A Unified Approach for Dealing with Ontology Mappings and their Defects

Muhammad Aun Abbas

THESE / Université de Bretagne-Sud

sous le sceau de L'université Bretagne Loire

pour obtenir le titre de :
DOCTEUR DE L'UNIVERSITE DE BRETAGNE-SUD
Mention:
Sciences et Technologies de l'Information et de la Communication
École Doctorale : SICMA

présentée par
Muhammad Aun ABBAS
Préparée à l'unité mixte de recherche
(n°) 6074
Etablissement de rattachement
Universite de Bretagne-Sud
Nom développé de l'unité
Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA)

A Unified Approach for Dealing with Ontology Mappings and their Defects

Thèse soutenue le 14-12-2016

devant le jury composé de :

Nacer Boudjlida Professeur, Université de Lorraine (Rapporteur)

Michaël PETIT Professeur, Université de Namur (Rapporteur)

Mounira Harzallah Maitre de Conferences, Université de Nantes (Examinatrice)

Pierre-François Marteau Professeur, Université de Bretagne Sud (Examinateur)

Giuseppe Berio Professeur, Université de Bretagne Sud *(Directeur de Thèse)*

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

Dedication

I dedicate my thesis work to my parents and family.

Acknowledgements

I am thankful to Almighty Allah, who bestows me the abilities and capabilities to complete this thesis. While I alone am responsible for this thesis as it is my own work, it is nonetheless at least a product of an inspiration, motivation, support, help, and guidance of many others and the blessings of Almighty Allah. For this reason, I express my gratitude to all those who help me in any way either by comments, questions, criticize or support during this thesis work. Regrettably, but inevitably, the following list of names will be incomplete, and I hope that those who are missing will forgive me, and will still accept my sincere appreciation of their influence on my work.

My deepest gratitude to Professor Giuseppe Berio, my thesis supervisor, for his supervision and support throughout all my years in the university. I am indebted to him, his continued guidance, feedback and comments have always been helpful in improving and completing my thesis. He is indeed the mentor. It has been a pleasure for me to complete this thesis under his supervision and guidance.

I am thankful to Nacer Boudjlida and Michaël PETIT for accepting the request of becoming my PhD thesis reviewers.

I am thankful to Dr. Pnina Soffer, Dr. Manfred Reichert, Dr. Sandeep Purao and Dr. Barbara Weber. They provided valuable comments and advice on my thesis work. Their comments and advice have helped me to complete this thesis.

I am thankful to my colleagues at LabSTICC and IRISA labs. I spent good time with Ayle Dahir at IUT. We discuss our works along with discussing extra-curricular things. His encouragement and support are always with me during this thesis work.

I am thankful to my friends Aamir, Amanullah yasin, Abdul haq, Hassan, Kamran Jaffri, Quaid Zaman, Usman who helped me in difficult circumstances during my stay in France. Beside them, it was pleasure to spend some time with Naeem shehzad, Adeel, Zeeshan, Fasih, Kamran Jaffri and Zahoor.

I am indebted to Mr. Irfan Haider who helped me in fulfilling some official formalities.

I thank my friends Murtaza Sabir, Samiullah Khan, Kamran Javed, Mubashir Sohail, Awais Khan, Asad Aslam, Arif Sattar, Shozeb Sharif, Asif Akbar, Imran Ahmed, Muhammad Tariq, Saif Islam and others.

I am grateful to my friend Iftikhar Haider who supported me in some critical juncture of my life. I cherished the memories of all my friends.

I am thankful to the Government of Islamic Republic of Pakistan for providing me the opportunity in getting admission in French Universities and providing the necessary funding.

I am also thankful to Madam Somia Ashraf and Anna Maneta of SFERE. They help me in some administrative works.

I am thankful to LabSTICC and IRISA for hosting me as a researcher during my thesis.

I am thankful to all those who helped me in any manner to complete my thesis.

I am also thankful to my wife Samna Robab and my son. She always encourages me to complete this thesis. My son, Muhammad Ali Abbas who always wants to play with me. His arrival and presence during my thesis work is another source of joy and happiness during this work.

I miss my father Muhammad Yaqoob on this occasion. He always supported and guided me. I felt sad that he is not in this world. I am indebted to my parents who are always my source of inspiration and motivation. My mother Jamila Bibi always prays for my success and my return. I am thankful for my brothers (Muhammad Azhar Abbas, Muhammad Baqir Abbas, Muhammad Tauqir Abbas) and sisters (Tehmina Robab, Samina Robab), and other family members for their support during this thesis work.

Abstract

An ontology mapping is a set of correspondences. Each correspondence relates artifacts, such as concepts and properties, of one ontology to artifacts of another ontology. In the last few years, a lot of attention has been paid to establish mappings between source ontologies. Ontology mapping is widely and effectively used for interoperability and integration tasks (data transformation, query answering, or web-service composition, to name a few), and in the creation of new ontologies.

On the one side, checking the (logical) correctness of ontology mappings has become a fundamental prerequisite of their use. On the other side, given two ontologies, there are several ontology mappings between them that can be obtained by using different ontology matching methods or just stated manually. Using ontology mappings between two ontologies in combination within a single application or for synthesizing one mapping taking the advantage of two original mappings, may cause errors in the application or in the synthesized mapping because those original mappings may be contradictory (conflicting).

In both situations, correctness is usually formalized and verified in the context of fully formalized ontologies (e.g. in logics), even if some "weak" notions of correctness have been proposed when ontologies are informally represented or represented in formalisms preventing a formalization of correctness (such as UML). Verifying correctness is usually performed within one single formalism, requiring on the one side that ontologies need to be represented in this unique formalism and, on the other side, a formal representation of mapping is provided, equipped with notions related to correctness (such as consistency).

In practice, there exist several heterogeneous formalisms for expressing ontologies, ranging from informal (text, UML and others) to formal (logical and algebraic). This implies that, willing to apply existing approaches, heterogeneous ontologies should be translated (or just transformed if, the original ontology is informally represented or when full translation, keeping equivalence, is not possible) in one common formalism, mappings need each time to be reformulated, and then correctness can be established. This is possible but possibly leading to correct mappings under one translation and incorrect mapping under another translation. Indeed, correctness (e.g. consistency) depends on the underlying employed formalism in which ontologies and mappings are expressed. Different interpretations of correctness are available within the formal or even informal approaches questioning about what correctness is indeed.

In the dissertation, correctness has been reformulated in the context of heterogeneous ontologies by using the theory of Galois connections. Specifically ontologies are represented as lattices and mappings as functions between those lattices. Lattices are natural structures for directly representing ontologies, without changing the original formalisms in which ontologies are expressed. As a consequence, the (unified) notion of correctness has been reformulated by using Galois connection condition, leading to the new notion of compatible and incompatible mappings.

It is formally shown that the new notion covers the reviewed correctness notions, provided in distinct state of the art formalisms, and, at the same time, can naturally cover heterogeneous ontologies.

The usage of the proposed unified approach is demonstrated by applying it to upper ontology mappings. Notion of compatible and incompatible ontology mappings is also applied on domain ontologies to highlight that incompatible ontology mappings give incorrect results when used for ontology merging.

Une Approche Unifiée de Traitement de "Mappings" d'Ontologies et de leurs Défauts

Résumé

Un mapping d'ontologies est un ensemble de correspondances. Chaque correspondance relie des artefacts, typiquement concepts et propriétés, d'une ontologie avec ceux d'une autre ontologie. Le mapping entre ontologies a suscité beaucoup d'intérêt durant ces dernières années. En effet, le mapping d'ontologies est largement utilisé pour mettre en œuvre de l'interopérabilité et intégration (transformation de données, réponse à la requête, composition de web service) dans les applications, et également dans la création de nouvelles ontologies.

D'une part, vérifier l'exactitude (logique) d'un mapping est devenu un prérequis fondamentale à son utilisation. D'autre part, pour deux ontologies données, plusieurs mappings peuvent être établis, obtenus par différentes méthodes d'alignement, ou définis manuellement. L'utilisation de plusieurs mappings entre deux ontologies dans une seule application ou pour synthétiser un seul mapping tirant profit de ces plusieurs mappings, peut générer des erreurs dans l'application ou dans le mapping synthétisé car ces plusieurs mappings peuvent être contradictoires.

Dans les deux situations décrites ci-dessus, l'exactitude, la non-contradiction et autres propriétés sont généralement exprimées de façon formelle et vérifiées dans le contexte des ontologies formelles (par exemple, lorsque les ontologies sont représentées en logique) La vérification de ces propriétés est généralement effectuée à l'aide d'un seul formalisme, exigeant d'une part que les ontologies soient représentées par ce seul formalisme et, d'autre part, qu'une représentation formelle des mappings soit fournie, complétée par des notions formalisant les propriétés recherchées.

Cependant, il existe une multitude de formalismes hétérogènes pour exprimer les ontologies, allant des plus informels (par exemple, du texte contrôlé, des modèles en UML) aux formels (par exemple, des logiques de description ou des catégories). Ceci implique que pour appliquer les approches existantes, les ontologies hétérogènes doivent être traduites (ou juste transformées, si l'ontologie source est exprimée de façon informelle ou si la traduction complète pour maintenir l'équivalence n'est pas possible) dans un seul formalisme commun et les mappings sont reformulés à chaque fois : seulement à l'issu de ce processus, les propriétés recherchées peuvent être établies. Même si cela est possible, ce processus peut produire à la fois des mappings corrects et incorrects vis-à-vis de ces propriétés, en fonction de la traduction (transformation) opérée. En effet, les propriétés recherchées dépendent du formalisme employé pour exprimer les ontologies et les mappings.

Dans cette dissertation, des différentes propriétés ont été a été reformulées d'une manière unifiée dans le contexte d'ontologies hétérogènes utilisant la théorie de Galois. Dans ce contexte, les ontologies sont représentées comme treillis, et les mappings sont reformulés

comme fonctions entre ces treillis. Les treillis sont des structures naturelles pour la représentation directe d'ontologies sans obligation de traduire ou transformer les formalismes dans lesquels les ontologies sont exprimées à l'origine.

Cette reformulation unifiée a permis d'introduire une nouvelle notion de mappings compatibles et incompatibles. Il est ensuite formellement démontré que cette nouvelle notion couvre plusieurs parmi les propriétés recherchées de mappings, mentionnées dans l'état de l'art.

L'utilisation directe de mappings compatibles et incompatibles est démontrée par l'application à des mappings d'ontologies de haut niveau. La notion de mappings compatibles et incompatibles est aussi appliquée sur des ontologies de domaine, mettant en évidence comment les mappings incompatibles génèrent des résultats incorrects pour la fusion d'ontologies.

Contents

List of Tables

List of Figures

Chapter 1.

Introduction

This thesis contributes to the unified approach for dealing with ontology mappings and their defects for the area of using ontology mappings collectively in a single application. This chapter presents the background of the thesis (Section 1.1) and motivates the relevance of the work reported here (Section 1.2). It also defines the main objectives of our research (Section 1.3) and its scope (Section 1.4). The chapter concludes with the presentation of the approach that we follow to accomplish these objectives along with an overview of the thesis structure (Section 1.5).

1.1 Background

In 21st century, research is focused in providing information in an easy and understandable way to people. To meet the requirements of people, various information systems are available and development of new information systems are on the rise. The basic component of these information systems is data. This data is processed for providing useful information. People are interested in getting useful information, mainly for decision-making purposes. World Wide Web is a masterpiece that has changed and improved our social, political and economic nature of activities.

Now researchers are trying to replace syntactical World Wide Web with semantic web. The main goal of the semantic web is to provide semantic information, while researchers are still in pursuit of this goal. One of the main challenges of the semantic web are heterogeneity of information that is often similar. Heterogeneity in data and information is due to the use of languages having different syntax and semantics and by inherent natural language problems such as homonyms, heteronyms and vagueness. Therefore, it is required to use a clear and understandable representation of data and information.

Ontologies are used to represent a clear and unambiguous information. The term ontology is a philosophical term, but it is now widely used in computer science with slightly different meaning. In this work, we treat the term 'ontology' in the context of computer science and not in a philosophical context.

In the field of computer science, the definition of ontology evolves, but Thomas Gruber's definition of ontology is mostly cited in literature.

Definition 1-1 (Ontology): (Gruber, 1993): An Ontology is an explicit representation of a conceptualization

This definition is further extended and refined in the following definition by Studer and colleagues as

Definition 1-2 (Ontology): (Studer et al., 1998) A formal, explicit specification of a shared conceptualization.

A 'conceptualization' refers to an abstract model of some concepts in the real world which user wants to represent. 'Explicit' means that the concepts, their relations and constraints are explicitly defined. 'Formal' refers to the fact that the ontology is able to be read and processed by machine and it has a clear and well-founded semantics. 'Shared' reflects the notion that a group accepts the conceptualization of the ontology. Hence, ontologies represent information in a clear and unambiguous way that is accepted by a group of individuals where ontology will be used.

Ontologies are classified into domain and core/upper ontologies. Ontologies are generally built for specific domain (only some part of the world) and are often called domain ontologies. Ideally, there should be single domain ontology for each domain, but there exist several domain ontologies for a single domain. The reason, in part, is that the 'degree of preciseness' varies in defining conceptualization because of the specific needs of a group and creator of ontology does not have complete knowledge about the domain.

Upper ontologies and Core ontologies are often distinguished from domain ontologies. A Core ontology defines the precise conceptualization of world by defining minimal ontology that is used for defining the conceptualization of a particular domain. Upper ontology is more generic and it defines conceptualization of the world that is applicable to several domains. However, the problem of having a single ontology for each domain is not solved even by using core ontologies or upper ontologies. The reason is the same as in the case of domain ontologies, the varied degree of preciseness of conceptualization of the world, and this reason results into several core and upper ontologies. Some of the upper ontologies are Cyc (Lenat, 1995), BWW (Wand & Weber, 1990), BFO (Grenon, 2003), DOLCE (Masolo et al., 2003), and UFO (Guizzardi & Wagner, 2004). Even though upper ontologies are created, for the purpose, that their conceptualizations will be applicable to several domains, but their conceptualizations are based on particular theories. For instance, DOLCE's common sense and BFO naïve realism are in the same spirit but they have cognitive bias; DOLCE allows distinction between abstract and concrete entities but BFO does not make this distinction and is aligned towards realism. The difference in the theoretical choices of upper ontologies leads to a different conceptualization of the same topic.

An ontology, in general, is not a mere taxonomy; instead, it is mostly based on rigorous logical theories that is often used for reasoning purposes. According to Gruber (Gruber, 1993) (Gruber, 1995), main components of an ontology are:

a) Concepts, which represent sets of objects with common properties within the domain of interest;

b) Relations, which represent relationships among concepts by means of the notion of mathematical relation;

c) Functions, which are functional relations;

d) Axioms, which are sentences that are always true and are used in general to add more information about properties of classes, relations, and individuals;

e) Instances, which are individual objects in the domain of interest.

Nowadays, ontologies are widely used in various fields. Deborah L. McGuinness lists the uses of ontologies (McGuinness, 2003). Some of these uses are

- Provide controlled vocabulary, users use the same set of terms defined in the ontology for the same concepts;
- Sense disambiguation support, ontology provides clear semantics of the terms and users can understand that the same term having different conceptualization and they will use the terms accordingly;
- Search support and completion of terms; a query expansion method is used to expand the search query of the users from the most specific categories in a hierarchy;
- Consistency checking, ontology is treated as logical theories and reasoning services are used to check consistency in ontology.
- Interoperability support, ontology provides interoperability support as the two applications based on same ontology and so they are using the same sets of terms. While if applications are based on different ontologies, interoperability among applications is achieved by using ontology mappings (we will explain ontology mapping later in this chapter).
- Support validation and verification of testing of data (and schema), by using the axioms of ontology for constraining the interpretation of terms and relations, one can validate and verify the data and schema.
- Exploit generalization/specialization information, ontologies exploit generalization/specialization information of ontology for constraining the search result, query expansion and consistency checking.

Ontologies can be represented in various languages ranging from less formal to more formal. However, the choice of language influences the semantics of ontology. For instance, some ontology languages only allow single model, i.e., these languages allow exact interpretation about the domain, while some other languages allow multiple models, i.e., these languages allow incomplete information about the domain. Examples of languages that allow single models are entity-relationship model (Chen, 1976), UML (OMG, 2016), OKBC (Chaudhri et al., 1998), XML (W3C, 2016), and Semantic networks (Sowa, 2014); while examples of languages allowing multiple models are Ontolingua (Farquhar et al., 1997), KIF (Genesereth & Fikes, 1992), PSL (Schlenoff et al., 2000), RDF(S), DL (Baader & Nutt, 2003), and OWL (W3C, 2016). Ontology language also varies in terms of expressivity. For instance, UML can represent part-whole and is-a relationship while OWL can represent is-a relationship. In this thesis, we have often represented is-a relation of ontology artifacts by UML generalization link symbol.

Ontology O_B is an ontology representing different roles working in research organization. O_B have concepts 'Research Org.' which represent the research organization, 'Research Staff' representing the information about research staff in the organization, 'Computer Scientist' and 'Social Scientist' represents the information about research staff associated to computer science and social science research activities, 'Research Officer' represents staff who assists research staff. There is also an administrative staff who manages the administration of the research organization which are 'Admin. Staff' and 'Director Admin'. Some of the relations of this ontology are $work\_with$ and $work\_on$. Relation $work\_with$ is used to represent two or more research staff working together, while relation $work\_on$ is used to represent research staff working on some project. A fragment of this ontology is presented in Figure 1-1.

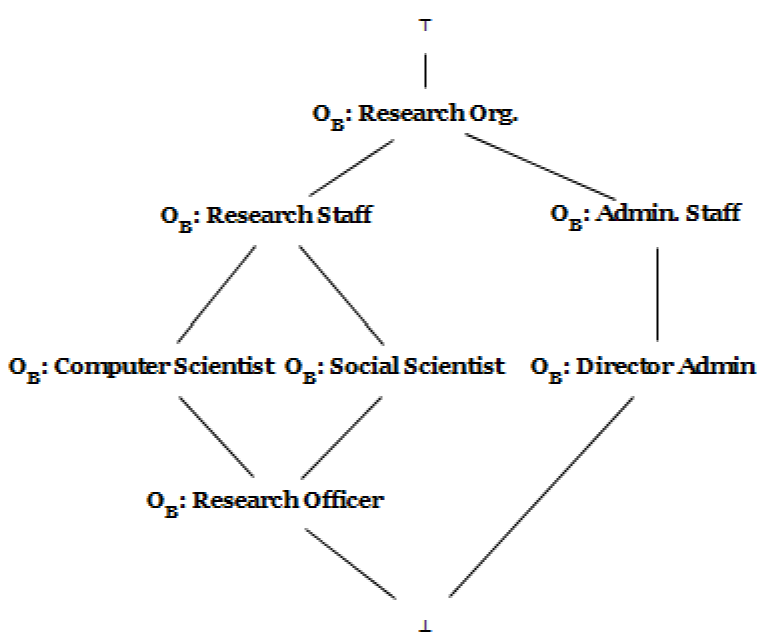

Figure 1-1. Fragment of Ontology of Research Organization

The backbone of an ontology consists of generalization and specialization hierarchy of concepts and relations. In the ontology of Research organization, Research staff is the super concept of Computer Scientist and Social Scientist. Ontologies have special concepts 'top' $\top$ is the most generic concept in the ontology and 'bottom' $\perp$ is the most specialized concept of the ontology. Similarly, there are 'top' and 'bottom' relations in the ontology. In Figure 1-1, the prefix 'O_B:' for each term in the ontology refers to the name of the ontology.

In open or evolving systems, such as semantic web, different parties often use different ontologies of the same domain based on their preferences, availability, and functional requirements. Additionally, there does not exist a single ontology which can be used in every application. Thus, the issue of describing and using clear and understandable information is not solved by using ontologies rather it raises heterogeneity to a higher level (Euzenat & Shvaiko, 2007). In this open and evolving environment, information systems are usually distributed in nature. In distributed systems, there are many situations when one system has to interact with other system. Applications using different ontologies might want to communicate, users want to integrate data that are structured on the basis of ontologies are examples of scenarios where two distributed systems communicate with each other. There is a necessary and essential step of finding or relating artifacts of involved ontologies, this step is called ontology matching. Artifacts of ontologies are concepts, relations, and instances. The relation between artifacts of two ontologies is used as a basis for interaction between two ontology based systems. We present here some of the key definitions related to ontology matching and they are listed below.

Definition 1-3 (Ontology correspondence) (Euzenat & Shvaiko, 2007): An ontology correspondence is the relation holding or supposed to hold between artifacts of different ontologies. Relations can be equivalence, subclass, superclass relationships, or transformation rules between artifacts of source ontologies.

Definition 1-4 (Ontology mapping): (Noy, 2009): An ontology mapping is a set of correspondences between artifacts of two ontologies.

We consider that artifacts of the ontologies are not only the axiomatic part of the ontology but they can represent intermediate concepts or properties that are not the part of the original ontologies. For Instance, concept A, B and C are part of the orginal ontology but $A \sqcap B \sqcup C$ is not part of the original ontology and it is an intermediate part. In this thesis, Correspondence represents both axiomatic part and intermediate part of the ontology.

We will discuss different kinds of semantics of ontology mappings in Chapter 2. Some authors used the terms of 'ontology matching' and 'ontology alignment' for 'ontology mapping'. We will use the term of 'ontology mapping' in our work.

Definition 1-5 (Ontology matching): (Noy, 2009): A process of finding ontology mapping is often referred to as ontology matching.

To illustrate the need of ontology mappings, we present here an example.

Example 1-1: Suppose that there are two ontologies: first ontology belongs to the university domain while second ontology belongs to a research organization domain. Figure 1-1 and Figure 1-2 present small portion of these ontologies.

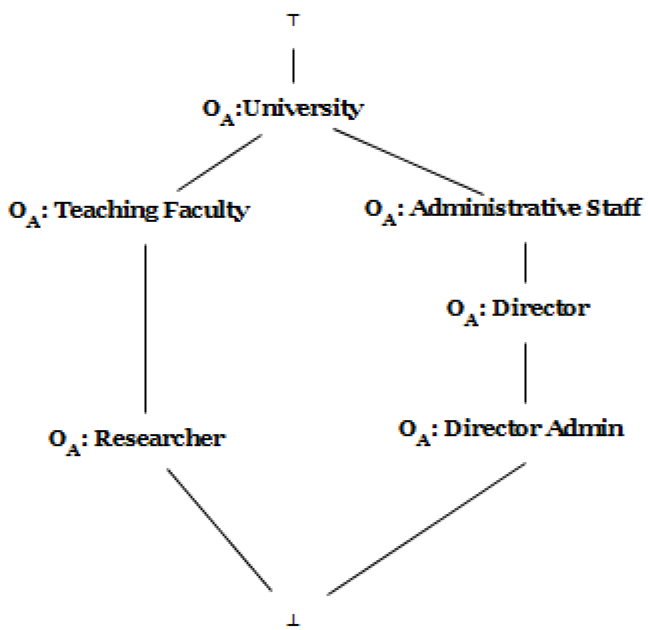

Figure 1-2. Fragment of Ontology of University domain

Ontology belonging to the University has concepts 'University' which represents the University, 'Teaching Faculty' which represents the information concerning Teaching Faculty and it has a sub concept 'Researcher' which represents the teaching faculty involve in research related activities. Apart from Teaching faculty in the university, administrative staff manages the administration that is represented by 'Administrative Staff'. The administrative staff has a sub concept 'Director' and 'Director' has a sub concept 'Director Admin'.

These ontologies have some overlapping information such as research activity is performed in both domains, while they have some different or not similar information such as in 'university' emphasis on teaching while in 'research organization' emphasis is on research. These ontologies may be related/mapped because of several reasons such as collaboration between research organization and university. In a scenario, where persons of two organizations may share resources of two domains or they may switch their roles. Hence, there is a need to know what are the similarities or correspondences between the two ontologies. Correspondences specify the relationships between artifacts of two ontologies. It is not necessary that all the artifacts of two ontologies be related to each other, as there may be some artifacts in both ontologies that are not related to each other.

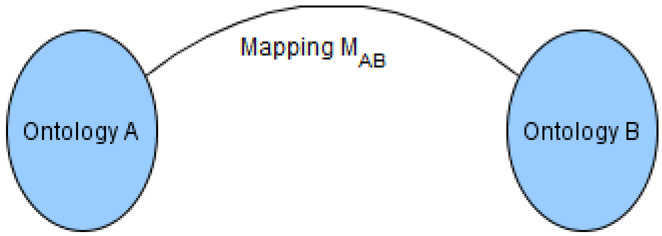

Figure 1-3. Ontologies and their mappings

Ontologies A and B and their mappings M_{AB} are abstracted in Figure 1-3. When artifacts of two ontologies are not completely mapped to each other, then such a mapping is called *partial mapping*. When there is a mapping M_{AB} that is established only from either A to B or B to A, then such a mapping is called *directional mapping*. In most of the cases, the mapping is *bidirectional*. The mapping M_{AB} and M_{BA} are not required to be *symmetric,* i.e., mappings between ontology A to B and B to A can be different. When the artifacts of one ontology are mapped by relating the artifacts of one ontology to another by some role, then such a mapping is called '*link' or 'connection'*.

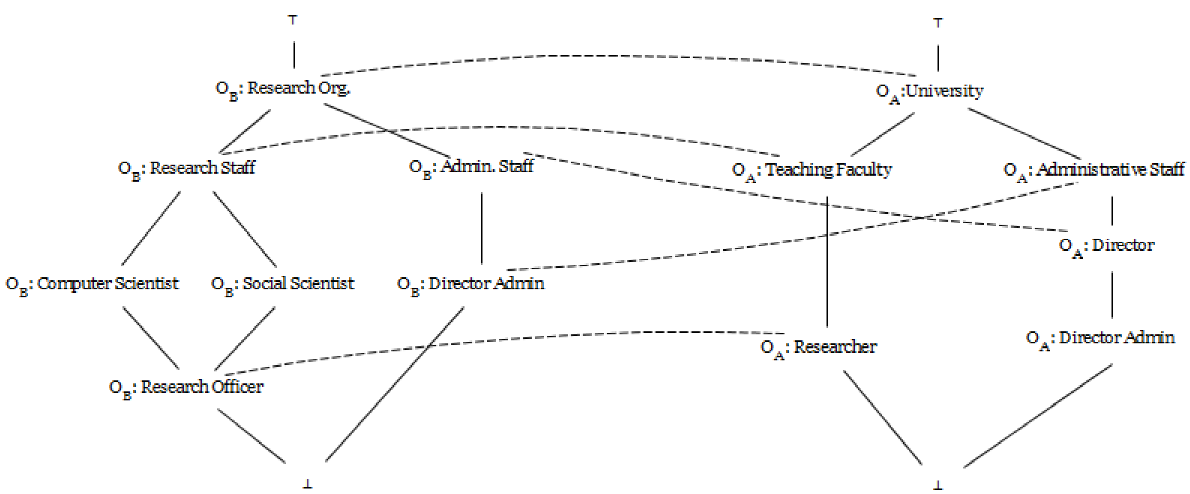

Figure 1-4. Ontologies and their Mappings

Mappings between two ontologies are shown in Figure 1-4 by dashed lines. 'Research Org.' is similar to 'University' in terms of research performed in these institutions. 'Research Staff' and 'Teaching Faculty' is similar since teaching faculty can perform research activities as performed by research staff. 'Research Officer' and 'Researcher' are similar in terms of label similarity. 'Director Admin' and 'Administrative Staff' are similar and 'Admin. Staff' and 'Director' are similar since their roles are same in their organizations, but they have different names in the two ontologies.

Euzenat and colleagues show extensive range of applications in which ontology mapping plays a key role (Euzenat & Shvaiko, 2007). Some of these are ontology evolution, Data integration, P2P information sharing, Web service composition, Autonomous communication systems, and Query answering.

Ontology mappings and ontology axioms are implication in their nature. One can infer implicit information from ontology axioms and ontology mappings. On the one hand, this feature is very useful for extracting implicit information, but on the other hand, this feature is error prone. If there is some inconsistency in the ontology, it can make whole ontology inconsistent. Since in most of the cases ontologies are represented in Description logics or languages founded on description logics such as OWL (Web Ontology Language) and it is known that Description logics obey the principle of explosion (ex falso quodlibet), so inconsistencies propagates to whole ontology. Nowadays, research community focus on identifying the cause of the inconsistency in ontology and ontology mapping. In this thesis, our focus is not on identifying inconsistencies in ontology mappings.

One of the basic components for integration of systems and even in the interaction between systems is an ontology mapping. Generally, the size of ontologies is big, so it is very difficult and non-trivial for human experts to discover ontology mappings between ontologies. There exist numerous semi-automated approaches for discovering ontology mappings between ontologies. These approaches find similarities between ontologies based on the structure and/or terminology of the ontologies and the background knowledge. Upper ontologies are also used as an intermediate or background source for discovering ontology mapping as it is recognized in (Gehlert & Esswein, 2007). Even though upper ontologies does not solve the issue of precise conceptualization, but they are still built by using well-founded theories. Thus, upper ontologies are still worthy enough to be used in applications, Mascardi and colleagues present various situations where role of upper ontology is very effective (Mascardi et al., 2007). The other approaches of discovering ontology mapping mainly use some heuristics, machine learning and graph algorithms.

Ontology mappings can be expressed as the language in which ontologies are expressed (e.g., using OWL (Hitzler et al., 2009)); 'bridge rules' which are not part of the ontology (Bouquet et al., 2003), (Dou et al., 2005); as 'bridge ontology' (Maedche et al., 2002), (Crubézy & Musen, 2004); as views to describe mapping between global ontology and local ontology (Calvanese et al., 2002).

In this thesis, we are interested in studying the problem of combined use of ontology mappings, as opposed to, for instance, devising or ameliorating a method for discovering ontology mapping between ontologies. We want to identify ontology mappings that contradict each other and we call such mappings as "incompatible mappings". Incompatible mappings result in logical errors and/or unexpected results when they are used in combination. Some definitions concerning absolute defects that will be frequently used in this thesis are described below.

Definition 1-6 (Inconsistency): When a theory has formulas β and its negation $\neg\beta$, i.e., a formula and its negation both are true, then such theory is inconsistent. One can infer anything (α any formula) from such theories $\beta \rightarrow \neg\beta \rightarrow \alpha$.

In this work, we treat ontologies as logical theories, and inconsistency as logical inconsistency according to above definition.

Definition 1-7 (Unsatisfiability): If σ is a concept in the ontology $O \vDash \sigma$ and there is no model (object) in the ontology that satisfy this axiom.

A concept in an ontology is unsatisfiable if it cannot be instantiated without causing inconsistency in the ontology.

Definition 1-8 (Incoherent Ontology): An ontology is Incoherent if it contains at least one unsatisfiable concept.

The concept of incoherence and inconsistency are related to each other, but they are different concepts (Haase & Qi., 2007). If an incoherent ontology is not inconsistent, one can easily make this ontology as inconsistent; just by adding an assertion of instantiating an unsatisfiable concept of the incoherent ontology (Flouris et al., 2006).

There are ontologies which are incoherent but consistent. For instance, $O = \{C \sqsubseteq D, C \sqsubseteq \neg D\}$, here C is unsatisfiable, so the ontology O is incoherent but consistent. There are ontologies which are coherent but inconsistent, for instance, $O = \{a = b, a \neq b\}$. There are ontologies which are incoherent and inconsistent, for instance, $O = \{C, C(a), C(b), a = b, a \neq b\}$, This ontology does not have any model even empty set $\emptyset$ because there is no model that satisfy $a = b, a \neq b$.

We present, here, an example to highlight that when ontology mappings used in combination cause not only logical errors but may also give unwanted results.

Example 1-2: Suppose that there are two ontologies, ontology A deals with Falcon's diet and ontology B deals with fruit and animals. In Figure 1-5 fragments of two ontologies are shown along with two ontology mappings (shown here for few artifacts).

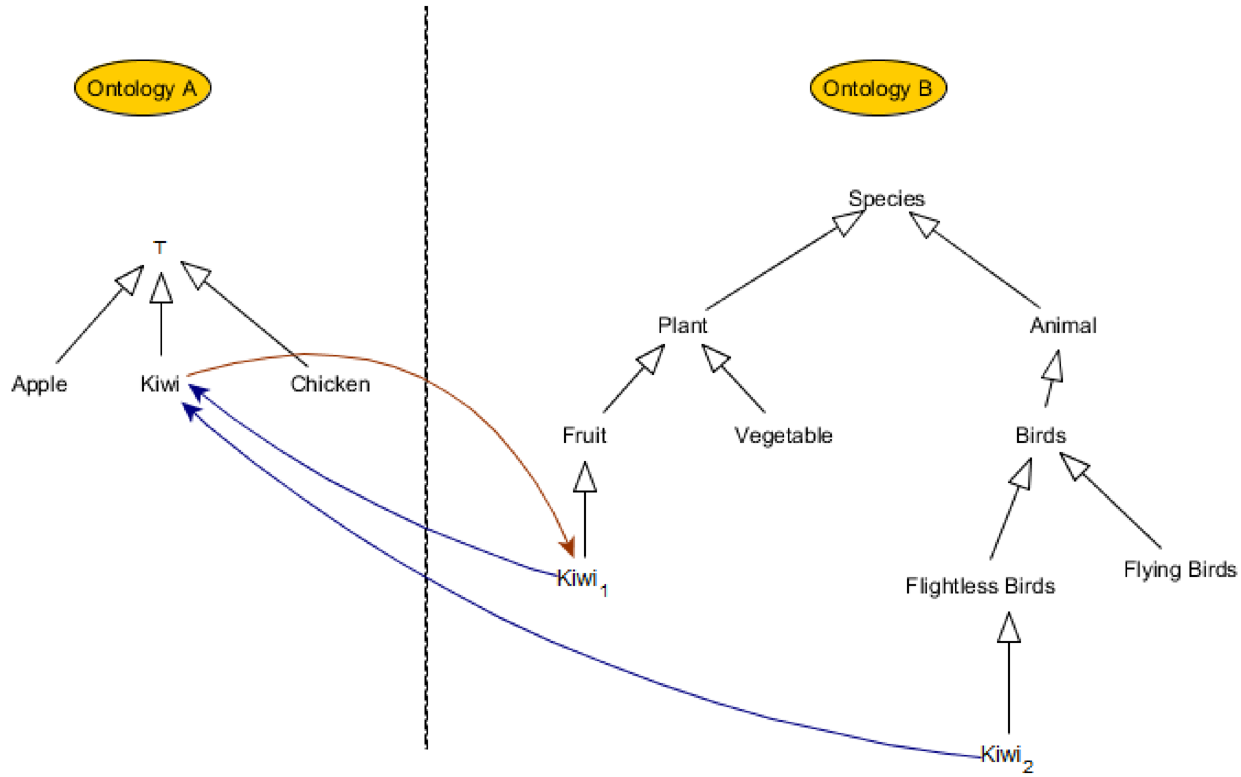

Figure 1-5. Example of two Conflicting mappings

Mappings used in the example are obtained by using two different matching methods. Mapping from ontology A to ontology B is established on lexical similarity basis, while mapping from ontology B to ontology A is established on structural similarity basis. Few ontology correspondences of the two ontology mappings is shown here, there also exists other ontology correspondences in both ontology mappings.

Here, two cases of ontology mappings based on their relations are presented. Mappings are formalized as $(O_A: a, r, O_B: b)$, where a is artifact of ontology O_A and b is artifact of ontology O_B and r is the relation between artifacts a and b. Relation r is defined here in two ways.

Firstly, when both mappings are established based on equivalence ≡ relation.

$$M_A = \{O_A: Kiwi \equiv, O_B: Kiwi_1\}$$

$$M_B = \{O_B: Kiwi_1 \equiv O_A: Kiwi;\ O_B: Kiwi_2 \equiv O_A: Kiwi\}$$

Secondly, when both mappings are established on ⊒ and ⊑ relations.

$$M'_A = \{O_A: Kiwi, \sqsupseteq, O_B: Kiwi_1\}$$

$$M'_B = \{O_B: Kiwi_2, \sqsubseteq, O_A: Kiwi; O_B: Kiwi_1, \sqsubseteq, O_A: Kiwi\}$$

$$M''_B = \{O_B: Kiwi_1 \sqsupseteq O_A: Kiwi;\ O_B: Kiwi_2 \sqsupseteq O_A: Kiwi\}$$

When mappings M_A and M_B are used in combination, one ontology mapping M_A describes that a fruit '$Kiwi$' in ontology A is equivalent to a fruit in '$Kiwi$' in ontology B by considering $'Kiwi'$ of ontology A as a fruit. While other mapping M_B represents that a bird '$Kiwi$' in ontology A is equivalent to a bird '$Kiwi$' in ontology B by considering that $'Kiwi'$ is a bird in ontology A and also a bird '$Kiwi$' in ontology A is equivalent to a fruit '$Kiwi$' in ontology B. Where fruit '$Kiwi$' and bird '$Kiwi$' in ontology B are not same concept and assume they are disjoint. But these mappings make fruit $'Kiwi'$ and bird $'Kiwi'$ of ontology B as equivalent. This shows that these ontology mappings are contradicting each other $\{O_A{:}Kiwi, \equiv, O_B{:}Kiwi_1;\ O_A{:}Kiwi, \equiv, O_B{:}Kiwi_2;\ O_B{:}Kiwi_1, \perp, O_B{:}Kiwi_2\}$. It may be the case that both mappings are correct but their combined use result into an absolute defect. However, when ontology O_B omitted or may not have the axiom of disjointness then this defect will not occur. This is often the case as ontologies are not well-formalized i.e. not all axioms are given in ontologies.

While when mappings M_A' and M_B' or M'_A and M''_B are used in combination, then there will not be any absolute defect. However, these mappings cause some other issues (named here as relative defects) that are considered as defect in some (and not all) situations. We, here, use (Guarino, 1999) to classify these situations.

In case of combination of mappings O_A, O_B, M_A' and M_B'.

$\{O_A, O_B, \langle O_A{:}Kiwi, \sqsupseteq, O_B{:}Kiwi_1\rangle_{M_A'};\ \langle O_B{:}Kiwi_1, \sqsubseteq, O_A{:}Kiwi\rangle_{M_B'}, \langle O_B{:}Kiwi_2, \sqsubseteq, O_A{:}Kiwi\rangle_{\mathrm{M_B}'};\ O_B{:}Kiwi_1 \perp O_B{:}Kiwi_2\}$.

Mapping M'_A complements one of the correspondence of M'_B. This combination results in a situation of 'reduction of senses' to that extent where an artifact $O_A{:}Kiwi$ that is more than $O_B{:}Kiwi_1$ as in the other mappings $O_A{:}Kiwi$ is $O_B{:}Kiwi_1$ and $O_B{:}Kiwi_2$. And this combination M_A' and M_B' with source ontology completely ignores $O_B{:}Kiwi_2$ which is disjoint with $O_B{:}Kiwi_1$. It is the case that an artifact that consists of two disjoint artifacts ($O_A{:}Kiwi$ is $O_B{:}Kiwi_1$ and $O_B{:}Kiwi_2$), while at the same time it reduces to one of the disjoint artifacts ($O_A{:}Kiwi$ that is more than $O_B{:}Kiwi_1$). Guarino describes this situation as ontological misconception and classifies it as 'reduction of senses'.

In case of a combination of mappings O_A, O_B, M_A' and M_B''.

$\{\langle O_A{:}Kiwi, \sqsubseteq, O_B{:}Kiwi_1\rangle_{M_A'};\ \langle O_B{:}Kiwi_1, \sqsupseteq, O_A{:}Kiwi\rangle_{M_B}, \langle O_B{:}Kiwi_2, \sqsupseteq, O_A{:}Kiwi\rangle_{\mathrm{M_B}'};\ O_B{:}Kiwi_1 \perp O_B{:}Kiwi_2\}$.

Guarino classifies this situation as 'confusion of senses' and 'clash of senses'. It is a situation of 'confusion of senses' as different senses of word are collapsed into a single concept, inheriting from different parents as $O_B{:}Kiwi_1$ 'a fruit' and $O_B{:}Kiwi_2$ an animal have become $O_A{:}Kiwi$ in the combination of O_A, O_B, M_A' and M_B''. In a precise conceptualization, two different senses should be represented as two disjoint artifacts. It is also a situation of 'clash of senses' as parent artifacts of O_A: $Kiwi$ are disjoint and have incompatible meanings and they have no common identity criteria in the ontology. Additionally, the situation becomes $O_A: Kiwi \subseteq (O_B: Kiwi_1 \cap O_B: Kiwi_2)$, i.e., $O_A: Kiwi \subseteq \emptyset$, in the combination of O_A, O_B, M'_C when mapping is $M'_C = \{O_A: Kiwi, \sqsubseteq, O_B: Kiwi_2; O_A: Kiwi, \sqsubseteq, O_B: Kiwi_1\}$, which is acceptable in original Distributed Description Logics semantics but is considered as defect in classical logic.

In this thesis, we name mappings that cause either 'absolute' or 'relative' defects when they are used in combination as 'incompatible' mappings.

In summary, we propose a unified approach in this thesis is to check whether mappings are incompatible mappings (contradicting with each other when used together). Therefore two central questions are: How to define compatible and incompatible mappings? How to develop a unified approach for identifying compatible and incompatible mappings is a unified approach? We will answer these questions in this thesis.

1.2 Motivation

Ontology mappings are mainly used in data integration, interoperability of applications and systems and information sharing applications and they play pivotal part in these applications. If ontology mapping contains some defects (incorrect information) then applications using them are not correct and are unreliable.

We will discuss the importance of ontology mappings and also availability of more than one ontology mappings by presenting following scenarios:

1. Peer to peer applications, distributed applications and agent based applications often comprise several autonomous entities without any central point of control and even without any common management procedure: each of these entities may use its own vocabulary, reference schema, ontology and so on to map each message, flow, variables and so on coming from other entities. This is the situation when at run time different applications, systems, agents communicate with each other without prior knowledge about them. There is a need to check that entities involved in communications or interoperability must not get conflicting and contradicting messages. For instance, one system S_1 sends a message 'A' to another system S_2 considering that S_2 will interpret it as A'. But other system interpret it as $\neg A'$. In this situation, the message is in contradicting to the assumption (mapping) of the involved systems.
2. At design-time, more than one mappings can be used to understand distinct perspectives underlying interpretations of ontological artifacts. For instance, to accomplish mapping, one artifact can be mapped on to another concept belonging to another ontology because of label based similarities (and therefore the mapping is quite loose) while another mapping can map one artifact on to another one because their logical equivalence (within some theoretical frameworks) can be established.

Within the former mapping, the various ontological artifacts are interesting because of their labels; within the second mapping, the various ontological artifacts are interesting as logical artifacts (to perform for instance, reasoning).

3. Data integration requires both design and run time communication or interoperability. View based approach (Lenzerini, 2002) are generally used in data integration. In this situation, it is required that mappings used at design time and run time do not contradict each other. Some of the Scenarios in which mappings are used at design time and/or run time are listed in Table 1-1.
4. Demand of matching system increases as ontologies are used in different fields. Combining several mappings is naturally useful for getting benefits of each mapping. Several matching systems such as ASMOV (Jean-Mary et al., 2009), and RiMOM (Wang et al., 2010) combine and aggregate the mappings obtained by using different matching discovery approaches. Matching systems like ASMOV and RiMOM show better result in Ontology Matching Evaluation Initiative (OAEI) (Ferrara et al., 2013). When there are more than one matching systems, they may give different matching results. When these matching results are combined and used with source ontologies, they may cause inconsistencies in ontologies. P. Shvaiko and J. Euzenat describe future challenges for ontology mapping and one of them is finding novel ways for combining ontology matching systems (Shvaiko & Euzenat, 2013).

Table 1-1. Scenarios in which mappings are used at design time and/or run time

Scenarios	**Using mappings at Design time**	**Using mappings at Run time**
Peer to peer applications	No	Yes
Understanding distinct perspectives of ontology mappings	Yes	No
Data integration	Yes	Yes
Ontology matching systems	No	Yes

There can be n mappings between two ontologies as shown in Figure 1-6. These n mappings may not be the same since they are obtained by using different algorithms or even by using same algorithm but by using different parameters and tools. Each mapping may have no contradicting correspondences in it. However, when these mappings are combined as shown in Figure 1-7, some of the correspondences may conflict with other correspondences and causes contradiction. Hence, it is interesting to know which of the correspondences are conflicting with each other and in addition which of the mappings may not be combined due to the conflicting correspondences and if still they are combined, these conflicting correspondences need special treatment otherwise they result in inconsistent results. For dealing with n mappings we propose to check them for contradiction in pair-wise manner. We will show that our approach can easily check compatibility of n ontology mappings by using transitive property and composition operation.

