	0
ServOMap_ v104-confof-edas	32	4	7	0	7	0
ServOMap_ v104-confof-ekaw	34	0	0	0	0	0
ServOMap_ v104-confof-iasted	16	0	0	0	0	0
ServOMap_ v104-confof-sigkdd	10	0	0	0	0	0
ServOMap_ v104-edas-ekaw	38	72	10	0	3	0
ServOMap_ v104-edas-iasted	22	0	0	0	0	0
ServOMap_ v104-edas-sigkdd	18	0	0	0	0	0
ServOMap_ v104-ekaw-iasted	24	0	0	0	0	0
ServOMap_ v104-ekaw-sigkdd	24	0	0	0	0	0
ServOMap_ v104-iasted-sigkdd	28	0	0	0	0	0
StringsAuto-cmt-conference	22	2	4	0	4	0
StringsAuto-cmt-confof	12	0	2	0	2	0
StringsAuto-cmt-edas	24	4	6	0	6	0
StringsAuto-cmt-ekaw	16	0	2	0	2	0
StringsAuto-cmt-iasted	12	0	2	0	2	0
StringsAuto-cmt-sigkdd	22	0	2	0	2	0
StringsAuto-conference-confof	26	4	3	0	4	2
StringsAuto-conference-edas	26	6	3	0	3	0
StringsAuto-conference-ekaw	32	2	4	0	4	0
StringsAuto-conference-iasted	14	0	2	0	2	0
StringsAuto-conference-sigkdd	26	0	2	0	2	0
StringsAuto-confof-edas	24	0	2	0	2	0
StringsAuto-confof-ekaw	36	7	8	0	6	0
StringsAuto-confof-iasted	12	0	2	0	2	0
StringsAuto-confof-sigkdd	12	41	3	0	4	0
StringsAuto-edas-ekaw	30	0	2	0	2	0
StringsAuto-edas-iasted	28	0	2	0	2	0
StringsAuto-edas-sigkdd	20	27	6	0	3	0
StringsAuto-ekaw-iasted	16	0	2	0	2	0
StringsAuto-ekaw-sigkdd	20	18	4	0	3	0
StringsAuto-iasted-sigkdd	34	0	2	0	2	0
SYNTHESIS-cmt-conference	10	0	0	0	0	0
SYNTHESIS-cmt-confof	16	6	0	6	2	1
SYNTHESIS-cmt-edas	22	0	0	0	0	0
SYNTHESIS-cmt-ekaw	12	2	0	2	2	0
SYNTHESIS-cmt-iasted	10	0	0	0	0	0
SYNTHESIS-cmt-sigkdd	16	3	0	3	2	3
SYNTHESIS-conference-confof	16	0	0	0	0	0
SYNTHESIS-conference-edas	16	0	0	0	0	0

SYNTHESIS-conference-ekaw	26	7	1	0	1	0
SYNTHESIS-conference-iasted	12	0	0	0	0	0
SYNTHESIS-conference-sigkdd	18	0	0	0	0	0
SYNTHESIS-confof-edas	38	29	4	29	6	3
SYNTHESIS-confof-ekaw	24	0	0	0	0	0
SYNTHESIS-confof-iasted	8	0	0	0	0	0
SYNTHESIS-confof-sigkdd	8	0	0	0	0	0
SYNTHESIS-edas-ekaw	18	0	0	0	0	0
SYNTHESIS-edas-iasted	14	5	2	0	2	0
SYNTHESIS-edas-sigkdd	16	5	0	5	2	0
SYNTHESIS-ekaw-iasted	12	0	0	0	0	0
SYNTHESIS-ekaw-sigkdd	14	0	0	0	0	0
SYNTHESIS-iasted-sigkdd	28	5	0	5	2	0
WeSeEMatch-cmt-conference	12	0	0	0	0	0
WeSeEMatch-cmt-confof	14	0	0	0	0	0
WeSeEMatch-cmt-edas	18	0	0	0	0	0
WeSeEMatch-cmt-ekaw	10	0	0	0	0	0
WeSeEMatch-cmt-iasted	10	0	0	0	0	0
WeSeEMatch-cmt-sigkdd	16	0	0	0	0	0
WeSeEMatch-conference-confof	18	0	0	0	0	0
WeSeEMatch-conference-edas	18	0	0	0	0	0
WeSeEMatch-conference-ekaw	26	7	0	7	0	7
WeSeEMatch-conference-iasted	10	0	0	0	0	0
WeSeEMatch-conference-sigkdd	18	0	0	0	0	0
WeSeEMatch-confof-edas	22	0	0	0	0	0