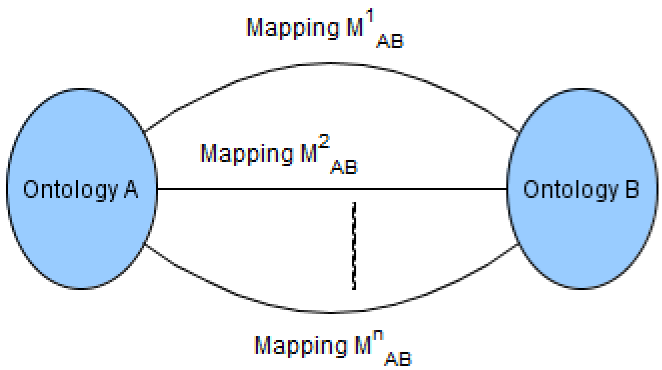

Figure 1-6. n Mappings of two ontologies

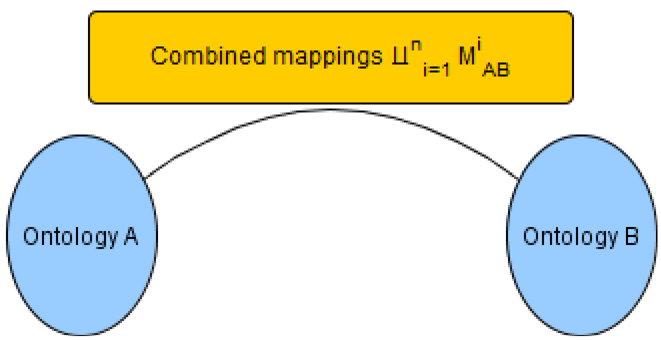

Figure 1-7. Combined Ontology mappings

The task of 'debugging' used for finding and repairing errors in or caused by ontology mappings is a current and important research topic (Shvaiko & Euzenat, 2013), (Meilicke, 2011), (Jiménez-Ruiz et al., 2011). In Chapter 6, we will present that some of the result of this thesis can contribute to the area of debugging ontology mappings. Nonetheless, the focus of our work is not on debugging ontology mappings. Our focus is on identifying conflicting ontology mappings or correspondences for which we have introduced the notion of incompatibility of ontology mappings. By discussing debugging ontology mappings in Chapter 4 , we will show the need of a unified approach to deal with different formalism of ontology mappings.

Since the integration of systems and interoperability between systems are widely used in the distributed systems such as semantic web, the results developed in this thesis contribute to these fields. Nowadays ontology mapping is the main component in distributed systems, our work is applicable to the distributed systems such as multi agent systems, web service integration, and peer to peer communication. In broader terms, the results presented here contribute to computer science in which ontology mapping plays an essential role.

In summary, we defend the position in this thesis that conflicting mappings cause undesired effects in integration and interoperability of the system, and use of other scenarios of using ontology mappings and it is required to identify the conflicting mappings to avoid undesirable consequences. We also defend that a unified approach is required to deal with such a problem having too much heterogeneity; as heterogeneity between formalisms of ontology and ontology mapping, between methods and algorithms of extracting ontology mappings, between formalism of ontology mappings, and mapping between different kinds of artifacts of ontologies (between concept and properties). The need of a unified approach is shown in this work as when one changes the underlying formalism of source ontologies and ontology mappings, defects arise in one formalism may not remain a defect in the new formalism or there may arise new defects which were not earlier present in the previous underlying formalism. Therefore, two central research questions are: How can we identify conflicting mappings? How can we devise a method that is applicable to different mapping languages and ontologies expressed by different formalisms? These questions are answered throughout this thesis.

There exist three major approaches in the context of dealing conflicts in ontology mappings. These are logic-based approaches for debugging ontology mappings, algebraic approaches of ontology merging, and reaching at a consensus on ontology mappings by using argumentation framework. We will discuss these approaches in Chapter 2, Chapter 3 and Chapter 4 respectively.

In logic based debugging approaches, ontology mapping $\mathcal{M}$ is combined with ontologies $\mathcal{O}_1$and $\mathcal{O}_2$ represented in this thesis as $f(0_i, O_j, M)$. This combination $f(0_i, O_j, M)$ are then analyzed whether there is some unsatisfiability of the artifact in any of the ontology or are there any undesired inferences in the ontologies. In the case of comparing two ontology mappings $\mathcal{M}_1$ and $\mathcal{M}_2$, $f(0_i, O_j, M)$ where $\mathcal{M}$ is the union of $\mathcal{M}_1$ and $\mathcal{M}_2$, i.e., $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$.

We will show that the expressiveness and operations of these approaches cannot be considered as adequate for identifying compatible and incompatible ontology mappings. They fall short in providing a unified approach for dealing with ontology mappings and their defects.

In algebraic approaches, ontologies and ontology mappings are treated as algebraic structures. They are generally treated as category theory (Adámek et al., 1990). Ontologies are merged by using ontology mapping by using categorical *push out* and *colimit*. If the colimit is inconsistent, then mappings are treated as erroneous otherwise mappings contain no errors. In case of using two ontology mappings, there are two cases: both mappings are combined and treated as one mapping, treat separately both mappings. In the former case, the problem is that there may exist conflicting correspondences; for instance, if there exists a correspondence $O_i: A \perp O_j: B$ in one mapping, while in another mapping there exists a correspondence $O_i: A \sqsubseteq O_j: B$. When these two mappings are combined then choice would be either to choose one of the relation or consider the mappings as incompatible without using Category theory operations. If the choice would be to choose one of the relation, then we have to treat such ontology mappings separately and we fall in the latter case. In the latter case, we need to check the resultant ontology obtained after colimit operations for each ontology mapping and compare the colimits of all ontology mappings (if exists) to check their compatibility. We discuss algebraic approaches in detail in Chapter 3.

In Argumentation Framework, correspondences of ontology mappings are used for generating arguments on the basis like structure similarity and terminological similarity. Then, when correspondences have such arguments, arguments become *attack* to the arguments that do not have such arguments. Finally, a consensus ontology mapping is achieved by *argumentation* on the basis of preferences on the matching criteria which is conflict free.

The approach presented in this thesis is different from others approaches mentioned above in two ways. First, these approaches treat only a single ontology mapping. Although several ontology mappings can be combined and treated as a single ontology mapping to identify and correct conflicts in ontology mappings, but there are some limitations in doing that. These limitations are mainly redundancy in mappings which will increase complexity in finding conflicting mappings; and no distinction of ontology mappings that remove the idea of existence of two ontology mappings because both are considered as single ontology mapping, whereas we treat ontology mappings as pairwise and they are not combined to make single ontology mapping. After the verification of these pair of ontology mappings and if they do not contain conflicting correspondences, they can be combined.

Second, we take an intermediate approach of treating ontology mapping for checking compatibility. Instead of arriving at consensus or discarding complete ontology mapping, we subscribe to the idea that both ontology mappings may be correct and after verification of conflicts in the ontology mapping, if they contain some conflicts then we termed such mappings as 'incompatible ontology mappings'. Our claim is that these incompatible mappings cannot be used together in a single application. If the ontology mappings do not contradict each other, then such mappings are 'compatible ontology mappings'.

Our proposed method can be applicable to both domain and upper ontology mappings. In Chapter 6, we have shown how our proposed approach can be applied to upper ontology mappings.

There exist several formalisms for representing Ontologies. Generally, formalisms for representing ontology mapping needs more expressiveness. It is not necessary that ontologies and their ontology mappings are in the same formalisms. We will discuss logic-based formalisms used for expressing ontology mapping in Chapter 2. The complexity further increases, since, there exist several formalisms for expressing ontology mappings such as OWL (Hitzler et al., 2009), C-OWL (Bouquet et al., 2003), DDL (Borgida & Serafini, 2003), and $\epsilon - connections$ (Kutz et al., 2004). Hence, there is a need of more general and more expressive formalism that can handle this kind of heterogeneity and expressiveness.

On the one side, checking the (logical) correctness of ontology mappings has become a fundamental prerequisite of their use. On the other side, given two ontologies, there are several ontology mappings between them that can be obtained by using different ontology matching methods or just stated manually. Using ontology mappings between two ontologies in combination within a single application or for synthesizing one mapping taking the advantage of two original mappings, may cause errors in the application or in the synthesized mapping because those original mappings may be contradictory (conflicting).

In both situations, correctness is usually formalized and verified in the context of fully formalized ontologies (e.g. in logics), even is some "weak" notions of correctness have been proposed when ontologies are informally represented or represented in formalisms preventing a formalization of correctness (such as UML). Verifying correctness is usually performed within one single formalism, requiring on the one side that ontologies need to be represented in this unique formalism and, on the other side, a formal representation of mapping is provided, equipped with notions related to correctness (such as consistency).

In practice, there exist several heterogeneous formalisms for expressing ontologies, ranging from informal (text, UML and others) to formal (logical and algebraic). This implies that, willing to apply existing approaches, heterogeneous ontologies should be translated (or just transformed if, the original ontology is informally represented or when full translation, keeping equivalence, is not possible) in one common formalism, mappings need each time to be reformulated, and then correctness can be established. This is possible but possibly leading to correct mappings under one translation and incorrect mapping under another translation. Indeed, correctness (e.g. consistency) depends on the underlying employed formalism in which ontologies and mappings are expressed. Different interpretations of correctness are available within the formal or even informal approaches questioning about what correctness is indeed.

In the dissertation, correctness is reformulated in the context of heterogeneous ontologies by using the theory of Galois connections. Specifically ontologies are represented as lattices and mappings as functions between those lattices. Lattices are natural structures for directly representing ontologies, without changing the original formalisms in which ontologies are expressed. As a consequence, the (unified) notion of correctness is reformulated by using Galois connection condition, leading to the new notion of compatible and incompatible mappings presented in Chapter 5.

1.3 Objectives

In summary the objectives of this thesis are

(O 1) To show the Importance of compatibility and incompatibility of ontology mappings;

(O 2) To define compatibility and incompatibility of ontology mappings;

(O 3) To check whether existing state of the art work provides a solution for proposing unified approach of identifying compatibility and incompatibility of ontology mappings;

(O 4) To develop a unified approach for dealing with ontology mappings and their defects that is applicable to various formalisms of ontology and ontology mappings;

(O 5) To relate notions of compatible and incompatible ontology mappings with mapping correctness;

(O 6) To deal with upper ontology mappings.

1.4 Scope

There may be defects in ontology before the use of ontology mapping. However, we treat and consider ontologies are free from conflicts, and any defect if present in the source ontologies were removed before their use. The scope of our work is to find conflicts in ontology mappings or caused by ontology mappings in source ontologies.

We show that conflicts arising by ontology mappings are problematic to the areas where mappings are used. Although ontology mapping can be used for interoperability and other fields, but in this work we focus on ontology merging. The reason is mere matter of scope. The results shown for ontology merging can also be applicable to interoperability scenarios.

One of the objective of this thesis is to find conflicts in upper ontology mappings. We do not establish new ontology mappings for upper ontologies; instead, we use available upper ontology mappings that exist in literature. The reason is that creating upper ontology mappings is different from domain ontologies where some semi-automated tools help in establishing mappings. In establishing upper ontology mappings, usually mappings are created manually by experts. Using domain ontology matching tools for establishing mappings between upper ontologies may result nothing or establish wrong mappings. The available upper ontology mappings for various upper ontologies are sufficient to be analyzed by our method.

Finally, the target of our work is to propose a unified approach that deals with ontology mappings and their defects. Therefore, computational efficiency and tractability of this approach falls outside the scope of our work.

1.5 Approach and Structure

The structure of this thesis reflects the successive elaboration of the objectives identified in Section 1.4. The approach followed here to accomplish these objectives is detailed in the sequel.

(O1). Objective 1: Importance of identifying compatibility and incompatibility of ontology mappings.

Various scenarios are presented here where the importance of identifying compatibility and incompatibility of ontology mappings is described. We also present an example showing that contradictions which logical approaches do not consider are important enough to be consider as contradiction in case of ontology mapping. This objective is achieved in Chapter 1.

(O2). Objective 2: Define compatibility and incompatibility of ontology mappings.

Objective 1 and 2 will be achieved in Chapter 5 of this thesis. We start by describing the syntax and semantics of ontology mappings in Chapter 2. After briefly describing the representation and interpretation of ontology mappings, we concentrate on the combined use of ontology mappings in a single application. We introduce the notions of compatible and incompatible ontology mappings. In our approach, these notions are used to systematically describe the presence or absence of defects in ontology mappings. We introduce the semantics of these notions.

In Chapter 6, we will present formal proofs of some important characteristics of compatible and incompatible ontology mappings. We differentiate these notions of compatible and incompatible ontology mappings with other notions used in the context of ontology mappings, however we will relate these notions with mapping correctness to show that these notions cover the case of mapping correctness.

(O3). Objective 3: To check whether existing state of the art work provides a solution for proposing unified approach of identifying compatibility and incompatibility of ontology mappings.

This objective will be achieved in Chapter 2, Chapter 3 and Chapter 4. We analyze existing work to solve the problem of combined use of ontology mappings.

In Chapter 2, we will present various formalisms used for representing and interpreting ontology mappings. We try to find one formalism that is general enough to capture the representation of other formalism, so that it can be used for finding compatible and incompatible ontology mappings.

In Chapter 3, we will analyze algebraic approaches for representing ontology mappings and various operations performed by these approaches. We will identify the inadequacy of these approaches to be used for identifying compatible and incompatible ontology mappings.

In Chapter 4, we will analyze debugging and verification based approaches for identifying conflicts in ontology mappings. We will investigate whether these approaches identify conflicts among ontology mappings. We will analyze argumentation framework and evaluate its suitability to solve our problem.

(O4). Objective 4: To develop a unified approach for dealing with ontology mappings and their defects that is applicable to various formalisms of ontology and ontology mappings.

In Chapter 5, we will present our Galois connections based approach for finding compatibility and incompatibility between ontology mappings. We will describe the basic definitions that are relevant and useful in our proposed approach. We will present different kinds of lattices that can be used for representing ontology and define the notions of compatibility and incompatibility for dealing with ontology mappings and their defects. We will present a sketch of detecting defects by using our unified approach for dealing with ontology mappings and their defects.

(O5). Objective 5: To relate notions of compatible and incompatible ontology mappings with ontology mapping correctness.

In Chapter 6, we will relate our notion of compatible ontology mappings with theory interpretation and principle of conservativity, coherence and consistency and present proofs of theorem about the key characteristics of compatible ontology mappings. Agreed consensus mappings may still have defects and this will be highlighted in Chapter 4.

(O6). Objective 6: To demonstrate the adequacy of the compatible and incompatible ontology mappings in the context of domain and upper ontologies.

In Chapter 6, we will present how our method can be applicable to mappings of upper ontologies. Upper ontologies are different from domain ontologies in various ways, one reason is that upper ontologies have abstract concepts and they have not individuals in general. Hence, methods used for domain ontology mappings, which in general consider individuals for mapping artifacts of domain ontologies, are not applicable to upper ontologies. We will present a methodology for dealing with upper ontology mappings. We will show that when approaches not considering compatible and incompatible ontology mappings are used, it result in incorrect results in domain ontologies.

In Chapter 8, we present conclusion.

An overview of structure of this thesis is presented in Figure 1-8.

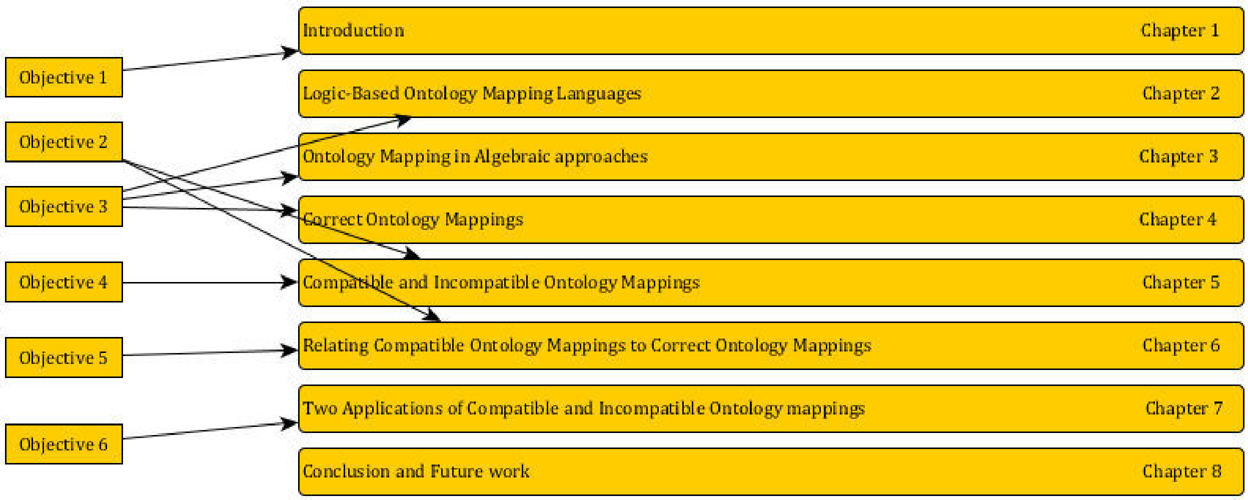

Figure 1-8. An overview of the thesis structure relating the objectives of the thesis with the chapters in which they are accomplished

Chapter 2.

Logic-Based Ontology Mapping Languages

Ontology mapping relates artifacts of source ontologies and it is represented in some formalization. In most cases, mappings are represented as declarative syntax having no semantics. However, there exist few formalisms having predefined semantics that are used for expressing ontology mappings. We focus on those mapping languages that have predefined semantics and in this chapter we present only logic-based ontology mapping languages since mappings expressed in declarative syntax have no precise semantics.

This chapter is organized as follows: Section 2.1 presents prevalent logic-based ontology mapping languages and explain their syntax, semantics and other related features; and finally Section 2.2 presents synthesis of ontology mapping languages described in this chapter and also list the key features of ontology mappings.

2.1 Syntax and semantics of Logic based Ontology mapping Languages

In this section, we present syntax and semantics of logic based ontology mapping languages. We do not present all logic-based ontology mapping languages, we only present those that are generally used for ontology mapping in our point of view. The focus of the presentation of each language is on semantics, since it constrains the meaning of models, and local inconsistency, since it can propagate to other source ontologies. In the following, we focus on mappings between two source ontologies.

2.1.1 Web Ontology Language (OWL 2)

Web otology Language OWL (Hitzler et al., 2009) is an ontology language having formal semantics. It is mainly used for representing ontologies but it has some constructs that are used for expressing ontology mappings. In OWL, Ontology mapping between ontologies is carried out in two steps; by importing an ontology and then by defining equivalence relation between artifacts of two ontologies (importing and imported ontologies).

Artifacts of ontologies are used and accessed by direct references to ontologies' artifacts such as $O_A{:}\,Car$ and $O_B{:}\,Honda$. In OWL, one can represent concept mappings and role mappings, but can't represent concept/role-mappings or role/concept-mappings. OWL's two constructs owl:equivalentClasses and owl:equivalentObjectProperties are used for relating concepts and for relating roles respectively in expressing ontology mappings.

owl:equivalentClasses construct is mainly used for defining class and expressing relatedness of classes in ontologies. An example that relates classes Parent and Guardian is

EquivalentClasses(:Parent :Guardian)

equivalentClasses or equivalentObjectProperties represents only relatedness and not equality (same intentional meaning). This means that equivalent classes in OWL have same set of individuals (extension) but they may have different properties (intention) and same is the case for equivalent object properties. owl:equivalentClasses and owl:equivalentObjectProperties provide necessary and sufficient conditions for membership of a class and property respectively.

These two constructs are also used for expressing relatedness among artifacts of two ontologies. For instance,

$$EquivalentClasses(O_1{:}Teacher\ O_2{:}Professor)$$
$$EquivalentObjectProperties(O_1{:}teaching\ O_2{:}presenting\_seminar\ )$$

represent ontology mapping of source ontologies O_1 and O_2. Class $Teacher$ in O_1 has correspondence with $Professor$ in O_2 and Properties $teaching$ in O_1 has correspondence with $presenting\_seminar$ in O_2.

An advantage of using OWL as a mapping language is that powerful reasoning approaches for OWL ontology are available and no new algorithm is needed for reasoning.

In OWL, interpretation function defines model-theoretic semantics; an interpretation $\langle \Delta^{\mathcal{I}}, .^{\mathcal{I}} \rangle$ consists of non-empty set Δ called interpretation domain and mapping function $.^{\mathcal{I}}$ which maps each concept C with a set $C^{\mathcal{I}} \subset \Delta^{\mathcal{I}}$, each object property P with a binary relation $P^{\mathcal{I}} \subset \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and each individual i with an element $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

In OWL, Ontology has an interpretation domain. An ontology that is imported for mapping purposes does not have its own separate interpretation domain since after import operation its previous interpretation domain becomes part of the interpretation domain of the importing ontology. Hence, there is no distinction between importing and imported ontology interpretation domains; they have same interpretation domain.

If a source ontology involved in mappings has some inconsistent concept or property then the new ontology obtained after the import operation and defining equivalence among artifacts becomes inconsistent too.

OWL is mainly used for ontology representation and reasoning on a single ontology, not suitable for ontology mappings. OWL supports very limited expressivity of mappings; it can only handle mapping consisting of correspondences with equivalence relation. Other types of mapping involving subsumption and overlap relation can only be expressed in OWL as annotations, but they are of no use for semantic purposes because this information cannot be accessed by using OWL constructs. Ontology mappings in OWL are symmetric and transitive, and composition of mappings is also possible. OWL does not allow source ontologies to have different views (different mappings) about each other. Limited expressivity of OWL for ontology mappings and same interpretation domain for source ontologies makes it unsuitable for using it as a mapping language in most cases.

2.1.2 Distributive First-order Logic (DFOL)

First-order logic does not support distributed features directly, which are often required when using ontologies. Distributed First-order Logic (DFOL) (Ghidini & Serafini, 1998) is the distributed version of first-order logic.

The design goal of DFOL is to formalize relations among objects such as by predicates and functions, as well as formulae of different subsystems (here ontologies) by logical consequence. The relation represents a contextual point of view of ontology about another ontology.

The syntax of DFOL is primarily based on first-order logic (FOL). Each ontology is represented by a language L_i and complete set of ontologies are represented by family of first-order logics $\{L_i\}_{i \in I \wedge I \neq \emptyset}$. An ontology may have some knowledge that is common with other ontologies. A formula ϕ may appear in L_i and L_j, but its interpretation may not be same in both languages. For instance, i is vegetarian and j is non-vegetarian. The formula "he eats" for i, means eating vegetarian food while for j, means eating non-vegetarian foods.

Let M_i be set of all possible models of L_i and $m \in M_i$ be a local model of L_i. Each m is a pair $\langle dom, I\rangle$ where dom is a domain and I is an interpretation function. In DFOL, ontologies S_i represent partial information about the system since they do not have complete knowledge of the distributed system represented as $S_i \subseteq M_i$ for domain dom_i.

A *set of local models* is a set of first-order interpretations which agree on interpretations of L_i having complete knowledge about ontology O_i. A domain relation r_{ij} which is from i to j represents the capability of the j-th subsystem to represent in its domain the elements of domain of i-th subsystem. It is not necessary that $r_{ij} = r_{ji}$, i.e., domain relation is not necessarily symmetric. For d in dom_i, $r_{ij}(d)$ denotes $\{d' \in dom_j | \langle d, d'\rangle \in r_{ij}\}$ and it should not be considered as if d and d' are the same objects in a domain shared by i and j. Domain relation r_{ij} allows relating one object of domain dom_i with several objects of dom_j.

In DFOL, for expressing relations between objects of one ontology with other objects of some other ontology, language L_i is extended by arrow variables. For each variable x and each index $j \in I$, DFOL has two arrow variables: (i) $x^{\rightarrow j}$ represents that it is a placeholder of object(s) of domain dom_i which is a pre-image of object(s) of j-th ontology via relation r_{ij}, and (ii) $x^{j\rightarrow}$ represents that it is a place holder of object(s) of dom_i which is an image of objects of j-th ontology via relation r_{ji}. To distinguish occurrence of terms and formulae in different languages, they are prefixed by their index such as ϕ is a formula of L_i, so it is represented by $i{:}\,\phi$.

DFOL model is defined as

Definition 2-1 (DFOL Model): A DFOL model $\mathcal{M}$ is a pair $\langle\{\mathcal{S}_i\},\{r_{ij}\}\rangle$, where $i \neq j \in J_i$: $S_i \subseteq M_i$, is a set of possible local models for L_i, and r_{ij} is a domain relation from dom_i to dom_j.

$|P(x)|_i$ syntax is used for specifying objects that belongs to the interpretation of a predicate P in all the local models of S_i. While for dealing arrow variables, it is required to extend the definition of assignment. An assignment a for each system i provides an interpretation for all the variables and for some arrow variables (as there is no consistent way to assign arrow variables; for instance, if $a_i(x) = d$ and $r_{ij}(d) = \emptyset$ then for representing this situation, it does not exists any assignment for $x^{i\rightarrow}$).

Assignment of variable is defined as

Definition 2-2 (Assignment): (Hitzler et al., 2006) Let $\langle\{\mathcal{M}_i\},\{r_{ij}\}\rangle$ be a model for L_i. An assignment a is a family $\{a_i\}$ of partial functions from the set of variables and arrow variables to dom_i, such that for each variable x and all $i \neq j$:

$a_i(x) \in dom_i$, i.e., variable x of L_i has objects in dom_i

$a_i\left(x^{j\rightarrow}\right) \in r_{ji}\left(a_j(x)\right)$, i.e., objects of dom_i are images of objects of dom_j via relation r_{ji}

$a_j(x) \in r_{ij}\left(a_i\left(x^{\rightarrow j}\right)\right)$, i.e., objects of dom_i are pre-images of objects of dom_i via relation r_{ij}

An assignment a is ***admissible*** for formula $i{:}\,\phi$ if a_i assigns all the arrow variables occurring in ϕ.

Using the definition of admissibility, satisfiability in DFOL is defined as

Definition 2-3 (Distributed Satisfiable): (Hitzler et al., 2006) Let $\mathcal{M} = \langle|\mathcal{M}_i|,\left|r_{ij}\right|\rangle$ be a model for $\{L_i\}$, $m \in \mathcal{M}_i$, and a an assignment. An i-formula ϕ is satisfied by $m \in \mathcal{M}_i$ w.r.t. a, $m \vDash_D \phi[a]$ if

a is admissible, i.e., assigns interpretation for all the arrow variables of $i{:}\,\phi$ and

$m \vDash \phi[a]$, according to the definition of satisfiability for first-order logic.

$\mathcal{M} \vDash \Gamma[a]$ if for all $i{:}\,\phi \in \Gamma$ and $m \in \mathcal{M}_i$, $m \vDash_D \phi[a_i]$, where $\vDash_D$ represents satisfiability in DFOL.

Mappings between different ontologies are formalized in DFOL with the help of interpretation constraints that involve more than one ontology.

Definition 2-4 (Interpretation Constraint): Interpretation constraint from $i_1, \ldots i_n$ to i with $i_k \neq i$ is an expression of the form

$$i_1: \phi_1, \dots, i_n: \phi_n \to i: \phi$$

Interpretation constraints restricts the set of possible DFOL models to those which satisfy it.

Definition 2-5 (Satisfiability of Interpretation Constraint): A model $\mathcal{M}$ satisfies the interpretation constraint $\mathcal{M} \vDash i_1: \phi_1, \dots, i_n: \phi_n \to i: \phi$

If for any assignment a strictly admissible, i.e., only assigns all the arrow variables of $\{i_1: \phi_1, \dots, i_n: \phi_n\}$, if $\mathcal{M} \vDash i_k: \phi_k[a]$ for $1 \leq k \leq n$, then a can be extended to an assignment a' admissible, i.e., assign all the arrow variables of $i: \phi$ s.t. $\mathcal{M} \vDash i: \phi[a']$.

Interpretation constraints act like axioms that restrict the set of possible models of DFOL to those which satisfy it.

Following are few interpretation constraints (Hitzler et al., 2006). We first present interpretation constraint, and then its translation in FOL (in loose sense) followed by an example. We use Ontologies described in Example 1-1 presented in Chapter 1 for giving examples of these constraints.

a) $\mathcal{M} \vDash i: P(x^{\to j}) \to j: Q(x)$ iff For all $d \in |P|_i$ and for all $d' \in r_{ij}(d), d' \in |Q|_j$.

$$\forall a_i(x^{\to j}) \in |P|_i \to \forall a_j(x) \in r_{ij}\left(a_i(x^{\to j})\right) \wedge a_j(x) \in |Q|_j$$

Example 2-1: Every Research Organizations of $O_A$which are mapped to O_B are university and are represented by $O_A: Research\ Org(x^{\to B}) \to O_B: University(x)$

b) $\mathcal{M} \vDash i: P(x) \to j: Q(x^{i\to})$ iff For all $d \in |P|_i$ and there is a $d' \in r_{ij}(d)$, s.t. $d' \in |Q|_j$

$$\forall a_i(x) \in |P|_i \to \exists a_j(x^{i\to}) \in r_{ij}(a_i(x)) \wedge a_j(x) \in |Q|_j$$

Example 2-2: Every Researcher of $O_A$is mapped to Research officer of O_B are university and are represented by $O_A: Researcher \to O_B: Reserch\ Officer(x^{A\to})$

c) $\mathcal{M} \vDash j: Q(x^{i\to}) \to i: P(x)$ iff For all $d \in |Q|_j$ and for all d' with $d \in r_{ij}(d'), d' \in |P|_i$

$$\forall a_j(x^{i\to}) \in |Q|_j \to \forall a_i(x) \in |P|_i \wedge a_j(x^{i\to}) \in r_{ij}(a_i(x))$$

Example 2-3: Every Director Admin of ontology $O_B$which is mapped to O_A are Administrative Staff and it is represented by O_B: Director Admin $Org(x^{A\to}) \to O_A$: Administrative Staff(x)

d) $\mathcal{M} \vDash j: Q(x) \to i: P(x^{\to j})$ iff For all $d \in |Q|_j$ there is a d' with $d \in r_{ij}(d')$s.t. $d' \in |P|_i$

$$\forall a_j(x) \in |Q|_j \to \exists a_i(x^{\to j}) \in |P|_i \wedge a_j(x) \in r_{ij}(a_i(x^{\to j}))$$

Example 2-4: Every Admin Staff of $O_B$is a Director in O_B and it is represented by O_A: Admin Staff $(x) \to O_B: University(x^{\to B})$

Since domain relations r_{ij} and r_{ji} are independent, so, analogously, interpretation constraints from L_i to L_j never affects the logical consequence in the opposite direction.

Ontologies may have inconsistencies, it is desired that whole distributed system does not become inconsistent. DFOL permits local inconsistency (only local ontology is inconsistent and whole distributed system consisting of several ontologies may not be inconsistent) by using some variant of multi-model epistemic semantics in which inconsistent knowledge has models when the set of accessible worlds is empty. DFOL deals with propagation of local inconsistency by allowing it when there is an *explicit specification* of the following constraint

$\mathcal{M} \vDash IP_{ij}$ iff $\mathcal{M}_i = \emptyset$ implies that $\mathcal{M}_j = \emptyset$

where $\mathcal{M}$ is a DFOL model and IP refers to 'Inconsistency propagation'.

Inconsistency propagates from ontology i to ontology j only if i is inconsistent and there is an implication from i to j. Inconsistency propagates from inconsistent ontology i to ontology j only when there is an interpretation constraint $i{:}\phi \rightarrow j{:}\psi$ between them. Consistent ontologies do not become inconsistent if there is no interpretation constraint from inconsistent ontologies to them.

Main drawback of DFOL is that ontology mapping become additional ontology axioms (constraints), and also it requires ontologies to be formalized in FOL.

2.1.3 Distributive Description Logics (DDL)

Ontologies are often represented in decidable fragments of first-order logic, which is Description Logic. Distributed Description Logics (DDL) (Borgida & Serafini, 2003) has been proposed to deal with multiple ontologies which may be interconnected with each other.

Syntax of DDL is inspired from DFOL and it is also composed of two components $\langle \{\mathcal{T}_i\}_{i \in I \wedge I \neq \emptyset}, \mathfrak{B} \rangle$. A family of local ontologies $\mathcal{T}_i$ and ontology being expressed in Description Logic, while $\mathfrak{B}$ is a union of 'bridge rules'. Bridge rules are axioms that are used for expressing mapping, a semantic association between artifacts of different ontologies. There are two types of bridge rules: Into bridge rule and Onto bridge rule.

- $i{:}A \xrightarrow{\sqsubseteq} j{:}B$ (*into bridge rule*), i.e., according to ontology j's point of view artifact A of ontology i is a sub-artifact of artifact B of ontology j
- $i{:}A \xrightarrow{\sqsupseteq} j{:}B$ (*onto bridge rule*), i.e., according to ontology j's point of view artifact A of ontology i is a super-artifact of artifact B of ontology j

where A and B are either concepts or roles.

Semantically, bridge rules are treated as domain relations. Domain relation r_{ij} is a relation between Δ^{I_i} and Δ^{I_j} as $r_{ij} \subseteq \Delta^{I_i} \times \Delta^{I_j}$. The basic intuition of DDL is that each ontology has its own view of domain and this intuition is implemented by permitting any type of domain relation, so domain relation need not to be symmetric, injective etc.

DDL also allow overlapping relation to map artifacts of ontologies as $i: A \overset{*}{\rightarrow} j: B$.

Definition 2-6 (Satisfiability of Bridge rules): $\mathcal{I}_i$ and $\mathcal{I}_j$ are interpretations of ontology i and j respectively and a_{ij} is artifact

$$\langle \mathcal{I}_i, a_{ij}, \mathcal{I}_j \rangle \vDash i: A \overset{\sqsubseteq}{\rightarrow} j: B \ if \ r_{ij}(A^{\mathcal{I}_i}) \subseteq B^{\mathcal{I}_j}$$

$$\langle \mathcal{I}_i, a_{ij}, \mathcal{I}_j \rangle \vDash i: A \overset{\sqsupseteq}{\rightarrow} j: B \ if \ r_{ij}(A^{\mathcal{I}_i}) \supseteq B^{\mathcal{I}_j}$$

$$\langle \mathcal{I}_i, a_{ij}, \mathcal{I}_j \rangle \vDash i: A \overset{*}{\rightarrow} j: B \ if \ r_{ij}(A^{\mathcal{I}_i}) \cap B^{\mathcal{I}_j} \neq \emptyset$$

Later Serafini and colleagues extended DDL by adding , individual correspondences (Serafini & Tamilin, 2006)

- $i: a \mapsto j: b$ (Partial Individual Correspondence) individual a of ontology i is associated with individual b of ontology j, and there are other individuals besides b to which a is associated but, here, there is only information about association between a and b. Satisfiability of this mapping is as

$$\langle \mathcal{I}_i, a_{ij}, \mathcal{I}_j \rangle \vDash i: a \mapsto j: b \ if \ b^{\mathcal{I}_j} \in r_{ij}(a^{\mathcal{I}_i})$$

- $i: a \overset{=}{\mapsto} j: \{b_1, .. b_n\}$ (Complete Individual Correspondence), i.e., individual a of ontology i is associated with only individual $b_1, .. b_n$ of ontology j. Satisfiability of this mapping is as

$$\langle \mathcal{I}_i, a_{ij}, \mathcal{I}_j \rangle \vDash i: a \mapsto j: \{b_1, \dots, b_n\} \ if \ b^{\mathcal{I}_j} \in r_{ij}(a^{\mathcal{I}_i}) = \{b_1^{\mathcal{I}_j}, \dots, b_n^{\mathcal{I}_j}\}$$

There also exist heterogeneous mappings, i.e., concepts are mapped to roles (concept/role) and roles are mapped to concepts (role/concept). DDL has been extended to incorporate such mappings.

The domain relation cr_{ij} between ontology i's concept and ontology j's role is a subset of $\Delta^{I_i} \times \Sigma^{I_j}$ where $\Sigma^{I_j} = \Delta^{I_j} \times \Delta^{I_j}$. This means that for $\in c_i$, there is $(x_1, x_2) \in r_j$.

Example 2-5: An example of cr_{ij} is $i: Latitude \overset{\sqsubseteq}{\rightarrow} j: latitude$ where $Latitude$ is a concept and $latitude$ is a role. Whereas the domain relation between concepts and roles is rc_{ij} from ontology i to ontology j is a subset of $\Sigma^{I_i} \times \Delta^{I_j}$ where $\Sigma^{I_i} = \Delta^{I_i} \times \Delta^{I_i}$. This means that for $(x_1, x_2) \in r_i$ there is $x \in c_j$.

Example 2-6: An example of rc_{ij} is $i{:}player \overset{\sqsubseteq}{\rightarrow} j{:}Team$ where $player$ is a role and $Team$ is a concept.

- $i{:}R \overset{\sqsubseteq}{\rightarrow} j{:}C$ (*into bridge rule*), i.e., according to ontology j's point of view role R of ontology i is a sub-artifact of concept C of ontology j.

$$\langle \mathcal{I}_i, rc_{ij}, \mathcal{I}_j \rangle \vDash i{:}R \overset{\sqsubseteq}{\rightarrow} j{:}C \ if \ rc_{ij}(R^{\mathcal{I}_i}) \subseteq C^{\mathcal{I}_j}$$

- $i{:}R \overset{\sqsupseteq}{\rightarrow} j{:}C$ (*onto bridge rule*), i.e., according to ontology j's point of view role R of ontology i is a super-artifact of concept C of ontology j.

$$\langle \mathcal{I}_i, rc_{ij}, \mathcal{I}_j \rangle \vDash i{:}R \overset{\sqsupseteq}{\rightarrow} j{:}C \ if \ rc_{ij}(R^{\mathcal{I}_i}) \sqsupseteq C^{\mathcal{I}_j}$$

- $i{:}C \overset{\sqsubseteq}{\rightarrow} j{:}R$ (*into bridge rule*), i.e., according to ontology j's point of view concept C of ontology i is a sub-artifact of concept C of ontology j

$$\langle \mathcal{I}_i, cr_{ij}, \mathcal{I}_j \rangle \vDash i{:}C \overset{\sqsubseteq}{\rightarrow} j{:}R \ if \ cr_{ij}(C^{\mathcal{I}_i}) \subseteq R^{\mathcal{I}_j}$$

- $i{:}R \overset{\sqsupseteq}{\rightarrow} j{:}C$ (*onto bridge rule*), i.e., according to ontology j's point of view role R of ontology i is a super-artifact of concept C of ontology j.

$$\langle \mathcal{I}_i, cr_{ij}, \mathcal{I}_j \rangle \vDash i{:}C \overset{\sqsubseteq}{\rightarrow} j{:}R \ if \ cr_{ij}(C^{\mathcal{I}_i}) \sqsupseteq R^{\mathcal{I}_j}$$

All the bridge rules are directed and it is not necessary that $\mathfrak{B}_{ij} = \mathfrak{B}_{ji}$.

Example 2-7: An example of cr_{ij} is mapping of concept $Latitude$ of ontology i is mapped to role $latitude$ of ontology j in a *subsumption relation*, in symbols as $i{:}Latitude \overset{\sqsubseteq}{\rightarrow} j{:}latitude$.

Example 2-8: An example of rc_{ij} is role $player$ of ontology i is mapped to concept $Team$ of ontology j in a *subsumption relation*, in symbols as $i{:}player \overset{\sqsubseteq}{\rightarrow} j{:}Team$.

In the following, our focus is on concepts bridge rules. Bridge rules involving roles and heterogeneous bridge rules can be handled analogously.

A distributed T-box (DTB) $\mathfrak{T} = \langle \mathcal{T}_i, \mathcal{B}_i \rangle$ is a collection $\mathcal{T}_i$ of T-boxes, and a collection $\mathcal{B} = \{\mathcal{B}_i\}_{i \neq j \in I}$ of bridge rules between them.

There are two types of interpretations in DDL: (a) d-interpretation (b) ϵ-interpretation

Definition 2-7 (d-interpretation): For distributed ontology $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i \in I \wedge I \neq \emptyset}, \mathfrak{B} \rangle$, a d-interpretation $\mathcal{J} = \langle \{I_i\}_{i \in I \wedge I \neq \emptyset}, \{r_{ij}\}_{i,j \in I, i \neq j} \rangle$ consists of a set of local interpretations and a set of domain relations as $r_{ij} \subseteq \Delta^{I_i} \times \Delta^{I_j}$. Each interpretation $I_i = \langle \Delta^{I_i}, .^{I_j} \rangle$ has a non-empty domain. Local domains are mutually disjoint.

In order to avoid the situation when whole distributed ontology becomes inconsistent due to local inconsistency, ϵ-interpretation is introduced by using the idea of *hole*.

Definition 2-8 (Hole): (Borgida & Serafini, 2003) A Hole is a pair $\langle \emptyset, .^{\epsilon} \rangle$, such that its domain is empty set $\emptyset$ and interpretation function $.^{\epsilon}$ assigns $\emptyset$ to every concept, role or individual.

Definition 2-9 (ϵ-interpretation): For distributed ontology $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i \in I \wedge I \neq \emptyset}, \mathfrak{B} \rangle$, a d-interpretation $\mathcal{I} = \langle \{I_i\}_{i \in I \wedge I \neq \emptyset}, \{r_{ij}\}_{i,j \in I, i \neq j} \rangle$. consist of set of local interpretations and a set of domain relations as $r_{ij} \subseteq \Delta^{I_i} \times \Delta^{I_j}$. Each interpretation $\mathcal{I}_i$ is either $I_i = \langle \Delta^{I_i}, .^{I_i} \rangle$ or is a hole. Local domains are mutually disjoint or both are equal to $\emptyset$.

Definition 2-10 (DDL Distributed Model): A distributed model $\mathcal{M}$ represented by $\mathcal{M} \vDash_{\epsilon} \mathfrak{T}$ if $\mathcal{M} \vDash_{\epsilon} \mathcal{T}_i$ and $\mathcal{M} \vDash_{\epsilon} \mathfrak{B}$, i.e., there is ϵ-interpretation for every local ontology and ϵ-interpretation satisfies all bridge rules.

In DDL, bridge rules are directional and back flow does not occur in most of the cases except when $i{:}A \xrightarrow{\sqsupseteq} j{:}B$ where $j{:}\top \sqsubseteq j{:}B$, since it says that $r_{ij}(A^{\mathcal{I}_i}) \neq \emptyset$ because the extension of ANYTHING ($\top$) cannot be empty. This kind of reasoning can sometime cause new subsumptions in ontology i.

Subsumption in DDL propagates across ontologies by using into-bridge and onto-bridge rule. For instance, ontology i has axiom $C \sqsubseteq D$ and there are bridge rules $i{:}C \xrightarrow{\sqsupseteq} j{:}G$ and $i{:}D \xrightarrow{\sqsubseteq} j{:}H$, then consequently there is $j{:}G \sqsubseteq j{:}H$.

If there is a bridge rule $\mathfrak{B}_{ij}$, and ontology $\mathcal{T}_j$ is inconsistent and T_i having interpretation I_i is consistent, it remains consistent even with $\mathfrak{B}_{ij}$ since there exists a distributed interpretation $\mathcal{I} = \{I_i, I_j^{\epsilon}\}, \{r_{ij} = \Delta^{I_i} \times \Delta^{I_j}\}$.

If there is a bridge rule $\mathfrak{B}_{ij}$, and $\mathcal{T}_i$ is inconsistent, then T_j becomes inconsistent due to bridge rule. However, still there exists a $\epsilon -$interpretation for distributed ontology, hence distributed ontology is not inconsistent.

DDL has unusual semantics that some concepts are satisfiable when in intuitive logical sense they should be unsatisfiable. (Grau et al., 2004) identifies that Penguin in ontology j is satisfiable when there exist bridge rules $i{:}Bird \xrightarrow{\sqsupseteq} j{:}Penguin$ and $i{:}\neg Flying \xrightarrow{\sqsupseteq} j{:}Penguin$. Where ontology i has axiom $i{:}Bird \sqcap i{:}\neg Flying = \emptyset$. The reason is that DDL allows too many models and this combination of onto bridge rule does not make jth concept unsatisfiable. This problem occurs only with onto bridge rule. In literature, there are two solutions to avoid this kind of behavior but both solutions limit the original semantics of DDL. Firstly, use of injective function purposed by (Bao et al., 2006) which can be easily followed that this solution avoids that problem. Secondly, use conjunctive bridge rule (Homola, 2007) and operator $\overset{r}{\twoheadrightarrow}$ is used to relate artifacts of two ontologies, where r is a relation. The above mentioned problem occurs only in case of onto conjunctive bridge $i: A \overset{\sqsupseteq}{\twoheadrightarrow} j: B$. A distributed interpretation satisfies this rule as $\langle \mathcal{I}_i, cr_{ij}, \mathcal{I}_j \rangle \vDash i: A \overset{\sqsupseteq}{\twoheadrightarrow} j: B$ if for any given other conjuctive bridge rule $i: D_1 \overset{\sqsupseteq}{\twoheadrightarrow} j: H_1, \dots . i: D_n \overset{\sqsupseteq}{\twoheadrightarrow} j: H_n$ in $\mathfrak{B}_{ij}$, where $n \geq 0$ and $r_{ij}\left(C^{I_i} \cap D_1^{I_j} \cap \dots \cap D_n^{I_j}\right) \supseteq (G^{I_j} \cap D_1^{I_j} \cap \dots . \cap D_n^{I_j})$. To avoid above mentioned problem in every distributed model $\mathfrak{I}$ of whole distributed system is should hold for conjunctive onto bridge rules that $r_{ij}\left(C^{I_i} \cap D_1^{I_j} \cap \dots \cap D_n^{I_j}\right) \supseteq (G^{I_j} \cap D_1^{I_j} \cap \dots . \cap D_n^{I_j})$.

DDL can express subsumption, supersumption, equivalence, disjointness, and overlap relations between artifacts of different ontologies. C-OWL (Bouquet et al., 2003) is also a mapping language, which is OWL-based implementation of DDL.

The main drawback of DDL is same as that of DFOL that ontology mapping becomes part of the ontology axioms (act as constraints) and it require source ontologies to be formalized in DDL.

2.1.4 $\mathcal{E}$-Connections

$\mathcal{E}$-connections (Kutz et al., 2003) is an approach originally proposed for combining logics but later also used for defining mappings between ontologies (Grau et al., 2004). Here, we are focusing on mapping aspect of this approach and not on computational aspect.

Domains $D_1, D_2, \dots, D_n$ representing in appropriate language $L_1, L_2, \dots, L_n$ are linked by a set $\mathcal{E} = \{E_j | j \in J\}$ of links establishing certain relations $E_j \subseteq D_1 \times \dots \times D_n$ among objects of domains. In DFOL and DDL, there is single relation from ontology i to ontology j while in $\mathcal{E}$ −connections there are many possible relations. It is not required that domains are disjoint in $\mathcal{E}$ −connections.

Ontologies in $\mathcal{E}$ −connections may represent the same object but with different aspects of the same object. For instance, two ontologies describing about person; one is describing its temporal life such as year of birth, year of graduation, while other is describing its spatial life such as place of birth and his citizenship.

$\mathcal{E}$ −connections is defined as (Kutz et al., 2004)

Definition 2-11 (ϵ −connections); Links establishing between different domains are represented by a link set $\mathcal{E} = \{E_j | j \in J\}$, where $E_j \subseteq D_1 \times D_2 \times \dots \times D_n$. A new language L is formed which contains all of the L_i that describe about $\bigcup_{i=1}^{n} D_i$ where D_i are connected by links in $\mathcal{E}$ and the $(n-1)$-ary operator $\langle E_j \rangle^i, 1 \leq i \leq n, j \in J$ which given an input $(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n)$, for $X_l \subseteq D$, returns

$$\{x \in D_i \mid \forall l \neq i \; \exists x_l \in X_l (x_1, \dots, x_{i-1}, x, x_{i+1}, x_n) \in E_j\}$$

For instance, if $i = 2$ then, for all $X_1 \subseteq D_1$ and $X_2 \subseteq D_2$, there is

$$x_1 \in \langle E_j \rangle^1 (X_2) \Leftrightarrow \exists x_2 \in X_2, (x_1, x_2) \in E_j$$

$$x_2 \in \langle E_j \rangle^2 (X_1) \Leftrightarrow \exists x_1 \in X_1, (x_1, x_2) \in E_j$$

A non-empty set $\mathcal{E}$ is called a '*link set'* for domains (ontologies). The elements of ε are called *link relations* or '*links'*. Give a set ε the *'link operators'* generated by ε is the set $\{\langle E_j \rangle^i | 1 \leq i \leq n, n \in J\}$ of function symbols $\langle E_j \rangle^i$ of arity $(n-1)$. The terms of ϵ −connection $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \dots, \mathcal{S}_n)$, where $\mathcal{S}_i$ is the $i-th$ ontology, consists of terms of language L_i enriched with the new function symbols $\langle E_j \rangle^i$ for each $j \in J$. Link assertions is $(a_1, \dots, a_n)\colon E_j$, where $a_1, \dots, a_n)$: are object variables for $L_i$enrich and $j \in J$.

Definition 2-12 (Semantics of $\boldsymbol{\epsilon}$ −connections): The semantics of ε-connection is given by combined interpretation $\mathfrak{M} = \langle (\mathfrak{M}_i)_{i \leq n}, \mathcal{E}^{\mathfrak{M}} = (E_j^{\mathfrak{M}})_{j \in J}) \rangle$ Where $\mathfrak{M}_i = (\Delta_\mathrm{i}, .^{\mathfrak{M}_\mathrm{i}}) \in \mathcal{M}_\mathrm{i}$ for $1 \leq i \leq n$ and $E_j^{\mathfrak{M}} \subseteq \Delta_1 \times \dots \times \Delta_n$ for each $j \in \mathcal{M}$. Δ is a non-empty set. For term $t^{\mathfrak{M}} \subseteq \Delta_i$ is defined by induction; set of variables $X^{\mathfrak{M}} = X^{\mathfrak{M}_i}$, set of objects $a^{\mathfrak{M}} = a^{\mathfrak{M}_i}$, $\neg t_1 = \Delta_\mathrm{i} \backslash \mathrm{t_1}^{\mathfrak{M}}$, $(t_1 \wedge t_2)^{\mathfrak{M}} = t^{\mathfrak{M}} \wedge t^{\mathfrak{M}}$, $f(t_1, \dots . t_m)^{\mathfrak{M}} = f^{\mathfrak{M}}(t_1^{\mathfrak{M}}, \dots . t_m^{\mathfrak{M}})$, for relation $R^{\mathfrak{M}} = R^{\mathfrak{M}_i}$, for $\bar{t} = t_1, \dots t_{i-1}, t_{i+1}, \dots t_n$ be a sequence of j −terms t_j $i \neq j$,

$$(\langle E_j \rangle^i (\bar{t}_i))^{\mathfrak{M}} = \{x \in \Delta_i \underset{l \neq i}{\exists} x_l \in t_l^{\mathfrak{M}} (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \in E_j^{\mathfrak{M}}\}$$

Definition 2-13 (Satisfiability of $\boldsymbol{\mathcal{E}}$ −connection $\boldsymbol{C^{\mathcal{E}}(S_1, \dots, S_n)}$):

$\mathfrak{M} \vDash t_1 \sqsubseteq t_2 \; iff \; t_1^{\mathfrak{M}} \subseteq t_2^{\mathfrak{M}}$

$\mathfrak{M} \vDash a\colon t \; iff \; a^{\mathfrak{M}} \in t$

$\mathfrak{M} \vDash R(a_1, \dots, a_m) iff \; R^{\mathfrak{M}}(a_1^{\mathfrak{M}}, \dots a_m^{\mathfrak{M}})$

$\mathfrak{M} \vDash (a_1, \dots, a_m)\colon E_j \; iff \; E_j^{\mathfrak{M}}(a_1, \dots, a_m)$

A formula φ is satisfied in $\mathfrak{M}$ if $\mathfrak{M} \vDash \varphi$.

Entailment of term assertion can be reduced to satisfiability as in the case of Description logic.

Semantic mappings can also be represented by links: for instance, as $i\colon \phi \sqsubseteq \exists \langle E \rangle^i . j\colon \psi$ or by $i\colon \phi \sqsubseteq \exists \langle E^- \rangle^i . j\colon \psi$, analogously mapping may have universal quantifier and number restrictions.

Example 2-9: Let i and j are two ontologies. Ontology i deals with open space places, while ontology j deals with animals. There exist several links between these ontologies; one link is $lives \in \mathcal{E}_{ji}$. 'Wild jungle' is defined in ontology $i$as

$$Wild - Jungle \equiv Open - Space - Place \quad \sqcap \langle E_{j\text{-}} \rangle^1 . Lion$$

Or in Description logic as

$$Wild - Jungle \equiv Open - Space - Place \quad \sqcap \exists lives. Lion$$

And Frightened sheep as

$$Frightened - Sheep \equiv Sheep \sqcap \exists \langle E_{j\text{-}}^{-} \rangle^1 . (Wild - Jungle)$$

Or in Description logic as

$$Frightened - Sheep \equiv Sheep \sqcap \exists lives^{-} (Wild - Jungle)$$

Links in $\mathcal{E}$ −connection from ontology i to ontology j can be used to define concepts of ontology i in a way analogous for defining concepts by using roles.

$$(\exists r. C)^I = \{x \in \Delta_i | \exists y \in \Delta_j, (x, y) \in r^I \wedge y \in C^I \}$$

$$(\forall r. C)^I = \{x \in \Delta_i | \forall y \in \Delta_j, (x, y) \in r^I \rightarrow y \in C^I \}$$

There can be n different relations between same artifacts of ontologies O_i and O_j. It is possible to perform Boolean operations on links as $\langle E_1 \sqcup E_2 \rangle$ and $\langle E_1 \sqcap E_2 \rangle$, but then it is not possible to distinguish models of Boolean combination of links.

Interpretation domains are strictly non-empty whereas in DDL interpretation domains can be empty. If one of the ontology involved in ϵ −connection is inconsistent, other ontologies would only become inconsistent whenever there is a link between an inconsistent ontology and themselves. Since there are inverse links so it does not matter whether inconsistent ontology involved in ϵ −connection is source or target of the *link* relation, all ontologies involved in the link in which inconsistent ontology is a part of this link becomes inconsistent.

The main weak point concerning $\mathcal{E}$ −connection regarding their use as mapping is that there is no distinction between mappings and ontology axioms, moreover, new concepts can be created by combining *links*. In $\mathcal{E}$ −connection ontologies can be expressed in Abstract Description Systems (ADS), so source ontologies can be formalized in modal or temporal logics.

2.1.5 Ontology Integration Framework (OIS)

Ontology integration system (OIS) is proposed by (Calvanese et al., 2002). The main components of this framework $\langle \mathcal{G}, \mathcal{S}, \mathcal{M}_{\mathcal{G},\mathcal{S}} \rangle$ are a Global ontology $\mathcal{G}$, a set of local ontologies $\mathcal{S}$ and mappings between them $\mathcal{M}_{\mathcal{G},\mathcal{S}}$. The basic purpose of OIS approach is to integrate different data sources. Queries are posed over global ontology and then each query is reformulated in terms of queries over local ontologies then results obtained from local ontologies are collected and assembled to provide final result.

In OIS, global ontology is required into which all local ontologies are mapped. In semantic terms, interpretation domain of local ontologies are embedded in the interpretation domain of global ontology.

In OIS, there are three types of mappings in OIS

a) Global As View (GAV): artifacts of global ontology are represented by views over local ontologies in the form of $v_i(S_1, \dots S_n) \subseteq G_i$
b) Local As View (LAV): artifacts of local ontology are represented by views over global ontology relations in the form of $S_i \subseteq v_i(G_1, \dots G_m)$
c) Global Local As View (GLAV): combination of GAV and LAV

Example 2-10: A Global Ontology G consists of artifacts $Organization$, $Person$, $Hires(Organization, Person)$, and other axioms.

And Local ontologies consists of $S_1 = (Org)$, $S_2 = (Person)$, $S_3 = (Org, Per)$, $S_4 = (Person, Orga)$

And mapping $M_{G,S}$ in GAV ($g \rightsquigarrow q_S$)formalized in DATALOG is

$$Organization(x) \leftarrow S_1(x)$$

$$Person(x) \leftarrow S_2(x)$$

$$Hires(x, y) \leftarrow S_3(x, z) \wedge S_4(z, y)$$

Similarly, for each local ontology in LAV, mapping associates a *view* over the global ontology.

Inconsistency in one of the local ontology makes the whole system inconsistent as there is no mechanism of handling inconsistency.

In OIS, mapping is the set of correspondences between local and global ontologies. I is the interpretation of whole OIS, ϕ is query over local ontologies, and ψ is query over global ontology. How accurately the view over global ontology is related to view over local ontology in case of GLAV is described by using notion of s*ound, complete* and *exact* as (Calvanese et al., 2002)

1. I satisfies $\langle \phi, \psi, \boldsymbol{sound} \rangle$ w.r.t. the local interpretation D, if all the tuples satisfying ψ in D satisfy ϕ in I.

   In DFOL syntax, it is $s: \psi(x) \rightarrow g: \phi(x^{s\rightarrow})$
2. I satisfies $\langle \phi, \psi, \boldsymbol{complete} \rangle$ w.r.t. the local interpretation D if no tuple other than those satisfying ψ in D satisfies ϕ in I ,

In DFOL syntax, it is $g: \phi(x) \rightarrow s: \psi(x^{\rightarrow g})$

3. I satisfies $\langle \phi, \psi, \boldsymbol{exact} \rangle$ w.r.t. the local interpretation D if the set of tuples that satisfies ψ in D is exactly the set of tuples satisfying ϕ in I.

   In DFOL syntax, it is $s: \psi(x) \rightarrow g: \phi(x^{s\rightarrow})$ and $g: \phi(x) \rightarrow s: \psi(x^{\rightarrow g})$.

$\langle \phi, \psi, \boldsymbol{exact} \rangle$ is equivalent to the conjunction of both $\langle \phi, \psi, \boldsymbol{sound} \rangle$ and $\langle \phi, \psi, \boldsymbol{complete} \rangle$.

Analogously s*ound, complete* and *exact can be defined for LAV for describing* how accurately the content of source ontologies are characterized in terms of views over Global ontology in case of LAV and for describing how accurately each element of the global ontology is characterized in terms of local ontologies in case of GAV.

OIS provides a database method for dealing ontology mappings for checking whether they are sound, complete or exact. These mappings can be represented in DFOL and it provides another way to use methods and languages like DataLog can be used to formalize OIS mappings.

2.1.6 Integrated Distributed Description Logic (IDDL)

Integrated Distributed Description logic (IDDL) is proposed by (Zimmermann, 2007). It is different from mapping languages such as DDL and ϵ −connections in which mapping represents one ontology point of view about other ontology whereas in IDDL mapping represents relation between ontologies from third party point of view. The basic intuition behind IDDL is that mappings can be treated as first class citizens and mapping operations such as composition can be performed.

IDDL has global interpretation domain for whole network of ontologies and local interpretation domain of each ontology is related to the global interpretation domain by "*equalizing functions".* For all interpretations $I_i \in I$ equalizing function ϵ maps each element of local interpretation domain Δ^{I_i} to global interpretation domain Δ^{I_ϵ} as $\epsilon_i: \Delta^{I_i} \rightarrow \Delta^{I_\epsilon}$.

They (Zimmermann, 2007) define distributed interpretation of IDDL as

Definition 2-14 (Distributed interpretation of IDDL): Let $S = \langle O, A \rangle$ be a Distributed system, where O is the set of ontologies and A is the mapping of ontologies. A distributed interpretation is a pair $\langle I, \epsilon \rangle$ where I is a family of interpretation indexed by O, ϵ is an equalizing function for I, , such that for all $i \in O$, I_i interprets i and $\epsilon_i: \Delta^{I_i} \rightarrow \Delta_\epsilon$ (where Δ_ϵ is the global domain of interpretation of ϵ).

Local satisfiability of ontology is as in Description logic while correspondence satisfaction is as:

- cross-ontology concept subsumption;

$$I \vDash_d i{:}C \overset{\sqsubseteq}{\leftrightarrow} j{:}D \text{ iff } \varepsilon_i(C^{I_i}) \subseteq \varepsilon_j(D^{I_j});$$

- cross-ontology role subsumption;

$$I \vDash_d i{:}R \overset{\sqsubseteq}{\leftrightarrow} j{:}S \text{ if f}\varepsilon_i(R^{I_i}) \subseteq \varepsilon_j(S^{I_j});$$

- Cross-ontology concept disjunction

$$I \vDash_d i{:}C \overset{\perp}{\leftrightarrow} j{:}D \text{ iff } \varepsilon_i(C^{I_i}) \cap \varepsilon_j(D^{I_j}) = \emptyset;$$

- Cross-ontology role disjunction

$$I \vDash_d i{:}R \overset{\perp}{\leftrightarrow} j{:}S \text{ iff } \varepsilon_i(R^{I_i}) \cap \varepsilon_j(S^{I_j}) = \emptyset;$$

- Cross-ontology membership;

$$I \vDash_d i{:}a \overset{\epsilon}{\leftrightarrow} j{:}C \text{ iff } \varepsilon_i(a^{I_i}) \in \varepsilon_j(C^{I_j})$$

- cross-ontology identity

$$I \vDash_d i{:}a \overset{=}{\leftrightarrow} j{:}b \text{ iff } \varepsilon_i(a^{I_i}) = \varepsilon_j(b^{I_j})$$

A distributed interpretation I satisfies mapping A iff it satisfies all correspondences of A and satisfies all axioms of O_i.

In IDDL, mappings are not injective, DL inferences are valid for local ontologies and correspondences while axioms deduce new axioms or correspondences. Correspondences can make distributed ontology incoherent. Inconsistency in a local ontology makes the global ontology inconsistent, since all elements of local ontologies are mapped to global ontology and global ontology has single interpretation domain.

2.1.7 Weighted mappings

A weighted ontology mapping is an approach for interpreting assigned weights to ontology mappings (Atencia et al., 2012). Mapping is interpreted as how elements of X (a common set of items X are classified by two source ontologies) classified in concepts of O_i are reclassified in concepts of O_j and weight measures preciseness and completeness of classification. The semantics of this approach is a conservative extension of the semantics of crisp mappings (mappings with set theoretic relations) for a specific class of DDLs.

Definition 2-15 (Weighted mappings): Let $O_{i_{\{i \in I\}}}$ be a family of ontologies. A weighted mapping from O_i to O_j is an expression of the form $i : C\ r_{[a,b]}\ j : D$ where C and D are concepts of O_i and O_j respectively, $r \in \{\sqsubseteq, \equiv, \sqsupseteq, \perp\}$ and a, b are real numbers in the unit interval [0, 1].

The weighted mapping $O_i{:}C \sqsubseteq_{[a,b]} O_j{:}D$ is used to express the fact that the proportion of items of X classified under C according to interpretation $\mathcal{I}_i$ which are re-classified under D according to interpretation $\mathcal{I}_j$ lies in the interval $[a,b]$. The satisfiability of $O_i\!: C \sqsubseteq_{[a,b]} O_j\!: D$ is $\frac{\left|C_X^{\mathcal{I}_i} \cap D_X^{\mathcal{I}_j}\right|}{\left|C_X^{\mathcal{I}_i}\right|} \in [a,b]$ which is 'recall' of $D_X^{I_j}$ w.r.t $C_X^{I_i}$ as $\frac{|C \cap D|}{|C|}$.

The weighted mapping $O_i\!: C \sqsupseteq_{[a,b]} O_j\!: D$ is used to express the fact that the proportion of items of X classified under D according to $\mathcal{I}_j$ is reclassified under C according to $\mathcal{I}_i$ lies in the interval $[a,b]$. The satisfiability of $O_i\!: C \sqsupseteq_{[a,b]} O_j\!: D$ as $\frac{\left|C_X^{\mathcal{I}_i} \cap D_X^{\mathcal{I}_j}\right|}{\left|D_X^{\mathcal{I}_i}\right|} \in [a,b]$ which is 'precision' of $D_X^{I_j}$ w.r.t $C_X^{I_i}$ as $\frac{|C \cap D|}{|D|}$.

Similarly, the weighted mapping $O_i\!: C \equiv_{[a,b]} O_j\!: D$ is interpreted as 'F-measure' $2.\frac{|C \cap D|}{|C|+|D|}$. It is rephrased as $F(C_X^{\mathcal{I}_1}, D_X^{\mathcal{I}_2}) \in [a,b]$.

Weighted mapping for disjointness in interpreted as $1 - F(C_X^{\mathcal{I}_1}, D_X^{\mathcal{I}_2}) \in [a,b]$.

Mapping entailment (Atencia et al., 2012) Let O_i and O_j be two ontologies and let X be a non-empty finite set of fresh individual constants. Also, let M be a set of weighted mappings from O_i to O_j . The set M entails $i : C\ r_{[a,b]}\ j : D$ ***modulo*** X, denoted $M \vDash_X\ O_i : C\ r_{[a,b]} O_j : D$, if for every interpretations I_i and I_j of O_i and O_j, respectively, such that (I_i, I_j) satisfies M ***modulo*** X, i.e., $I_i, I_j \vDash_M O_i : C\ r_{[a,b]}\ O_j\!: D.$ The set M entails $O_i : C\ r_{[a,b]}\ O_j : D$, in symbols, $M \vDash\ O_i : C\ r_{[a;b]}\ O_j : D$, if $\vDash_X O_i\!: C\ r_{[a,b]}\ O_j : D$ for every $X \neq \emptyset$:

When weights of mappings are $[0,0], [0,1], [1,0], [1,1]$, then they can be rewritten in an equivalent set of mappings in $\sqsubseteq_{[1,1]}-$normal form which also refers to DDL based mappings.

$O_i\!: C \sqsupseteq_{[1,1]} O_j\!: D$ is equivalent to $O_j\!: D\ \sqsubseteq_{[1,1]} O_i\!: C$

$O_i\!: C \equiv_{[1,1]} O_j\!: D$ is equivalent to $O_i\!: C\ \sqsubseteq_{[1,1]} O_j\!: D$ and $O_j\!: D\ \sqsubseteq_{[1,1]} O_i\!: C$

$O_i\!: C \perp_{[1,1]} O_j\!: D$ is equivalent to $O_i\!: C\ \sqsubseteq_{[1,1]} O_j\!: \neg D$

$O_i\!: C \sqsupseteq_{[0,0]} O_j\!: D$ is equivalent to $O_i\!: C\ \sqsubseteq_{[1,1]} \mathrm{O_j}\!: \neg D$

$O_i\!: C \equiv_{[0,0]} O_j\!: D$ is equivalent to $O_i\!: C\ \sqsubseteq_{[1,1]} O_j\!: \neg D$

$O_i\!: C \perp_{[1,1]} O_j\!: D$ is equivalent to $O_i\!: C\ \sqsubseteq_{[1,1]} O_j\!: D$ and $O_j\!: D\ \sqsubseteq_{[1,1]} O_i\!: C$

$O_i\!: C\ r_{[0,1]} O_j\!: D$ is equivalent to $O_i : \perp \sqsubseteq_{[1,1]} O_j\!: \top$

$O_i\!: C\ r_{[1,0]} O_j\!: D$ is equivalent to $O_i : \top\ \sqsubseteq_{[1,1]} O_j\!: \perp$

These –normal form weighted mappings can be expressed in bridge rules such as

$$(I_i, I_j) \vDash_X O_i : C \sqsubseteq_{[1,1]}\ O_j : D \text{ iff } (I_i, I_j, r_{ij}) \vDash O_i : C \overset{\sqsubseteq}{\rightarrow} O_j : D$$

Where I_i, I_j are finite non-empty interpretation domains.

This approach is an extension of DDL that deal with weighted mappings so limitations of DDL approach remains the limitation of this approach for checking defects in mappings.

2.1.8 Interpretation between theories

A theory is a set of facts that is always true in a knowledge base. Ontologies are often treated as logical theories. Language of a theory $\mathcal{T}$ denoted by $\mathcal{L}(T)$ consists of non-logical symbols $\Sigma(T)$ (often called *signature* of a theory $\mathcal{T}$). Signatures of first-order logic and Second-order logic consist of predicate (except equality), functions and individual constants. A logical theory is formally defined as set of sentences closed under logical consequence, in symbols it is represented as $T = Cn(T)$.

Using mathematical logic, Enderton (Enderton, 2001) shows that first-order theories can be compared and can be judged whether one theory is as powerful as other theory even when the languages of theories are different but are in FOL. Here, we present Enderton approach of comparing theories and how mappings are treated in this approach. There are two cases on the basis of language(s) of theories.

When language of both theories is same, then theory T_0 is as strong as another theory T_1 if T_0 is a subset of T_1. A special case when signature of smaller theory coincides with signatures of bigger theory that is referred as conservative extension. The notion of conservative extension is now widely used for comparing ontologies.

Definition 2-16 (Conservative Extension): Let T_0 and T_1 are first-order theories s.t $sig(T_0) \subseteq sig(T_1)$, T_2 is conservative extension of T_0 iff for any $\sigma \in \mathcal{L}(T_0)$, $T_1 \vDash \sigma$ iff $T_0 \vDash \sigma$.

Conservative extension means that T_1 does not add any information about the sentences expressed in $\mathcal{L}_0$. Above definition is based on deduction and it is often referred as 'Deductive Conservative Extension'.

Conservative extension is also defined on models by (Lutz et al., 2007) and is named as model conservative extension.

Definition 2-17 (Model Conservative Extension): Let T_0 and T_1 be two theories with signature S in language $\mathcal{L}$. T_1 is a model S −conservative extension of T_0 if for every model I of T_0, there exists a model $\mathcal{I}$ of T_1 such tat $I|_S = \mathcal{I}|_S$. T_1 is *model conservative extension* of T_0 if T_1 is a model S −conservative extension of T_0 for signature $S = \Sigma(T_0)$.

Model conservative extension is stronger notion than deductive conservative extension. If T_1 is model conservative extension of T_0, then it is clearly deductive conservative extension of T_1. However, converse does not hold. It is explained by (Lutz et al., 2007) as suppose there are two theories T_0 and T_1 with $T_0 = \{\exists r.\top \sqcap \exists s.\top \equiv \top\}$, $T_1 = \{\exists r.A \sqcap \exists s.\neg A \equiv \top\}$; one can easily note that $T_0 \cup T_1$ is the deductive conservative extension of T_0 when the language is $\mathcal{ALC}$, but it is not a model conservative extension of T_0.

When signatures of theories are not same, then there may exist a way in which one theory T_0 is translated into another theory T_1 such that members of T_0 are translated as members of T_1. Enderton proposes an approach for checking whether a theory is strong enough as another theory, even if they are represented in different languages and this approach is referred as "*relative interpretation between theories*", *theory interpretation* or *Enderton mapping*. It is defined in (Enderton, 2001) as

Definition 2-18 (Relative interpretation): An interpretation π of theory T_0 with language $\mathcal{L}_0$ into a theory T_1 with language $\mathcal{L}_1$ different from $\mathcal{L}_0$ and T_1 includes equality is a function on the set of parameters of $\mathcal{L}_0$ such that

π assigns to $\forall$ a formula $\pi_\forall$ of $\mathcal{L}_1$ in which at most the variable v_1 occurs free,

such that $T_1 \vDash \exists v_1 \pi_\forall$

π assigns to each n-place relation symbol P a formula π_P of $\mathcal{L}_1$ in which at most the variables $v_1, \dots, v_n$ occur free.

π assigns to each n-place function symbol f a formula π_f of $\mathcal{L}_1$ in which at most the variables $v_1, \dots, v_n$ occur free, such that

$$T_1 \vDash \forall v_1 \dots \forall v_n \Big(\pi_\forall(v_1) \to \cdots \to \pi_\forall(v_n) \\ \to \exists x \Big(\pi_\forall(x) \land \forall v_{n+1}\big(\pi_f(v_1 \dots v_{n+1}) \leftrightarrow v_{n+1} = x\big)\Big)\Big)$$

In other words, π_f defines in such a way that its members are in $\pi_\forall$.

For any atomic sentence σ with predicate P in signature of $\mathcal{L}_0$, $\pi(\sigma) = \pi(P)$

For any sentence σ in signature of $\mathcal{L}_0$, $\pi(\neg\sigma) = \neg\big(\pi(\sigma)\big)$

For any sentence σ in signature of $\mathcal{L}_0$, $\pi(\sigma \to \tau) = \pi(\sigma) \to \pi(\tau)$

For any sentence σ in signature of $\mathcal{L}_0$, $\pi(\forall x\sigma) = \forall x \pi_\forall \to \pi(\sigma)$

For any sentence σ in signature of $\mathcal{L}_0$, $T_0 \vDash \sigma \Longrightarrow T_1 \vDash \pi(\sigma)$

Thus, the mapping π is an interpretation of T_0 and T_0 is interpretable in T_1. If $\mathcal{L}_0$ coincides with $\mathcal{L}_1$, trivially π is the identity interpretation.

Structure of T_1 for $\mathcal{L}_0$ can be extracted. Suppose $\mathfrak{B}$ be a model of T_1, one can extract from $\mathfrak{B}$ a structure ${}^{\pi}\mathfrak{B}$ for $\mathcal{L}_0$ as

$|{}^{\pi}\mathfrak{B}|$ = the set defined in $\mathfrak{B}$ by $\pi_\forall$

$P^{{}^{\pi}\mathfrak{B}}$ = the relation defined in $\mathfrak{B}$ by π_P, restricted to $|{}^{\pi}\mathfrak{B}|$,

$f^{\,\pi\mathfrak{B}}(a_1, \dots a_n) =$ the unique b such that $\vDash_{\mathfrak{B}} \pi_f[a_1, \dots a_n, b]$ where $a_1, \dots a_n$ are in $|\,{}^{\pi}\mathfrak{B}|$.

Also set $\pi^{-1}[T_1]$ of $\mathcal{L}_0$-sentences can be defined by the equation

$$\pi^{-1}[T_1] = Th\{\,{}^{\pi}\mathfrak{B} | \mathfrak{B} \in ModT_1\}$$

$=\{\sigma|\sigma$ is an $\mathcal{L}_0$-sentence true in every structure ${}^{\pi}\mathfrak{B}$ obtainable from a model $\mathfrak{B}$ of $T_1\}$

$\pi^{-1}[T_1]$ is a satisfiable theory iff T_1 is a satisfiable.

For every formula φ of $\mathcal{L}_0$, there is a formula φ^{π} in $\mathcal{L}_1$, which corresponds exactly to φ.

Gruninger and colleagues show in (Grüninger et al., 2012) that theory T_0 is interpretable in theory T_1 iff there exists a set of translation definitions Δ for T_0 into T_1 such that

$$T_0 \cup \Delta \vDash T_1$$

Ontologies are also treated as logical theories. Enderton approach of interpretation between theories describes that one theory is interpretable in another theory. Following are the main cases in which one theory is interpretable in another theory.

Case 1: When one theory is more general than other theory as $T_0 \subseteq \pi^{-1}[T_1]$, i.e., there is a situation where σ is a sentence of Theory T_0, i.e., $\sigma \in T_0$ and $\sigma \in T_0 \Rightarrow \sigma^{\pi} \in T_1$

Case 2: When one theory is faithfully interpretable in another theory as

Definition 2-19 (Faithful Interpretation): An interpretation π of a theory T_0 into a theory T_1 is called faithful iff $T_0 \subseteq \pi^{-1}[T_1]$, i.e., $\sigma \in T_0 \Leftrightarrow \sigma^{\pi} \in T_1$

Case 3: When two theories are logical equivalent.

Definition 2-20 (Logically equivalent): Two theories T_0 and T_1 are are logically equivalent iff Theory T_0 is interpretable in theory T_1 and theory T_1 is interpretable in theory T_0.

Case 4: When two theories are logically synonymous (Gruninger & Aameri, 2014).

Definition 2-21 (Definitional Extension): Let us consider an ontology O with langague L, while C an artifact not in L. then if there exists another ontology O' resulting from L by adding the new artifact C to L. O' is *definitional extension* of O if it result from adding to O an axiom of the form

$$O(p_1, \dots, p_n) \leftrightarrow A$$

where A is a formula of L containing no variable other than $p_1, \dots p_n$

Definition 2-22 (Logically synonymous): Two theories T_0 and T_1 are logically synonymous iff there exists a third theory T with signature $\Sigma(T_0) \cup \Sigma(T_1)$ that is a definitional extension of T_0. Logical synonymy implies logical equivalence of theories.

If Δ is a translation definition for T_0 into T_1 and Π is a translation definition for T_0 into T_1 and $T_0 \cup \Delta \vDash T_1$ and $T_1 \cup \Pi \vDash T_2$, then $T_0 \cup \Delta \cup \Pi$ is a definitional extension of T_0.

In Enderton approach of mapping, only equivalent relation between terms of theories can be expressed and it is not possible to express subsumption, overlap relation.

Combination of theories and mapping is expressed as $f(T_0, T_1, \Delta)$ and it is considered here a new theory. This new resultant theory can be inconsistent.

Example 2-11: Suppose that, there are two theories T_0 and T_1. $T_0 = \{\sigma \in \mathcal{L}_0 | T_0 \vDash \sigma\}$ and $T_1 = \{\tau \in \mathcal{L}_1 | T_1 \vDash \tau\}$ and there are mappings $\Delta: T_0 \mapsto T_1$ and $\Pi: T_1 \mapsto T_0$. $f(T_0, T_1, \Delta)$ and $f'(T_0, T_1, \Pi)$ are two functions that create new theories T and T' respectively and they are defined as

$$f(T_0, T_1, \Delta) = \{\Delta(\sigma) \in \mathcal{L}_1 \,| \forall M \vDash T_2 \; \exists M \vDash \Delta(\sigma)\}$$

and

$$f(T_0, T_1, \Pi) = \{\Pi(\tau) \in \mathcal{L}_0 \,| \forall M \vDash T_1 \; \exists M \vDash \Pi(\tau)\}$$

$f(T_0, T_1, \Delta)$ and $f(T_0, T_1, \Pi)$ can be inconsistent.

Interpretation between theories is also used in ontology integration. Schorlemmer and colleagues (Schorlemmer & Kalfoglou, 2008) use relative interpretation between theories for semantic integration of theories and define semantic integration of theories as

Definition 2-23 (Semantic integration of theories): Let T_1,T_2 and T be first-order theories and T_1 and T_2 are semantically integrated with respect to T , if there exist *theory interpretations* $\alpha_1: T_1 \to T$ and $\alpha_2: T_2 \to T$; there exist structure reducts $\beta_1: Mod(T) \to Mod(T_1)$ and $\beta_2: Mod(T) \to Mod(T_2)$ and $Mod(T) \neq \emptyset$. Semantic Integration I of local theories T_1, T_2 w.r.t. T is $I = \{\alpha_i: T_i \to T;\ \beta_i: Mod(T) \to Mod(T_i)\}_{i=1,2}$

A priori knowledge of inconsistency is required to avoid the situation when semantic integration of theories is inconsistent. This shows that it requires a-priori information about mappings that they do not cause inconsistencies. In Chapter 4, we will show that there exist some other defects and they can cause problems or give undesired results. User should have the information about mappings that whether there exist some defects in them or not before using them in applications.

2.2 Synthesis

It is evident from the previous section that there exist several types of different mapping languages. Mappings are interpreted and treated differently in various formalisms and it is not necessary that these languages define mappings in explicit sense. In languages like DFOL, DDL, $\mathcal{E}$ −connections, mapping is treated as constraints that limits the possible distribution models, while mapping is treated as interpretation between theories in Enderton approach where mapping is used to translate one theory into another theory. So comparing different mappings having precise semantics is not a trivial task and it requires either there exists a universal semantic of mappings, which is not the case since mappings are formalized in different formalism so they have different semantics about mappings, or an approach independent of any precise semantics. Our proposed approach in this thesis falls in later case.

Mapping between ontology can be one of these type (i) symbol to symbol mapping (ii) symbol to formula (iii) formula to symbol iv) formula to formula (v) or any combination of (i), (ii), (iii), and (iv). DFOL mapping maps formula of one ontology to another ontology, DDL maps allows all five types, whereas in interpretation between theory mapping is carried out by mapping symbols of one ontology to formulas of another ontology. A translation of logical symbol performed by standard translation of logical symbols of one theory into logical symbols of another theory, whereas translation of non-logical symbols is not a trivial task. Symbol to Symbol translation of ontology i into ontology j is performed by finding a symbol of theory j which can interpret symbol of ontology i, but sometimes there is no symbol in j which can interpret one or more symbols of i then symbols to formula translation is required. A formula of ontology j is then used to map symbol of ontology i. All these types of mapping can be converted to (i); there exists several ways by which translation of symbols to formula can be reduced to translation as symbols to symbols, generally this is performed by definitional extension. When one ontology (theory) has a symbol which is a formula (inference) in other ontology then the symbol of one ontology is said to be definable in another ontology. In definitional extension, a ontology j is extended by adding new terms to the language L_i which can interpret symbols of ontology i in such a way that these added terms represent definitioin of one of the formula of j.

From the mapping languages presented in this chapter, we have identified four kinds of mapping operators and they are presented inTable 2-1.

Table 2-1. Mapping relation for expressing Ontology Mappings

Mapping operators	Description
Equal function	There is only one interpretation domain for source ontologies. Mapping relates artifacts of ontologies in this interpretation domain. It is used in OWL.
Equalizing function	There is a global interpretation domain. All the local domains are moved to global interpretation domain by *equalizing function*, this function also relates (mappings) the elements of local domains in the global domain. It is used in IDDL.
Domain relation	Local interpretation domains are related by *domain relation ,* which is binary. A domain relation r_{ij} relates domain i to domain j and it is not necessary that $r_{ij} \neq rji$. It is used in DDL, DFOL
Link	The interpretation domains are connected by link. This link is n −ary. The mapping is expressed by *link called* ϵ −*connection*. It is used in $\mathcal{E}$ −connection.
Equivalence	Enderton approach allows mapping is represented as equivalence relation such that $\vDash_{\mathfrak{B}} \sigma^{\pi} iff \vDash_{\pi_{\mathfrak{B}}} \sigma$

We have not found any existing language that covers all aspect of mappings. Indeed, research community accepts the diversification of mapping languages, since the semantics of global ontology and contextual ontologies are not same. Hitzler and colleagues compare different ontology mapping languages and try to map them in DFOL, but there are some mappings (for instance, artifacts that are mapped by overlapping relation as in DDL) that are not expressible in DFOL (Hitzler et al., 2006).

There does not exist a unique way to map ontologies as there exist different mapping formalisms and operators that have different semantics for ontology mappings. We summarize these approaches in Table 2-2 based on the following features.

- **Interpretation domains:** Every ontology has its own interpretation domains. After establishing ontology mappings, semantics of mappings are based on either combined or individual interpretation domains.
- **Expressivity:** Expressivity of mappings are : a) mapping between concepts (C-C) b) mapping between roles (R-R), c) mapping between concept and role (C-R) d) mapping

between role and concept (R-C) e) partial individual mappings by ∈ relation (O-O) f) complete individual mappings by = relation (O-O).

- **Mapping relations:** Ontology mappings do not only express equality between artifacts of ontology but also there exist other relations used in relating artifacts of ontologies. Logic based mapping languages use relations $\sqsubseteq$ (subsumption), $\sqsupseteq$ (supersumption), $\equiv$ (equality), $\perp$ (disjointness), $\between$ (overlap), $\in$ (membership relation), $=$ (individual eqals to), where $\between$ (overlap) is equal to $\neg(\perp)$ (not disjoint). Logic based mapping languages may not have the same symbol for expressing the same relation between artifacts of ontologies, but they have the expressivity to represent these relations using some other symbol. For instance, the syntax of DFOL does not have $\sqsubseteq$ symbol, but DFOL has the ability to represent this relation by $\rightarrow$ (implication relation). In Table 2-2, we use relations for showing expressivity of the language and not the relation supported by the syntax of the languages.
- **Symmetric and transitive property:** Some ontology mapping languages allow symmetric and/or transitive property while others do not.
- **Local unsatisfiability:** Whether the unsatisfiability present in one ontology make the other ontology unsatisfiable or not?
- **Reasoning with mapping:** Whether mappings allows to reason in mapping language or it is just declarative? What existing reasoning services available for these languages.
- **Weights or confidence:** Whether the mappings approaches can express confidence or weights about the mappings. For instance, two concepts are similar to each other and having similarity equal to 75% based on the number of instances they cover.
- **Heterogeneity of languages:** Which languages are allowed in the logic based mapping languages.

Table 2-2. Features of Ontology Mapping Languages

	Interpretation domain	**Expressivity**	**Mapping relations**	**Symmetric, Transitive**
OWL	1 Each ontology interpretation domain is merged into global interpretation domain	C-C, R-R (formulas to formulas)	$\sqsubseteq, \sqsupseteq, \equiv, \perp$	yes
DFOL	n Each ontology has its own interpretation domain, they are disjoint	C-C, R-R, C-R, R-C, O-O (formulas to formulas)	$\sqsubseteq, \sqsupseteq, \equiv, \perp$, $\between, \in, =$	Generally no
DDL/C-OWL	n Each ontology has its own interpretation domain that are disjoint	C-C, R-R, C-R, R-C, O-O (formulas to formulas)	$\sqsubseteq, \sqsupseteq, \equiv, \perp$, $\between, \in, =$	Generally no
OIS	1 Local domains are embedded in global domain	C-C, R-R, C-R, R-C (formulas to formulas)	$\sqsubseteq, \sqsupseteq, \equiv, \between$	Generally no
E-conn	n Each ontology has its own interpretation domain, they are disjoint	C-C, O-O (formulas to formulas)	$\sqsubseteq, \sqsupseteq, \equiv, \perp$, $\in, =$	Generally no
IDDL	1 Each ontology interpretation domain is merged into global interpretation domain	C-C, R-R (formulas to formulas)	$\sqsubseteq, \sqsupseteq, \equiv, \perp$, $=$	yes
Weighted mappings (Atencia et al., 2012)	n Each ontology has its own interpretation domain that are disjoint	C-C, R-R (formulas to formulas)	$\sqsubseteq, \sqsupseteq, \equiv, \perp$	Generally no
Enderton's approach	One's ontology interpretation domain is embedded into more general ontology's interpretation domain	symbols to formulas	$\equiv$	N/A

	Local inconsistency propagation (if ontology O_i is inconsistent then distributed system	Automated reasoning tools	Weight or confidenc e	Heterogeneous (different Languages)
OWL	becomes inconsistent and ontology O_j also becomes inconsistent in the distributed system	OWL reasoners such as Pellet, HermiT, Fact++, RacerPro, KAON2	No	Only OWL
DFOL	does not become inconsistent due to the DFOL multi-model epistemic semantics but other ontology O_j becomes inconsistent only if there is a domain relation r_{ij} but domain relation r_{ji} does not affect ontology O_j	Not available	No	Only FOL
DDL/C-OWL	does not become inconsistent due to the semantics of DDL which allows holes to deal with that, however, ontology O_j may become inconsistent only if there is a domain relation r_{ij} but domain relation r_{ji} does not affect ontology O_j	Yes (Ghidini, 2008), 'DRAGO'	No	Only DL
OIS	become inconsistent	N/A	No	Generally used for mapping involving relational schema, XML schema. mappings are mostly expressed in DataLog

	Local inconsistency propagation (if ontology O_i is inconsistent then distributed system	**Automated reasoning tools**	**Weight or confidenc e**	**Heterogeneous (different Languages)**
E-conn	become inconsistent, while other ontology O_j becomes inconsistent only if there is link from inconsistent ontology to O_j	No complete reasoning support, earlier version of 'Pellet' provides small reasoning support for E-conn	No	ADS languages
IDDL	becomes inconsistent	Yes (Chan, et al., 2013), 'DRAOn'	No	DL
Weighted mappings (Atencia et al., 2012)	does not become inconsistent, while other ontology O_j becomes inconsistent only if there is a domain relation r_{ij} but domain relation r_{ji} does not affect ontology O_j	No	Yes, created especially for dealing with weights	DL
Enderton's approach	N/A if one of the ontology is inconsistent then mapping is not possible	No	No	FOL

Ontology mapping languages and approaches described in this chapter express certain aspects of mappings and there does not exist a mapping language that can express all types of mappings and mapping semantics. Indeed, certain aspects of mappings are not ordered. Hence, when mapping approach subscribe to specific aspects of mapping other aspects cannot be covered by that mapping approach.

We conclude this chapter by listing the following four points about Logic-based ontology mappings.

1. Heterogeneity is poorly taken into account, but heterogeneity can be implemented by translation. This translation may not express certain kind of mappings. For instance, DFOL cannot express mappings of artifacts that are related by overlapping relation.
2. Except the case of Enderton, mappings are used to specify additional constraints or knowledge between ontologies. This may be a problem if a mapping should represent how original mapping of an artifact of one ontology is represented in different ontology. Indeed, in Enderton mapping, specific properties that are sometimes referred as principle of conservativty in Literature should be respected.
3. Same syntax of ontology mappings does not have the same meaning in different logic-based ontology mapping formalism.
4. The notion of inconsistency widely used and implemented in different approach is not universal.

Chapter 3.

Ontology Mapping in Algebraic Approaches

Algebraic approaches study the structure of arbitrary set and the operations defined on that set. Category theory (Eilenberg & Lane., 1945), is an algebraic approach and computer scientists are interested in it for its generality. It is a basic conceptual and a notational framework like set theory and graph theory, but with more abstraction. Goguen explains how 'category theory' can be useful for fields involving mathematical structure by presenting seven dogmas (guidelines) about basic concepts of category theory: category, functor, natural transformation, limit, adjoint, colimit and comma category (Goguen, 1991). He emphasizes that each mathematical structure corresponds to a category.

Category theory has been effectively used in the field of computer science for last two decades. It is used in different domains of computer science such as specification languages, programming languages, and ontologies. In the field of ontology, category theory is used especially in the context of ontology merging (Jannink et al., 1998), (Bench-Capon & Malcolm, 1999), interoperability with the help of ontologies (Michael & Dampney, 2001), (Cafezeiro & Haeusler, 2007), and modularization (Grüninger et al., 2010), (Kutz et al., 2010). Category theory is useful in the context of ontologies, since it provides many benefits including generality in specifying ontologies and ontology mappings and the composition operation is available for composing ontology mappings.

Researchers also use Category theory while dealing with tasks related to ontology mapping. Other approaches such as Institution theory (Goguen, 1984) and Information flow (Kent., 2001) are based on category theory. Institution theory (Goguen, 1984) is based on Category theory and incorporate logic with emphasis on truth is invariant from the choice of logic. Institutions are proposed for generalizing the logical systems, but now it is also used for translating different ontologies and ontology languages to a generalized ontology and ontology language DOL (Mossakowski & Kutz, 2011), (Lange et al., 2012).

This chapter is organized as follows: Section 3.1and Section 3.2 discuss how ontologies are formalized in category theory and Institution theory, respectively, describe how ontology based operations are defined in these theories, What are the limitations of using these theories, Are these theories useful in solving the problem of checking whether two ontology mappings are contradicting or not; and finally Section 3.3 presents a synthesis of this chapter.

3.1 Ontology Mapping in Category Theory

In this section, we present basic concepts of Category theory and the state of the art of category theory in the field of mappings.

3.1.1 Category Theory Fundamentals

In this section, we present definitions, adapted from (Adámek et al., 1990).

Definition 3-1 (Category): A category is a quadruple A $= (\mathcal{O}, hom, id, \circ)$ consisting of

1. a class $\mathcal{O}$, whose members are called A-*objects,*
2. for each pair (A, B) of A-objects, a set $hom(A, B)$, whose members are called A-*morphisms* from A to B —expressed as by using arrows; $A \xrightarrow{f} B$ is a morphism,
3. For each A-object A, a morphism $A \xrightarrow{id_A} A$, called the A-*identity* on A,
4. a *composition law* associating with each A-morphism $A \xrightarrow{f} B$ and each A-morphism $B \xrightarrow{g} C$ an A-morphism $A \xrightarrow{g \circ f} C$, called the composite of f and g, subject to the following conditions:
   a. composition is associative, i.e., for morphisms $A \xrightarrow{f} B$, $B \xrightarrow{g} C$, and $C \xrightarrow{h} D$, the equation $h \circ (g \circ f) = (h \circ g) \circ f$ holds,
   b. A-identities act as identities with respect to composition, i.e., for A-morphisms $A \xrightarrow{f} B$, and there are $id_B \circ f = f$ and $f \circ id_A = f$,
   c. the sets $hom(A, B)$ are pairwise disjoint.