WeSeEMatch-confof-ekaw	22	0	0	0	0	0
WeSeEMatch-confof-iasted	10	0	0	0	0	0
WeSeEMatch-confof-sigkdd	8	0	0	0	0	0
WeSeEMatch-edas-ekaw	24	0	0	0	0	0
WeSeEMatch-edas-iasted	16	0	0	0	0	0
WeSeEMatch-edas-sigkdd	14	0	0	0	0	0
WeSeEMatch-ekaw-iasted	12	0	0	0	0	0
WeSeEMatch-ekaw-sigkdd	14	0	0	0	0	0
WeSeEMatch-iasted-sigkdd	24	0	0	0	0	0
WikiMatch-cmt-conference	18	1	0	1	0	1
WikiMatch-cmt-confof	18	10	0	10	2	1
WikiMatch-cmt-edas	22	0	0	0	0	0
WikiMatch-cmt-ekaw	12	2	0	2	2	0
WikiMatch-cmt-iasted	10	0	0	0	0	0
WikiMatch-cmt-sigkdd	20	3	0	3	2	3
WikiMatch-conference-confof	24	8	0	8	2	0
WikiMatch-conference-edas	26	4	0	4	2	0

WikiMatch-conference-ekaw	26	0	0	0	0	0
WikiMatch-conference-iasted	12	0	0	0	0	0
WikiMatch-conference-sigkdd	22	0	0	0	0	0
WikiMatch-confof-edas	38	29	0	29	2	3
WikiMatch-confof-ekaw	20	0	0	0	0	0
WikiMatch-confof-iasted	10	0	0	0	0	0
WikiMatch-confof-sigkdd	10	41	2	0	2	0
WikiMatch-edas-ekaw	30	8	0	8	2	0
WikiMatch-edas-iasted	18	0	0	0	0	0
WikiMatch-edas-sigkdd	18	22	2	5	4	0
WikiMatch-ekaw-iasted	12	0	0	0	0	0
WikiMatch-ekaw-sigkdd	14	0	0	0	0	0
WikiMatch-iasted-sigkdd	30	5	0	5	2	0
XMapGen-cmt-conference	12	0	2	0	2	0
XMapGen-cmt-confof	24	6	2	6	4	1
XMapGen-cmt-edas	38	8	6	4	6	4
XMapGen-cmt-ekaw	22	70	4	2	6	0
XMapGen-cmt-iasted	16	0	2	0	2	0
XMapGen-cmt-sigkdd	22	1	2	1	4	1
XMapGen-conference-confof	20	0	2	0	2	0
XMapGen-conference-edas	16	7	5	0	5	0
XMapGen-conference-ekaw	26	7	1	0	1	0
XMapGen-conference-iasted	18	0	2	0	2	0
XMapGen-conference-sigkdd	24	0	2	0	2	0
XMapGen-confof-edas	50	31	8	29	10	3
XMapGen-confof-ekaw	36	0	2	0	2	0
XMapGen-confof-iasted	20	0	2	0	2	0
XMapGen-confof-sigkdd	10	0	2	0	2	0
XMapGen-edas-ekaw	32	157	2	0	2	0
XMapGen-edas-iasted	22	0	2	0	2	0
XMapGen-edas-sigkdd	20	3	2	3	4	0
XMapGen-ekaw-iasted	14	0	0	0	0	0
XMapGen-ekaw-sigkdd	14	0	0	0	0	0
XMapGen1_4-cmt-conference	30	44	7	0	7	0
XMapGen1_4-cmt-confof	32	17	7	6	9	1
XMapGen1_4-cmt-edas	42	34	7	4	6	4
XMapGen1_4-cmt-ekaw	38	86	14	2	17	0
XMapGen1_4-cmt-iasted	24	0	2	0	2	0
XMapGen1_4-cmt-sigkdd	36	23	6	3	8	3
XMapGen1_4-conference-confof	30	9	4	0	5	0
XMapGen1_4-conference-edas	30	14	6	0	6	0
XMapGen1_4-conference-ekaw	38	25	8	0	8	0

XMapGen1_ 4-conference-iasted	22	0	2	0	2	0
XMapGen1_ 4-conference-sigkdd	28	0	2	0	2	0
XMapGen1_ 4-confof-edas	64	47	17	29	19	3
XMapGen1_ 4-confof-ekaw	52	83	14	0	14	0
XMapGen1_ 4-confof-iasted	26	0	2	0	2	0
XMapGen1_ 4-confof-sigkdd	20	-	7	0	7	0
XMapGen1_ 4-edas-ekaw	56	-	15	0	15	0
XMapGen1_ 4-edas-iasted	40	6	4	0	4	0