Different from Set theory that mainly focuses on objects, category theory mainly focuses on relations between objects.

Examples of Category Theory are

Example 3-1: The category of sets, Set; has sets as *objects* and functions as *morphisms* and it respects composition and identity property.

Example 3-2: A poset (partially ordered set) is a pair $(X, \leq)$ that consists of a set X and a transitive, reflexive, and antisymmetric relation $\leq$ on X. A Poset P can be viewed as a category whose *objects* are the elements of P; and for any $x, y \in P$ satisfying $x \leq y$, there is a unique *morphism* that has x as source and y as target.

Definition 3-2 (Dual Category): For any category $A = (\mathcal{O}, hom, id, \circ)$ the dual (or opposite) category of A is the category $A^{op} = (O, \mathrm{hom}_{\mathrm{A}}^{\mathrm{op}}, id, \circ^{op})$, where $hom_A(A, B) = hom_A(B, A)$ and $f \circ^{op} g = g \circ f$. The category and its dual have same objects and morphisms but the direction of morphism is in opposite directions.

Definition 3-3 (isomorphism in Category): A Category morphism $f: A \to B$ is an isomorphism if there exists a morphism $f^{-1} : B \to A$ such that $f^{-1} \circ f = id_A$ and $f \circ f^{-1} = id_B$. The morphism f^{-1} is called the inverse of f; and objects A and B are called isomorphic.

Definition 3-4 (Functor): If A and B are categories, then a functor F from A to B is a function that assigns to each A-object A a B-object $F(A)$, and to each A-morphism $A: \xrightarrow{f} A'$ a B-morphism $F(A): \xrightarrow{F(f)} F(A')$ in such a way that

1. F preserves composition, i.e., $F(f \circ g) = F(f) \circ F(g)$ whenever $f \circ g$ is defined, and
2. F preserves identity morphisms, i.e., $F(id_A) = id_{F(A)}$ for each A-object A.

Definition 3-5 (Diagram): A diagram is a collection of objects and morphisms, indexed by a fixed category; or in other words a diagram is a functor from a fixed index category to some category.

Definition 3-6 (Commutative diagram): A diagram is said to be a commutative diagram if and only if m is the composite of f, g as $m = f \circ g$ shown in the Figure 3-1.

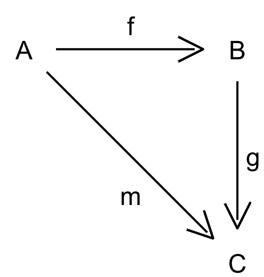

Figure 3-1. Categorical commutative diagram

Definition 3-7 (Cone): Let $F: J \to C$ be a diagram in C and N be an object of C and a *cone* from N to F is a family of morphisms $\psi_X: N \to F(X)$ for each object X of J such that for every morphism $f: X \to Y$ in J and there is $F(f) \circ \psi_X = \psi_Y$, as shown in Figure 3-2 the diagram commutes.

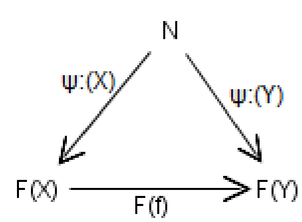

Figure 3-2. Categorical Cone

Definition 3-8 (Cocone): Cocone is a categorical dual of cone.

Definition 3-9 (Limit): A limit of the diagram $F: J \rightarrow C$ is a cone (L, φ) to F such that for any other cone (N, ψ) to F there exists a unique morphism $u: N \rightarrow L$ such that $\varphi_X \circ u = \psi_X$ for all X in J as shown in Figure 3-3.

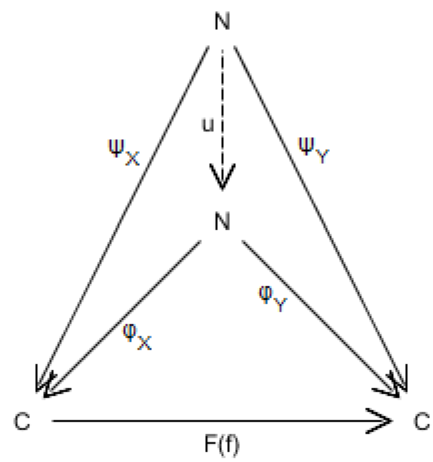

Figure 3-3. Categorical Limit

Definition 3-10 (CoLimit): Colimit is a categorical dual of Limit.

Definition 3-11 (Pushout): A pushout of a pair of Categorical-morphisms $f: C \rightarrow A$ and $g: C \rightarrow B$ is an object $S \in |C|$ together with a pair of morphisms $h: A \rightarrow S$ and $i: B \rightarrow S$ such that: $h \circ f = i \circ g$ for any $S' \in |\mathcal{C}|$ and pair of morphisms $h': A \rightarrow S'$ and $i': B \rightarrow S'$ satisfying $j \circ h = h'$ and $j \circ i = i'$, there is a unique morphism $j: S \rightarrow S'$ such that the diagram in Figure 3-4 commutes.

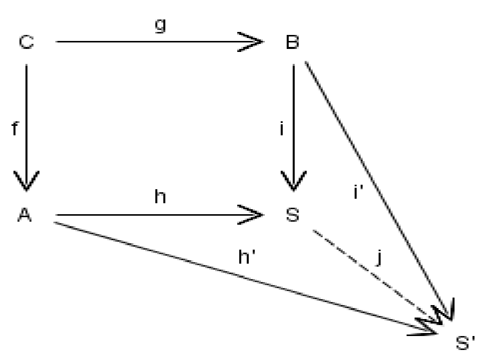

Figure 3-4. Categorical Pushout

Pushouts are the special case of colimits consisting of pair of morphisms with a common domain.

Definition 3-12 (Pullback): Pullback is the categorical dual of the pushout.

For more details about category theory concepts, (Adámek et al., 1990), (Saunders, 1998) are good references.

3.1.2 Ontology mappings and Category Theory

Algebraic approaches are used for ontology mapping based operations like ontology merging, translations and others. The basic idea behind these approaches is to use a more expressive language for representing ontology mappings and to use the operations of a mathematical framework for ontology related operations.

In the state of the art of use of ontology mapping with category theory, ontologies are used as objects of a category. Ontology mapping is also treated as another ontology and an object of a category. It is shown in Figure 3-5, where O_A and O_B are ontologies and M_{AB} is the mapping between O_A and O_B ontologies are objects of a category and morphism $f_A: M_{AB} \rightarrow O_A$ and $f_B: M_{AB} \rightarrow O_B$. Axioms of ontology are within the ontology; an object of category while morphisms are structure preserving, i.e., homorphism form one object to another. So if there is a morphism from one ontology to another it should preserve structure of ontology. For preserving axioms of ontologies, morphisms are defined in such a way that axioms of ontologies are preserved in categorical operation.

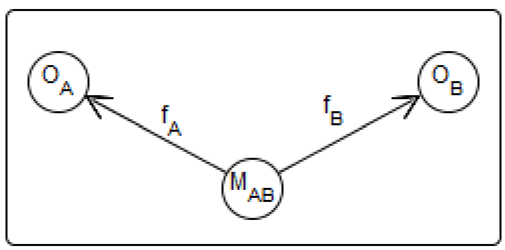

Figure 3-5. Category of ontologies and ontology mappings

Several researchers use category theory while dealing with ontology merging. Some of these approaches are discussed in this section.

Jannink and colleagues use category theory for defining translation and combination operations (Jannink et al., 1998). They use pullback for extracting information and product for combining information of ontologies. Michael and colleagues propose that category theory can be used as meta-ontology for the structural aspects of semantic web applications (Michael & Dampney, 2001). Category theory has been proposed as a foundation ontology for the IEEE Standard Upper Ontology in the Information Flow Framework (Kent., 2001).

Krötzsch and colleagues describe how to use category theory while dealing with ontology merging. They treat ontologies as categories by apply restrictions on ontology and ontology mapping is directional and the composition of ontology is associative in nature and there exists identity relationship to ontology itself (Krötzsch et al., 2005). They show how categorical co-product and pushout operations for merging ontologies can be performed.

Hitzler and colleagues explain the use of categorical theory based approach for ontology merging. They use categorical 'Pushouts' (Hitzler et al., 2005).

An example is presented below to show how ontologies and ontology mapping are used in category theory.

Example 3-3: Ontology Merging using category theory approach

Ontologies and ontology mappings are objects, of a category. Ontology merging is carried out by pushout. This is depicted in Figure 3-6. Ontology mapping is M_{AB} = {(University, Research Org.,≡), (Teaching Faculty, Research Staff,≡), (Director, Admin. Staff,≡), (Researcher, Research Officer,≡), (Administrative Staff, Director Admin,≡)}. We abbreviate subsumption relation between artifacts of ontologies and they are labeled in Figure 3-6 by morphisms f, g, h, i in Ontology merging by Categorical Pushout as

$f(University, Research\ Org., \equiv) \mapsto University$

$f(Teaching\ Faculty, Research\ Staff, \equiv) \mapsto Teaching\ Faculty$

$f(Director, Admin. Staff, \equiv) \mapsto Director$

f(Researcher, Research Officer, ≡) ↦ Reasearcher

$f(Administrative\ Staff, Director\ Admin, \equiv) \mapsto Administrative\ Staff$

$f(ut) \mapsto utr$

$f(ua) \mapsto udaad$

$f(tr) \mapsto trc \circ crr$

$f(ad) \mapsto udaad$

$f(da) \mapsto udaad$

$g(rr) \mapsto utr$

$g(ra) \mapsto udaad$

$g(rc) \mapsto trc$

$g(rs) \mapsto trs$

$g(cr) \mapsto crr$

$g(sr) \mapsto srrr$

$g(rad) \mapsto daada$

$g(University, Research\ Org., \equiv) \mapsto Research\ Org.$

$g(Teaching\ Faculty, Research\ Staff, \equiv) \mapsto Research\ Staff$

$g(Director, Admin. Staff, \equiv) \mapsto Admin. Staff$

$g(Researcher, Research\ Officer, \equiv) \mapsto Research\ Officer$

$g(Administrative\ Staff, Director\ Admin, \equiv) \mapsto Director\ Admin$ $h(University) \mapsto University, Research\ Org.$

$h(Teaching\ Faculty) \mapsto Teaching\ Faculty, Research\ Staff$

$h(Administrative\ Staff) \mapsto Director, Admin. Staff, Administrative\ Staff,$

$Director\ Admin$

$h(Researcher) \mapsto Researcher, Research\ Officer$

$h(Director) \mapsto Director, Admin. Staff, Administrative\ Staff,$

$Director\ Admin$

$h(Asst. Director) \mapsto Asst. Director$

$i(Research\ Org.) \mapsto University, Research\ Org.$

$i(Research\ Staff) \mapsto Teaching\ Faculty, Research\ Staff$

$i(Admin. Staff) \mapsto Director, Admin. Staff, Administrative\ Staff,$

$Director\ Admin$

$i(Research\ Officer) \mapsto Researcher, Research\ Officer$

$i(Director\ Admin) \mapsto Director, Admin. Staff, Administrative\ Staff,$

$Director\ Admin$

$i(Computer\ Scientist) \mapsto Computer\ Scient$

$i(Social\ Scientist) \mapsto Social\ Scient$

$i(Research\ Officer) \mapsto Researcher, Research\ Officer$

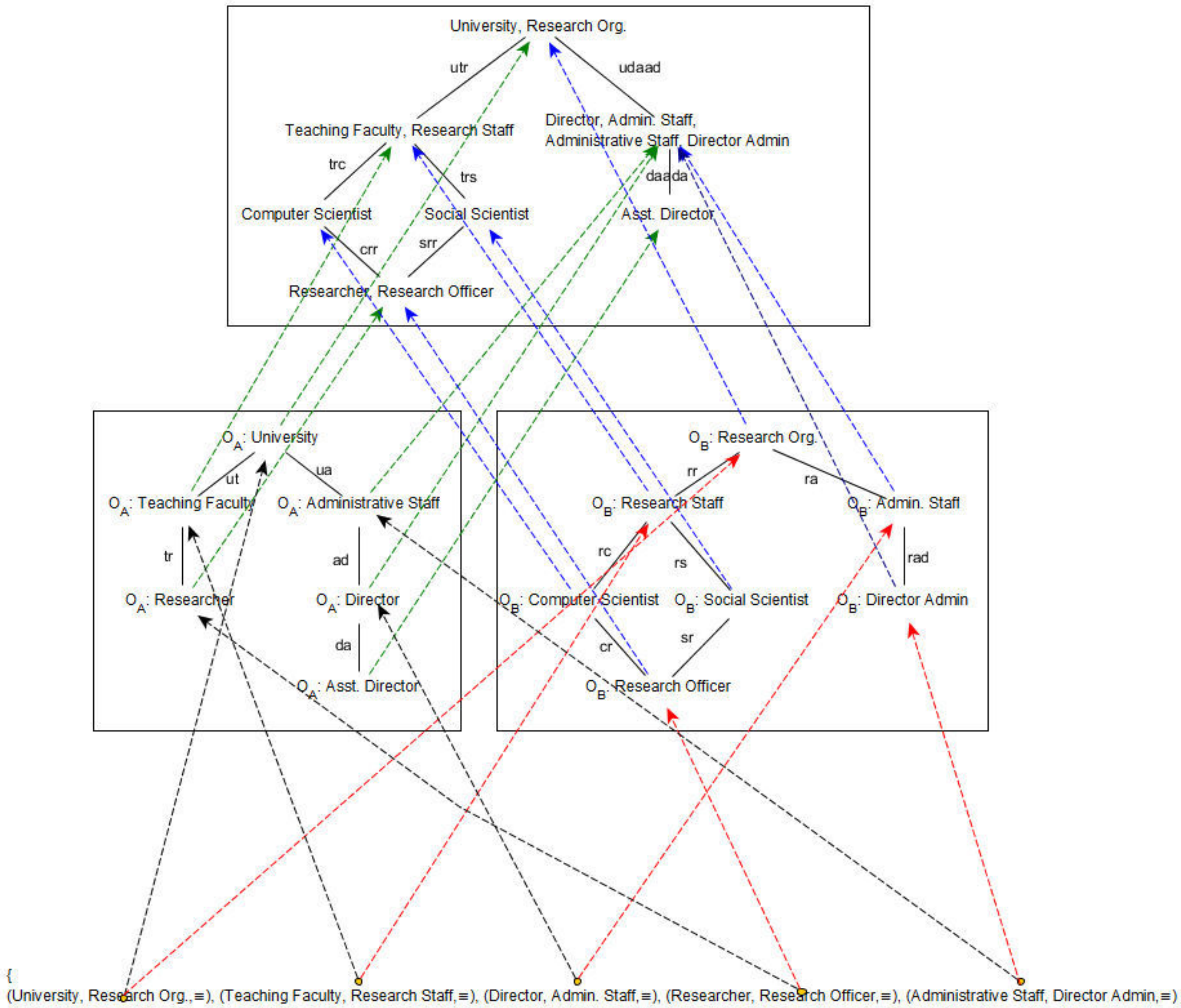

Figure 3-6. Ontology merging by Categorical Pushout

Mapped artifacts are renamed in the merged ontology, subsumption relation between artifacts in the merged ontology is derived from source ontologies.

Mapped concepts in the resulting merged ontology is preserved by pushout condition of $h \circ f = i \circ g$ condition, there are many objects satisfying this condition, so excessive identification is prevented by requiring the existence of unique morphism j. Irrelevant information in the merged ontology is prevented by imposing pushout condition of $j \circ h = h'$ and $j \circ i = i'$. Part of the merged operation involving j is not shown in Figure 3-6.

3.1.2.1 Compatibility of ontologies

Bench-Capon and colleagues present an approach of using category theory for ontology specification and relating ontologies. They use signature and also models (Bench-Capon &

Malcolm, 1999). They treat ontology as a pair $\langle \Sigma, Ax \rangle$; Σ as signature and Ax as axioms of ontology. A signature morphism is defined in terms of morphism of data, type and attributes of two ontologies. A signature morphism induces a functor between category of models of two ontologies. They use ontology morphism for relating ontologies. They define ontology morphism as $\chi: O_1 \rightarrow O_2$ as $\chi: \langle \Sigma, Ax \rangle \rightarrow \langle \Sigma', Ax' \rangle$, i.e., $\chi: \Sigma \rightarrow \Sigma'$ s.t. $\chi M \vDash Ax$ when $M \vDash Ax'$. In other words, ontology morphism is translation of signatures such that axioms of the translated ontology are respected after translation. And defines a *relation* between ontologies O and O_i as a pair of morphisms $\chi_i: O \rightarrow O_i$ for $i = 1,2$.

A morphism $\chi : O_1 \rightarrow O_2$ is a special case of a relation where $O = O_1$.

They define the compatibility between ontologies as:

Let $\chi_i : O \rightarrow O_i$ for $i = 1,2$ be a relation between ontologies. O_1 and O_2 are *compatible* (over O) iff their colimit is consistent. Here consistency refers to logical consistency.

This definition depends upon colimit and pushout, since in some cases colimit and pushout do not exist. In addition, the coproducts in category theory are not unique in general. Therefore, it is possible that either there is no or more than one colimit. Whenever colimit exists, then it is required to perform another step of checking of the consistency of colimit. In this approach, ontology mappings are treated as ontology. However, for checking consistency, it requires logical approach because inconsistency is defined in logical terms. Since there can be more than one colimit, so there is no guarantee that ontologies and ontology mappings are logically consistent in general. Only it can be said that ontologies and ontology mappings are logical consistent for a particular resultant colimit when resultant colimit is logical consistent.

A category is considered as ***'co-complete'*** if all of its colimits exist. In the case of a category of partially ordered set (poset), it is cocomplete if and only if it is a lattice (Adámek et al., 1990).

Cases when there exists no or more than one pushout are described below.

Case: (No colimit)

An ontology can be considered as a partially ordered set, if the ontology is not a lattice, then it does not have colimit. For instance, suppose that there exists an ontology i as shown is in the Figure 3-7, it does not have colimit since it is not a lattice as it does not have supremum since ontology i has axioms $i{:}C \sqsubseteq i{:}A$, $i{:}D \sqsubseteq i{:}B$, $i{:}A \sqsubseteq i{:}D$ and $i{:}B \sqsubseteq i{:}C$.

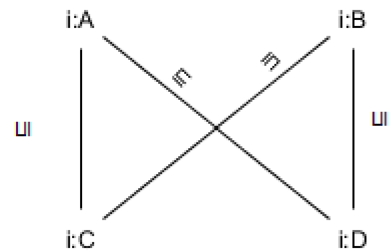

Figure 3-7. Ontology having no colimit

Case: (more than one solution by pushout)

Suppose that there are two ontologies i and j. Ontology i consists of artifacts 1 and 2, while Ontology i consists of artifacts a and b. There is a mapping in which artifact 1 of ontology i is maped to artifact a of ontology B, i.e., $i{:}\,1 \equiv j{:}\,a$.

Using pushout may result in multiple solutions as shown in Figure 3-8.

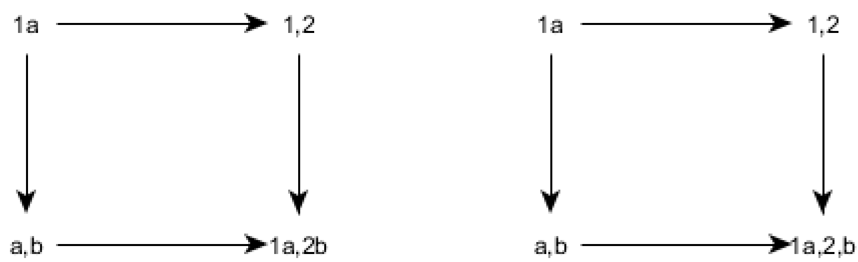

Figure 3-8. more than one solution by pushout

Although there exists only one correspondence in the mapping $1 \equiv a$, but pushout does not refrain us to map $2 \equiv b$ in the final result. In fig (a) $2 \equiv b$ mapping is used in the final result (which has two elements) while in fig (b) $2 \equiv b$ is not used and they are added in the final result (which has three elements) as it is.

3.1.2.2 Algebraic operations of Ontology Mappings

Some authors used the term alignment for mapping and they define some notions which uses the term alignment, so we have used the term mapping and alignment interchangeably in this chapter.

Ontologies are connected by mapping and different types of connection are named according to the shape they form in connection such as V-alignment and W-alignment (Zimmermann et al., 2006), M-alignment (Kutz et al., 2010) and their basic details and characteristics are presented below.

Definition 3-13 (V-alignment) (Zimmermann et al., 2006)**:** O_1, O_2 are ontologies and A is an alignment of O_1, O_2 forms objects of a category. π_1, π_2 are morphisms, where $\pi_1{:}\, A \rightarrow O_1$ and

$\pi_2: A \rightarrow O_2$ as shown in Figure 3-9. It is a basic form of alignment, since there is no reference ontology involved in this form of alignment.

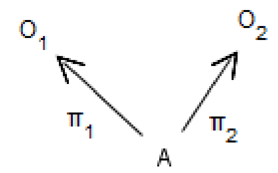

Figure 3-9. V-alignment

Sometimes it is required that merged ontology obtained by categorical pushout operation should have mapped concepts related in some relation such as subsumption. For instance, if a concept 'Woman' in one ontology is subsumed by a concept 'Person' in another ontology, then it is not possible in merged ontology obtained by using pushout operation and V-alignment because of not having any ability to deal with such cases.

One of the basic operation of category theory is composition operation. It is used to compose ontology alignments and obtaining new alignments. If A_1 is alignment between O_1 and O_2 and A_2 is alignment between O_2 and O_3 then by composing A_1 and A_2 then by using categorical pullback operation the resultant alignment A_3 is between O_1 and O_3. Composition of alignments by V-Alignment is shown in Figure 3-10. Similarly, W-alignments and M-Alignments can be composed.

Definition 3-14 (Composition of alignment): (Zimmermann et al., 2006) If there is an alignments A between ontologies O_1 and O_2, and B between O_2 and O_3, then it should be possible to obtain an alignment C of O_1 and O_3. Figure 3-10 shows the composition of two V-alignments $\langle A, \alpha_1, \alpha_2 \rangle$, $\langle B, \beta_2, \beta_3 \rangle$ is $\langle C, \alpha_1 \circ f_A, \beta_2 \circ f_B \rangle$, where $\langle C, f_A, f_B \rangle$ are pullback of α_2 and β_2.

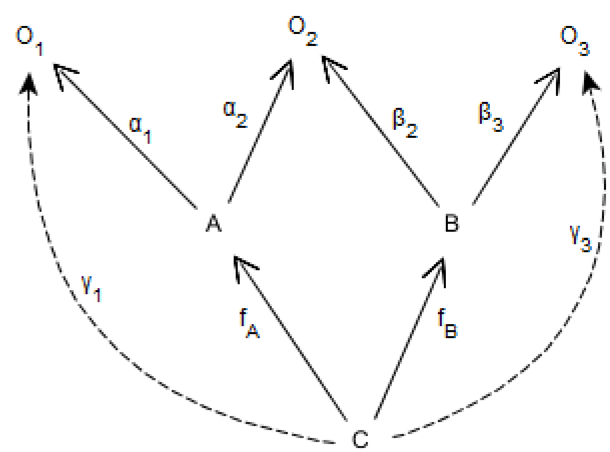

Figure 3-10. Composition by V-Alignment

Definition 3-15 (Intersection of alignment): (Zimmermann et al., 2006) Figure 3-11 shows the diagram of intersected alignments $\langle A, f_1, f_2 \rangle$ and $\langle B, g_1, g_2 \rangle$. Object C together with morphisms k_A, k_B, h_1 and h_2 make the limit of the diagram consisting of the two alignments. The resulting alignment of intersected alignments is $\langle C, h_1, h_2 \rangle$.

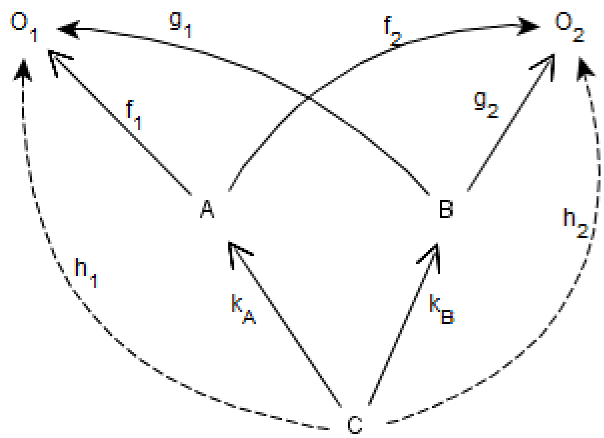

Figure 3-11. Intersection of V-alignments

Definition 3-16 (Union of alignment): (Zimmermann et al., 2006) Union of alignment is defined by using common part (intersection) of alignments. Union is a disjoint union of this common part and non-common part between alignments. In Figure 3-12, Union is achieved by way of a categorical pushout of $\langle k_A, k_B \rangle$. Morphisms u_1 (resp. u_2) is obtained by factorizing f_1 (resp. f_2) through i_A (resp. i_B). So informally, union is the pushout of intersection. Union gathers all asserted relations specified in two alignments.

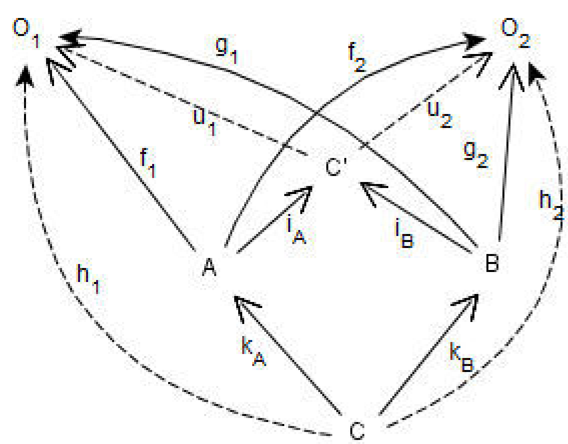

Figure 3-12. Union of V-alignments

Definition 3-17 (W-alignment): (Zimmermann et al., 2006) W-alignment consists of two V-Alignments and an intermediate bridge ontology. O_1, O_2 are ontologies, B is a bridge ontology, and A_1, A_2 are alignments and O_1, O_2 forms objects of a category. $\pi_1, \sigma_1, \sigma_2, \pi_2$ are morphisms, where $\pi_1: A_1 \rightarrow O_1$, $\sigma_1: A_1 \rightarrow B$, $\sigma_2: A_2 \rightarrow B$ and $\pi_2: A_2 \rightarrow O_2$.

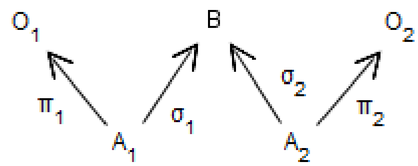

Figure 3-13. W-alignment

W-alignment removes the V-alignment's limitation of relating mapped artifacts in a particular relation. A bridge ontology is used containing the additional information about the desired result of merged ontology such as 'Woman is subsumed by 'Person' and W-alignment are used to achieve the desired merge operation. Merged operation is obtained by successive V-alignments as shown in Figure 3-14. W-alignment solve the limitations of V-alignment but creates some new problems. B is a bridge ontology and it can contain any information that may not be related to neither O_1 nor O_2, hence, this makes loose coupling between ontologies O_1 and O_2 (Kutz et al., 2010).

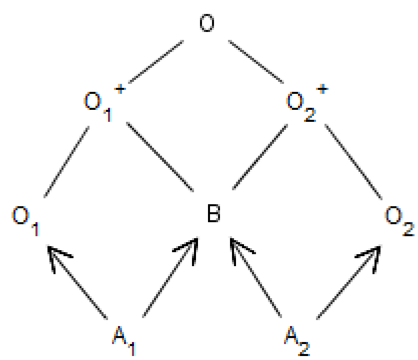

Figure 3-14. Merging with W-alignment

When composing W-alignments all the axioms are embedded into new bridge ontology even when two ontologies involve in the composition operation is disjoint, which is not the desired result. The composition of two W-alignments is shown in Figure 3-15.

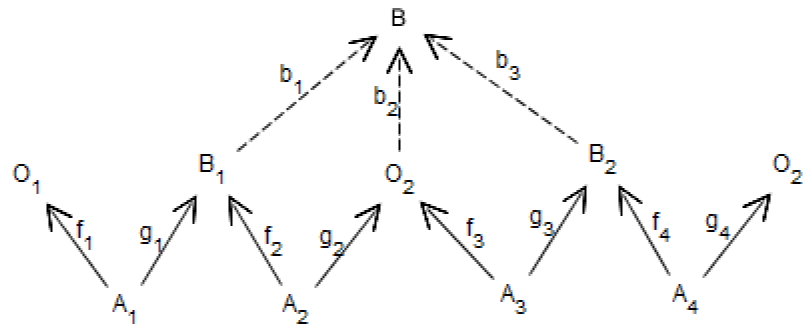

Figure 3-15. Composing W-alignment

Since the merge of W-alignment is obtained by successive pushouts, so it falls in the same class of complexity as V-alignments. The composition, union, and intersection of W-alignments suffer from the fact that bridge ontology requires minimal non-redundant set of axioms that gives desired merge results, while such a bridge ontology is not easily known or built.

Definition 3-18 (M-Alignment): (Kutz et al., 2010) It is a generalization of V-Alignment. O_1 and O_2 are two ontologies and $O_1^\sharp$ and $O_2^\sharp$ by the extension of O_1 and O_2 respectively and this extension defines new symbols and new relationship such as 'Woman' is subsumed by 'Person'. Σ contains signature of ontologies. M-alignment is shown in Figure 3-16.

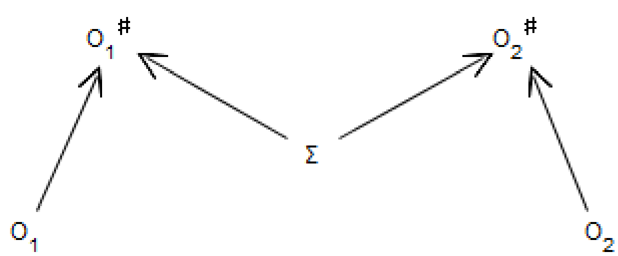

Figure 3-16. M- Alignment

Example 3-4: Integration of two source ontologies O_1 and O_2 by relating the mapped artifacts in subsumption relation $Woman \sqsubseteq Person$ is shown in Figure 3-17.

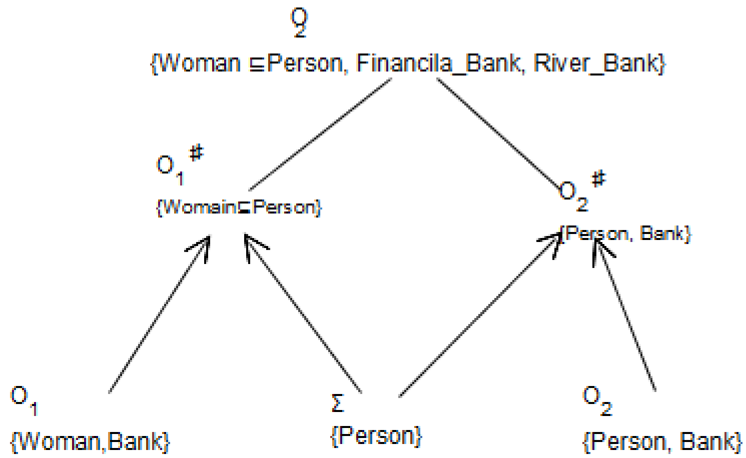

Figure 3-17. Merging by M-Alignment

V-alignment does not represent complex relationships between ontologies and W-alignment and M-alignment solve the problem of adding desired complex relationship between mapped artifacts in the merged ontology but there remains a problem of finding minimal bridge ontology. In some cases, this minimal bridge ontology is not available and user has to build this ontology. Zimmermann and colleagues (Zimmermann et al., 2006) introduce an enhanced form of category Ont^+ to overcome the limitations of V-alignment and W-alignment especially dealing with relations other than equivalence relation such as subsumption, supersumption, overlap, disjointness, label similarity. They define Ont^+ as

Definition 3-19 (Ont^+ category): A category Ont^+ consists of ontologies as objects and particularly elaborated morphisms.

Definition 3-20 (Morphism in Ont^+): Let a morphism $f : O_1 \rightarrow O_2$ in Ont^+ is a set of triples $\langle e_1, e_2, R \rangle$ such that:

e_1 and e_2 are syntactic entities (concepts, relations, individuals, etc.) from ontologies O_1 and O_2 respectively,

R denotes a relationship that holds between e_1 and e_2 (e.g., subsumption, equivalence, temporal relations, etc.). The set of available relations will be denoted R.

This category defines modulo the set of available relations R, so there is a category of ontologies with relations such as *subClass*, *superClass*, *equivalentClass*, *disjointClass*, *partiallyOverlappingClass*.

Definition 3-21 (Composition in Ont^+): Let $f: O_1 \rightarrow O_2$ and $g: O_2 \rightarrow O_3$ be two morphisms in Ont^+.

The composition of f and g, noted $g \circ f$ is the set of triples $\langle e_1, e_3, R_1 \rangle$ such that there exist e_2, R_1, R_2 such that $\langle e_1, e_2, R_1 \rangle \in f, \langle \mathrm{e}_2, \mathrm{e}_3, \mathrm{R}_2 \rangle \in$ g and $R = \phi(R_1, R_2)$

with $\phi : R \times R \rightarrow R$ shown in Figure 3-18.

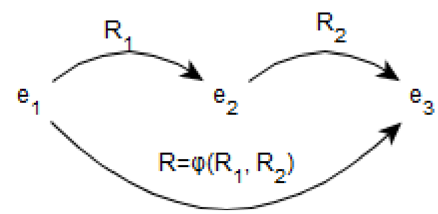

Figure 3-18. Composition in Ont^+

Table 3-1 shows the composition of relations in category Ont^+.

Table 3-1. Composition of relations in Ont^+

	$=$	$\subset$	$\supset$	$\perp$	$\between$
$=$	$\{=\}$	$\{\subset\}$	$\{\supset\}$	$\{\perp\}$	$\{\between\}$
$\subset$	$\{\subset\}$	$\{\subset\}$	$\{=, \subset, \supset, \perp, \between\}$	$\{\perp\}$	$\{\subset, \perp, \between\}$
$\supset$	$\{\supset\}$	$\{=, \subset, \supset, \between\}$	$\{\supset\}$	$\{\supset, \perp, \between\}$	$\{\supset, \between\}$
$\perp$	$\{\perp\}$	$\{\subset, \perp, \between\}$	$\{\perp\}$	$\{=, \subset, \supset, \perp, \between\}$	$\{\subset, \perp, \between\}$
$\between$	$\{\between\}$	$\{\subset, \between\}$	$\{\supset, \perp, \between\}$	$\{\supset, \perp, \between\}$	$\{=, \subset, \supset, \perp, \between\}$

The associative property is preserved, since all the relations respect associative property. The identity constraint is preserved by enforcing the presence of equality relation.

This category has strong advantage of its expressivity and the algebra of V-alignments is still applicable to this category. The drawback of this category is that pushouts do not generally coincide with expected merge operation.

The composition of relations does not result into a base relation, therefore, in some cases these results might be interpreted as disjunction of base relations. Such composition of relations increases the complexity and may lead to multiple different solutions, moreover the correspondences which normally represent one relation between artifacts of two ontologies, are representing more than one relation, so the ontology mapping does not remain ontology mapping. If these relations are treated as single resultant relation then new relations may not be true in logical sense. For instance, in the case of the relation $\{\subset, \perp, \between\}$, it is not possible that artifacts are disjoint and overlapping at the same time. One of the solution is to treat only base relation and select one of the relation, but in this case the choice of selecting relation varies on preference basis and there will be different solution on the basis of different choice.

Ont^{+} adds expressivity but issues related to colimit (may be there exists no colimit or more than one colimit) remains.

3.2 Ontology mapping in Institution Theory

In this section, we present syntax and interpretation independent theory named as institution theory and how it is used in the context of ontology mapping.

3.2.1 Institution Theory Fundamentals

There are numerous logical systems and still they are growing. Researchers in late 20th century were trying to treat different logical systems in a generic way so that different logical systems can be used together. Goguen and colleagues identify that the implementation of many logical systems is independent of the actual details of logical system and to bring uniformity they envision that these systems can be developed in a generic way (Goguen, 1984), (Goguen & Burstall, 1992). They provide a very general formal definition to the logical system and to represent this generality, category theory is the underlying tool of institutions. Institutions provide a meta-mathematical framework for building model theory that is free from the commitment to a specific logical system. This led to formalize various logics as institutions. Barwise has also presented abstract model theory (Jon Barwise, 1974) an extension of some conventional logics, but institutions provide true independence from actual logical system. Institution is defined as

Definition 3-22 (Institution): Formally, an institution (Goguen & Burstall, 1992) is a quadruple $I = \langle Sign, Sen, Mod, \vDash \rangle$ consists of

1. a category $Sign$, whose objects are called signatures,
2. a functor $Sen: Sign \rightarrow Set$, giving for each signature Σ the set of sentences $Sen(\Sigma)$, and for each signature morphism $\sigma: \Sigma \rightarrow \Sigma'$ the sentence translation map $Sen(\sigma): Sen(\Sigma) \rightarrow Sen(\Sigma')$ where often $Sen(\sigma)(\varphi)$ is written as $\sigma(\varphi)$,
3. a functor $Mod : Sign^{op} \rightarrow Cat$ giving for each signature Σ, the category of models $Mod(\Sigma)$, and for each signature morphism $\sigma: \Sigma \rightarrow \Sigma'$, the reduct functor $Mod(\sigma): Mod(\Sigma') \rightarrow Mod(\Sigma)$ where often $Mod(\sigma)(\varphi)$ is written as $M' \upharpoonright_{\sigma}$, and $M' \upharpoonright_{\sigma}$ is called σ-reduct of M', while M' is called σ expansion of $M' \upharpoonright_{\sigma}$
4. a satisfaction relation assigning to each $\Sigma \in |Sign|$ a binary relation $\vDash_{\Sigma} \subseteq |Mod(\Sigma)| \times Sen(\Sigma)$ such that for each $\sigma: \Sigma \rightarrow \Sigma'$ the following condition holds
$$M' \vDash_{\Sigma'} \sigma(\varphi) \text{ iff } M' \upharpoonright_{\sigma} \vDash_{\Sigma} \varphi$$
i.e., for each $M' \in Mod(\Sigma')$ and each $\varphi \in Sen(\Sigma)$, expressing that truth is invariant under change of notation and context.

Σ-satisfaction relationship $\models_\Sigma$ is between sentences and models, such that when signatures are changed satisfaction is preserved. It represents the fact that truth is invariant under change of notation. Sentences are translated in the same direction as of signature morphism, whereas models are translated in the opposite direction as of signature morphism. It is shown in Figure 3-19.

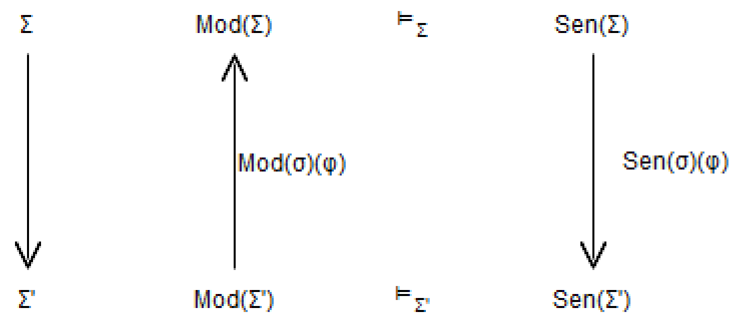

Figure 3-19. Institutions (Goguen & Burstall, 1992)

Example showing how logical systems can be represented as institutions are presented below.

Example 3-5: (Institution of Propositional logic) has propositional symbols as signature Σ and signature morphism σ is $\sigma : \Sigma_1 \to \Sigma_2$ between such sets of signature. Sentences are formed by using usual propositional connectives and sentence translation replaces the propositional symbol along the morphism. A Σ-model M is a mapping from Σ to {true, false}. The reduct of a Σ_2-model M_2 along $\sigma: \Sigma_1 \to \Sigma_2$ is the Σ_1-model given by the composition $M_2 \circ \sigma$. Satisfaction of a sentence in a model is defined by standard truth table semantics. It can be easily viewed that such satisfaction condition holds for this institution.

Example 3-6: (Institution of First-order logic with equality $FL^{=}$) signature of institution $FL^{=}$ consists of set of functions with arities and predicates with arities. Sentences are formed by using usual first-order logic connectives and are first-order formulas. Signature morphism map symbols in such a way that arities are preserved. Sentence translation is performed by translating symbols. Models are first-order structures. Model reduct is reassembling the model components according to signature morphism. Satisfaction is the usual satisfaction of first-order sentences in first-order structures.

Institution theory provides a way to translates both syntax and semantics of different logical formalism.

Definition 3-23 (Institution Comorphism): (Goguen & Roşu, 2002) Given two institutions I and J with $I = \langle Sign^I, Mod^I, Sen^I \rangle$ and $J = \langle Sign^J, Mod^J, Sen^J \rangle$, an institution comorphism from I to J consists of a functor $\Phi: Sign^I \rightarrow Sign^J$, and natural transformations $\beta: Mod^J \circ \Phi \Rightarrow Mod^I$ and $\alpha: Sen^I \Rightarrow Sen^J \circ \Phi$, such that satisfaction condition.

$$M' \vDash^{J}_{\phi(\Sigma)} \alpha_{\Sigma}(\phi) \Leftrightarrow \beta_{\Sigma}(M') \vDash^{I}_{\Sigma} \phi$$

holds. Here, $\phi(\Sigma)$ is the translation of signature Σ from institution I to institution J, $\alpha_{\Sigma}(\phi)$ is the translation of the Σ-sentence ϕ to a $\Phi(\Sigma)$-sentence, and $\beta_{\Sigma}(M')$ is the reduction of the $\Phi(\Sigma)$-model M' to a Σ-model.

Subinstitution is defined in terms of Institution comorphism as

Definition 3-24 (Subinstitution): (Kutz et al., 2010) A subinstitution is an institution comorphism with Φ an embedding of categories, α_{Σ} injective and β_{Σ} an isomorphism for each Σ.

Definition 3-25 (model-expansive Institution): A comorphism is model-expansive if each β_{Σ} is surjective;

Definition 3-26 (faithful Institution): A comorphism is faithful if logical consequence is preserved and reflected along the comorphism: $\Gamma \vDash^{I} \phi$ iff $\alpha(\Gamma) \vDash^{J} \alpha(\phi)$

Each subinstitution comorphism is model-expansive and each model-expansive comorphism is also faithful.

The signature morphisms are different from mappings expressed by logic based mapping languages. In logic based mapping languages, mapping is obtained on the basis of meaningful relations of artifacts of the involved ontologies in the mapping, while in institution mapping (morphism) is independent of translating symbols to any meaningful symbol since truth in institution is not dependent on having any particular signature. Furthermore, sentence translation in institution is induced by signature morphism and in this way such translations dependence on a specific context, if sentences are translated directly then any sentence can be mapped to *true.* In Institutions, any kind of standard mapping (i.e., non logical symbols can be mapped to any non-logical symbols by keeping these non-logical symbols distinct in both ontologies and mapped logical symbols to equivalent logical symbols) is possible, the only requirement is that it fulfils the satisfaction condition of the institutions.

When the underlying logics of theories are same and are in FOL or its sub-languages (such as Description logic, propositional logic) then Enderton approach is similar to Institution theory, but when underlying logics of theories are different or more expressive than FOL then Institution approach is more general than the Enderton approach of relative interpretation of theories.

3.2.2 Operations on Ontologies using Institution theory

Ontologies are treated as logical theory and institution theory can be applied on them. Languages, which are used for formalizing ontologies, are formalized as institutions. Non-

Logical symbols of ontologies are the signatures of ontologies. These signatures are used in the sentences and are interpreted by the model. An essential part of the matching and alignment process is to relate and identify signature elements from different ontologies (possibly formalized in different ontology languages). When mappings are established between ontologies in one institution, then signature morphisms are used for expressing mappings. While when mappings are established between different institutions representing different logical systems, then mapping is carried out by 'Institution comorphism'.

Institutions are also used for semantic integration. Semantic integration of two ontologies O_1 and O_2 into third ontology O_0 is embedding of O_1 and O_2 into existing reference ontology O. In other words, semantic integration is re-interpretation of O_1 and O_2 from the point of view of O. Schorlemmer and colleagues propose semantic integration is carried out when ontologies are treated as institutions (Schorlemmer & Kalfoglou, 2008).

Definition 3-27 (Semantic integration by Institution Comorphism): Let $\mathcal{I}^1 = \langle Sign^1, Sen^1, Mod^1, \vDash^1 \rangle$, $\mathcal{I}^1 = \langle Sign^2, Sen^2, Mod^2, \vDash^2 \rangle$ and $\mathcal{I} = \langle Sign, Sen, Mod, \vDash \rangle$ be three instiutions for ontologies O_1, O_2 and O respectively. Theories representing in two institutions $\mathcal{I}^1$ and $\mathcal{I}^2$ are considered semantically integrated with respect to the theory representing in institution $\mathcal{I}$ as

- sentence translation $\alpha_1 : O_1 \rightarrow O$ and $\alpha_2 : O_2 \rightarrow O$ preserve logical entailments
- There must be structure reduct $\beta_1 : Mod(O) \rightarrow Mod(O_1)$ and $\beta_2 : Mod(O) \rightarrow Mod(O_2)$
- And O is consistent.

Grothendieck institutions are used for dealing with heterogeneous ontologies (theories). In Grothendieck institution signature consists of a pair $\langle L, \Sigma \rangle$ where L is the logic and Σ is the signature in logic L. Signature morphism $(\rho, \sigma) : \langle L_1, \Sigma_1 \rangle \rightarrow \langle L_2, \Sigma_2 \rangle$ consists of logic translation (institution comorphism) $\rho = (\Phi, \alpha, \beta) : L_1 \rightarrow L_2$ and L_2-signature morphism $\sigma : \Phi(\Sigma_1) \rightarrow \Sigma_2$. Sentences, models and satisfaction are defined in componentwise manner (involving both logics L_1 and L_2 and both signatures Σ_1 and Σ_2).

We, here, provide a customized example of (Schorlemmer & Kalfoglou, 2008) about heterogeneous theories where one theory logically entails another theory using institution theory.

Example 3-7: Two institutions are semantically integrated by using Institution co-morphism

Suppose that there is an ontology O_1 that deals with students of the university and there is a relational schema O_2 that also deals with similar domain. We have a reference ontology O expressed in FOL. O_1, O_2 and O are institutions.

Signature of O_1 institution are that of Description Logic.

Let O_1 expressed in Description Logic has following axioms

$$Student \sqsubseteq \exists name.\top \sqcap \exists studies.\top \sqcap \exists degree.\top$$

$$Degree \sqsubseteq \exists degree.\top \sqcap \exists title.\top \sqcap \exists issuedBy.\top$$

$$University\text{-} \sqsubseteq \exists name.\top \sqcap location.\top$$

Let O_2 expressed in relational schema is

A signature in the institution of relational schema (O_2) consists of a set of sorts and a set of relation symbols.

$$student(key\ id: integer, name: string)$$

$$institute(key\ id: integer, name: string, location: string)$$

$$enrolledIn(student: integer, diploma: integer)$$

$$diploma(key\ id: integer, title: string, issuedbBy: string)$$

Since it is not possible literally to merge relational schema being based on closed world assumption and ontology being based on open world assumption in a way that entailments of relational schema and ontology are preserved in merged ontology, so we treat them that they are two different formalism representing same situation and ignoring this issue in this example.

We want to semantically integrate these two institutions in an ontology O which is formalized in FOL.

O is specified in first-order logic as

$$\forall s \exists n, e, d\ (Student(s, n, e, d) \Leftrightarrow studentID(s) \wedge name(s, n) \wedge studies(s, e) \wedge degree(s, d))$$

$$\forall u \exists n, l\ (University(u, n, l) \Leftrightarrow university(u) \wedge name(u, n) \wedge location(u, l))$$

$$\forall s \exists d\ (EnrolledIn(s, d) \Leftrightarrow studentID(s) \wedge degree(s, d))$$

$$\forall d \exists t, i\ (Degree(d, t, i) \Leftrightarrow Degree(d) \wedge title(d, t) \wedge issuedBy(d, i))$$

Sentence translation $\alpha_1: O_1 \rightarrow O$ is as

$$student(key\ id: integer, name: string) \mapsto Student(s, n, e, d)$$

$$institute(key\ id: integer, name: string, location: string) \mapsto University(u, n, l)$$

$$enrolledIn(student: integer, diploma: integer) \mapsto EnrolledIn(s, d)$$

$$diploma(key\ id: integer, title: string, issuedbBy: string) \mapsto Degree(d, t, i)$$

and Sentence translation $\alpha_2: O_2 \rightarrow O$ is as

$$Student \mapsto Student(s, n, e, d)$$

$$Degree \mapsto Degree(d, t, i)$$

$$University \mapsto University(u, n, l)$$

β_i structure of ontologies extracted from reference ontology O by using α_i.

It is easy to check that α and β are well-defined and satisfy the property of Institution comorphism. However, it is not possible to preserve all the entailments of relational schema in first-order logic due to different point of view particularly in dealing with negation. Example 3-7 put in evidence the difficulty of dealing with heterogeneity.

Institution are needed when we have ontologies formalized in some formalism for which first-order logic based semantic integration (by using Enderton Interpretation between theories) is not possible. For instance, when one ontology is formalized in modal logic.

A Distributed Ontology, modelling and specification language (DOL) is proposed by (Lange et al., 2012) and it is based on institution theory. DOL relates ontologies that are distributed and expressed in different languages. The purpose of DOL is to compare and integrate ontologies that are expressed in different formalism. In DOL, several logics are arranged as a graph which shows that logic can be translated into other logic. Some logical languages could not completely be translated to some other logical languages; for instance, first-order logic (FOL) cannot be completely translated into OWL because OWL is less expressive than FOL. Descriptive ontology language DOL is going to be submitted Object Management Group (OMG) standard (Kutz et al., 2015).

One of the advantages of Institution theory is that it is based both on category theory and model theoretic semantics thus provide abstract semantics independent of any specific logic. It can handle heterogeneous ontologies. Different ontology languages and ontology mappings can be expressed as institution. In ontologies, institutions are mainly used in ontology for ontology merging and ontology translation.

Ontology mappings in institution theory is considered as signature morphism of one ontology to another ontology, so it is not necessary that mappings from Ontology O_1 to Ontology O_2 is same as mappings from O_2 to O_1, i.e., it is not necessary that ontology mappings are symmetric.

3.3 Synthesis

We summarize Algebraic Approaches used in Ontology Mappings based on the following parameters and it is shown in Table 3-2.

- **Basic idea:** Category theory is a basic approach used for abstract representation of mappings. Category theory is also used in other algebraic approaches such as Institution theory. What is the underlying basic idea of these approaches?

- **Heterogeneity:** no algebraic approaches presented in this chapter depend on specific language, since they support abstract representation of mappings.
- **Relations:** Generally, equality relation is used for representing mappings in category theory, but approaches proposed by (Zimmermann et al., 2006) also supports other operations such as disjointness and overlap for representing mappings. Institution theory used translation of symbols and sentences for representing mappings.
- **Condition required:** What are the conditions that are required in these approaches.
- **Language Support:** DOL (Distributive Ontology Language) is the only language that is based on Institution theory. based on these algebraic approaches.

Table 3-2. Summary of Algebraic Approaches used in Ontology Mappings

	Category theory	**Institution Theory**
Basic idea	Abstract representation and pushouts/colimits operations are used for ontology merging	Logic independence and translation and satisfiability operations are used
Heterogeneity	Yes	Yes
Relations (implicit)	Mostly Equality relation, but other relations such as disjointness, overlap, subsumption are also supported in category proposed by (Zimmermann et al., 2006)	Equality or substitution used for translation
Representation of equivalence and asymmetric mappings	equivalence mappings	equivalence mappings
Conditions required	When colimit of ontologies is consistent, then ontologies are compatible. But when there is no pushout (colimit) then compatibility of ontologies are not tested and when there are multiple colimits for source ontologies then compatibility of ontologies are not ascertained.	If one ontology (institution) is a Subinstitution of another ontology (institution) then both ontologies can be considered compatible, i.e., they do not contradict each other.
Language Support	No	DOL

The algebraic representation of ontologies and ontology mappings as in Category theory and Institution theory provides abstract representation of mappings independent of any specific logic. However, we have noted in the state of the art that category theory has some limitations and there are some undefined cases when there does not exist any colimit or several colimit exists in category theory while in institution theory. This abstract representation can be used to represent concrete mappings which are normally represented by ontology mapping languages. However, we have found that existing approaches do not provide solution for checking compatibility of ontology mappings especially when mappings caused relative defects. We have used some of the features such as abstract representation of algebraic approaches and comorphism or translations of institution theory (relating theories that are represented in different formalism) in our proposed solution.

Various formalisms of ontology mappings have been discussed in Chapter 2 and Chapter 3. We have found an important characteristic of these formalisms by which we can classify these formalisms. This characteristic is dependency on contents of ontologies.

Ontology mapping formalisms are classified in terms of dependence of contents and logic and it is shown in the Figure 3-20.

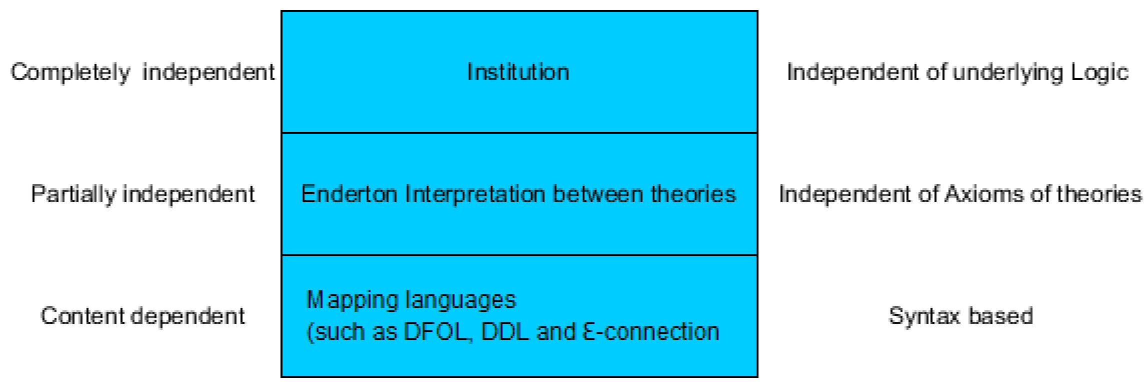

Figure 3-20. Characteristics of various formalism used in ontology mappings

Ontology mapping formalisms like DFOL, DDL, $\epsilon-$connections depend on ontologies involved in mapping. Mappings defined using these formalisms are only meant for ontologies involved in mappings. Distributed models in mapping languages like DFOL, DDL, $\epsilon-$connections are dependent on the contents of the involved ontologies.

Enderton approach of interpretation between theories is partially independent from contents of theories. In this approach, mapping is not only restricted between two theories, rather mappings are defined between signature of language of one theory and the formulas of another theory which may have signature of another language. So correspondences of the mapping in this approach can be reused for some other theories of the languages involved in mappings.

Institutions provide complete independence from contents and from underlying formalisms (logics) of ontologies. Mappings are defined by translating signature of one language to the signature of another language. Institution of $DDL-OWL$ can be formed in which components theories (ontologies) are in OWL and mappings are expressed in DDL. Similarly, institution $\epsilon conn - FOL$C can be built. Ontology language translation is performed by Institution comorphism. For instance, translation of an OWL institution to FOL institution. However, discovering the existence of relative defect requires extra overhead and requires some patterns or heuristics or logical mechanism to discover such entailments (relative defects).

Chapter 4.

Correct Ontology Mappings

Applications using ontology mappings require that mappings do not cause contradiction among source ontologies. However, matching systems or experts establish mappings without considering this important point, therefore ontology mappings may have some correspondences that cause contradiction, i.e., incoherence, inconsistencies or other defects among source ontologies. These correspondences need to be located and corrected either by removing or by modifying these correspondences.

In some situations, when users (agents) want to accept only those mappings that meet their preference criteria such as one agent prefers lexical similarity to structural similarity while other prefers structural similarity to lexical similarity, Argumentation Framework is used to reach at acceptable mappings.

This chapter is organized as follows: Section 4.1 describes debugging ontology mappings and its different phases; Section 4.2 describes the notations to describe various types of defects of ontology mappings; Section 4.3 describes classification of ontology and ontology mappings defects; Section 4.4 presents formal approaches, patterns and empirical ways to locate and repair defects caused by ontology mappings; Section 4.5 presents a synthesis about ontology correctness; and finally, Section 4.6 describes argumentation framework and how logical inconsistency is handled in this framework.

4.1 Debugging Mappings

The term Correctness is used for things that are conforming to truth and such things are considered as defect free. In literature there are many terms such as bug, defect, failure that are used to refer to things which are not correct. Here, we adhere to software engineering's terminology while referring incorrect things in the context of ontology mappings. Generally in the field of software engineering, *errors* are mistakes in coding that occurred due to software developers such as coding errors and requirement gaps and they are found before moving on to the 'next phase'. A *defect* refers to those mistakes which and are found in 'next phase'. When a defect is executed it produces wrong results. The inability of a system to perform correctly is termed as a *failure*. The main difference between error and defect is in terms of timing when they are found. In literature, *bug* is also used to refer both defects and errors.

In the context of mapping, we use the term *defect* for referring incorrectness of mappings as we consider that mappings are already created and we are in the *'next phase' of* checking correctness of ontology mappings.

In the field software engineering, correctness is checked in two steps. Firstly, by checking the presence of defects and it is termed as *testing*. Secondly, by locating and repairing those defects and it is called *debugging*.

Testing: Test cases are developed for checking the existence of defects. There are two main types of testing activities: (a) Validation Testing; (b) Defect Testing. Validation testing is used for checking for fulfillment of requirements specified for the system. When system does not meet any requirement then this fact shows existence of the defect. While in defect testing, existence of any defect in the system is checked.

For testing ontologies 'Reasoners' such as Pellet, Racer Pro, and HermiT are used. Although Reasoners are used for inferring implicit knowledge from ontologies but they are also employed for existence of unsatisfiable artifacts (unsatisfiability is considered as major defect of ontology). Similarly, test cases can be developed for finding the existence of defects other than unsatisfiability. In this chapter our focus is on unsatisfiability and inconsistency. Patterns are also used for defect testing that not only check for unsatisfiable artifacts, but also check those defects which do not conform to user requirements. We present some patterns in Section 4.4.2.3.

Debugging: Debugging is a process that is used for finding and fixing defects identified in testing. Debugging is extensively used in computer programming and, currently, debugging tools become an integrated part of almost all software development environments. Debugging tools help software developers in finding and fixing many programming errors. In the field of database, debugging is used to debug stored procedure (Anon., 2001), views (Caballero et al., 2012) and database schema mapping (Chiticariu & Tan, 2006). Debugging can be used for checking integrated constraints of database schema including normalization, but normalization process is user dependent, i.e., user may not interested in following all normalization rules based on his/her own requirements.

Debugging process becomes complex, when it comes to ontology and ontology mapping, since there is a lot of dependencies and entailments in the ontology. Defects found in ontology or ontology mapping may depend on other artifacts or axioms, therefore focus is on finding the root cause of defects.

Example 4-1: For instance, there is an ontology O with axioms $p \rightarrow q$, $q \rightarrow s$, $q \rightarrow \neg s$, and $p \rightarrow s$ and $p \rightarrow \neg s$ are also the entailments of O. Even though these entailments are contradicting with each other and they are defects but the root cause of these defects are axioms $q \rightarrow s$ and $q \rightarrow \neg s$ because $p \rightarrow q$.

The seminal work on debugging ontology is of Schlobach and colleagues (Schlobach & Cornet, 2003). While Meilicke and colleagues (Meilicke & Stuckenschmidt, 2009) and Qi and colleagues (Qi et al., 2009) bring limelight to debugging ontology mapping.

Most of the Defects in ontology mapping are discovered when they are used with source ontologies. We discuss debugging ontology mapping while not taking ontologies into account in Section 4.4.1 and while taking ontologies into account in Section 4.4.2.

In general, Debugging is performed in four steps; a) Symptoms for Defects (b) Locate Defect c) Repair Defect d) Re-test. It is shown in Figure 4-1.

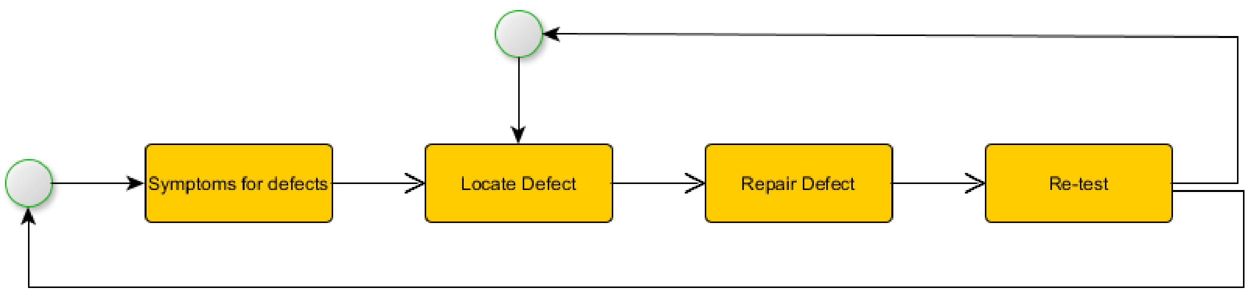

Figure 4-1. Key steps in debugging approaches

We describe these steps in the context of ontology and ontology mapping.

Symptoms for defects: There are several situations, which when occur cause defect(s) in an ontology, these situations are sometime referred as symptoms. Symptoms refer either to the existence of defect or incorrect situation. Baclawski and colleagues (Baclawski et al., 2004) categorize common symptoms of ontologies formalized in OWL such as incomplete union, missing value, ambiguous statement. Knowledge about symptoms of ontology helps in finding defect in ontology.

Locate Defect: In ontologies, it is not easy to locate defects since an unsatisfiable artifact may cause unsatisfiability of other artifacts that depends on the source unsatisfiable artifact. Similarly, in ontology mapping, correspondences may depend on each other and also multiple correspondences may cause incoherence/inconsistency. In this phase, it is desired to find defects, particularly that defect which is the root cause of other defects.

Repair Defect: Generally, there is more than one solution to repair artifact. It is desired that repair is performed in such a way that a minimal change occurs. Repair is mostly performed by removing the defect so that minimal change can occur. An alternative can be modifying the axioms or correspondences, but this choice requires expert knowledge of source ontologies.

Re-test: Re-test is performed to check the presence of new and remaining defects. If there is any defect then debugging process is performed again. Otherwise, it is considered that the debugged source is correct and it does not contain any defect.

In this chapter, our focus is on what is a defect in ontology and ontology mapping and how to locate this defect. We discuss in detail 'locate defect' by presenting two main approaches (theory of diagnosis, belief theory) and common patterns available in the state of the art.

4.2 Notations

When source ontologies are combined with ontology mapping, we use the notation $f(O_i, O_j, M_{ij})$ to express this situation.

There are two main approaches for combining mappings with source ontologies $f(O_i, O_j, M_{ij})$:

a) When mappings are applied to source ontology to obtain new ontology (merged ontology), i.e., $f\left(O_i, O_j, M_{ij}\right)$ results in new ontology. (materialized merge)

b) When mappings are combined with source ontologies and mappings and source ontologies are treated as a logical theory, i.e., $f\left(O_i, O_j, M_{ij}\right)$ treated as new logical theory. (virtual merge)

There can be a combination of one source ontology and ontology mapping represented as $f(O_i, M)$, while second is when both source ontologies are combined with mapping represented as $f(O_i, O_j, M)$.

When there are two ontology mappings M_{ij} and M_{ji} that are combined with source ontologies, this is represented as $f\left(O_i, O_j, M_{ij}, M_{ji}\right)$.

4.3 Defects related to Ontology Mapping

In this section, we present symptoms and defects of ontology mappings and classify these defects.

Having information about ontology or ontology mapping that it is Incoherent, is Inconsistent or contains other defects is as having the information about the symptoms of defects. Formal approaches, patterns or empirical ways are required for locating defects.

We describe state of the art about ontology's defects, since some of the ontology mapping defects occur in source ontology(ies), and then we classify ontology mappings' defects using the insight gain from state of the art.

4.3.1 Ontology Defects

In literature, there exist several classifications of ontology defects (Kalyanpur et al., 2005), (Gangemi et al., 2006), (Corcho et al., 2009), (Poveda Villalon et al., 2010), These classifications do not take ontology mappings into account.

Kalyanpur and colleagues (Kalyanpur et al., 2005) classify ontology defects into three classes: a) Syntactic defects b) Semantic defects, and c) Modeling/style defects.

1. *Syntactic defects:* These defects occurred when ontology does not adhere to the syntax of formalization used to represent ontology. Syntactic errors are easy to correct and Parsers are used to validate the syntax of ontology.

   Example 4-2: Example of syntactical defect in OWL is using ontology constructs that are not supported by OWL such as using SWRL rules in ontology and treated them as OWL constructs.

2. *Semantic defects:* These defects occurred when there are inconsistencies or incoherence in ontologies. These defects are more critical since classical logic (underlying logic of formalization used for expressing ontologies is generally classical) follows the principal of explosion, i.e., any statement can be proven/entailed from a contradiction even $\top \sqsubseteq \bot$ can be entailed.

3. *Modelling/style defects:* These are defects that are not necessarily invalid syntactically and semantically. They may be perceived as *user-defined defects,* one user may be interested in finding and removing such defects while other may not interested in finding such defects or even she does not considering them as defects. Modelling/style defects are important since they cause confusion in the use of ontology containing these defects that may ultimately result in semantic defects. Reasoners are not generally equipped to catch such kind of defects.

   Example 4-3: Examples of such defects are using disjunction in place of conjunction, use of existential quantifier for representing functional property and others. Modelling defects are significant since they can result in unintended results.

Gangemi and colleagues propose to evaluate ontologies on three measures (Gangemi et al., 2006); a) structural dimensional (concerning syntax and semantic of ontologies) b) functional dimensional (concerning the use of ontology with focus on how well conceptualization is formalized c) usability profile (concerning to the documentation and annotation of ontology and this helps the user in using the ontology). Relating this to the classification of Kalyanpur and colleagues. (Kalyanpur et al., 2005), *Structural dimensional* concerning semantics is related to evaluating the presence of *semantic* defects, while *functiona*l and *usability* dimensions are related to evaluation of *modelling/style* defects (Gangemi et al., 2006).

Gomez classify ontology's defects into three classes a) inconsistency (concerned to both logical and semantic inconsistency; semantic inconsistency, here, means that semantically unrelated things are related in the ontology), b) incompleteness (concerned with imprecise conceptualization of ontology) and c) redundancy (concerned with redundant information in the ontology, i.e., information which is implicitly or explicitly already part of the ontology is added to ontology superfluously) (Gómez-Pérez, 2001).

Relating this to the classification of Kalyanpur and colleagues (Kalyanpur et al., 2005), *Inconsistency* is concerned with *semantic* defects while *incompleteness* and *redundancy* are concerned with *modelling/style* defects.

In the literature, there are some terms that are used for specific defects in ontology like design anomalies, pitfalls and anti-patterns. Baumeister and colleagues (Baumeister & Seipel., 2006) uses *anomalies* for defects. Poveda Villalon and colleagues (Poveda Villalon et al., 2010) distinguishes between *pitfalls* and *anti-patterns*. Pitfalls are defects that occur due to the worst practices in creating and modifying ontology, while anti-patterns are those defects that are caused due to not following of Ontology Design Patterns (ODP). Corcho and colleagues (Corcho et al., 2009) uses the terms of *anti-patterns* for set of patterns used by practitioners to identify defects that causes inconsistency, however, we use the term defects to refer such defects.

Ontology defects occur due to uncertainty, ambiguity, imprecision vagueness, or imperfection. In this work, our focus is on semantic and modelling defects.

4.3.2 Ontology mapping Defects

We classify Ontology mappings defects into *syntax defects*, *absolute defects,* and *relative defects*. In this section, we are focusing on *absolute and relative defects*. These defects are checked on an ontology $\mathfrak{O}$ obtained by the combination of source ontologies (O_i and O_j) and mapping (M_{ij}), in symbols $f(O_i, O_j, M_{ij})$.

A. Syntactic defects

Defects that occurred in ontology mapping due to the use of incorrect language syntax in representing mappings or using incorrect names of artifacts of ontologies involved in the mapping. Labels used in mappings may be completely wrong. Mapping may contain some labels that are not present in source ontologies.

These defects can be easily checked with parsers and tracing back to source ontologies.

B. Absolute defects

In ontology mappings, absolute defects are referred to those defects that are universally accepted and that include logical incoherence and logical inconsistency.

Mapping causes incoherence/inconsistency: Ontology mappings may have correspondences which cause incoherence/inconsistency in ontology $\mathfrak{O}$. For instance, two ontologies O_i and O_j are coherent and consistent and two artifacts of an ontology that are disjoint are mapped to same artifact of another ontology. For instance, ontology O_i has $O_i: A \perp B$ and ontology O_j has $O_j: \alpha$, while mapping $M = \{O_i: A \overset{\equiv}{\leftrightarrow} O_j: \alpha, O_i: B \overset{\equiv}{\leftrightarrow} O_j: \alpha\}$. This mapping gives new inference when combined with source ontologies as $f(O_i, O_j, M_{ij})$ which is $O_i: A \equiv O_i: B$ and this new inference is contradicting with ontology axiom $O_i: A \perp B$ and this makes artifacts A and B unsatisfiable and ontology $\mathfrak{O}$ incoherent. When unsatisfiable artifacts are instantiated then inconsistency occurs.

C. Relative defects

These are defects which do not necessarily make source ontologies incoherent and/or inconsistent, but user many be interested that mapping should not have specific characteristics. For instance, user may be interested that mapping should not add additional information (no matter whether this information is true of false) to source ontologies that was not present in source ontologies before combining ontology mappings with source ontologies. This added additional information may not necessarily contradict source ontologies axioms.

Relative defects are described below.

a. **Imprecise mappings:** When establishing mappings between ontologies, it is not clear that how artifacts of source ontologies are exactly related. For instance, an artifact A of ontology O_i is mapped to α of O_j in subsumption relation $\sqsubseteq$ in one correspondence while in other correspondence A is mapped to β in subsumption relation $\sqsubseteq$ as $M = \{O_i: A \overset{\sqsubseteq}{\leftrightarrow} O_j: \alpha; O_i: A \overset{\sqsubseteq}{\leftrightarrow} O_j: \beta\}$ then it is not clear that $M' = \{O_i: A \overset{\sqsubseteq}{\leftrightarrow} O_j: (\alpha \wedge \beta)\}$ or $M' = \{O_i: A \overset{\sqsubseteq}{\leftrightarrow} O_j: (\alpha \vee \beta)\}$. Impreciseness may occur when no relation is specified in the mapping, For instance, an artifact of one ontology is mapped to pair of artifact in other ontology as $M = \{O_i: A \mapsto \alpha, \gamma\}$ then it is not clear in logical term that how these artifacts are related. Whether they are related as $O_i: A \overset{\sqsubseteq}{\leftrightarrow} O_j: \alpha$ and $O_i: A \overset{\sqsubseteq}{\leftrightarrow} O_j: \gamma$ or $O_i: A \overset{\sqsupseteq}{\leftrightarrow} O_j: \alpha$ and $O_i: A \overset{\sqsupseteq}{\leftrightarrow} O_j: \gamma$ or $O_i: A \overset{\sqsubseteq}{\leftrightarrow} O_j: \alpha$ and $O_i: A \overset{\sqsupseteq}{\leftrightarrow} O_j: \gamma$ or in some other way.

Then such mappings are termed as imprecise mappings.

b. **Redundant mappings:** When a correspondence can be inferred from other correspondences, then such kind of correspondence is termed as redundant. For instance, there is an axiom in ontology O_i $\{B \sqsubseteq A\}$ and mappings are $\{O_i: A \overset{\equiv}{\leftrightarrow} O_j: \alpha, O_i: B \overset{\sqsubseteq}{\leftrightarrow} O_j: \alpha\}$. In this case, mapping $O_i: B \overset{\sqsubseteq}{\leftrightarrow} O_j: \alpha$ is redundant since it can be inferred by correspondence $O_i: A \overset{\equiv}{\leftrightarrow} O_j: \alpha$ with ontology axiom of $O_i B \sqsubseteq O_i: A$. Although redundant mappings are not defect in itself but such mappings are source for defect, i.e., they can result in some defect. We do not cover such mappings in our approach; the reason for this is that they are not themselves defect and whenever redundant mappings are causing absolute and/or relative defect then we check for absolute and/or relative defects and ignore the defect of redundancy.

c. **Abnormal mappings:** Sometimes a mapping when combined with source ontologies results into new inferences such as adding new information to ontology that is unwanted, then such mappings are termed as unintentional mappings. For instance,

O_i has axioms $\{B \sqsubseteq A, C \sqsubseteq B\}$ and mapping M is $\{O_i{:}B \overset{\equiv}{\leftrightarrow} O_j{:}\alpha, O_i{:}C \overset{\equiv}{\leftrightarrow} O_j{:}\alpha\}$, when mapping is combined with source ontology O_i as $f(O_i, M)$, this results in new entailment in ontology $f(O_i, M)$ which is $A \equiv C$ and this makes is-a circle (an artifact is defined as generalization or specialization of itself) as $C \sqsubseteq B \sqsubseteq A \sqsubseteq C$. Mapping that results into unintentional inferences in ontology is termed as *unintentional* mapping. It is considered as violation of *principle of conservativity, and this principle* states that Mapping should not add new information, i.e., information which was not available in the source ontology.

d. **Semantic Inconsistency:** This defect occurs in a situation when creator of the mapping relates artifacts of ontologies that she does not intend to do. This situation may not cause absolute defect in some situation. Such defects may not be easily discovered through logical approaches. Example 1-2 is an example of this defect, in which $Kiwi_1$ is a fruit and $Kiwi_2$ is bird where $Kiwi_1$ and $Kiwi_2$, here assume that $Kiwi_1$ and $Kiwi_2$ are not declared disjoint in the ontology, and both of them are mapped to an artifact $Kiwi$ of another ontology and $Kiwi$ is mapped to $Kiwi_1$ in another mapping. This means that artifacts having distinct meanings are being semantically interpreted as same or similar.

e. **Mapping is not an interpretation of a theory (ontology) into another theory (ontology):** Mapping $m{:}\, O_i \rightarrow O_j$ is a theory interpretation if mapping m translates all the inferences of ontology O_i into formulas that are inferences in theory O_j (Enderton, 2001). If mapping cannot become an interpretation of one theory into another theory then, we term this defect as not an interpretation.

f. **User defined defects:** Sometimes user may add some condition (axiom) to source ontology(ies) or to combination of source ontologies and mappings $f(O_i, O_j, M)$ and then check logical consistency or coherence. Since these defects are neither part of source ontologies nor ontology mappings, so we do not tackle this case directly. However, it should be noted that when these added conditions (axioms) are treated as part of ontology then this situation is implicitly covered when we check for absolute or other relative defects of ontology mappings. Example of such defect is $f(O_i, O_j, M)$ should not have a particular property for instance, depth of $f(O_i, O_j, M)$ should not exceed n.

Wang and colleagues (Wang & Xu, 2012) classify ontology mapping defects as (a) Redundant mapping (b) Imprecise mapping (c) Inconsistent mapping (d) Abnormal mappings. Inconsistent mapping is an *absolute* defects while remaining three classes are *relative* defects.

We classify Ontology mapping defects into two classes on the basis of their presence.

a. Defects in ontology mapping

b. Defects in the combination of source ontologies and mappings caused by ontology mapping.

a. **Defects in ontology mapping:** This kind of defects occurred in ontology mapping. When ontology mapping is treated as logical theory then defects that are found are considered as defects in ontology mappings. These defects can be absolute defects such as incoherence and inconsistency or relative defects such as redundancy or semantic inconsistency.

Example 4-4 *Absolute defects*

Suppose that there is an ontology mapping M={O_i: $A \overset{\sqsubseteq}{\leftrightarrow} O_j$: B, O_i: $A \overset{\perp}{\leftrightarrow} O_j$: B}. These two correspondences of mappings are contradicting each other. It is not possible that simultaneously artifact A of ontology O_i is in subsumption and disjointness relation with artifact B of ontology O_j.

Above example is also an example of imprecise and abnormal mappings.

Example 4-5: Relative *defects:*

Suppose that there is an ontology mapping M={O_i: $A \overset{\sqsubseteq}{\leftrightarrow} O_j$: B, O_i: $A \overset{\equiv}{\leftrightarrow} O_j$: B} O_i: $A \overset{\sqsubseteq}{\leftrightarrow} O_j$: B can be inferred from O_i: $A \overset{\equiv}{\leftrightarrow} O_j$: B so it is considered as redundant correspondence, a relative defect.

b. **Defects caused by ontology mapping in the combination of source ontologies and ontology mapping :** This kind of defects occurred when either source ontologies and mappings or single source ontology and mappings are combined. We can classify ontology mappings into two classes on the basis of ontology mapping combined with number of source ontologies. Firstly, when single ontology is combined with mapping $f(O_i, M)$. Secondly, when both source ontologies are combined with mapping $f(O_i, O_j, M)$. $f(O_i, M)$ and $f(O_i, O_j, M)$ are treated as logical theory.

Example 4-6: *Defects in ontology caused by ontology mapping taking source ontologies into account*

For instance, there is a mapping $M = \{O_i: A \overset{\equiv}{\leftrightarrow} O_j: \alpha, O_i: B \overset{\equiv}{\leftrightarrow} O_j: \beta\}$ and axiom of ontology O_i as $B \sqsubseteq A$ and axiom of ontology O_j as $\alpha \sqsubseteq \neg\beta$. When mapping M is combined with ontology axioms it will result new entailment a $O_i: B \sqsubseteq O_i: \neg A$ which makes artifact B of O_i as unsatisfiable (because of $O_i: B \sqsubseteq O_i: \neg A \wedge O_j: A$) and therefore ontology $f(O_i, O_j, M_{ij}$ becomes incoherent. Besides incoherence, there are other types of defects caused by ontology mapping and these are discussed in this section.

This example is also a case of violation of principle of conservativity and abnormal mapping.

Example 4-7: *Defects in ontology caused by ontology mapping taking single source ontologies into account*: $M = \{O_i : A \overset{\equiv}{\leftrightarrow} O_j : \alpha, O_i : A \overset{\equiv}{\leftrightarrow} O_j : \beta\}$ and while taking only O_j into account which has axiom $\alpha \sqsubseteq \neg\beta$. This clearly means that $O_i : A$ is unsatisfiable (because $O_i : A \sqsubseteq \beta \wedge \neg\beta$) and $f(O_i, O_j, M_{ij})$ becomes incoherent. Also, inference $\alpha \sqsubseteq \neg\beta$ can be translated into ontology O_i and mapping M is not an interpreatoin of ontology O_j into ontology O_i. Mapping M can be treated in both direction from $O_i \rightarrow O_j$ and also from $O_j \rightarrow O_i$. However, if mapping is in only one direction then its theory interpretation in that direction can only be checked and not in the reverse direction.

This example is also a case of of violation of principle of conservativity and abnormal mapping.

We classify defects in different contexts in Table 4-1

Table 4-1. Ontology mapping defects in different contexts

	Syntactical defects	**Absolute defects**	**Relative defects**
Mapping $\boldsymbol{M}$	Yes	Incoherence Inconsistency	Redundant mappings, Imprecise mappings, Abnormal mappings
Combination of source ontology and mapping $\boldsymbol{f(O_1, M)}$	Yes	Incoherence Inconsistency	Conservativity, Not an interpretation, Redundant mappings, Imprecise mappings, Abnormal mappings, User defined defects
Combination of source ontologies and mapping $\boldsymbol{f(O_1, O_2, M)}$	Yes	Incoherence Inconsistency	Conservativity, Not an interpretation, Redundant mappings, Imprecise mappings, Abnormal mappings, User defined defects

4.4 Locate Defects

Most of the state of the art is focused on debugging incoherence but there exist few works that focus on debugging other defects. Various approaches and patterns to debug defects of ontologies and ontology mappings are presented in the following subsections.

4.4.1 Debugging Ontology mappings without taking into account ontologies

4.4.1.1 Logical reasoning

Reasoning on ontology mapping alone may leads to infer more correspondences between source ontologies and identify defects in ontology mapping. Ontology mapping may contain correspondences that are conflicting with each other. These correspondences can be debugged without taking source ontologies into account; just check whether each artifact of first ontology is related to another artifact of the second ontology in more than one way. If there are such correspondences then check whether or not the relations involved in correspondences are contradicting each other.

Example 4-8: Suppose that there is a mapping M and it contains following two correspondences among others

$$O_i: A \overset{\sqsubseteq}{\leftrightarrow} O_j: B$$

$$O_i: A \overset{\perp}{\leftrightarrow} O_j: B$$

Ontology mapping can be treated as a logical theory and Reasoners (logical tools) can be used to debug ontology mappings for identifying logical inconsistencies. Reasoners identify above correspondences as absolute defects, since it is logically invalid that artifact A is subsumed by artifact B and at the same time these two artifacts are disjoint.

When correspondences involve roles, even the relations involved in mapping are same or do not contradict each other, it is still require checking for the existence of contradicting existential and universal quantifiers .

Example 4-9: Mapping $M = \{O_i: A \overset{\equiv}{\leftrightarrow} O_j: \forall r.\alpha$ and $O_i: A \overset{\equiv}{\leftrightarrow} O_j: \exists r.\alpha)\}$ has contradicting existential and universal quantifiers for the involved artifacts of source ontologies so this mapping will cause defect.

When correspondences involve roles, number restrictions can also cause contradiction.

Example 4-10: Mapping $M = \{O_i: A \overset{\sqsubseteq}{\leftrightarrow} O_j: \geq 3\, \exists r.\alpha$ and $O_i: A \overset{\sqsubseteq}{\leftrightarrow} O_j: \leq 2\exists r.\alpha)\}$ has contradicting number restrictions on role and make the artifact $O_i: A$ unsatisfiable.

4.4.1.2 Formal ways frequently used for debugging ontology mappings

Euzenat (Euzenat, 2008) proposes an algebra of relations for handling ontology mapping operations. He also highlights that reasoning on ontology mapping may be used to infer more correspondences between ontologies and identify inconsistencies in ontology mappings. His proposed algebra for ontology mapping relations results in new relations that are not base relations. We have presented Table 3-2 about this in chapter 3. Composition of relations $<$ and $>$ is $\{<, >, =, \between\}$ and this is not a base relation. The basic assumption is that relation between artifacts of ontologies are jointly exhaustive and pair-wise disjoint, so there is one and only one relation between artifacts of ontologies. Set of relation is used to express uncertainty. He also incorporated confidence measure; correspondences having higher confidence value are more reliable than correspondences having lesser confidence value. Confidence measures are used for mapping reduction.

Example 4-11: For example, $A\,\{<\}_{.9}\,B, A\,\{<,=\}_{.8}\,B$ can be reduced to $A\,\{<\}_{.9}\,B$, because $\{<\} \subseteq \{<,=\}$ and $.9 \geq .8$, hence $A\,\{<\}_{.9}\,B$ *entails* $A\,\{<,=\}_{.8}\,B$. The mapping reduction can be used in removing redundant ontology correspondences.

Defects found by debugging ontology mappings without taking ontologies into account can also be debugged by using approaches of debugging ontology mappings that take ontologies into account, but there is an extra overhead of dealing with ontologies. However, debugging only ontology mapping without taking source ontologies into account will not detect all defects of ontology mapping since some defects arises only when source ontologies are involved.

Example 4-12: Mapping $M = \{O_i{:}\,A \overset{\sqsubseteq}{\leftrightarrow} O_j{:}\,\alpha, O_i{:}\,A \overset{\sqsubseteq}{\leftrightarrow} O_j{:}\,\beta\}$, if α and β are disjoint in O_j then one cannot discover that mapping makes artifact A and B unsatisfiable without taking source ontologies into account.

Hence, approaches of debugging ontology mappings without taking ontologies into account may be used as a first step in debugging ontology mappings. However, approaches that take ontologies into account while debugging ontology mappings are required for debugging other existing defects in ontology mappings.

4.4.2 Debugging ontology mappings while taking ontologies into account

Most of the defects in ontology mapping are identified when mappings are combined with ontologies. Currently Reasoners are not completely capable to detect all relative defects and to describe root cause of defects. Reasoners, in most cases, only highlight unsatisfiable artifacts but this information is not sufficient since defects in ontology are interdependent on each other and single unsatisfiable artifact can make remaining artifacts of ontology unsatisfiable.

Combining source ontologies and mappings provides a way to debug ontology mappings and repair defects caused by ontology mappings. There are several ways to combine mapping with source ontologies. Hence, it is not necessary that defects occurred in one way of combining mapping with source ontologies also occurred in other ways of combining mappings with ontology mappings. From the work presented in Chapter 2, and Chapter 3, we represent combination of source ontologies O_i and O_j and mappings M_{ij} as $f(O_i, O_j, M_{ij})$. We have described that $f(O_i, O_j, M_{ij})$ can be interpreted in two ways: (a) logical theory (b) materialized merge.

Entailments inferred by (a) can be inferred by (b) and vice versa. There are several ways to merge ontologies (Raunich & Rahm, 2012) and they may give different result. When merged operation is perform which add new artifacts and/or axioms other than $f\left(O_i, O_j, M_{ij}\right)$ and that requires user choice or expert advice or personal preference, then it will be different from (b). However in this chapter, we are dealing (a) and (b) in a similar manner and considered that ontology obtained after merge operation in (a) is same as the ontology in (b).

Mapping may represent one party's (ontology) point of view about another ontology or represent third party's point of view about source ontologies. In the following we consider that mapping takes third party's point of view about source ontologies (similar to IDDL approach) and represent mapping relation as $\overset{r}{\leftrightarrow}$ where r can be $\sqsubseteq, \equiv, \perp$ $and\ other\ relations$. But we have added related comments about mapping representing one party's (ontology) point of view about another source ontology where it is appropriate to mention.

Debugging ontology mapping is a complex task due to interdependency of artifacts and axioms of ontologies, hence it is required to know how a specific defect occurs in ontology. There may be several reasons that describe how a specific defect occurs but in debugging ontology mapping focus is on finding the root cause of that defect. This root cause is termed as *justification* (Kalyanpur et al., 2005). McGuinness and colleagues (McGuinness & Pinheiro da Silva, 2004) describe that justification should include knowledge provenance and information about how a specific entailment or defect can be derived or retrieved.

Definition 4-1 (Justification in Ontology): A justification of entailment in ontology is defined as a minimal set of axioms of an ontology that is sufficient to hold the entailment.

Emphasis in locating defect phase is on finding minimal and relevant justifications of defects. Horridge and colleagues (Horridge et al., 2008) identify some difficulties in identifying justifications.

(a) justification contains irrelevant parts e.g., $O = \{B \sqsubseteq C \sqcap D, D \sqsubseteq E\} \vDash B \sqsubseteq E$ while artifact C does not play any part in the entailment of $B \sqsubseteq E$;

(b) a single justification can mask other justifications e.g. $\{B \sqsubseteq \neg C \sqcap D, B \sqsubseteq C \sqcap \neg D\} \vDash B \sqsubseteq \perp$ but $B \sqsubseteq \perp$ is responsible for two reasons $B \sqsubseteq \neg C, B \sqsubseteq C$; and $B \sqsubseteq D, B \sqsubseteq \neg D$ while $\{B \sqsubseteq \neg C \sqcap D, B \sqsubseteq C \sqcap \neg D\}$ mask these two distinct reasons; and

(c) multiple justifications can conceal a fine grained core e.g.$\{A \sqsubseteq B \sqcap C, B \sqsubseteq D\}$ and $\{A \sqsubseteq B \sqcap F, B \sqsubseteq D\}$ are two justifications for $A \sqsubseteq B$. But there is only one fine grained justification $\{A \sqsubseteq B, B \sqsubseteq D\}$. These difficulties of finding fine-grained justification highlight the fact that locating defects in ontologies is a complex task.

Precondition for debugging ontology mapping is that source ontologies are free from any absolute defect. When source ontologies are combined with mappings and if there arise any defect then this defect is only caused by ontology mappings. Here, we present a seminal work of Schlobach and colleagues in debugging ontology.

We discussed here the work of (Schlobach & Cornet, 2003) because most of the recent work in the field of debugging ontology is based on this work. They propose a solution to debug unfoldable $\mathcal{ALC}$ TBox based on minimization of axioms for computing *minimal unsatisfiability-preserving sub-TBoxes* (MUPS) and *minimal incoherence-preserving sub-TBoxes* (MIPS), and use a bottom-up method for *generalized incoherence-preserving terminologies* (GIT). GIT is a set of incoherent axioms, which are syntactically related to the original axioms, are more general and have minimal structural complexity. Schlobach and colleagues exclude those axioms which are irrelevant to incoherence in ontology and then pinpoint the axioms and concepts responsible for incoherence by identifying unsatisfiable concepts, compute MUPS for each concept and then compute MIPS. Subsequently they compute cores for each axiom to count the number of occurrences of axiom in different MIPS and GIT is used to generalized the MIPS in a way that keep only related information that reflects unsatisfiability and remove those concepts that are not involved in unsatisfiability. GIT is then used to repair incoherence in the ontology.

MUPS and MIPS are defined below with illustration of how they can be computed.

Definition 4-2 (Minimal Unsatisfiability Preserving subTBox): (Schlobach & Cornet, 2003) Let $\mathcal{O}$ be an ontology, let $\mathcal{T}' \sqsubseteq \mathcal{O}$ be the TBox of $\mathcal{O}$, and let C be the unsatisfiable concept in the ontology $\mathcal{O}$. A set $\mathcal{T}' \subseteq \mathcal{T}$ is a minimal unsatisfiability preserving sub-TBox (MUPS) in $\mathcal{T}$ for C if C is unsatisfiable in $\mathcal{T}'$and C is satisfiable in every $\mathcal{T}'' \subset \mathcal{T}'$. The set of all MUPS with respect to C is referred to as $mups(O, C)$.

Example 4-13: An ontology O has axioms (1) $A \sqsubseteq B \sqcap C$, (2) $B \sqsubseteq \neg C$, (3) $D \sqsubseteq B$ then $\text{mups}(O, A) = \{1,2\}$, i.e., $\{1{:}\, A \sqsubseteq B \sqcap C, 2{:}\, B \sqsubseteq \neg C\}$ makes the concept A unsatisfiable. Axiom 3 does not play any role for making concept A unsatisfiable.

Definition 4-3 (Minimal Incoherence Preserving subTBox): (Schlobach & Cornet, 2003) Let $\mathcal{T}$ be an incoherent TBox (ontology). A TBox $\mathcal{T}' \subseteq \mathcal{T}$ is a minimal incoherence preserving sub-TBox (MIPS) of $\mathcal{T}$, if $\mathcal{T}'$ is incoherent and every subTBox $\mathcal{T}'' \subset \mathcal{T}'$ is coherent.