XMapGen1_ 4-edas-sigkdd	26	7	4	5	6	0
XMapGen1_ 4-ekaw-iasted	30	2	3	0	3	0
XMapGen1_ 4-ekaw-sigkdd	24	20	4	0	4	0
XMapGen1_ 4-iasted-sigkdd	92	181	4	9	7	0
XMapSiG1_ 3-cmt-conference	14	0	2	0	2	0
XMapSiG1_ 3-cmt-confof	20	1	2	1	2	1
XMapSiG1_ 3-cmt-edas	20	0	2	0	2	0
XMapSiG1_ 3-cmt-ekaw	18	70	4	2	6	0
XMapSiG1_ 3-cmt-iasted	16	0	2	0	2	0
XMapSiG1_ 3-cmt-sigkdd	26	3	2	3	4	3
XMapSiG1_ 3-conference-confof	18	0	2	0	2	0
XMapSiG1_ 3-conference-edas	22	2	3	0	3	0
XMapSiG1_ 3-conference-ekaw	30	20	5	0	5	0
XMapSiG1_ 3-conference-iasted	18	0	2	0	2	0
XMapSiG1_ 3-conference-sigkdd	24	0	2	0	2	0
XMapSiG1_ 3-confof-edas	40	29	4	29	6	3
XMapSiG1_ 3-confof-ekaw	28	0	2	0	2	0
XMapSiG1_ 3-confof-iasted	10	0	2	0	2	0
XMapSiG1_ 3-confof-sigkdd	10	0	2	0	2	0
XMapSiG1_ 3-edas-ekaw	28	151	2	0	2	0
XMapSiG1_ 3-edas-iasted	20	0	2	0	2	0
XMapSiG1_ 3-edas-sigkdd	16	0	2	0	2	0
XMapSiG1_ 3-ekaw-iasted	16	0	2	0	2	0
XMapSiG1_ 3-ekaw-sigkdd	18	0	2	0	2	0
XMapSiG1_ 3-iasted-sigkdd	38	6	2	6	4	0
XMapSiG1_ 4-cmt-conference	10	0	2	0	2	0
XMapSiG1_ 4-cmt-confof	14	1	2	1	2	1
XMapSiG1_ 4-cmt-edas	16	0	2	0	2	0
XMapSiG1_ 4-cmt-ekaw	12	2	0	2	2	0
XMapSiG1_ 4-cmt-iasted	12	0	2	0	2	0
XMapSiG1_ 4-cmt-sigkdd	18	1	2	1	4	1
XMapSiG1_ 4-conference-confof	14	0	2	0	2	0
XMapSiG1_ 4-conference-edas	12	0	2	0	2	0
XMapSiG1_ 4-conference-ekaw	20	0	0	0	0	0

XMapSiG1_4-conference-iasted	14	0	2	0	2	0
XMapSiG1_4-conference-sigkdd	18	0	2	0	2	0
XMapSiG1_4-confof-edas	38	29	4	29	6	3
XMapSiG1_4-confof-ekaw	20	0	0	0	0	0
XMapSiG1_4-confof-iasted	10	0	2	0	2	0
XMapSiG1_4-confof-sigkdd	8	0	2	0	2	0
XMapSiG1_4-edas-ekaw	24	0	0	0	0	0
XMapSiG1_4-edas-iasted	14	0	2	0	2	0
XMapSiG1_4-edas-sigkdd	12	0	2	0	2	0
XMapSiG1_4-ekaw-iasted	10	0	0	0	0	0
XMapSiG1_4-ekaw-sigkdd	12	0	0	0	0	0
XMapSiG1_4-iasted-sigkdd	30	5	2	5	4	0
YAM++-cmt-conference	24	0	0	0	0	0
YAM++-cmt-confof	26	0	0	0	0	0
YAM++-cmt-edas	34	0	0	0	0	0
YAM++-cmt-ekaw	16	0	0	0	0	0
YAM++-cmt-iasted	10	0	0	0	0	0
YAM++-cmt-sigkdd	24	3	0	3	3	0
YAM++-conference-confof	26	0	0	0	0	0
YAM++-conference-edas	34	0	0	0	0	0
YAM++-conference-ekaw	54	0	0	0	0	0
YAM++-conference-iasted	12	0	0	0	0	0
YAM++-conference-sigkdd	24	0	0	0	0	0
YAM++-confof-edas	30	0	0	0	0	0
YAM++-confof-ekaw	34	0	0	0	0	0
YAM++-confof-iasted	16	0	0	0	0	0
YAM++-confof-sigkdd	14	0	0	0	0	0
YAM++-edas-ekaw	34	0	0	0	0	0
YAM++-edas-iasted	24	0	0	0	0	0
YAM++-edas-sigkdd	16	0	0	0	0	0
YAM++-ekaw-iasted	18	0	0	0	0	0
YAM++-ekaw-sigkdd	20	0	0	0	0	0
YAM++-iasted-sigkdd	36	0	0	0	0	0