MUPS are used in computing MIPS. MIPS are the smallest subsets of an original TBox preserving unsatisfiability of at least one atomic concept.

Example 4-14: (Schlobach & Cornet, 2003) Suppose that there is an ontology with axioms $1{:}\, A_1 \sqsubseteq \neg A \sqcap A_2 \sqcap A_3$, $2{:}\, A_2 \sqsubseteq A \sqcap A_4$, $3{:}\, A_3 \sqsubseteq A_4 \sqcap A_5$ $4{:}\, A_4 \sqsubseteq \forall s.B \sqcap C$, $5{:}\, A_5 \sqsubseteq \exists s.\neg B$, $6{:}\, A_6 \sqsubseteq A_1 \sqcup \exists r.(A_3 \sqcap \neg C \sqcap A_4$, $7{:}\, A_7 \sqsubseteq A_4 \sqcap \exists s.\neg B$.

$MUPS(O, A_1) = \{\{1,2\}, \{1,3,4,5\}\}$ $MUPS(O, A_3) = \{\{3,4,5\}\}$ $MUPS(O, A_6) = \{\{4,7\}\}$ $MUPS(O, A_7) = \{\{1,2,4,6\}, \{1,3,4,5,6\}\}$

and $MIPS(O) = \{\{1,2\}, \{3,4,5\}, \{4,7\}\}$. There are 3 $MIPS$ of O and each $MIPS$ locates axioms responsible for at least one unsatisfiable concept.

$MIPS$ only identify axioms of ontology but each axiom may have irrelevant information which has nothing to do with the unsatisfiability of the artifact. Schlobach and colleagues use GIT for identifying only those parts of axioms that are responsible for defect.

Example 4-15: For {1,2}, the axioms are $\{A_1 \sqsubseteq \neg A \sqcap A_2 \sqcap A_3,\ A_2 \sqsubseteq A \sqcap A_4\}$. GIT for these axiom is $A_1 \sqsubseteq \neg A \sqcap A$.

In the literature there are different ways to debug defects in ontology caused by ontology mapping. In Section 4.4.2.3, we present some patterns that are used to debug some of the defects of ontology. We discuss two debugging approaches which are predominately used in debugging ontology; One of which is based on theory of diagnosis and this is discussed in Section 4.4.2.1, and other is based on belief revision theory and this is discussed in Section 4.4.2.2.

4.4.2.1 Approaches based on Theory of Diagnosis Theories

Rieter proposes theory of diagnosis in terms of *system* and *components*. He describes that when *observer* observes behavior of system different from expected behavior then it means that at least one of the component is functioning abnormally (Reiter, 1987). He defines *diagnosis* as a minimum set of components responsible for abnormal behavior. Theory of diagnosis is widely used in debugging.

After obtaining MIPS, there are several ways to make ontology coherent. For instance, in the example 4-14, one can remove axiom set {1, 3, 7} or {2, 4} or {2,3,7} to make ontology coherent. Thus, it is required to find that which choice is better choice. Reiter's theory of diagnosis (Reiter, 1987) and belief revision theory of (Gaerdenfors, 1992) are also used for debugging ontology mappings (Qi et al., 2009), (Meilicke & Stuckenschmidt, 2009). A principle of minimal change (Gärdenfors, 2003) is followed for making the ontology coherent when removing axioms from MIPS. In debugging ontology and ontology mapping, minimal change is mostly computed by finding the Hitting Set and it is defined as

Definition 4-4 (Hitting-set): Given a set T and a collection $S = \{S_1, \dots, S_n\}$ with $S_i \subseteq T$ for $i = 1 \dots n$. $H \subseteq T$ is a hitting-set for S iff $H \cap S \neq \emptyset$ for $i = 1 \dots n$. $H \subseteq T$ is a minimal hitting-set for S iff H is a hitting-set for S and there exists no $H_0 \subset H$ such that H_0 is a hitting-set for S.

A Minimal Hitting Set Tree (Minimal HST) is used for identifying minimal part of the ontology responsible for incoherence also called diagnosis. Reiter proposes to use minimal HST for computing the minimum part of the system and when this minimal part is removed, it makes the system consistent. $MUPS$ and $MIPS$ corresponds to conflict set of Reiter's theory of diagnosis.

Example 4-16: A minimal Hitting set for $MIPS(O) = \{\{1,2\},\{3,4,5\},\{4,7\}\}$ is $\Delta(O) = \{\{1,4\},\{2,4\},\{1,3,7\},\{2,3,7\},\{1,5,7\},\{2,5,7\}\}$. Coherent ontology can be obtained by removing any element of $\Delta(O)$.

Now it is required to identify that which element of the diagnosis is to be removed so that minimal change is occurred in the ontology. It can be seen that if $\{1,3,7\}$ is removed then we have to remove 3 axioms of ontology, while if $\{1,4\}$ is removed then we have to remove 2 axioms of ontology. Obvious choice is to remove less number of axioms. Even this can be problematic since the selected less number of axioms for deletion are more important than large number of axioms that are not selected for deletion. In general, when diagnosis has same number of correspondences as $\{1,4\},\{2,4\}$ then choice is not simple because some axioms are more important than others. Confidence values are associated to each axioms on the basis of their frequency of occurrence in justification, provenance information about axioms (like authors, date), relevance to ontology in terms of their use, and impact of removal or alteration of axiom ontology.

In (Schlobach, 2005), they showed that computing all MIPS and MUPS are time consuming and inefficient as computing minimal Hitting set for $MIPS$ is $NP - COMPLETE$. Some approximation and optimization techniques are proposed (Schlobach, 2005), (Kalyanpur et al., 2005), (Qi et al., 2009) and (Meilicke & Stuckenschmidt, 2009).

(Schlobach, 2005) has proposed to use an approximation measure to pinpoint minimal set of diagnosis. He suggests selecting axioms having maximum frequency in $MIPS$, but it turns out that this solution is not always minimal. For instance, suppose that $MIPS(O) = \{\{1,2\},\{3,4\},\{5,1\},\{5,3\}\}$, by pinpointing approach we get $\{1,3,5\}$. But by using HST, we get minimal set as $\{1,3\}$. Their solution is *linear* as compared to $NP - COMPLETE$ but it is not minimal.

Kalyanpur and colleagues (Kalyanpur et al., 2006) also uses HST for finding diagnosis and they pinpoint axioms on the basis of ranks. Ranks are associated to axioms on provenance information and their importance in ontology. For instance, ranks r for axiom $r(1) = 0.1, r(2) = 0.2, r(3) = 0.3, r(4) = 0.4, r(5) = 0.5$ and minimal HST results in $\{2,4,1\}$ and $\{5,3,1\}$ But when ranks are considered with these minimal HST results, then the choice will be $\{2,4,1\}$.

Schlobach and Kalyanpur and colleagues work is related to debugging ontology and they do not explicitly talk about ontology mappings. However, their work can be used for debugging ontology mapping when ontology mapping is treated as axioms of ontology. We present here two approaches which explicitly focus on ontology mapping with the goal of debugging and repairing ontology mapping.

A. Meilicke and colleagues (Meilicke & Stuckenschmidt, 2009) work is formalized in the Reiter's theory of diagnosis but they do not use minimal hitting set for computing diagnosis.
B. Qi and colleagues (Qi et al., 2009) work is formalized in the AGM theory of Belief revision.

In both approaches, the underlying assumption is that ontologies are coherent and if there is incoherence in the combination of source ontologies and ontology mapping then it is only because of ontology mapping.

An incoherent mapping is defined by (Meilicke & Stuckenschmidt, 2009) as

Definition 4-5 (Incoherent mapping): A mapping M is incoherent if an artifact which is unsatisfiable due to M w.r.t. source ontologies O_i and O_j (i.e., unsatisfiable in $O_i \cup O_j \cup \bigcup_{c \in M} \xi(c)$ where ξ maps a correspondence between O_i and O_j to an ontology with signatures of O_i and O_j).

Meilicke proposes two approaches for computing diagnosis and they are (a) Local optimal diagnosis; (b) Global optimal diagnosis. Both of these approaches compute hitting set implicitly. They use confidence value of correspondences and select correspondence having minimal confidence value that cause incoherence for computing diagnosis. They do not compute all $MIPS$ and $MUPS$.

In Local optimal diagnosis, mappings are arranged in descending order with respect to their associated confidence values. Each correspondence is iteratively checked (starting from correspondence with higher confidence value) whether it causes incoherence according to definition 5. If correspondence causes incoherence then it is saved to diagnosis set Δ otherwise it is not saved to Δ. Subsequently, mappings that do not cause incoherence are combined with next mapping in iteration and this combination is then checked for incoherence. This check continues until all mappings are not checked for incoherence. In the end Δ is diagnosis for incoherence. This diagnosis is generally not it is not a smallest confidence weighted diagnosis. Global optimal diagnosis is identified on the basis of mapping sets with minimal aggregated confidence value.

Example 4-17: (Local optimal diagnosis) Let there are four correspondences with confidence values $(m_1, 0.9)$, $(m_2, 0.8)$, $(m_3, 0.7)$, and $(m_4, 0.5)$. For reference, $MIPS$ are $\{m_1, m_2\}, \{m_1, m_3\}, \{m_2, m_4\}, \{m_3, m_4\}$. After arranging correspondences in descending order w.r.t. their confidence values, incoherence is checked iteratively by starting from correspondence with highest confidence value, i.e., m_1. since there is no incoherence then m_2 is added in next iteration for checking incoherence. Addition of m_2 causes incoherence so it is removed and is identified as part of the diagnosis. In the next iteration m_3 is added and it also results in incoherence then it is also removed and make part of the diagnosis. In next iteration m_4 is added, it does not cause incoherence and there is no more correspondence for checking coherence. The diagnosis is $\{m_2, m_3\}$. This shows that there is no need to compute $\{m_2, m_4\}, \{m_3, m_4\}$.

Global Optimal diagnosis for this example when there are two diagnosis $\{m_1, m_4\}$ and $\{m_2, m_3\}$ the aggregated confidence value for $\{m_1, m_4\}$ is 1.4 while aggregated confidence value for $\{m_2, m_3\}$ is 1.5. So, the choice in global optimal diagnosis is $\{m_1, m_4\}$ which is different from local optimal diagnosis.

Local and global optimal diagnosis works for mappings having distinct confidence values and they do not handle mappings having same confidence value correspondences. Their proposed work always selects minimal number of correspondences for diagnosis. But, as local optimal and global optimal diagnosis may give different results, it remains unclear which is the best choice. However, they empirically show that global optimal diagnosis is better in most of the cases.

4.4.2.2 Approach based on Belief revision Theory

Although belief revision theory is applicable to set of formulas closed under logical consequence while ontology expressed in formalism like DL is not necessarily closed under operators $\neg$ and $\wedge$ (Flouris et al., 2006) but still this theory is used in ontology by adopting it according to the requirements. Restriction of set of formulas closed under logical consequence can be ignored (Hansson, 1999), so in this way theory of belief revision can be applied for debugging ontologies (Qi et al., 2009).

Qi and colleagues (Qi et al., 2009) treat mappings as axioms of an ontology and formalize their work according to Belief revision theory. A mapping revision operator is defined similarly as of theory of belief revision's internal revision operator and it is used for removing a subset of mapping so that remaining mapping becomes coherent. They do not compute all $MUPS$ and $MIPS$.

In their algorithm, they arrange all correspondences in descending order on the basis of their weightage and check Incoherence iteratively by checking whether the correspondence having highest value is combined with source ontologies causes incoherence or not. If this correspondence does not cause incoherence, then the next correspondence in order is combined with already combined correspondences and source ontologies for checking coherence, if this correspondence does not cause incoherence then this step is repeated till incoherence occur or no more correspondences are left for checking. If the correspondence that is combined in the last iteration causes incoherence, then that correspondence is removed from already combined correspondences and this step of checking for incoherence is performed on remaining correspondences.

This algorithm does not always remove minimal number of correspondences when some of the correspondences have same confidence value since this algorithm construct hitting set in depth search manner. This algorithm is optimized in another algorithm by starting from selecting *relevant* correspondence responsible for first discovered unsatisfiable artifact and it is then expanded systematically to avoid reasoning of complete ontology.

Example 4-18: (Qi et al., 2009) Let there are five correspondences $m1$ having confidence value of 0.9 with confidence m_2 and m_3 both have confidence value of 0.8, $m4$, $m5$ both have confidence value of 0.63. Qi and colleagues algorithm arranges the correspondences in decreasing order. It is started with checking coherence of mapping from 0.9, it results in coherence, then correspondence with 0.8 is also added to already added correspondence of 0.9, this results in incoherence. Then correspondence with 0.8 confidence value should be removed, but there are two correspondences having 0.8 confidence value. So *relevant* correspondence is identified for first discovered unsatisfiable artifact let it be m_3, then m_3 is removed and then remaining correspondences are checked and dealt for incoherence in similar manner as it is checked and dealt for m_3.

Qi and colleagues relate their approach to belief theory by considering that ontologies are more reliable than mapping and remove some of the correspondences in the mapping to restore consistency and define mapping revision operators.

The main drawback of Qi and colleagues work is that it does not remove minimal number of correspondences.

4.4.2.3 Patterns used for Debugging Ontology Mappings

Usually debugging of all defects caused by ontology mapping take more time and this in some circumstances is not acceptable. Patterns are used to speed up debugging ontology mapping but it is an incomplete solution, as not all defects are removed by using specific patterns. Patterns act like a query in debugging ontology mapping for locating defects. In this section, we discuss various patterns used for debugging ontology mappings. We classify patterns in two classes: patterns related to absolute defects and patterns related to relative defects.

Instead of treating complete ontology for checking defects caused by ontology mapping, Jiménez-Ruiz and colleagues (Jiménez-Ruiz et al., 2011) and Kutz and othres (Kutz et al., 2010) propose to use a modularize ontology. They propose to create module of each ontology involved in mapping. Each module consists of artifacts and axioms of only those signatures that are used in ontology mapping for respective ontologies.

In the following, we use two ontologies O_i and O_j. Artifacts of ontologies are distinguished by their index. We express ontology axioms in DL syntax and mapping as $\langle i{:}A, j{:}\alpha, r\rangle$ in examples relating to various patterns discussed below.

Mapping $\langle i{:}\, A\ j{:}\, \alpha\ r\rangle$ describe an artifact A of ontology O_i is related to an artifact α of ontology O_j by relation ($\sqsubseteq, \equiv$ $and\ others$). Disjointness axiom is expressed by using relation $\perp$ as $O_i{:}\, B \sqsubseteq O_i{:}\, \neg C$ expressed as $O_i{:}\, B \perp O_i{:}\, C$.

Combination of mapping and source ontologies $f(O_i, O_j, M_{ij})$ is treated as a logical theory and reasoning is performed on it by using OWL semantics. When some other type of semantics is used for $f(O_i, O_j, M_{ij})$ then we mention this explicitly.

We classify patterns into two classes (a) Patterns that debug absolute defects (b) Patterns that debug relative defects.

4.4.2.3.1 Patterns that debug absolute defects

Wang and colleagues named mappings that cause incoherence and/or inconsistency when mappings are combined with consistent and coherent source ontologies as inconsistent mappings (Wang & Xu, 2012).

Some of the patterns used for detecting absolute defects in ontologies caused by ontology mapping are listed below.

1. **Subsumption Disjointness Contradiction** (Jiménez-Ruiz et al., 2011)

When artifacts that are disjoint in ontology O_i are mapped to artifacts of ontology O_j that are in subsumption relation, then such mapping results in incoherence in ontology $f(O_i, O_j, M_{ij})$.

Example 4-19: Ontology O_i has artifacts B and C, where B and C are disjoint, are mapped to artifacts β and γ respectively of ontology O_j, where β and γ are in subsumption relation.

$$O_i{:}\, B \perp O_i{:}\, C$$

$$O_i{:}\, B \sqsubseteq O_i{:}\, A$$

$$O_i{:}\, C \sqsubseteq O_i{:}\, A$$

Axiom of ontology O_j is

$$O_j{:}\, \gamma \sqsubseteq O_j{:}\, \beta$$

$$O_j{:}\, \beta \sqsubseteq O_j{:}\, \alpha$$

Mapping $M = \langle \{O_i{:}\, A, O_j{:}\, \alpha, \equiv\}\{O_i{:}\, B, O_j{:}\, \beta, \equiv\}, \{O_i{:}\, C, O_j{:}\, \gamma, \equiv\}\rangle$.

Source Ontologies and mapping M are show in Figure 4-2.

Artifact $O_i{:}\, C$ becomes unsatisfiable in Combination of mapping and source ontologies $f(O_i, O_j, M_{ij})$.

$O_i: C \equiv O_j: \gamma \sqsubseteq O_j: \beta \equiv O_i: B$, i.e., $O_i: C \sqsubseteq O_i: B$ which is contradictory with axiom $O_i: B \perp O_i: C$, so artifact C becomes unsatisfiable and ontology $f(O_i, O_j, M_{ij})$ becomes incoherent.

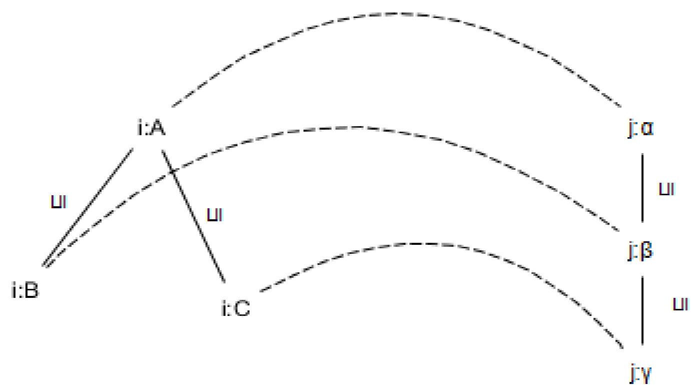

Figure 4-2. Subsumption-Disjointness contradiction

These errors are repaired by removing or modifying those correspondences that cause incoherence and/or logical inconsistencies.

2. **Subsumption propagation** (Meilicke & Stuckenschmidt, 2009)

This pattern is used by Meilicke and colleagues (Meilicke & Stuckenschmidt, 2009) for finding the violation of disjointness axiom. Mapping may be used to propagate subsumption in source ontologies and then check whether this propagation conflicts with existing axioms expressing disjointness in combination of source ontologies with mapping.

Example 4-20: Ontology O_i has axioms

$$i: B \sqsubseteq i: A$$

Axioms of Ontology O_j are

$$j: \gamma \sqsubseteq j: \beta$$

$$O_j \vDash j: \gamma \perp j: \alpha$$

Mapping is

$$M_{ij} = \langle \{i: A, j: \alpha, \sqsubseteq\}, \{i: B, j: \beta, \sqsupseteq\}$$

In combination of mapping and source ontologies $f(O_i, O_j, M_{ij})$, mapping M_{ij} add new entailments

$$f(O_i, O_j, M_{ij}) \vDash j: \gamma \sqsubseteq j: \alpha$$

$$f(O_i, O_j, M_{ij}) \vDash j: \beta \sqsubseteq j: \alpha$$

However, now we have $j{:}\gamma \sqsubseteq j{:}\alpha$ and $j{:}\gamma \sqsubseteq j{:}\neg\alpha$ and they are contradicting and this makes γ unsatisfiable and makes ontology $f(O_i, O_j, M_{ij})$ incoherent. This is shown in Figure 4-3.

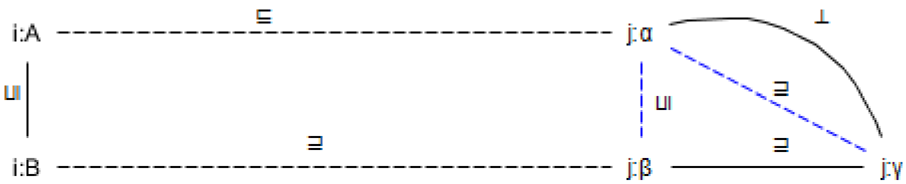

Figure 4-3. Subsumption propagation

Analogously, we can extend this pattern by using mappings involving equivalence relation and subsumption relation of artifacts propagates subsumption relation as above. For instance, $O_i{:}B \sqsubseteq A$, mapping $M = \langle\{i{:}A \equiv j{:}\alpha\}, \{i{:}B \sqsupseteq j{:}\beta\}\rangle$ will result in $j{:}\beta \sqsubseteq j{:}\alpha$. However, if in ontology O_j artifacts β and α are disjoint then contradiction arises and combination of source ontologies and mappings becomes incoherent.

3. **Disjointness propagation** (Meilicke & Stuckenschmidt, 2009)

This pattern is also used by Meilicke and colleagues (Meilicke & Stuckenschmidt, 2009) for finding the violation of disjointness axiom. Mapping may be used to propagate subsumption or disjointness in source ontologies and then check whether this propagation conflicts with existing axioms of source ontologies.

Example 4-21: Ontology O_i has axioms

$$i{:}B \perp i{:}A$$

Axioms of Ontology O_j are

$$j{:}\beta \perp j{:}\alpha$$

$$j{:}\gamma \sqsubseteq j{:}\beta$$

$$j{:}\gamma \sqsubseteq j{:}\alpha$$

Mapping is

$$M = \langle\{i{:}A \sqsupseteq j{:}\alpha\}, \{i{:}B \sqsupseteq j{:}\beta\}\rangle$$

In combination of mapping and source ontologies $f(O_i, O_j, M_{ij})$, mapping M add new entailments.

$$f(O_i, O_j, M_{ij}) \vDash j{:}\gamma \perp j{:}\alpha$$

However, $j{:}\gamma \sqsubseteq j{:}\alpha$ is contradicting with $j{:}\gamma \perp j{:}\alpha$. So, $j{:}\gamma$ becomes unsatisfiable and makes the combination of source ontologies and mappings $f(O_i, O_j, M_{ij})$ incoherent. This is shown in Figure 4-4.

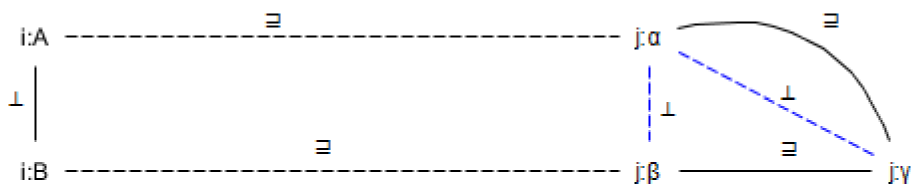

Figure 4-4. Disjointness propagation

It is not possible that such errors can be discovered by directional semantic mappings based on DDL logics since semantics of DDL allows artifacts of one ontology can be mapped to disjoint artifacts of another ontology (Homola, 2007) without causing inconsistency.

4.4.2.3.2 Patterns that locate relative defects

Principle of conservativity (sometimes also called as faithful interpretation between theories in the context of mapping) are used for locating some of the relative defects.

Principle of Conservativity (Jiménez-Ruiz et al., 2011)**:** It states that ontology mapping does not add any new information when source ontology(ies) are combined with ontology mapping while this added information is not available in source ontology(ies) alone.

In symbols, we represent this principle as

$f(O_i, O_j, M_{ij}) \vDash \alpha$ but $O_i \nvDash \alpha \vee O_j \nvDash \alpha \vee M \nvDash \alpha$

Example 4-22: Suppose that there are ontologies O_i and O_j with mapping M_{ij}.

Ontology O_i has axioms

$$i{:}B \sqsubseteq i{:}A$$

Ontology O_j consists of artifacts α and β among others and there is no implicit or explicit information about $\beta \sqsubseteq \alpha$.

Mapping M_{ij} contains $M_{ij} = \langle\{O_i{:}A \equiv O_j{:}\alpha\},\{O_i{:}B \equiv O_j{:}\beta\}\rangle$

When mapping M_{ij} is combined with source ontologies as $f(O_i, O_j, M_{ij})$, it results in entailment $\beta \sqsubseteq \alpha$. This shows that mapping add new information about O_j.

This principle can be used for checking incoherence. If for example O_j contains axiom $\beta \sqsubseteq \neg\alpha$. Then entailment $\beta \sqsubseteq \alpha$ obtained from $f(O_i, O_j, M_{ij})$ is contradicting with $\beta \sqsubseteq \neg\alpha$ and makes β unsatisfiable and make ontology $f(O_i, O_j, M_{ij})$.

In case of single source ontology having axiom $\beta \sqsubseteq \neg\alpha$ and mapping $M = \{O_i{:}A \equiv O_j{:}\alpha, O_i{:}A \equiv O_j{:}\beta\}$, violation of principle of conservativity in $f(O_i, M_{ij})$ can be checked. A solution to this problem is either to choose one of the correspondence or correct both mappings.

Deductive difference is computed for checking conservativity. Deductive difference refers to difference between entailments which are entailed by one ontology while it's not possible to entail same sentence in other ontology. Since there are many entailments and it is very difficult to handle all these entailments, Jiménez-Ruiz and colleagues (Jiménez-Ruiz et al., 2011) and Solimando and colleagues (Solimando et al., 2014) propose to use lighter version of conservativity principle which only deals with new subsumption relation. They only check for axioms of the form $A \sqsubseteq B$, i.e., subsumption. $O_i \nvDash A \sqsubseteq B$, $O_j \vDash A \sqsubseteq B$, in this case $A \sqsubseteq B$ is a difference between O_i an O_j. We can find the deductive difference between each source ontology and $f(O_i, O_j, M_{ij})$ for checking the principle of conservativity.

Subsumption propagation and Disjointness propagation patterns can also be used for checking conservativity of ontologies in some cases.

In Enderton's approach of interpretation between theories, faithful interpretation corresponds to principle of conservativity. Mapping $\Pi: T_0 \rightarrow T_1$ Π is faithful interpretation of T_0 into T_1 if for every σ there is $T_0 \vDash \sigma \leftrightarrow T_1 \vDash \pi(\sigma)$, i.e., T_1does not entail any new inference for the signatures Σ_0 which is not entailed by T_0.

Many patterns that are based on the intuition of principle of conservativity are frequently used for locating both absolute and relative defects.

1. **Bow-tie** (Hovy, 1998), (Wang & Xu, 2012)

   When artifacts of ontology O_i which are in subsumption relation are mapped to artifacts of ontology O_j which are in reverse subsumption relation, they form bow-tie like structure.

Example 4-23: Suppose that Ontology O_i has axioms

$$i{:}C \sqsubseteq i{:}B$$

$$i{:}B \sqsubseteq i{:}A$$

$$j{:}\gamma \sqsubseteq j{:}\beta, j{:}\beta \sqsubseteq j{:}\alpha$$

$$M = \{i{:}C \equiv j{:}\beta, i{:}B \equiv j{:}\gamma\}$$

Mapping M and artifacts of source ontologies involved in mappings make Bow-tie structure as shown in Figure 4-5. Note that, artifacts of a source ontologies forming bow-tie structure may not be in direct parent-child relation. This pattern is commonly used (Hovy, 1998), (Wang & Xu, 2012).

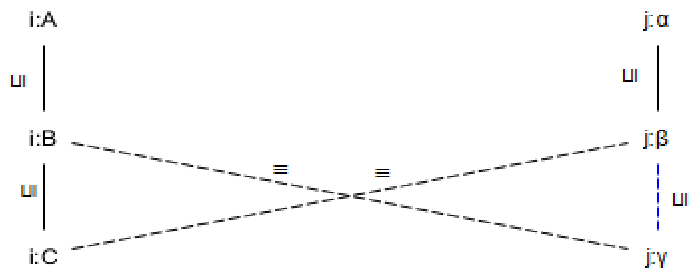

Figure 4-5. Mapping formed Bow-tie structure

In Figure 4-5, mappings formed bow-tie (sometimes it is also referred as is-a circle/cycle or criss-cross) (Wang & Xu, 2012). There are two cycles one in each source ontology which are $i{:}\, C \sqsubseteq i{:}\, B \sqsubseteq i{:}\, C$ and $j{:}\, \gamma \sqsubseteq j{:}\, \beta \sqsubseteq j{:}\, \gamma$.

Bow-tie structure collapse hierarchies, in this case $i{:}\, B \equiv i{:}\, C$ and $j{:}\, \beta \equiv j{:}\, \gamma$, this pattern also tells that mapping adds new information in ontology $f(O_i, O_j, M_{ij})$ and this new information is not present in source ontology(ies).

2. **Equivalent artifacts mapped to not equivalent artifacts** (Jean-Mary et al., 2010)

   When artifacts that are equivalent in ontology O_i are mapped to artifacts that are not equivalent of ontology O_j in equivalence relation, then such mapping do not result in incoherence in ontology if and only if mapped artifact of ontology O_j are in a relation other than the disjointness relationship.

Example 4-24: Suppose that there are ontologies O_i and O_j with mapping M_{ij}.

Ontology O_i has axiom

$$i{:}\, A \equiv i{:}\, B$$

Ontology O_j has artifacts α and β and there is a relation r between them.

$j{:}\, \alpha\ r\ j{:}\, \beta$, where r is a relation other than equivalence and disjointness.

Mapping M_{ij} is

$$\langle i{:}\, A \equiv j{:}\, \alpha\}, \{\, i{:}\, B \equiv j{:}\, \beta\}\rangle$$

When $j{:}\, \alpha$ and $j{:}\, \beta$ are not equivalent, then user may consider this as unwanted result in $f(O_i, O_j, M_{ij})$ as mapping adds new information about α and β that they are equal by changing the existing relation r between artifacts α and β to equivalence relation. This is shown in Figure 4-6.

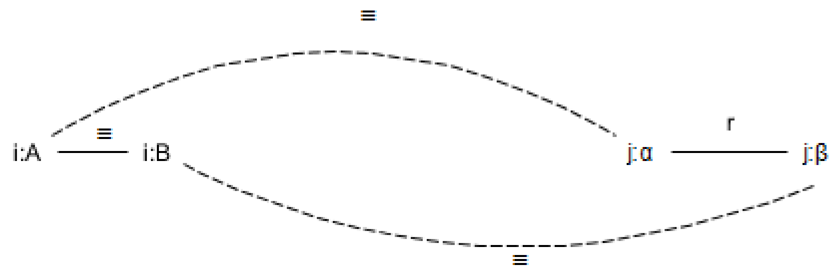

Figure 4-6. Equivalence axiom violation; r represents relation other than equivalence

3. **Domain range incompleteness** (Jean-Mary et al., 2009)

   Sometimes it is required to enrich ontology with new information from ontology mapping. We consider this as abuse of ontology mapping and consider it as a defect.

Example 4-25: Suppose that there are ontologies O_i and O_j with mapping M_{ij}.

Ontology O_i has artifacts $i{:}\,A$ as concept and $i{:}\,s$ as property. Ontology O_j has artifacts $i{:}\,A$ as concept and $i{:}\,\sigma$ as property.

In mapping M_{ij}, when concept $i{:}\,A$ is mapped to concept $j{:}\,\alpha$ and property s is mapped to property σ. If concept $i{:}\,A$ is domain (range) of property s then α should be domain (range) of σ. If this is not the case, then mapping indicates that ontology O_j is incomplete. ASMOV (Jean-Mary et al., 2010) software detects such defects in $f(O_i, O_j, M_{ij})$ by finding mapped properties and then check whether concepts that are their domain (range) are equivalent, if they are not equivalent, then there is an error.

ASMOV (Jean-Mary et al., 2010) software detects new subsumption relations in $f(O_i, O_j, M_{ij})$, when artifact $i{:}\,A$ is mapped to artifact $j{:}\,\alpha$ and artifact $i{:}\,B$ is mapped to artifact $j{:}\,\beta$ and ontology O_i has axiom that A is subsumed by B and when α is not subsumed by B then such mappings indicate that subsumption relation is missing in ontology O_j.

4.4.2.4 Empirical ways

Sometimes empirical ways and heuristics are used to detect some of the relative defects in ontology.

1. **Principle of Locality** (Wang & Xu, 2012)**,** (Jiménez-Ruiz et al., 2011)

   Jiménez-Ruiz and colleagues (Jiménez-Ruiz et al., 2011) use a locality principle which states that artifacts which are semantically similar in one ontology should be mapped to semantically related artifacts of another ontology. They propose to calculate whether mapped artifacts are semantically related by calculating the confidence measure by adding the number of artifacts that are mapped to semantically related artifacts of another ontology and divided this by number of semantically related artifacts in both ontologies.

(Wang & Xu, 2012) terms mappings that violate locality principle as *abnormal*. When artifacts of ontology that are closely related are mapped to artifacts of another ontology which are not closely related.

Example 4-26: Suppose that there are ontologies O_i and O_j with mapping M_{ij}.

Ontology O_i has axioms

$$i: B \sqsubseteq i: A$$

$$i: C \sqsubseteq i: A$$

$$i: B \perp i: C$$

Ontology O_j has artifact $j: \alpha$

Mapping M_{ij} is

$$i: A: \equiv j: \alpha$$

$$i: B \equiv j: \alpha$$

$$i: C \equiv j: \alpha$$

Source ontologies and mappings are shown in Figure 4-7.

In this example B and C are not closely related, in fact, they are disjoint.

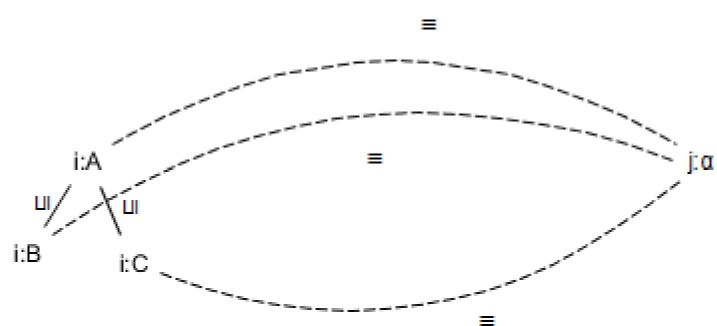

Figure 4-7. Equivalence axiom violation

Then mapping when combined with source ontologies as $f(O_i, O_j, M_{ij})$ destroys taxonomy of ontology O_i and making artifacts equivalent $i: C \equiv i: A \equiv i: C$. This also causes incoherence since B and C are disjoint in O_i but here we are considering this in different point of view. Here, B and C are semantically unrelated and are mapped to an artifact; principle of locality is checked on all the mapped artifacts and then checked whether it corss the threshold value and if this is the case then it is considered as defect. In this example, mapping causes incoherence but it is not necessary that when principle of locality is violated incoherence occurs. Violation of principle of locality roughly represents that either mapping is not *imprecise* or *abnormal*.

Defect of semantic inconsistency can be detected to some extend by using principle of locality. Kiwi Fruit and Kiwi bird in Example 1-2 are disjoint artifact in the ontology and if they are mapped to same artifact or closely related artifact then it can be judged as a defect on the basis of principle of locality.

2. **Heuristics to deal with Imprecise mappings** (Wang & Xu, 2012)

   If mappings are imprecise and one concept is mapped to multiple concepts in a relation r as in $C \overset{r}{\mapsto} \{D_i\}$ where $2 \leq i \leq k$ then if relation r is $\sqsubseteq$ then more precise mapping is $C \sqsubseteq \wedge D_i$, if relation r is $\sqsupseteq$ then more precise mapping is $C \sqsubseteq \vee D_i$ and if relation r is $=$ then more precise mapping is $C \sqsubseteq \vee D_i$.

Some of the defects can be checked only by combining one ontology with mapping $f(O_{j,}M)$. For instance, Suppose that ontology O_j has artifacts α and β which are disjoint $\alpha \sqsubseteq \neg\beta$, when they are combined with mapping $M = \{O_i: A \equiv O_j: \alpha, O_i: A \equiv O_j: \beta, \}$ as $O_j \cup M$ then it can be infered that artifact A and B are equivalent as $\alpha \equiv \beta$ which is contradicting with ontology axioms of O_j and in this case make α and β unsatisfiable.

Jiménez-Ruiz and colleagues present conservativity principle as $f(O_j, M)$ should not infer α when α cannot be inferred from O_j alone (Jiménez-Ruiz et al., 2011). This principle may be used not only locating absolute defects but also relative defects like mapping add new information to source ontologies.

4.5 Synthesis

We have identified that there are three kinds of ontology mapping defects; syntactical, absolute, and relative defects. Syntactical defect is very easy to correct; it just require a parser to verify that the syntax of ontology mapping are correct. Specific techniques are required to correct absolute and relative defects. Absolute defects are more critical while relative defects are user-dependent, i.e., some user may ignore these defects.

We conclude that ontology mapping should not cause incoherence and inconsistency and this feature should be present in ontology mappings. Additionally, debugging relative defects sometimes requires domain knowledge about the ontology while in case of debugging absolute defects does not require domain knowledge about the ontology.

Interdependency of axioms of ontology and interdependency of correspondences in mapping make debugging ontology and ontology mapping a complex task. A trivial solution in debugging is to remove all axioms and/or mappings but then it is better to not debug. Instead of this, the focus of debugging ontology and ontology mapping is to correct the defects with *minimal change*.

The basic assumption in debugging ontology mapping is that ontologies are coherent and consistent and when mappings are combined with source ontologies arises any defect then it is caused by ontology mapping. There are some debugging approaches propose to repair defects by repairing ontologies instead of ontology mappings (Fahad et al., 2012). Also, these days ontologies are organized in repositories. The limitation of these approaches is that ontologies are shared and especially ontology owner does not allow modifying the ontologies. The reason of not allowing modification in ontology is that ontologies are shared and some users consider that they do not need to modify ontology or they do not agree that there is any defect in the ontology.

We analyze how mappings in a particular formalism behave. We study questions as whether defects in one formalism are also consider as defect in other formalism? How mappings are expressed in a particular formalism? Here, we discuss formalisms IDDL, DDL, Institution theory, Enderton Interpretation between theories. We discuss IDDL since it takes third party point of view about source ontologies in mapping which in some sense relates to OWL in which mappings act as axiom of new merged ontology. We discuss DDL since it consider mappings as a point of view of one particular ontology about another ontology, Institution theory since it is algebraic formalism which involves both category theory and logics. Enderton approach of interpretation between theory is discussed here since it provides a way to check whether one ontology is as powerful as another and mappings are used to check this condition.

Mappings are represented by a set of correspondences (e_i, e_j, r) where e_i represents artifact (simple or complex) of ontology O_i and e_j represents artifact (simple or complex) of ontology O_j where r is the relation between e_i and e_j where relations like subsumption $\sqsubseteq$ are not symmetric so such relations represents that artifact of ontology O_i are in subsumption $\sqsubseteq$ relation with artifact of ontology O_j.

However, in some formalisms this information of relation r is generally treated as an implicit equivalence relation. In such cases, one can argue that this information represents that when mappings are established between artifacts it was not sure that whether mapped artifacts are equivalent or not so they want to coerce this information by using some weaker relation than equivalence. In Institution theory, morphisms are used to treat mappings; morphisms are very general and they can be considered as a function between two mapped artifacts and this function can be any relation but in general mapping represents that mapped artifact of an ontology is completely represented by artifacts of another ontology involved in a mapping. In Enderton approach of interpretation between theory, mapping relation is not explicitly used but the guideline is that functions are mapped to functions and predicates are mapped to predicates in such a way that models of one ontology involved in mapping become models of another ontology.

Here, an example is presented in which ontologies are formalized in description logic and mappings are formalized in respected formalism of each approach considered here. Then it is checked how defects in one formalism behave in another formalism.

Example 4-27: Suppose that there are two ontologies O_i that deals with ontology of research organization while O_j deals with ontology of university. Mappings are shown under a tabular form. We do not take all correspondences listed below when we are analyzing a specific situation, we only mention that which correspondences are used in that situation.

(i1) $O_i{:}\,Research\ Org. \sqsubseteq O_i{:}\,\top$
(i2) $O_i{:}\,Research\ Staff \sqsubseteq O_i{:}\,Research\ Org.$
(i3) $O_i{:}\,Admin\ Staff \sqsubseteq O_i{:}\,Research\ Org.$
(i4) $O_i{:}\,Director\ Admin \sqsubseteq O_i{:}\,Admin\ Staff$
(i5) $O_i{:}\,Computer\ Scientist \sqsubseteq O_i{:}\,Research\ Staff$
(i6) $O_i{:}\,Social\ Scientist \sqsubseteq O_i{:}\,Research\ Staff$
(i7) $O_i{:}\,Research\ Officer \sqsubseteq O_i{:}\,Computer\ Scientist$
(i8) $O_i{:}\,Research\ Officer \sqsubseteq O_i{:}\,Social\ Scientist$
(i9) $O_i{:}\,\bot \sqsubseteq O_i{:}\,Research\ Officer$
(i10) $O_i{:}\,\bot \sqsubseteq O_i{:}\,Director\ Admin$
(i11) $O_i{:}\,Research\ Staff \sqcap O_i{:}\,Admin\ Staff \sqsubseteq \bot$
(i12) $O_i{:}\,Research\ Officer(Abraham)$

(j1) $O_j{:}\,University \sqsubseteq O_j{:}\,\top$
(j2) $O_j{:}\,Teaching\ Faculty \sqsubseteq O_j{:}\,University$
(j3) $O_j{:}\,Admininstrative\ Staff \sqsubseteq O_j{:}\,University$
(j4) $O_j{:}\,Director \sqsubseteq O_j{:}\,Administrative\ Staff$
(j5) $O_j{:}\,Director\ Admin \sqsubseteq O_j{:}\,Director$
(j6) $O_j{:}\,Researcher \sqsubseteq O_j{:}\,Teaching\ Faculty$
(j7) $O_j{:}\,\bot \sqsubseteq O_j{:}\,Researcher$
(j8) $O_j{:}\,\bot \sqsubseteq O_j{:}\,Director\ Admin$
(j9) $O_j{:}\,Admin\ Staff \sqcap O_j{:}\,Teaching\ Faculty \sqsubseteq \bot$
(j10) $O_j{:}\,Director(Adam)$

Correspondences of the mapping in various formalisms is shown in Table 4-2. In the table e_i and e_j are artifacts of ontology and they may be a complex concept (formed by the combination of atomic concepts using logical connectives). r is a relationship. We write mapping in DDL formalism according to Ontology O_i perspective and note that mapping according to ontology O_j perspective can be different from O_i perspective. Mapping represented in one formalism may not be exactly translated into another formalism. But a user is expecting that the formalism she/he is using corresponds to her perspective and this may results into semantic inconsistencies or some other defects. Especially, Institution and Enderton approach of interpretation between theories do not exactly correspond to mappings in which relation $\sqsubseteq$ and $\sqsupseteq$ are differentiated. In these approaches, same translation is possible for both case or it should be considered that these approaches are not capable to deal such mappings. We, here, translate them to same syntax.

Table 4-2. Mappings in various formalisms

	Mapping $\langle e_i, e_j, r\rangle$	**OWL** $e_i\ r\ e_j$	**IDDL** $e_i \overset{r}{\leftrightarrow} e_j$	**DDL** $e_i \overset{r}{\rightarrow} e_j$	**Institution** $\sigma: e_i \mapsto e_j$	**Enderton** $\pi_{e_i}(x) = e_j(x)$, where e_i is symbol and $e_j(x)$ is a formula
m1	O_i:Research Org, O_j:University, $\sqsubseteq$	O_i:Research Org $\sqsubseteq$ O_j:University	O_i:Research Org $\overset{\sqsubseteq}{\leftrightarrow}$ O_j:University	O_i:Research Org $\overset{\sqsubseteq}{\rightarrow}$ O_j:University	O_i:Research $\mapsto$Org, O_j:Univ ersity	O_i:Research = Org, O_j:University
m2	O_i:Research Staff, O_j:Teaching Faculty, $\sqsubseteq$	O_i:Research Staff $\sqsubseteq$ O_j:Teaching Faculty	O_i:Research Staff $\overset{\sqsubseteq}{\leftrightarrow}$ O_j:Teaching Fa culty	O_i: *Research* Staff $\overset{\sqsubseteq}{\rightarrow}$ O_j:Teaching Facult*y*	O_i:Research Staff $\mapsto$ O_j:Teaching Faculty	O_i:Research Staff = O_j:Teaching Faculty
m3	O_i:Admin Staff , O_j:Director , $\sqsupseteq$	O_i:Admin Staff $\sqsupseteq$ O_j:Director	O_i:Admin Staff $\overset{\sqsupseteq}{\leftrightarrow}$ O_j:Director	O_i:Admin Staff $\overset{\sqsupseteq}{\rightarrow}$ O_j: Director	O_i:Admin Staff$\mapsto$, O_j:Director	O_i:Admin Staff = O_j:Director
m4	O_i:Director Admin , O_j:Admin istrative Staff, $\sqsupseteq$	O_i:Director Admin $\sqsupseteq$ O_j:Administrative Staff	O_i:Director Admin $\overset{\sqsupseteq}{\leftrightarrow}$ O_j:Administrat ive Staff	O_i:Director Admin $\overset{\sqsupseteq}{\rightarrow}$ O_j:Administrativ e Staff	O_i:Director Admin $\mapsto$ O_j:Administra tive Staff	O_i:Director Admin = O_j:Administrative Staff
m5	O_i:Admin Staff , O_j:Director , $\sqsubseteq$	O_i:Admin Staff $\sqsubseteq$ O_j:Director ,	O_i:Admin Staff $\overset{\sqsubseteq}{\leftrightarrow}$ O_j:Director	O_i:Admin Staff $\overset{\sqsubseteq}{\rightarrow}$ O_j: Director	O_i:Admin Staff$\mapsto$, O_j:Director	O_i:Admin Staff = O_j:Director
m6	O_i: *Research* Offic er, O_j:Director, $\sqsupseteq$	O_i: *Research* Offic er $\sqsupseteq$ O_j:Director	O_i: *Research* Offic er$\overset{\sqsupseteq}{\leftrightarrow}$ O_j:Director	O_i:Research Officer $\overset{\sqsupseteq}{\rightarrow}$ O_j:Director	O_i: *Research* Officer$\mapsto$ O_j:Director	O_i: *Research* Offic er = O_j:Director

	Mapping $\langle e_i, e_j, r\rangle$	**OWL** $e_i\ r\ e_j$	**IDDL** $e_i \overset{r}{\leftrightarrow} e_j$	**DDL** $e_i \overset{r}{\rightarrow} e_j$	**Institution** $\sigma: e_i \mapsto e_j$	**Enderton** $\pi_{e_i}(x) = e_j(x)$, where e_i is symbol and $e_j(x)$ is a formula
m7	O_i:Rerchercher , O_j:Director, $\ggg$	not possible	not possible	not possible	not possible	not possible
m8	O_i:Computer Scie ntist, O_j:Research er, $\sqsubseteq$	O_i:Computer Scie ntist $\sqsubseteq$ O_j:Researcher	O_i:Computer Scie ntist$\overset{\sqsubseteq}{\leftrightarrow}$ O_j:Researc her	O_i:Computer Scienti st$\overset{\sqsubseteq}{\rightarrow}$ O_j:Researcher	O_i:Computer S cientist$\mapsto$ O_j:Researcher	O_i:Computer Scie ntist = O_j:Researcher
m9	O_i:Social Scientist , O_j:Researcher, $\sqsubseteq$	O_i:Social Scientist $\sqsubseteq$ O_j:Researcher	O_i:Social Scientist $\overset{\sqsubseteq}{\leftrightarrow}$ O_j:Researcher	O_i:Social Scientist $\overset{\sqsubseteq}{\rightarrow}$ O_j:Researcher	O_i: *Social* Scie ntist$\mapsto$ O_j:Researcher	O_i: *Social* Scientis t = O_j:Researcher
m10	O_i:Computer Scie ntist, O_j:Teaching Faculty, $\sqsubseteq$	O_i:Computer Scie ntist $\sqsubseteq$ O_j:Teaching Facul ty	O_i:Computer Scie ntist$\overset{\sqsubseteq}{\leftrightarrow}$ O_j:Teachin g Faculty	O_i:Computer Scienti st$\overset{\sqsubseteq}{\rightarrow}$ O_j:Teaching Fac ulty	O_i:Computer S cientist, O_j:Tea ching Faculty	O_i:Computer Scie ntist, O_j:Teaching Faculty
m11	O_i:Computer Scie ntist, O_j:Research er, =	O_i:Computer Scie ntist $\equiv$ O_j:Researcher	O_i:Computer Scie ntist$\overset{\equiv}{\leftrightarrow}$ O_j:Researc her	O_i:Computer Scienti st$\overset{\sqsubseteq}{\rightarrow}$ O_j:Researcher O_i:Computer Scienti st$\overset{\sqsupseteq}{\rightarrow}$ O_j:Researcher	O_i:Computer S cientist$\mapsto$ O_j:Researcher	O_i:Computer Scie ntist = O_j:Researcher
m12	O_i:Social Scientist , O_j:Researcher, =	O_i:Social Scientist $\equiv$ O_j:Researcher	O_i:Social Scientist $\overset{\equiv}{\leftrightarrow}$ O_j:Researcher	O_i:Social Scientist $\overset{\sqsubseteq}{\rightarrow}$ O_j:Researcher O_i:Social Scientist $\overset{\sqsupseteq}{\rightarrow}$ O_j:Researcher	O_i: *Social* Scie ntist$\mapsto$ O_j:Researcher	O_i: *Social* Scientis t = O_j:Researcher

We present different situations in which a particular ontology mapping defect in one formalism may be either considered as defect (may have different name) or even not considered as defect in some other formalism. Cases of logical inconsistency, violation of principle of conservativity, imprecise mappings, and abnormal mappings are presented below.

1. **Case of Logical Incoherence:**

Example 4-28:

Mapping (m3,m6) when combined with source ontologies O_i and O_j, we check what happen in each formalism in the ontology $f(O_i, O_j, M_{ij})$

In IDDL, it results in $O_j{:}Director \sqsubseteq O_i{:}ResearchOfficer$ and $O_j{:}Director \sqsubseteq O_i{:}Admin\ Staff$, but $Adminin\ Staff$ and $Research\ Staff$ are disjoint which results into $Research\ Officer$ and $Admininstrative\ Staff$ are disjoint (by i5,i7,i11), i.e.,

$$O_i{:}Research\ Staff \sqcap O_i{:}Admin\ Staff \sqsubseteq \bot$$

But mapping m3, m6 results into

$$O_j{:}Director \sqsubseteq O_i{:}Research\ Officer \sqcap O_i{:}Admin\ Staff$$

So, these two axioms cannot be satisfied simultaneously and $O_j{:}Director$ becomes unsatisfiable and this makes ontology $f(O_i, O_j, M_{ij})$ incoherent.

In OWL, same situation as presented above in IDDL for this case.

In DDL, correspondence (m3) is treated as $r_{ij}(O_i{:}Admin\ Staff) \sqsupseteq O_j{:}Director$ and correspondence $(m6)$ is treated as $r_{ij}\left(O_j{:}Research\ Officer\right) \sqsupseteq O_j{:}Director$. Even though $O_i{:}Admin\ Staff$ and $O_j{:}Research\ Officer$ are disjoint but semantics of DDL permits inconsistent and partial knowledge.

In **institution theory**, mappings are treated as signature morphism and this induces sentence translation. For $\phi \in Sen(\Sigma)$ and $\sigma{:}\,\Sigma \mapsto \Sigma'$ is a morphism, satisfiability condition in institution theory is $M' \vDash_{\Sigma'} \sigma(\phi) \Leftrightarrow M \upharpoonright_\sigma \vDash_\Sigma \phi$. So with correspondences (m3,m6) satisfiability condition is not respected as $M' \vDash_{\Sigma'} Director$ is true but ($M \upharpoonright_\sigma \vDash_\Sigma Admin\ Staff$ and $M \upharpoonright_\sigma \vDash_\Sigma Research\ Officer$) is not true. So, this is a defect in which $f(O_i, O_j, M_{ij})$ is not an institution. Incoherence is not a terminology of Institution, but this can be consider a defect as in such setting (with these mappings and ontologies) there can be no institution and in IDDL and DDL referred to as logical inconsistency.

In **Enderton approach of interpretation between theories**, mapping (m3,m6) will map artifacts of ontology O_i to O_j. Enderton approach requires that all the mappings by which O_i maps to O_j *considering that other relevant correspondences are there* (since for interpretation between theories, Enderton approach requires that all symbols of a theory are translated into formulas of other theory) then correspondences m3,m6 will result into a situation when O_i is not interpreted in O_j. Since, we have mapping $O_i{:}\,Admin\ Staff = O_j{:}\,Director$ and $O_i{:}\,Research\ Officer = O_j{:}\,Director$, so the ontology O_j satisfies mapping of $O_i{:}\,Admin\ Staff$ and mapping of $O_i{:}\,Research\ Officer$, however, the inference $O_i{:}\,Admin\ Staff \perp O_i{:}\,Research\ Officer$ cannot be satisfiable in ontology O_j under the given mapping. Hence, it shows that with this mapping, ontology O_i is not interpretable in O_j.

2. **Case of Logical Inconsistency:**

Example 4-29: This example is extension of Example 4-28, now (j10) Director(Adam) is also considered.

Mapping (m3,m6) when combined with source ontologies O_i and O_j (especially to see the effect of (j10) Director(Adam)), we check what happen in each formalism in the ontology $f(O_i, O_j, M_{ij})$

In IDDL, instantiating unsatisfiable concept causes inconsistency. So, when we have (j10) Director(Adam), ontology $f(O_i, O_j, M_{ij})$ becomes inconsistent.

In OWL, same situation as presented above in IDDL for this case.

In DDL, (j10) Director(Adam) does not make ontology $f(O_i, O_j, M_{ij})$ inconsistence. The reason is that DDL permits empty set of models called hole. Even though local models of O_j are empty, but $f(O_i, O_j, M_{ij})$ is not inconsistent since models of O_i are not empty. It is because according to the semantics of DDL, ther exists a model $\mathcal{M}$ such that $\mathcal{M} \vDash i{:}\perp$ and $\mathcal{M} \nvDash j{:}\perp$.

In **institution theory**, correspondences (m3,m6) does not respect the conditions of institution. So adding new information (j10) Director(Adam) does not make this institution. And $f(O_i, O_j, M_{ij})$ is not an institution. Inconsistency is not a terminology used in Institution to refer the fact that something is not an institution, however, this case remains a defect in institution.

In **Enderton approach of interpretation between theories**, mapping (m3,m6) makes ontology O_i is not interpretable in O_j, so adding new information (j10) Director(Adam) does not make ontology O_i is interpretable in O_j.

3. **Case of violation of Principle of Conservativity and Abnormal mappings**

Example 4-30: When mapping (m4,m5) are combined with source ontologies O_i and O_j, then

In IDDL, it results in adding new information for source ontology O_j and this information was not there in O_j before combining mapping with source ontologies.

Ontologies have axioms

$$O_i{:}\,Director\ Admin \sqsubseteq O_i{:}\,Admin\ Staff$$

$$O_j: Director \sqsubseteq O_j: Administrative\ Staff$$

And mapping

$$O_i: Admin\ Staff \xrightarrow{\sqsubseteq} O_j: Director$$

$$O_i: Director\ Admin \xrightarrow{\sqsupseteq} O_j: Administrative\ Staff$$

When mapping is combined with source ontologies as $f(O_i, O_j, M_{ij})$ then we have

$$O_j: Director \sqsubseteq O_i: Admin\ Staff$$

$$O_i: Admin\ Staff \sqsubseteq O_j: Administrative\ Staff$$

And this results in the following new entailment

$$O_j: Admininstrative\ Staff \sqsubseteq O_j: Director$$

There is some additional information in ontology O_j which is $O_j: Admin\ Staff \equiv O_j: Director$ (because now we have $O_j: Director \sqsubseteq O_j: Administrative\ Staff$ and $O_j: Admininstrative\ Staff \sqsubseteq O_j: Director$) and this information is not the present before combining source ontologies and mappings. So, this is violation of principle of conservativity.

In OWL, same situation as presented above in IDDL for this case.

In DDL, same information $O_j: Admin\ Staff \equiv O_j: Director$ is inferred in this case.

In OWL, IDDL and DDL this situation is not an absolute defect, rather it is a defect due to violation of principle of conservativity. Mapping should not add new information. Some kind of deductive difference is used to trace this defect in logical approaches.

In institution theory, this mapping respects the satisfiability condition of institution. In Institution theory, it requires some additional mechanism to trace this defect since it does not violate satisfiability condition of institution theory.

In Enderton approach of interpretation between theories, mappings that contain these correspondences *considering that other relevant correspondences are there* (since for interpretation between theories, Enderton approach requires that all symbols of a theory are translated into formulas of other theory) are not interpretation, because maping of $O_i: Director \sqsubseteq O_i: Admin\ Staff$ as $O_j: Admininstrative\ Staff \sqsubseteq O_j: Director$ axiom in O_j is not true in ontology O_j.

In Enderton approach of interpretation between theories, the violation of principle of conservativity can be detected by checking that mapping is a faithful interpretation or not.

Example 4-28 is also a special case of violation of principle of conservativity in which ontology becomes incoherent.

4. **Case of Imprecise mappings**

Example 4-31: When there is a mapping with correspondences (m8,m10) then it is not clear from mapping that how to treat these mappings as conjunctive or disjunctive when same artifact of one ontology map to artifacts of another ontology in two way. In mapping m8, $O_i\text{: }Computer\ Science$ is mapped to $O_i\text{: }Researcher$ while in correspondence it is mapped to $O_j\text{: }Teaching\ Faculty$ respectively. This case depicts impreciseness of mapping.

This case persists in all the formalisms discussed in this subsection, some additional information is required before using such mappings to know that whether they should be interpreted as conjunction or disjunction.

5. **Case of Syntactic defects:**

Example 4-32: When there is a mapping with correspondence (m7) as $O_i\text{: }Rechercher \ggg O_j\text{: }Director$ it is a syntactical defect; since we don't have artifact $O_i\text{: }Rechercher$ and operator ⋙ is undefined in the language representing mapping. In all these formalisms, it is a syntactical defect because ⋙ is not a well-defined relation to be used in mapping.

This case persists in all the formalisms discussed in this subsection, but could be removed before using mappings.

6. **Case of Semantic inconsistency:**

Example 4-33: When there is a mapping with correspondences (m11,m12) then it results into making semantically different artifacts $O_i\text{: }Computer\ Scientist$ and $O_i\text{: }Social\ Scientist$ equal. Although, this mapping also results into logical inconsistency but this also destroy the taxonomy of ontology and violate principle of conservatity. However, we are taking this example from other perspective, we are viewing this case that semantically distinct artifacts are mapped to same or semantically similar artifacts.

Principle of locality can be partially helpful in detecting such defects, as it may be the case that two artifacts of an ontology are at same semantic distance but only one of them is more appropriate for mapping in given situation. This case is not detectable in formalism discussed in this subsection. It requires putting some threshold value; if that threshold is not achieved by mapping then it should be perceived as a defect.

Mapping defects in each formalism discovered in Example 4-28 to Example 4-33 are listed in Table 4-3.

Table 4-3. Mapping defects in each formalism

Mapping	IDDL	DDL	Institution	Enderton approach on interpretation between theories
m3,m6	Logical Incoherence	No defect	Not an Institution	O_i is not Interpretable in O_j
m3,m6 (when seen with (j10))	Logical Inconsistency	Only Hole interpretation satisfies ontologies and mappings	Not an Institution	O_i is not Interpretable in O_j
m4, m5	Violation of Principle of conservativity	Violation of Principle of conservativity	Violation of Principle of conservativity	O_i is not Interpretable in O_j
m8,m9	Impreciseness (should made precise before using mappings)	Impreciseness (should made precise before using mappings)	Impreciseness (should made precise before using mappings)	Impreciseness (should made precise before using mappings)
m11,m12	Semantic Inconsistency (not detected)	Semantic Inconsistency (not detected)	Semantic Inconsistency (not detected)	Semantic Inconsistency (not detected)
m7	Syntax error	Syntax error	Syntax error	Syntax error

This table shows that defect in one formalism may not be a defect in other formalism. Debugging defect depends on specific logic when using logic based formalism and when user change the underlying formalism of ontologies (i.e., move from one logic to another logic), then defects may not remain defects or some new defect arises. While in case of Institution theory it requires some other mechanism to debug relative defects like violation of principle of conservativity. In case of Enderton approach of interpretation between theories, it is required to check for whether mapping is an interpretation and/or faithful interpretation for debugging defects.

4.6 Ontology Mapping Acceptability

Ability to argue logically is a key skill. It is valuable not only in the field of research but also in everyday life. Argumentation techniques provide good understanding of reasoning and it is widely recognized as an important mechanism for practical reasoning. Based on argumentation involved parties reach on some common acceptable point.

An *argument* consists of a set of assumptions (*support*) together with a conclusion (*claim*). The support of an argument provides reason (justification) of an argument. Arguments can rebut each other. A *rebutting argument* is an argument with the claim that is negation of another argument. An argument can contradict some of the assumptions of another argument and such arguments are called *undercutting argument*. A *counter argument* is either rebutting argument or undercutting argument. *Argumentation* is a process in which arguments and counter arguments are handled. Argument has *proponents* and *audience; proponents* are those who put forward arguments and *audience* is recipients of arguments.

Example 4-34: Let α, β and γ are arguments. We discuss arguments here from classical logic point of view.

The claim $\neg\alpha \vee \beta$ is a *contradiction* of another claim $\alpha \wedge \neg\beta$.

If an argument A_1 has a claim $\neg\alpha \vee \beta$ and argument A_2 has a claim $\alpha \wedge \neg\beta$ then A_1 and A_2 *rebut* each other.

If an argument A_1 has information $\{\neg\alpha \vee \beta, \alpha\}$ and claim that β holds, then an argument A_2 having a claim $\neg\alpha$ is an *undercutting* argument for A_1.

Argumentation Framework is a logical framework proposed by Dung (Dung, 1995) which is used for arriving at a consensus whenever there are conflicting arguments. Dung's Framework provides semantics to argumentation based logics and allows a precise comparison between different systems on the basis of arguments. Basic concepts about Dung's Argumentation Framework are described below.

Definition 4-6 (Argumentation Framework): (Dung, 1995) An Argumentation Framework (AF) is defined as $AF = \langle AR, A\rangle$, where AR is a set of arguments and $A \subset AR \times AR$ is a binary relation on AR called as attacks, i.e., A is a set of ordered pairs of distinct arguments in AR. A pair $\langle x, y\rangle$ is attack and it represents "x attacks y".

Let R, S be subsets of AR, then

a) $s \in S$ is *attacked* by R if there is some $r \in R$ such that $\langle r, s\rangle \in A$.
b) $x \in AR$ is *acceptable* with respect to S if $\forall y \in AR$ that attacks x, $\exists z \in S$ that attacks y (i.e., z in S defends x against y).
c) S is *conflict free* if no argument in S is attacked by any other argument in S.
d) A conflict free set is *admissible* if every argument in S is acceptable with respect to S.
e) S is a *preferred extension* if it is a maximal (with respect to set inclusion) admissible subset of AR.

In Dung's framework, attacks always succeed, this is reasonable when dealing with deductive arguments but in many situations arguments lack this coercive force especially where the arguments have varied confidence and there are different preferences for the acceptability of arguments by each audience.

4.6.1 Ontology mapping and Argumentation Framework

There exist various matching algorithms for establishing mappings between two ontologies yielding different set of ontology mappings. Sometimes it is required to reach an agreement between audiences who have different preferences for matching algorithms. Argumentation framework provides a systematic way to reach an agreement.

Various approaches use Argumentation Framework for reaching at an agreement about ontology mapping (Laera et al., 2006), (Trojahn et al., 2008), (Isaac et al., 2008). Arguments are generated or reused for each ontology correspondence and argumentation framework supports reasoning on identifying acceptable arguments. Acceptable arguments are then used for identifying agreed ontology correspondences. Argumentation frameworks vary on the notion of acceptability, such as some use *confidence values* of correspondences (Trojahn et al., 2008) while some *use support (vote based)* in favor to that correspondence (Isaac et al., 2008).

Concepts related to value based argumentation framework that are used in the context of ontology mappings are defined below.

Definition 4-7 (*Argument in ontology mapping*): (Laera et al., 2006). An argument $a \in AF$ is a tuple $a = \langle G, m, \sigma \rangle$, such that m is a correspondence $\langle e;\ e', r, n \rangle$; ($e, e'$ are artifacts of two ontologies, r is a relation between these artifacts and n is confidence measure of a mapping relation between artifacts of two ontologies). G is the ground justifying the prima facie that the correspondence does or does not hold, for instance, lexical similarity or structural similarity of involved artifacts. σ is one of $\{+, -\}$, depending on whether the argument is that m does or does not hold.

For instance, suppose that a correspondence $m = (bank, bank, \equiv, 1)$, where ground G for this correspondence is lexical similarity then σ is positive. So, argument $a = \langle lexicalSimilarity, m, + \rangle$ while when ground is semantic (where *bank* in one ontology represents river bank and in other ontology *bank* represents commercial bank) then σ is negative since these two artifacts are different semantically.

$$a = \langle semanticSimilarity, m, - \rangle$$

Arguments interact based on the notion of attack relation.

Definition 4-8 (*Attack*): An argument $a \in A$ attacks another argument $b \in A$, if a and b are arguments of same mappings but with different signs as $a = \langle G_1, m, + \rangle$ and $b = \langle G_2, m, - \rangle$. Attack is represented by $\ltimes$ and $a \ltimes b$ also $b \ltimes a$, b is the counter-argument of a, and a is the counter-argument of b.

In Dung Argumentation Framework, all arguments have equal value, and an attack always succeeds. Amgoud et al. (Amgoud & Cayrol, 1998) has introduced the notion of preference between arguments, where an argument can defend itself against weaker arguments. Bench-Capon propose preferences for giving importance to one audience over another (Bench-Capon, 2003). Laera and colleagues use value based argumentation framework in ontology mapping and it is defined as

Definition 4-9 (Value based Argumentation Framework for ontology mapping): (Laera et al., 2006) A value-based argumentation framework (VAF) for audience is a 5-tuple: VAF $= \langle \mathrm{AR}, \mathrm{A}, \mathrm{V}, \eta, \mathrm{P} \rangle$, where AR and A are the standard notions of Argumentation Framework, V is a non-empty set of values that represents types of arguments (e.g; lexical, structural, extensional or semantical), $\eta: \mathrm{AR} \rightarrow \mathrm{V}$ is a function which maps elements of AR to elements of V and P is the set of possible audiences (e.g., audience that prefer structural over semantic, another audience prefer lexical over semantic).

a) For arguments x, y in AR, x is a *successful* attack on y (or x defeats y) with respect to the audience R if $(x, y) \in A$ and it is not the case that $\eta(y) \succ_R \eta(x)$.
b) An argument x is *acceptable* to the subset S with respect to an audience R if: for every $y \in AR$ that successfully attacks x with respect to R, there is some $z \in S$ that successfully attacks y with respect to R.
c) A subset S of AR is *conflict-free* with respect to the audience R if: for each $(x, y) \in S \times S$, either $(x, y) \notin A$ or $\eta(y) \succ_R \eta(x)$.
d) A subset S of AR is *admissible* with respect to the audience R if: S is conflict free with respect to R and every $x \in S$ is acceptable to S with respect to R.
e) A subset S is a *preferred extension* for the audience R if it is a maximal admissible set with respect to R.
f) A subset S is a *stable extension* for the audience R if S is admissible with respect to R and for all $y \notin S$ there is some $x \in S$ which successfully attacks y with respect to R.
g) $x \in AR$ is *objectively acceptable* (agreed) if x is acceptable by every audience and $x \in AR$ is *subjectively acceptable* if x is acceptable by at least one audience.

There are other kinds of argumentation framework such as Value based (Isaac et al., 2008) and strength based (Trojahn et al., 2008).

Ontology mappings are often available without their justifications (grounds of mappings), so justifications need to be created in order to apply Argumentation framework in ontology mapping, Arguments are generated for providing justifications of mappings such as neighbors of mapped artifacts involve in mappings are mapped, properties of mapped concepts, mapped instances, and terminological similarities. If there are justifications that satisfy mappings, then + is assigned to σ, otherwise − is assigned to σ. η assigns values to arguments these values are semantic (M), external structure (ES), internal structure (IS), terminology (T), and extensional (E) similarities. An audience is considered as agents or people who have certain preferences for the choice of mapping, such as one audience prefer

Terminological arguments over External structures $T \succcurlyeq ES$ and External structures over terminological $ES \succcurlyeq T$.

Preferred extensions for each audience is computed and then objectively accepted correspondences are identified. The mapping which is not in the list of 'objectively accepted correspondences' will be removed.

Arguments can be classified into terminological (T), extensional (ES), and intentional (IS) audience based on when arguments are generated based on terminological (lexical such as label), extension (external structure such as super and sub artifacts), intentional (internal structure such as properties and domain/range) similarities.

Example 4-35: Suppose that a mapping $m_1 = \langle O_i{:}A, O_j{:}\alpha, \equiv \rangle$. Arguments for this mapping is whether the sub/super artifacts of $O_i{:}A$ and $O_j{:}\alpha$ are also mapped, whether $O_i{:}A$ and $O_j{:}\alpha$ are lexically similar, whether roles involving these artifacts are mapped? In the following some arguments for m_1 is listed and $+/-$ shows whether argument for this mapping holds or not. $a_1, a_2, a_3, and\ a_4$ are arguments.

$$a_1{:}\langle\langle superconcept(O_i{:}A) \equiv\ superconcept\big(O_j{:}\alpha\big)\rangle, m_1, +\rangle \in ES$$

$$a_2{:}\langle\langle subconcept(O_i{:}A) \equiv\ subconcept(O_j{:}\alpha)\rangle, m_1 - \rangle\ \in ES$$

$$a_3{:}\langle\langle label(O_i{:}A) \equiv label\big(O_j{:}\alpha\big)\rangle, m_1, +\rangle \in T$$

$$a_4{:}\langle\langle prperty(O_i{:}A) \equiv prperty\big(O_j{:}\alpha\big)\rangle, m_1, -\rangle \in IS$$

a_1 attacks on a_2, a_4 and a_2 attacks on a_1, a_3, and a_3 attacks a_2, a_4 and a_4 attacks a_3, a_1.

It is depicted in Figure 4-8.

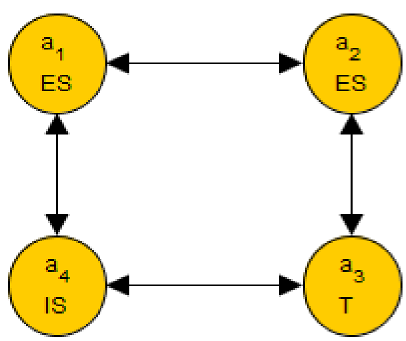

Figure 4-8. Pictorial representation of Argument Framework

Example 4-36: Suppose that mappings m_1, m_2 and m_3, arguments scheme, audience and attacks are as in the Table 4-4.

Table 4-4. Argument scheme, attack and audience of mappings,

Mapping	Argument	σ	Audience	Attack
m_1	a_1	-	ES	a_2
m_1	a_2	+	ES	a_1, a_3
m_1	a_3	-	T	a_2
m_2	a_4	+	T	a_5
m_2	a_5	-	ES	a_4, a_6
m_2	a_6	+	ES	a_6
m_3	a_7	-	IS	a_6, a_8
m_3	a_8	+	T	a_7
m_3	a_9	+	ES	a_7

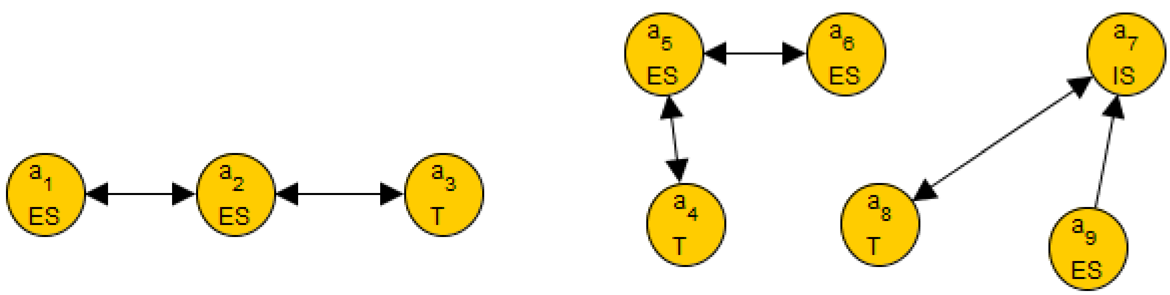

Figure 4-9. Value-based Argument Framework

Value-based argumentation framework is shown in Figure 4-9.

Suppose that there are two audiences $\mathcal{R}_1$and $\mathcal{R}_2$ where $\mathcal{R}_1$ prefers terminological arguments over extensional arguments $T \succ_{\mathcal{R}_1} ES$ and $\mathcal{R}_2$ prefers extensional arguments over terminological arguments $ES \succ_{\mathcal{R}_2} T$.

In value-based argumentation framework presented in Figure 4-9, preferred extension for audience $\mathcal{R}_1$ is $\{a_1, a_3, a_4, a_6, a_8, a_9\}$. While Preferred extensions for audience $\mathcal{R}_2$ are $\{a_1, a_3, a_5, a_8, a_9\}$, $\{a_1, a_3, a_4, a_6, a_8, a_9\}$, , $\{a_2, a_5, a_8, a_9\}$, $\{a_2, a_4, a_5, a_6, a_8, a_9\}$.

m_1 will be rejected since argument a_2 supporting $m_1$does not appear in any preferred extension of both audiences. m_3 is acceptable to both audiences since all the arguments supporting m_3 (a_8 and a_9) are present in preferred extension of both audiences. supporting m_3 m_2 is also acceptable since all the arguments supporting m_2 (a_4 and a_6) are present in $\mathcal{R}_1$ and some of the preferred extensions of $\mathcal{R}_2$. m_3 is agreed correspondences (objectively acceptable), m_2 is agreeable correspondence (subjectively acceptable) and m_1 is rejected correspondence.

Although, one can obtain acceptable correspondences from argumentation framework but still there is no guarantee that the acceptable correspondences are correct. Therefore, it requires some kind of mechanism to handle correctness of acceptable correspondences. In Section 4.6.2, we present state of the art about handling inconsistency in argumentation framework.

4.6.2 Handling Inconsistencies in Argumentation Framework:

There are two ways of resolving inconsistency in argumentation framework; (i) deal inconsistency with in argumentation framework (ii) deal individually logical and argumentation part of the problem. Dealing inconsistency within argumentation framework seems better choice, in this setting correspondences are arguments and two contradicting correspondences are attack for each other. However, when size of contradicting correspondences is large then encoding them argumentation framework would be exponential. That is why consistency is associated to *preferred extensions* and it is defined as

Definition 4-10 (Consistency of mapping in Argumentation Framework): (Trojahn dos Santos & Euzenat, 2010) A preferred extension S is said to be consistent iff its associated alignment $A(S)$ is consistent.

Trojahn dos Santos and colleagues consider two solutions for dealing inconsistency in argumentation framework. Firstly, consider only those preferred extensions that are consistent but such a set may be empty. Secondly, consider *maximal preferred consistent sub-extensions*. They opt for second choice.

Definition 4-11 (Maximal preferred consistent sub-extensions): (Trojahn dos Santos & Euzenat, 2010) A consistent extension S is a maximal preferred consistent sub-extension *iff* there exists a preferred extension S' such that $\mathrm{S} \subseteq \mathrm{S}'$ and $\forall \mathrm{S}''$; $\mathrm{S} \subset \mathrm{S}'' \subseteq \mathrm{S}'$, S'' is not consistent.

Their approach resolves inconsistency in ontology alignment through the result of argumentation process. They do not extended *attack* relation in argumentation framework and resolve inconsistencies after obtaining preferred extension through argumentation approach.

4.7 Synthesis

Argumentation Framework provides a way to reach agreement and it is employed in various fields including ontology mapping. The underlying idea of argumentation framework is that participants exchange arguments and counter arguments until some consensus is reached. Definition of acceptability of the argument can be defined in various ways depending on the context and requirements as there exist various kinds of argumentation framework like for instance, strength-based argumentation framework and voting based argumentation framework.

In the context of ontology mapping, weak point of argumentation framework is lack of arguments about ontology correspondences. In automated matching system, mapping is often generated without providing the arguments (justifications) about the correspondences of the mapping. It is very complex task to generate arguments for the ontology mapping under consideration, since generating arguments is like trying to create mapping once again as focus is on getting justifications of these mappings. When ontology mapping is large in size then generated arguments are numerous and are very difficult to handle.

However, still arguments can be generated easily on collecting the preferences of the audiences using some kind of voting mechanism.

Some potential contradiction such as logical incoherence, non-conservative mapping, which are mostly obtained by using debugging approaches, can also work as arguments in argumentation framework. The attack relation can also be defined on the basis of conflicting correspondences (causing ontology mapping defects).

Dealing with ontology mapping, Argumentation framework provides consensus mapping that is acceptable to all participating audiences. This agreed mapping meets audience acceptability criteria. However, it is not necessary that participating audiences consider consistency in their acceptability criteria and it can be seen in Example 4-36. This means that consensus proposal (arguments) can be wrong (in logical context can be logical inconsistent). However, there exist some ways to check logical consistency. For instance, Logical consistency can be checked after maximum preferred extension is known.

Chapter 5.

Compatible and Incompatible Ontology Mappings

The concept of compatibility is required for checking whether or not two or more things used in combination are not conflicting with each other. If two (or more than two) things are compatible to each other then they can be used together as they are not in conflict with each other, whereas incompatible things are in conflict with each other so either they should not be used together or they can be used by appropriately handling the conflict situation. In this thesis, our focus is on two or more ontology mappings of two ontologies and we are interested in knowing whether two (or more than two) ontology mappings are compatible or incompatible. If it is known that two (or more than two) ontology mappings are compatible, then one can take advantage of these ontology mappings and can use them in a single application. If it is known that two ontology mappings are incompatible, then either they should not be used together or they should be used in such a way that they do not contradict each other either by isolating them, or by removing or correcting incompatible correspondences of ontology mappings.

As we have described in Chapter 2, Chapter 3, and Chapter 4, there exist several formalisms and patterns that can be used for debugging absolute (such as incoherence and inconsistency) and relative (user-defined) defects. However, we have found that a defect in one formalism is not necessarily be considered as a defect in another formalism (as presented in Table 4-3), while patterns for debugging defects are applicable to specific formalisms only. Logical formalisms are widely used in the context of ontology and ontology mapping but there exist various kinds of logics and changing one logical formalism with another logical formalism can either generate defects which were not earlier present or there may be no defect. Algebraic formalisms provide independence of specific logical formalism but existing approaches such as Category theory and Institution theory help in providing independence from the use of (particular) logic but user have to revert back to logical approaches for checking absolute and relative defects (Section 3.1.2.1). Therefore, there is a need of an approach that provide independence of any logical formalism and check both absolute and relative defects. This prompts us to propose our approach.

Compatibility is a generic term in the context of ontology mapping, so we have to define it precisely. In this chapter, our aim is to give formal definition of compatible and incompatible ontology mappings. In particular, we give answer to questions like what are the conditions that should be presented for calling ontology mappings as compatible ontology mappings? Which framework is required to check compatibility of ontology mappings? What type of defects (contradiction) is detected by this definition? Our focus is to provide a unified approach that can handle mappings and ontologies formalized either in logical or algebraic and provide a way to identify both logical and categorical defects (contradictions) for classifying ontology mappings into compatible and incompatible mappings.

State of the art solutions (Meilicke, 2011) and (Qi et al., 2009) are mostly focused on detecting logical inconsistency and incoherence, even algebraic approaches (Bench-Capon & Malcolm, 1999) are used to detect logical inconsistencies. There exist few work in the literature for detecting relative defects, especially the violation of principle of conservativity (Solimando et al., 2014), however, this work is limited to checking deductive difference of source ontologies and combination of source ontologies with ontology mapping for a specific case of checking subsumptions $A \sqsubseteq B$ where A and B are atomic artifacts of the ontology. Algebraic approaches may solve the problem of heterogeneity of formalism used for source ontologies and ontology mapping, but still logical approaches are required for checking (logical) inconsistency. We have not found any work in which a single approach can handle both heterogeneity of formalism of ontologies & ontology mappings and can check absolute and relative defects. Our focus of this work is on presenting a unified approach for knowing the compatibility and incompatibility of ontology mappings based on presence or absence of absolute and relative defects.

The rest of this chapter is organized as follows: Section 5.1 describes a preliminary introduction about order relation; Section 5.2 describes lattices that can be built on ontology and accounts limitations of these lattices in checking the existence of Galois connection; Section 5.3 relates ontology mappings with mappings between lattices of ontologies; Section 5.4 presents a formal definition of compatibility and presents our approach for checking compatibility and incompatibility of ontology mappings; Section 5.5 illustrates a rough sketch how our notions of compatible and incompatible ontology mappings detect absolute and/or relative defects caused by ontology mappings; and finally, Section 5.6 describes conclusions drawn in this chapter.

5.1 Basic notions and properties related to order relation

Definitions of basic notations related to order relation are taken from (Ganter et al., 1997) and (Davey & Priestley, 2002).

Definition 5-1 (Binary relation): A binary relation R between two arbitrary sets A and B is defined on the Cartesian product $A \times B$ and it consists of paris (a, b) with $a \in A$ and $b \in B$. Binary relation is written usually as $\langle A\ R\ B \rangle$. When $(a, b) \in R$ it is usually written as aRb.

Definition 5-2 (Order relation): In mathematics, ordering is a binary relation on a set of objects. A binary relation R on a set A is called an order relation if it satisfies the following conditions for all elements $a, b, c \in A$

aRa (reflexivity)

aRb and $a \neq b \Rightarrow \neg bRa$ (antisymmetry)

aRb and $bRc \Rightarrow aRc$ (transitivity)

When the relation R is $\leq$ or $<$ over a set A which is reflexive, antisymmetric, and trasitive then it is called as *Partial order.* A set with partial order is called as *partially ordered set* or *poset*. When poset has a relation $\leq$ then such poset is named as *non-strict poset*. While when poset has a relation $<$ defined over a set then such poset is named as *strict poset*.

A power set, which is the set of subsets of a given set, ordered by inclusion is an example of poset. A power set of a set $\{a, b, c\}$ is shown as poset in Figure 5-1.

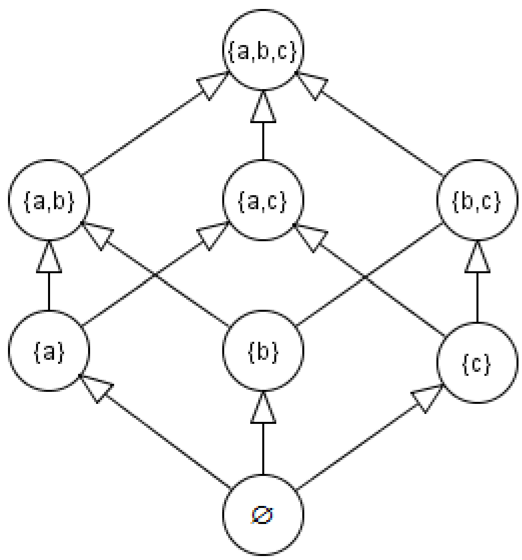

Figure 5-1. A poset of all subsets of 3 elements ordered by inclusion

A *Preorder* is a binary relation that is reflexive and transitive. Any collection of sets is preordered by their comparative sizes, $\{2,4\} \leq \{Monday, Tuesday, Friday\}$ is an example of preordered set.

The property $a \leq b$ or $b \leq a$ is called *totality* or a and b are *comparable*. A *total order* is a binary relation that is anti-symmetric, transitive, and total. A total order is also called as Linear order. The letters of the alphabet ordered by the standard dictionary order, e.g., $C < B < A$ etc. is an example of Ordered set.

Definition 5-3 (Converse of poset): If $\wp = \langle P, \leq \rangle$ is a poset then so is $\wp^{op} = \langle P, \leq^{op} \rangle$, where $\leq^{op}$ is the opposite or *converse* of $\leq$, i.e., for all $x, y \in P, x \leq^{op} y$ iff $y \leq x$.

Definition 5-4 (Antichain and chain): A subset of poset in which no two distinct elements are comparable is called as *antichain*. For example, the singletons $\{\{a\}, \{b\}, \{c\}\}$ in Figure 5-1.

While a subset of poset in which each pair of elements are comparable is called as chain. A chain is a totally ordered. For example the $\{\{a\},\{a,b\},\{a,b,c\}\}$ in Figure 5-1.

Definition 5-5 (Infimum and Supramum): Let $(M, \leq)$ be an ordered set and A a subset of M. A *lower bound* of A is an element s of M with $s \leq a, \forall a \in A$. If there exists a largest element in the set of all lower bounds of A, it is called the *infimum* of A and is denoted by "inf A" or $\bigwedge A$ or $\sqcap A$. An *upper bound* is defined as an alement s of M with $a \leq s, \forall a \in A$. If it exists a smallest in the set of all upper bounds of A, it is called the *supramum* of A and is denoted by "sup A" or $\bigvee A$.or $\sqcup A$.

Definition 5-6 (Join and meet): A function that returns *supramum* is called *join* and a function that returns *infimum* is called *meet*.

Definition 5-7 (Lattice): Let P be a non-empty poset. If join $x \vee y$ and meet $x \wedge y$ exist for all $x, y \in P$, then P is called a Lattice. While if $\bigvee S$ and $\bigwedge S$ exist for all $S \subseteq P$, then P is called a *Complete lattice*. Every complete lattice P has a largest element $\bigvee P$ called the *top element* and represented as $\top$, and has a smallest element $\bigwedge P$ called the *bottom element* and represented as $\perp$.

Definition 5-8 (Information ordering): The order relation can captures the notion of "y is more informative than x", "y is more defined than x", or "y is a better approximation than x" and can be represented as $x \leq y$.

Definition 5-9 (Upper set and Lower set): An *upper set* of a poset $(P, \leq)$ is a subset U with the property that, if x is in U and $x \leq y$, then y is in U. A *lower set* is a subset L with the property that, if x is in L and $y \leq x$, then y is in L.

Definition 5-10 (Maps between Ordered sets): Let P and Q be ordered sets. A map $\varphi: P \to Q$ is said to be

- *Order-preserving*: if $x \leq y$ in P implies $\varphi(x) \leq \varphi(y)$ in Q. An order preserving mapping is also called *montone* or *isotone* mapping.
- *Order-reversing*: if $x \leq y$ in P implies $\varphi(x) \geq \varphi(y)$ in Q. An order reversing mapping is also called *antitone* mapping.

- *Order-embedding*: if $x \leq y$ in P if and only if $\varphi(x) \leq \varphi(y)$ in Q
- *Order-isomorphism*: if it is an order-embedding which maps P onto Q.

Relating Poset with Category Theory: Posets can be treated as categories since they respect composition and associative property, and identity.

- Composition: If $f: P \to Q$ is a monotone map from $\mathcal{P} = \langle P, \leq \rangle$ to $\mathcal{Q} = \langle Q, \leq \rangle$ and $g: Q \to R$ is a monotone map from $\mathcal{Q} = \langle Q, \leq \rangle$ to $\mathcal{R} = \langle R, \leq \rangle$, then the composite map $g \circ f: P \to R$, is a monotone map from P to R.
- Composition of map is associative: If $f: P \to Q, g: Q \to R, h: R \to S$ are monotone, then $(f \circ g) \circ h$ and $f \circ (g \circ h)$ are monotone and equal.
- Identity: The identity map $Id_P: P \to P$, which maps every $p \in P$ to itself, is a monotone map from $\mathcal{P} = \langle P, \leq \rangle$ to itself.

Pos represents the Category of posets.

(Relating Lattice with Logic): The symbols used for infimum and supramum of Lattice are $\wedge$ and $\vee$ respectively. These symbols are also used for conjunction and disjunction in logic. Also, top element and bottom element are represented by $\top$ and $\perp$ resepectively. $\top$ refers to 'Truth' and $\perp$ refers to 'Falsity' in logic. Implication in logic $\alpha \to \beta$ refers to the relation $\leq$ in lattice.

5.1.1 Galois Connection

In Definition 5-10, we have described two types of mapping between ordered sets which are embedding, and order isomorphism. Another very special kind of mapping between ordered sets is Galois connection (Ore, 1944) due to rich in implications and properties. In this section, we describe the basic definitions and kinds of Galois connection and their important characteristics.

In the literature two types of Galois connections are reported: Order preserving and Order reversing.

Definition 5-11 (Order preserving Galois connection): Given ordered structures A, B with partial order relationship $\preccurlyeq$ and isotone mappings $\alpha : A \to B$ and $\gamma : B \to A$, the pair (α, γ) establishes an *order preserving Galois connection* between A and B if $\forall a \in \alpha, and\ \forall b \in \gamma$ the following condition satisfies for

$$\boldsymbol{\alpha(a) \preccurlyeq b \text{ iff } a \preccurlyeq \gamma(b)} \qquad \text{(A)}$$

α is said to be lower adjoint of the corresponding γ, and γ is the upper adjoint of α. Galois connection is also represented as in Figure 5-2.

$$\begin{tikzcd} \langle A, \subseteq \rangle \arrow[r, shift right, "\alpha"'] & \langle B, \subseteq \rangle \arrow[l, shift right, "\gamma"'] \end{tikzcd}$$

Figure 5-2. Galois Connection

Definition 5-12 (Order reversing Galois connection): Given ordered structures *A, B* with partial order relationship $\preccurlyeq$and antitone mappings $\alpha: A \rightarrow B$ and $\gamma: B \rightarrow A$, the pair (α, γ) establishes an *order reversing Galois connection* between A and B if $b \preccurlyeq \alpha(a)$ and $a \preccurlyeq \gamma(b)$, $\forall a \in \alpha, and\ b \in \gamma$.

The condition (A) consists of four components and they are two ordered sets and two mappings used for establishing Galois connection. Galois connection are not hard to find but sometimes these components and Galois condition are hidden and rarely recognized.

Galois connection is ubiquitous in mathematics and logics.

1. Let f is an order-isomporphim between $\langle A, \preccurlyeq \rangle$ and $\langle B, \preccurlyeq \rangle$, so the inverse function f^{-1} is also an order-isomorphism. Then $\langle f, f^{-1} \rangle$ is a Galois connection.

$$f(a) \preccurlyeq b \text{ iff } f^{-1}\big(f(a)\big) \preccurlyeq f^{-1}(b) \text{ iff } a \preccurlyeq f^{-1}(b)$$

2. Suppose $A = \langle \mathbb{N}, \leq \rangle$ and $B = \langle \mathbb{R}^+, \leq \rangle$, where $\mathbb{N}$ are set of Natural numbers and $\mathbb{R}$ is set of Real numbers and $\leq$ are the standard order relation. $\alpha: \mathbb{R}^+ \rightarrow N$ maps positive real number to its integral part (it can be interpreted either as floor or ceiling), and $\gamma: \mathbb{N} \rightarrow \mathbb{R}^+$ be the standard embedding of natural number to real numbers. In this situation, condition (A) trivially holds, $\langle \alpha, \gamma \rangle$ is a Galois connection between natural number and real number.

3. Let ϵ is an equivalent class of all well-formed formulas logically equivalent to ϕ. E be the set of all equivalent classes and $\rightarrow$ is a relation between equivalent classes $|\phi|$ and $|\psi|$ in E if $\phi \vDash \psi$. This shows that $\langle E, \rightarrow \rangle$ is a poset. Then mappings $\alpha: |\phi| \mapsto |\psi \wedge \phi|$ and $\gamma: |\phi| \mapsto (|\psi \rightarrow \phi|)$. Then these mappings form a Galois connection between E and itself since $|\psi| \wedge |\phi| \vdash |\chi|\ iff |\phi| \vdash (|\psi| \rightarrow |\chi|)$.

5.1.2 Relating order preserving and order reversing Galois connection

The order reversing Galois connection is symmetric when the two mappings γ and α cannot be differentiated (i.e., one cannot identify between upper and lower adjoint). If order relationships are information orders then if $a \preccurlyeq b$ means that a is less informed than b (the same as for instance, in subsumption); γ and α can be interpreted as ***abstraction mappings***, because applying one mapping result in some information loss. The order preserving Galois connection is not symmetric because γ and α can be differentiated (i.e., one can identify between upper and lower adjoint); under the same interpretation of order relationships, α is an ***abstraction mapping*** because resulting in information loss while γ is a ***concretization mapping*** because resulting in information enrichment. Abstraction α and concretization γ is shown in Figure 5-3.

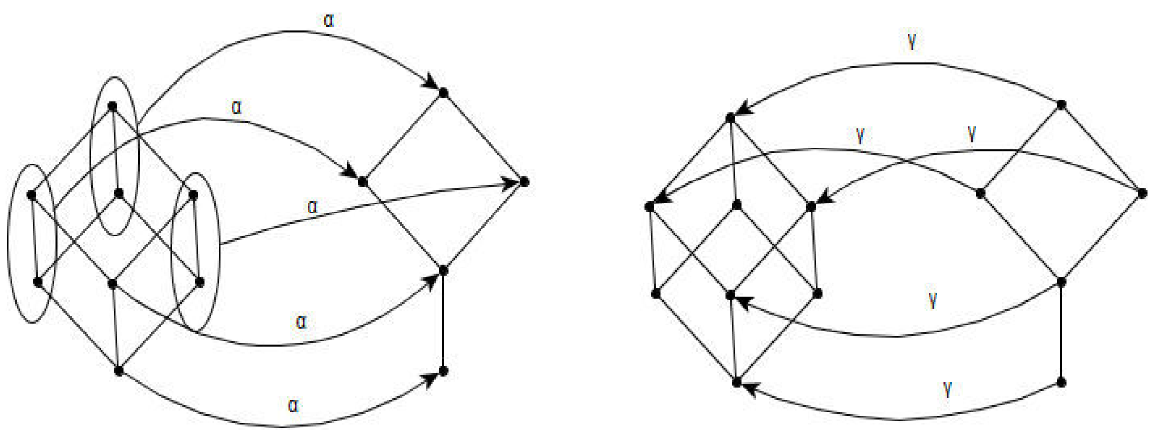

Figure 5-3. Abstraction and Concretization of α and γ mappings

One kind of Galois connection of can be translated to another kind with some modification. Suppose that there are two posets. $\mathcal{A} = \langle A, \preccurlyeq \rangle$ and $\mathcal{B} = \langle B, \preccurlyeq \rangle$ and there are order preserving Galois connection $\langle \alpha, \gamma \rangle$ between $\mathcal{A}$ and $\mathcal{B}$ such that $\alpha(a) \preccurlyeq_B b$ iff $a \preccurlyeq_A \gamma(b)$. For poset $\mathcal{B}^{op} = \langle B, \preccurlyeq^{op} \rangle$, there exists antitone Galois connection $\langle \alpha, \gamma \rangle$ between $\mathcal{A}$ and $\mathcal{B}^{op}$ such that $b \preccurlyeq_B^{op} \alpha(a)$ iff $a \preccurlyeq_A \gamma(b)$ where $\preccurlyeq^{op}$ represents the converse of $\preccurlyeq$ which can also be represented as $\succcurlyeq$. Since an isotone connection between $\mathcal{A}$ and $\mathcal{B}$ is an antitone connection between $\mathcal{A}$ and dual of $\mathcal{B}$ ($\mathcal{B}^{op}$). By reversing the order relation between posets involved in Galois connection, one can obtain the Galois connection of other kind. By converting order preserving Galois connection into order reversing Galois connection and vice versa, one can take advantage of properties of each kind of Galois connection.

5.1.3 Properties of Galois Connection

In the following, we describe some important properties of Galois connection. We consider that there exists a Galois connection $\langle \alpha, \gamma \rangle$ between $A, \preccurlyeq$ and $B, \preccurlyeq$ where $\alpha: A \rightarrow B$ and $\gamma: B \rightarrow A$

Idempotent:

Order preserving Galois connection preserves *Idempotence* property.

$$a \preccurlyeq \gamma(\alpha(a))$$

Duality:

The *dual* of a Galois connection is also a Galois connection in a reverse order. If $\langle A, \preccurlyeq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathcal{B}, \preccurlyeq \rangle$ is a Galois connection then it's *dual* is also a Galois connection $\langle A, \succcurlyeq \rangle \underset{\gamma}{\overset{\alpha}{\leftrightarrows}} \langle B, \succcurlyeq \rangle$.

Symmetry:

Galois connection is not necessarily symmetric.

Theorem 5-13

If $\langle P, \leq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle Q, \sqsubseteq \rangle$ and $\langle \alpha, \gamma \rangle$ is a Galois connection then it is not necessary that $\langle \gamma, \alpha \rangle$ is a Galois connection $\langle Q, \sqsubseteq \rangle \underset{\gamma}{\overset{\alpha}{\leftrightarrows}} \langle P, \leq \rangle$.

Proof:

By counter example, Suppose there is a Galois connection between $\langle \mathbb{N}, \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathbb{Q}^+, \leq \rangle$, where α mapping is a standard embedding of Natural number into the positive rationals, while γ map positive rational number to the natural corresponding to its integral part. Therefore, we have Galois connection between natural numbers and positive rational numbers in their natural ordering.

However, there is no Galois connection as $\langle \mathbb{N}, \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathbb{Q}^+, \leq \rangle$, since α can't appear as lower adjoint in Galois connection between $\mathbb{N}$ and $\mathbb{Q}^+$.

For instance,

$\alpha(1) \leq 1 \Leftrightarrow 1 \sqsubseteq \gamma(1)$, where $\gamma(1) = 1$ is true

But for

$\alpha(1.1) \leq 1 \Leftrightarrow 1.1 \not\sqsubseteq \gamma(1)$, where $\gamma(1) = 1$.

So α and γ in this setting do not form Galois connection.

Q.E.D.

Composition:

The composition of monotone Galois connection is a Galois connection.

Theorem 5-14

If $\langle P, \leq \rangle \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \langle Q, \sqsubseteq \rangle$ and $\langle Q, \sqsubseteq \rangle \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \langle R, \preccurlyeq \rangle$ then $\langle P, \leq \rangle \underset{\alpha_2 \circ \alpha_1}{\overset{\gamma_1 \circ \gamma_2}{\leftrightarrows}} \langle R, \preccurlyeq \rangle$.

Proof:

If $\langle P, \leq\rangle \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \langle Q, \sqsubseteq\rangle$ and $\langle Q, \sqsubseteq\rangle \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \langle R, \preccurlyeq\rangle$

then $\forall x \in P : \forall z \in R$:

$$\alpha_2 \circ \alpha_1(x) \preccurlyeq z$$

$\Leftrightarrow \alpha_1(x) \sqsubseteq \gamma_2(z)$ (by def. of Galois connection)

$\Leftrightarrow x \leq \gamma_1 \circ \gamma_2(z)$ (by def. of Galois connection)

Q.E.D.

However, antitone Galois connection does not compose.

Extensive and Reductive:

$\gamma \circ \alpha$ is *extensive*, i.e., $a \preccurlyeq \gamma \circ \alpha(a)$

$\alpha \circ \gamma$ is *reductive*, i.e., $\alpha \circ \gamma(b) \preccurlyeq b$

Theorem 5-15: *Let* $\langle P, \leq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle Q, \sqsubseteq\rangle$ *be a monotone Galois connection then* $\alpha \circ \gamma$ *is reductive and* $\gamma \circ \alpha$ *is extensive.*

Proof:

For all $x \in P$ and $y \in Q$

$\alpha(x) \sqsubseteq \alpha(y) \Rightarrow x \leq \gamma \circ \alpha(x)$ — $\gamma \circ \alpha$ is extensive

$\gamma(x) \leq \gamma(y) \Rightarrow \alpha \circ \gamma(y) \sqsubseteq y$ — $\alpha \circ \gamma$ is reductive

$x \leq y \Rightarrow x \leq \gamma \circ \alpha(x) \Rightarrow \alpha(x) \sqsubseteq \alpha(y)$ — α is isontone

$x \sqsubseteq y \Rightarrow \alpha \circ \gamma(y) \sqsubseteq y \Rightarrow \gamma(x) \leq \gamma(y)$ — γ is isotone

$\Longleftarrow$

For all $x \in P$ and $y \in Q$

$$\alpha(x) \sqsubseteq y$$

$\Longrightarrow \gamma \circ \alpha(x) \leq \gamma(y)$ — γ is isontone

$\Longrightarrow x \leq \gamma(y)$ — $\gamma \circ \alpha$ is extensive and transitivity

$\Longrightarrow \alpha(x) \sqsubseteq \alpha \circ \gamma(y)$ α is isontone

$\Longrightarrow \alpha(x) \sqsubseteq y$ $\alpha \circ \gamma$ is reductive and transitivity

Q.E.D.

This property tells that Galois connection may lose some information but it does not lose order relation.

Adjoint uniquely determines each other:

If $\langle \alpha, \gamma \rangle$ is a Galois connection then γ fixes what α has to be, and conversely α fixes what γ has to be (Smith, 2010).

Preservation of Infima and Suprema

Theorem 5-16: *Let* $\langle P, \leq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle Q, \sqsubseteq \rangle$ *be a Galois connection and* $X \subseteq P$ *such that its lub* $\bigvee X$ *does exist in P. Then* $\alpha(\bigvee X)$ *is the lub of* $\{\alpha(x) | x \in X\}$ *in Q, that is* $\alpha(\bigvee X) = \bigsqcup \alpha(X)$.

Proof: $\forall x \in X: x \leq \bigvee X$ by existence of the lub $\bigvee X$ so x$\forall x \in X: \alpha(x) \sqsubseteq \alpha(\bigvee X)$ by monotnoly of α proving that $\alpha(\bigvee X)$ is an upper bound of the set $\{\alpha(x) | x \in X\}$ in Q.

Let y be another upper bound of $\{\alpha(x) | x \in X\}$ in Q

$\forall x \in X: \alpha(x) \sqsubseteq y$ def. upper bound

$\Rightarrow \forall x \in X: x \sqsubseteq \gamma(y)$ def. Galois connection

$\Rightarrow \bigvee X \leq \gamma(y)$ def. lub

$\Rightarrow \alpha(\bigvee X) \sqsubseteq y$ def. Galois connection

Proving that $\alpha(\bigvee X)$ is the least of the upper bounds of $\{\alpha\ (x) | x \in X\}$.

If $\bigsqcup Y$ is the lub of $Y \subseteq Q$ in $\langle Q, \sqsubseteq \rangle$, whenever it exists, and we have proved that α preserves existing lubs.

If $\bigvee X$ exists in $\langle P, \leq \rangle$ then $\bigsqcup \alpha(X)$ does exists in $\langle Q, \sqsubseteq \rangle$ and $(\bigvee X) = \bigsqcup \alpha(X)$.

Q.E.D.

Dually, we can prove that gamma mapping of Galois connection preserves greatest lower bounds.

Preservation of all infimas and supremas guarantees that Galois connection best possible abstraction in one lattice and best possible concretization in other lattice.

Isotone Galois connection are mostly used in the computer science, where relative information preservation is important. Additionally operations like composition and others can be used with isotone Galois connections. Indeed, one can get isotone Galois connection from antitone Galois connection by reversing the order of one of the ontology. Therefore, We use only isotone Galois connection for dealing with ontology mapping in the later part of this thesis.

5.2 Lattices and Ontologies

There are several ways of building lattice (partially ordered set can be made lattices by adding $\perp$ and $\top$) of ontologies. In this section, we discuss these lattices and their potential use in verifying the compatibility of ontology mappings.

In this work, we restrict ourselves to classical logic $\mathfrak{L}$ having expressivity up to first-order logic. *Logic* $\mathfrak{L}$ consists of $\langle \mathcal{L}, \vDash \rangle$; $\mathcal{L}$ is *language* and $\vDash$ is *satisfiability*. *Signature*s of a language $\mathcal{L}$ is a set of symbols of the language $\mathcal{L}$. Signatures of language can be classified into non-logical, logical and variable signatures. *Non-Logical Signatures* are those symbols that do not have predefined meaning in the logic and here represented by Σ. *Logical Signatures* are sentential connectives having predefined meanings in the logic and here represented by Λ. *Variables* are placeholder that represent unspecified symbols, here, represented by Var. There are two kinds of variables: free and bounded variable. *Free variable* is a variable that is not bound to specific value, while *bound variable* is a variable which was previously free but later assigned a specific value or set of values. *Language* $\mathcal{L}$ can be defined as a set of symbols (signature) $\{\Sigma \cup \Lambda \cup Var\}$ and rules for forming the grammatically correct language. An *expression* is a finite sequence of symbol. Expressions are meaningful (terms or well-formed formulas) or nonsensical. A *Well-formed Formula* is either an atomic formula or a formula formed by combination atomic formula with connective symbols and quantifier symbols. *Terms* are expressions formed by constants c or functions $f(v_1, \dots, v_n)$ where v_i is variable. *Atomic formula* is a Formula that does not involve connectives and quantifiers. Predicates are atomic formula written as $P(t_1, \dots, t_n)$ where t_i is term. *Sentence* is a formula that does not have free variable.

In this section, we present various lattices built on (subset of) signatures of language $\mathcal{L}$ of an ontology. These lattices will be later used in defining the notions of compatible and incompatible ontology mappings. Lattice built on (subset of) signatures of language $\mathcal{L}$ can be infinite lattice, for instance, when variables and number of individuals involved in the ontology/theory are infinite, so we have to be aware of this issue of infiniteness while using lattices. Note that we are interested in making elements of lattice (which are inferences and their combination) finite and not in finiteness of lattice. Most of the lattices discussed in this section are based on inferences. We require that Inferences should be finite so that combination of inferences can be treated as logical formula by conjunction or by disjunction of all the elements of the combination as we required this in Section 6.1.

We present following well-known approaches that are mostly used in dealing with infiniteness of inferences, inferences are basic elements of some lattices described in this section.

Equivalence: Some inferences of the ontology/theory may be infinite. In some cases, such inferences can be made finite by using finite equivalent formula. An approach to reduce the cardinality of inferences is to built a lattice of inferences (logical formulas) of theories in which logical formulas are understood up to equivalence.$\Psi \Longleftarrow\!\!\Longrightarrow \Psi' \triangleq valid(\Psi \Leftrightarrow \Psi')$ (Cousot et al., 2013). Note that resulting quotient lattice may still be infinite due to the number of all inferences of the theory.

For instance, Let A, B and x are signature of theory T_0 and if $\{\exists x(Ax \wedge Bx) \vee (Ax \wedge Bx) \vee (Ax \wedge Bx \wedge Ax) \vee Ax \wedge Bx \wedge Ax \wedge Bx, \ldots\}$ is one of the inferences of theory T_0 and this inference is an infinite element. We can remove the infiniteness of this inference by using an equivalent inference which in this case is $(Ax \wedge Bx)$.

Compactness theorem of first-order logic: If above approach is not helpful in making infinite inferences finite then compactness theorem of first-order logic may be helpful in some situations. Let T is a theory and it has an inference S which is infinite then by using compactness theorem of first-order logic, we can try to make equivalent formula which should be union of infinite number of finite subsets, formally represented as

$S = \cup\, SI = union\ of\ subsets\ of\ infinite\ Inference\ s.t.\ each\ subset\ is\ finite.$

This approach works only in those situations in which we can derive infinite number of finite subsets and use them as union of infinite number of finite subsets such that this is equivalent to original infinite inference.

Approximation: If the above mentioned approaches do not work in making infinite inference to finite, then we can use approximation and use approximately equivalent finite formula for that inference.

An approach is using equivalent formula for infinite formula. For instance, we have an infinite formula $(A(x) \wedge B(x) \vee (C(x) \ldots\ldots)$, we can use $A(x) \wedge B(x)$ as its approximate formula.

Next, we describe various lattices that can be built from ontology/theory. Here, we are using the syntax of Description Logics (DLs) for artifacts of ontologies. Artifacts in DLs can be easily translated in first-order-Logic (FOL). We classify of artifacts of ontology as atomic artifacts and complex artifacts. In this section, we treat negation of atomic artifact and artifacts as combination of property and concepts as atomic artifact ($\exists R.C$) as atomic artifacts. While complex artifacts are formed by the combination (conjunction, disjunction) of atomic artifacts.

Note that, here, we are assuming that elements of the lattices are finite.

5.2.1 Lattice based on single taxonomy

Ontologies are mostly expressed in Description Logics (DL) or its variants especially in OWL. Taxonomy is a part of ontology and is generally considered as a backbone of ontology. In ontology, taxonomy is a generalization and specialization of Concepts. Formally, taxonomy of ontology is a *poset of concepts* $\langle C_i, \sqsubseteq\rangle$, and *ordered by* subsumption $\sqsubseteq$. This poset is not necessarily a lattice because there exist some ontologies which are expressed in a language like EL that do not support infimum ($\bot$). However, this is just a technical issue when treating ontology as a poset then logical formalism of ontology does not matter. When treating ontology as a *poset* (a pure mathematical structure), infimum ($\bot$) can be added to this *poset* and this poset can be made a lattice. Lattice based on single taxonomy is represented as $\langle C_i, \sqsubseteq, \top, \bot, \cup, \cap\rangle$.

Properties are also one of the main artifact types of Ontology. Concepts can also be represented in terms of properties. For instance, $C \sqsubseteq \exists R_1.D$ where C, D are concepts and R_1 is a relationship.

It is often the case that two ontologies do not have equivalent corresponding atomic concepts, in this case atomic concepts of one ontology are mapped to complex concepts. Original taxonomy consists of primitive concepts, however, it is possible to *extend original taxonomy* with complex concepts using methods like classification. For instance, in the original taxonomy we have $B \sqsubseteq A$ and $C \sqsubseteq A$, if we found that $B \sqcup C$, a complex concept is more precise than A then we may add it to the taxonomy as $B \sqsubseteq B \sqcup C$, $C \sqsubseteq B \sqcup C$ and $B \sqcup C \sqsubseteq A$.

Ontology taxonomy of concepts is a simple lattice that covers some inferences. Techniques like Formal Concept Analysis (FCA) can be used to build taxonomy and we will discuss in Section 5.2.7.

5.2.2 Lattice containing set of concepts (Power set Lattice of non-logical signature $\wp(\boldsymbol{\Sigma})$)

To cover complex concepts involved in ontology mappings, it is needed to build a more expressive lattice than $\langle C_i, \sqsubseteq, \top, \bot, \cup, \cap\rangle$. Our first attempt in this regard is $\langle \wp(\Sigma_C), \preccurlyeq, \top, \bot, \cup, \cap\rangle$, where $\wp$ is a power set, Σ_C is a set of concepts that also involves concepts of the form of $\exists R_x.C_i$ (DL syntax, a combination of concept and role but treated as atomic artifact) which can be renamed with a unique label, $\preccurlyeq$ is an order relation, $\bot$ bottom element, and $\top$ top element.

We define order relation $\preccurlyeq$ as

$V \preccurlyeq W\ iff\ \{C \in \mathcal{C}_i | S \in V\ and\ C\ \sqsubseteq_i\ S\} \subseteq \{C \in \mathcal{C}_i | S \in W\ and\ C\ \sqsubseteq_i\ S\}$ where V and W are ordered sets and $\sqsubseteq_i$ represents the concept taxonomies in C_i. This forms a complete lattice $\langle \wp(\Sigma_{\mathrm{C}}), \preccurlyeq, \top, \bot, \cup, \cap \rangle$ and every two elements have both least upper bound (LUB) and greatest lower bound (GLB) because of having $\top$ and $\bot$ elements.

Instead of using order relation as $\preccurlyeq$, inclusion relation $\subseteq$ can also be used in the lattice.

Situations when complex artifacts are involved in correspondences of the ontology mappings with no distinction of treating combination as disjunction or conjunction can be handled by lattices of the form $\langle \wp(\Sigma_{\mathrm{C}}), \subseteq, \top, \bot, \cup, \cap \rangle$, $\langle \wp(\Sigma_{\mathrm{C}}), \preccurlyeq, \top, \bot, \cup, \cap \rangle$.

An important feature of this lattice is that there is no distinction in interpreting the combination of signature used in power set of signature, i.e., it is not clear that whether they are interpreted as conjunction or as disjunction. If we commit ourselves to either conjunction or to disjunction, then it is not possible to express all the combinations since mapping may involve a formula that is a combination of conjunction and disjunction. Therefore, this lattice has limited expressivity but it works well in cases where this distinction is not needed and one meaning is used consistently for all mappings. For instance, some correspondences of the mapping use conjunction while some use disjunction as $O_1{:}A \sqsubseteq O_2{:}B \sqcup C$ and $O_1{:}D \sqsubseteq E \sqcap F$. This lattice does not cover such cases.

5.2.3 Lattice of theory (Lattice of inferences of theory)

Here, we treat theory as a set of all the inferences, i.e., closed and ground formulas, inferred from some given axioms of the theory.

We can build the lattice in two ways by changing order relation.

(a) We can define order relation $\overset{*}{\subseteq}$ on $\wp(T)$. Let S and W be two different elements of the $\wp(T)$, order relation $\overset{*}{\subseteq}$ between them is defined as

$S \overset{*}{\subseteq} W$ iff $S \vDash_T W$, where $S \vDash_T W\ iff\ \forall \phi \in S, \exists \psi \in W\ \phi \vDash_T \psi$

and the lattice is $\langle \wp(T), \overset{*}{\subseteq}, \top, \bot, \cup, \cap \rangle$.

(b) Instead of using $\overset{*}{\subseteq}$ order relation, elements of lattice can be ordered by inclusion relation $\subseteq$, and the lattice is $\langle \wp(T), \subseteq, \top, \bot, \cup, \cap \rangle$.

Lattices $\langle \wp(T), \subseteq \top, \bot \rangle$ and $\langle \wp(T), \overset{*}{\subseteq}, \top, \bot, \cup, \cap \rangle$ can represent any combination of conjunction and disjunction of atomic artifacts. This lattice provides a way to relate ontology mapping with mappings between these lattices (as we've done in Chapter 6) and that will become a basis to check the existence of defects in ontology mappings.

Lattice $\langle \wp(T), \subseteq \top, \perp \rangle$ is different from Lattice based on single taxonomy as this lattice directly represents inferences, while in Lattice based on single taxonomy represents concepts/relationships and the order relationship represents some inferences. We use this lattice most of the time in the rest of thesis particularly in Chapter 6. Remember that inferences used in Lattice $\langle \wp(T), \subseteq \top, \perp \rangle$ correspond to ground and closed formulas.

5.2.4 Lattice of Language (Power set Lattice of non-logical and logical signature $\wp(L)$

Lattice on signature $\{\Sigma \cup \Lambda\}$ covers all possible formulas generated by the language. Lattice built on signature $\{\Sigma \cup \Lambda\}$ differs with $\langle \wp(T), \subseteq, \top, \perp, \cup, \cap \rangle$ in terms of elements of lattice. In this lattice, elements of lattice are not based on one particular theory.

There are two possible lattices depending on the defined order relationship.

(a) Lattice $\langle \wp(L), \overset{*}{\subseteq}, \top, \perp, \cup, \cap \rangle$, where $\overset{*}{\subseteq}$ is defined as

Let S and W be two different elements of the $\wp(L)$, order relation $\overset{*}{\subseteq}$ between them is defined as

$S \overset{*}{\subseteq} W$ iff $S \vDash_T W$, where $S \vDash_T W\ iff\ \forall \phi \in S, \exists \psi \in W\ \phi \vDash_T \psi$.

However, partial order in the lattice $\langle \wp(L), \overset{*}{\subseteq}, \top, \perp, \cup, \cap \rangle$ is more sparse as compared to the lattice $\langle \wp(T), \overset{*}{\subseteq}, \top, \perp, \cup, \cap \rangle$. The reason is that elements of later lattice are inferences of theory T while elements of the former lattice are formulas that can be expressed in language L and elements in the former lattice are partially ordered which is more sparse as compared to the elements in the later lattice.

(b) Lattice $\langle \wp(L), \subseteq, \top, \perp, \cup, \cap \rangle$, where $\subseteq$ is inclusion relation $\subseteq$.

This lattice can be interesting. Any mapping defined between ontologies can be directly represented with this lattice. For instance, if $L_i: A \sqsubseteq L_j: B$ is a DDL mapping, because A/B are not formulas in each ontology, a Galois Connection on the lattice of language can represent directly $A \sqsubseteq B$ as $\alpha(A) = B$ and because B is potentially more general than A, some $\gamma(B) \supseteq A$. The same approach cannot be pursued whit lattice of theory where only inferences can be mapped.

5.2.5 Lattice of Logical Formulas

Taking the insight from the work of Cousot and colleagues (Cousot et al., 2013), in which they have built a lattice whose elements are logical formulas Φ and order relation in it is implication $\Rightarrow$, we can built similar lattice for formulas (inferences) of a theory as $\langle \Phi_{\mathrm{T}}, \Rightarrow, \perp, \top, \wedge, \vee \rangle$. Φ_{T} represents logical formulas (inferences) in theory T. We name this lattice a logical formula lattice.

Order in this lattice is defined as $(\psi \sqsubseteq \psi' \triangleq ((\forall \left(\vec{x}_\psi \cup \vec{x}_{\psi'}\right): \psi \Rightarrow \psi' \in T)$ (and can be quotiened to a partial order by $(\psi \equiv \psi' \triangleq ((\forall \left(\vec{x}_\psi \cup \vec{x}_{\psi'}\right): \psi \Leftrightarrow \psi' \in T)$.

$\langle \Phi_{\mathrm{T}}, \Rightarrow, \bot, \top, \wedge, \vee \rangle. \rangle$ is a lattice but it is not a complete lattice since infinite conjunction and infinite disjunction are missing in first-order logic.

To relate lattice of theory with the lattice of logical formulas of a theory, we highlight important similarities and differences between them. Both lattices cover all the inferences drawn from a theory. However, Lattice of logical formulas of a theory is more concise than lattice of theory due to arranging equivalent inferences at same place and without introducing (sub)set of inferences. These lattices are remarkably different in terms of order relation; In most of the cases, order relation in lattice of theory is subset while order relation in lattice of logical formulas of a theory is satisfiability. The potential drawback in lattice of logical formulas of a theory is that equivalence is defined differently in different logics.

5.2.6 Lattices of theories

Theories can be arranged into a Lattice of theories. A lattice of theories, as opposed to lattice of theory, provides an additional structuring of inferences by grouping them according to theories (i.e., in this case, axioms).

Theories can be ordered in different ways. Sowa arranges theories in lattice of theories on the basis of generalization and specification (Sowa, 2000). For example, in Sowa's lattice of theories *Abstract* is more general than *Intention*. Gruninger and colleagues arrange theories in a repository (a poset) and ordered them by non-conservative extension if they have same signature and placed them in the same hierarchy while relate (order) theories having different signature by the order relation of faithful interpretation and reducibility (Grüninger et al., 2012).

Gruninger define reducibility as

A theory T is *reducible* to a set $T_1, \dots, T_n$ iff T faithfully interprets each theory T_i and $T_1 \cup \dots \cup T_n$ faithfully interprets T.

In our case, this lattice is not useful because this lattice is difficult to be built starting from a flat ontology specification except when we have modular ontologies which are arranged, for instance, on the basis of reducibility. Since, building a lattice of theories based on ordering (sub)theories in the order of generalization and specialization is not a trivial task as one has to order each theory in a generalization (specialization) order.

5.2.7 Formal Concept Analysis (FCA)

Formal Concept Analysis (FCA) (Birkhoff, 1967) is widely used in the context of ontology. FCA is generally used to build a taxonomy of ontology and it is considered as a process of abstracting concept description from a set of objects described by attributes (Ganter et al., 1997).

Context $\mathcal{K}$ associates a set of objects G to a set of attributes M in an incidence relation I; $\mathcal{K} = (G, M, I)$.

For example, a context of Pizza is presented in Illustration of RCA where objects are pizza, attributes are thin and thick describing the shape, and incidence relation is interpreted as 'pizza shape'.

FCA focuses on grouping objects together on the basis of shared attributes.

Data in FCA is represented as a *formal context where*

$$A^I = \{m \in M | gIm \text{ for all } g \in A\} \text{ if } A \subseteq G$$

$$B^I = \{g \in G | gIm \text{ for all } m \in B\} \text{ if } B \subseteq M$$

And A is maximal sets of objects named as *extents* in FCA and B is maximal sets of attributes named as *intents* in FCA. A^I and B^I defined above shows that there is one-to-one correspondence between *extents* and *intents*.

There exists Galois connection $\langle \alpha, \gamma \rangle$ between power set lattice of G and power set lattice of M where

$$\alpha(X) = X^I \text{ for } X \subseteq G \text{ and } \gamma(Y) = Y^I \text{ for } Y \subseteq M$$

The set $\mathfrak{B}(G, M, I) \triangleq \{(A, B) | A \subseteq G, B \subseteq M, A^I = B, B^I = A\}$ ordered by $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow B_2 \subseteq B_1$ is a complete lattice.

Ganter (Ganter, 2007) extends this notion and uses Galois connection to ordered sets (G, M, I) and (H, N, J). $I \subseteq G \times M$ and $J \subseteq H \times N$ where M and N are sets of attributes and G and H are sets of objects, and I and J are binary relations as $I \subseteq G \times M$ and $J \subseteq H \times N$.

$\alpha: G \to N, \gamma: H \to M$

Satisfying

$g\ I\ \gamma(h)$ iff $h\ J\ \alpha(g)$ for antitone Galois connection

Where $g \in G, h \in H$, and I and J are binary relations and these are the generalization of Galois connection $\alpha(g) \preccurlyeq h$ iff g $\preccurlyeq \gamma(h)$). This is a Galois connection between two ordered sets $\langle G, M, \leq \rangle$ and $\langle H, N, \preccurlyeq \rangle$.

A monotone Galois connection is achieved by replacing (H, N, J) with (N, H, J^{-1}).

Ganter (Ganter, 2007) generalized α and γ mappings further for dealing with arbitrary binary relations. A pair of mapping by pair of relations $\alpha \subseteq G \times N$ and $\gamma \subseteq H \times M$. Galois connection condition remains the same $g\ I\ h^\gamma$ iff $h\ J\ g^\alpha$ with a new constraint to make Galois connection condition strong. This constraint is α is the largest relation satisfying Galois condition for the given γ and conversely.

To allow properties to be mixed in intents of FCA, an approach Relational Concept Analysis (RCA), a variant of FCA, has been introduced (Hacene, et al., 2013. We will illustrate the use of this approach in creating a taxonomy, a taxonomy which is more enriched than obtain from FCA.

Relational Concept Analysis (RCA) (Hacene et al., 2013) is an extension of FCA in which set of attributes consists of both concepts and properties. Main idea behind RCA approach is to express a relational model based on the entity-relationship model. Relational information is represented by cross-tables between objects $r \subseteq M \times N$, where M is domain $dom(r)$ and N is range $ran(r)$. A conceptual scaling process is used to represent relations between objects as relational attributes. Scaling meant to have a special interpretation. Generally, many-valued contexts are translated to one-valued context via concept scaling. To obtain relational attribute, we need scaling operators to be used on concepts. There are three forms of relational attributes obtained after the application of scaling operator.

Existential scaling: an object o is linked (by r) to at least one object of the extent of a concept $extent(C)$.

$$\exists r.C: r(o) \cap extent(C) \neq \emptyset$$

Universal scaling: an object is linked (by r) only to objects of the extent of a concept $extent(C)$.

$$\forall r.C: r(o) \subset extent(C)$$

Universal Existential scaling: an object is linked (by r) only to objects of the extent of a concept $extent(C)$ and $r(o)$ is not an empty set.

$$\forall\exists r.C: r(o) \subset extent(C) \text{ and } r(o) \neq \emptyset$$

In RCA, an iterative process is used for building a concept lattice where concept intents include binary and relational attributes. A main objective of RCA is to enrich the lattice obtained by FCA with relational attributes and this results in merging of two or more lattices obtained by FCA.

A unique relational context family consists of all the contexts $\mathcal{K}$ (set of object-attribute contexts $\mathcal{K}_i = \langle G, M, I \rangle$ and relations $\mathcal{R}$ (set of relations object-object contexts r_k ($r_k \subseteq M \times N$)).

A function π maps a context $\mathcal{K}$ to the set of relations $\mathcal{R}$. $\pi(\mathcal{K}) = \{r \in \mathcal{R} | dom(r) = G\}$. Similarly function π can be defined for mapping a context $\mathcal{K}$ to $ran(r)$.

Building a final lattice in RCA starts from building FCA lattice. In the next step, relational attributes $r(o_i, o_j)$ are taken into account and all relations are examined between objects.

When two objects are in relation for instance, object x is in relation with i through a relation r_1 and object a is in the extext of two or more concepts in the intital lattice here for instance, to C_1 and C_2 then object i is given two new relational attributes as $\exists r_1.C_1$ and $\exists r_1.C_2$ and same procedure is performed for all relations. Then new lattice is constructed by considering all these additional attributes. Creation of new lattices in the same manner is continue until a fix point is reached, i.e., no more modification in the lattice is possible.

We illustrate the use of RCA in building a taxonomy in the following example.

Illustration of RCA:

Extract a taxonomy from a dataset of Pizza. Objects in this dataset are Pizza and ingredients of Pizza. Pizza is described by attributes (thin, thick) about its shape while ingredients are described by their category (Fish, Dairy, Meat, Veg/Fruit). Relation hasTopping relates objects Pizza and Ingredients.

Relational context Family in this example is

Object-Attribute contexts

a. Pizza
b. Ingredients

Object-object contexts

a. $hasTopping \subseteq Pizza \times Ingredients$

Relational context Family Object-Attribute contexts

Pizza

Pizza	Thin	Thick
A	×	
B		×
C		×
D	×	
E		×
F	×	

Ingredients

Ingredients	Fish	Dairy	Meat	Veg, Fruit
Tomato Sauce				×
Shrimps	×			
Chicken			×	
Mutton			×	
Pineapple				×
Cream		×		
Cheese		×		

Relational context Family Object-Object contexts $R_j = (O_k, O_l, I_j)$,

hastopping	Tomato Sauce	Shrimps	Chicken	Mutton	Pineapple	Cream	Cheese
A	×		×				×
B	×	×					×
C			×		×	×	

D	×			×			×
E			×			×	
F	×				×	×	

RCA process starts with building context lattices (Lattice of Pizza and Lattice of Ingredients) using FCA approach.

We describe the step-by-step procedure for building Lattice of Pizza, while other lattices are built analogously.

X is a set of objects, Y is a set of attributes, $\langle x, y \rangle \in I$ object x has attribute y. We know that $G^I = M$ and $M^I = G$

For table describing Relational context Family Object-Attribute context of Pizza, we have

$\{A\}^I = \{Thin\}$, $\{B\}^I = \{Thick\}$, $\{C\}^I = \{Thick\}$, $\{D\}^I = \{Thin\}$, $\{E\}^I = \{Thick\}$, $\{F^I\} = \{Thin\}$, $\emptyset^I = \{A, B, C, D, E, F\}$, $\{A, B, C, D, E, F\}^I = \emptyset$

$\{Thin\}^I = \{A, D, F\}$, $\{Thick\}^I = \{B, C, E\}$, $\emptyset^I = \{Thin, Thick\}$, $\{Thin, Thick\}^I = \emptyset$

Sub-concept and Super concept ordering is defined as

For formal concepts $\langle X_1, Y_1 \rangle$ and $\langle X, Y_2 \rangle$ of $\langle G, M, I \rangle$

$\langle X_1, Y_1 \rangle \leq \langle X_2, Y_2 \rangle$ iff $X_1 \subseteq X_2$ and $Y_2 \subseteq Y_1$

$$\langle \emptyset, \{Thin, Thick\} \rangle \leq \langle \{B, C, E\}, \{Thick\} \rangle$$

$$\langle \emptyset, \{Thin, Thick\} \rangle \leq \langle \{A, D, F\}, \{Thin\} \rangle$$

$$\langle \{A, D, F\}, \{Thin\} \rangle \leq \langle \{A, B, C, D, E, F\}, \emptyset \rangle$$

$$\langle \{B, C, E\}, \{Thick\} \rangle \leq \langle \{A, B, C, D, E, F\}, \emptyset \rangle$$

And the resulting lattice is shown in Figure 5-4. Analogously, Lattice of Ingredients has been built and it is shown in Figure 5-5.

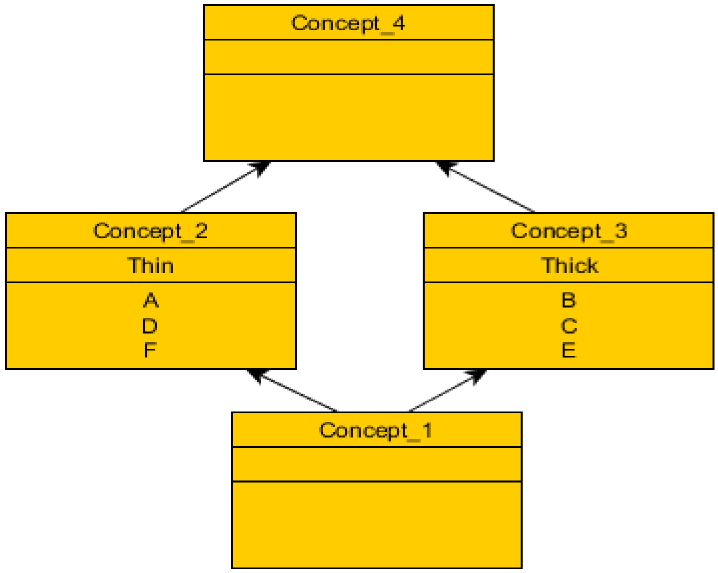

Figure 5-4. Lattice of Pizza

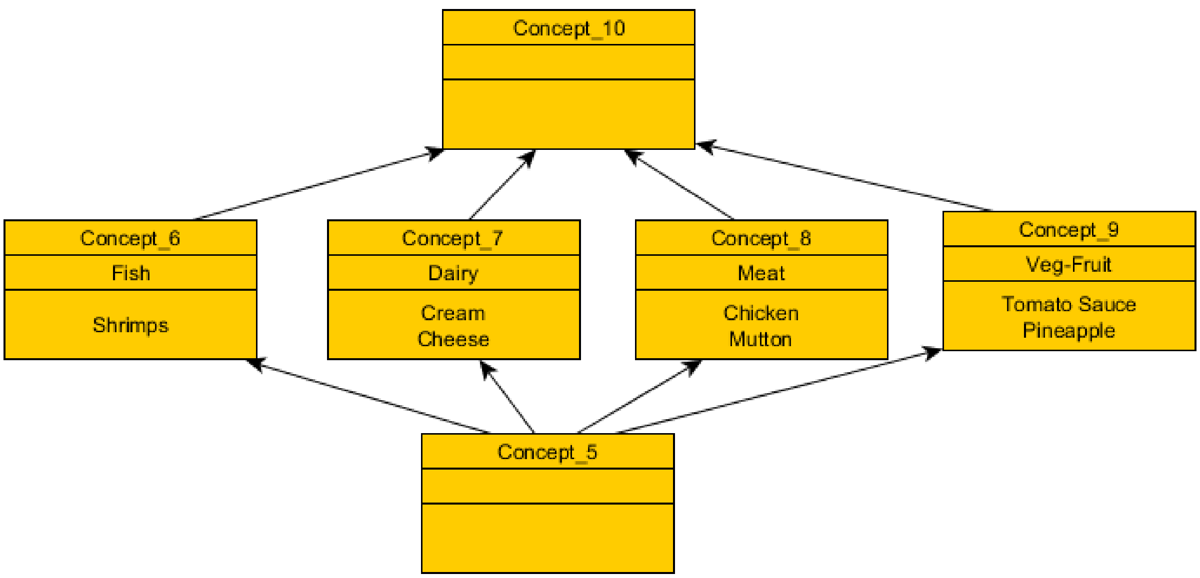

Figure 5-5. Lattice of Ingredients

Given an object-object context $R_j = (O_k, O_l, I_j)$,

We have a relation *hastopping* between an object of domain O_k (Pizza) and concepts formed on object O_l (Ingredients). To obtain relational attribute, we need scaling operators to be used on concepts.

In this example, use of scaling operator results in new concepts which are described as below.

An object B has one topping in $Concept6$. It has other links with extent of other concept.

$\exists hastopping.Concept6$ is assigned to B (according to the table of ingredients and lattice of ingredients)

An object A has one topping in $Concept7$. It has other links with extent of other concept.

$\exists hastopping.Concept7$ is assigned to A (according to the table of ingredients and lattice of ingredients).

Similarly, $\exists hastopping.Concept7$ is assigned to B, C, D, E, F.

An object A has one topping in $Concept8$. It has other links with extent of other concept.

$\exists hastopping.Concept8$ is assigned to A (according to the table of ingredients and lattice of ingredients).

Similarly, $\exists hastopping.Concept8$ is assigned to C, D, E.

An object A has one topping in $Concept9$. It has other links with extent of other concept.

$\exists hastopping.Concept9$ is assigned to A (according to the table of ingredients and lattice of ingredients).

Similarly, $\exists hastopping.Concept9$ is assigned to B, C, D, F.

Following table shows object-attribute context of Pizza.

	$\exists hastopping.Concept6$	$\exists hastopping.Concept7$	$\exists hastopping.Concept8$	$\exists hastopping.Concept9$
A		×	×	×
B	×	×		×
C		×	×	×
D		×	×	×
E		×	×	
F		×		×

Original lattice of pizza is extended with this added information of relation.

Subconcept and Super concept ordering for lattice is

$$\langle\emptyset, \left\{\begin{matrix}Thin, Thick, hasTopping.concept6, hasTopping.concept7, hasTopping.concept8, \\ ,hasTopping.concept9\end{matrix}\right\}\rangle \leq \langle\{A, D\}, \emptyset\rangle$$

$$\langle\emptyset, \left\{\begin{matrix}Thin, Thick, hasTopping.concept6, hasTopping.concept7, hasTopping.concept8, \\ ,hasTopping.concept9\end{matrix}\right\}\rangle \leq \langle\{F\}, \emptyset\rangle$$

$$\langle\emptyset, \left\{\begin{matrix}Thin, Thick, hasTopping.concept6, hasTopping.concept7, hasTopping.concept8, \\ ,hasTopping.concept9\end{matrix}\right\}\rangle \leq \langle\{B\}, \emptyset\rangle$$

$$\langle\emptyset, \left\{\begin{matrix}Thin, Thick, hasTopping.concept6, hasTopping.concept7, hasTopping.concept8, \\ ,hasTopping.concept9\end{matrix}\right\}\rangle \leq \langle\{C\}, \emptyset\rangle$$

$$\langle\emptyset, \left\{\begin{matrix}Thin, Thick, hasTopping.concept6, hasTopping.concept7, hasTopping.concept8, \\ ,hasTopping.concept9\end{matrix}\right\}\rangle \leq \langle\{E\}, \emptyset\rangle$$

$$\langle\{A, D\}, \emptyset\rangle \leq \langle\{A, D, F\}, \{Thin\}\rangle$$

$$\langle\{F\}, \emptyset\rangle \leq \langle\{A, D, F\}, \{Thin\}\rangle$$

$$\langle\{B\}, \emptyset\rangle \leq \langle\{B, C, E\}, \{Thick\}\rangle$$

$$\langle\{C\}, \emptyset\rangle \leq \langle\{B, C, E\}, \{Thick\}\rangle$$

$$\langle\{E\}, \emptyset\rangle \leq \langle\{B, C, E\}, \{Thick\}\rangle$$

$$\langle\{A, D\}, \emptyset\rangle \leq \langle\{A, C, D, E\}, \{hasToping.concept8\}\rangle$$

$$\langle\{C\}, \emptyset\rangle \leq \langle\{A, C, D, E\}, \{hasToping.concept8\}\rangle$$

$$\langle\{E\}, \emptyset\rangle \leq \langle\{A, C, D, E\}, \{hasToping.concept8\}\rangle$$

$$\langle\{A, D, F\}, \{Thin\}\rangle \leq \langle\{A, B, C, D, F\}, \{hasToping.concept9\}\rangle$$

$$\langle\{B\}, \{Thin\}\rangle \leq \langle\{A, B, C, D, F\}, \{hasToping.concept9\}\rangle$$

$$\langle\{C\}, \{Thin\}\rangle \leq \langle\{A, B, C, D, F\}, \{hasToping.concept9\}\rangle$$

$$\langle\{A, D, F\}, \{Thin\}\rangle \leq \langle\{A, B, C, D, E, F\}, \{hasToping.concept7\}\rangle$$

$$\langle\{A, C, D, E\}, \{hasToping.concept8\}\rangle \leq \langle\{A, B, C, D, E, F\}, \{hasToping.concept7\}\rangle$$

$$\langle\{B, C, E\}, \{Thick\}\rangle \leq \langle\{A, B, C, D, E, F\}, \{hasToping.concept7\}\rangle$$

$$\langle\{A, B, C, D, E, F\}, \{hasToping.concept7\}\rangle \leq \langle\{A, B, C, D, E, F\}, \emptyset\rangle$$

And the resulting lattice is shown in Figure 5-6.

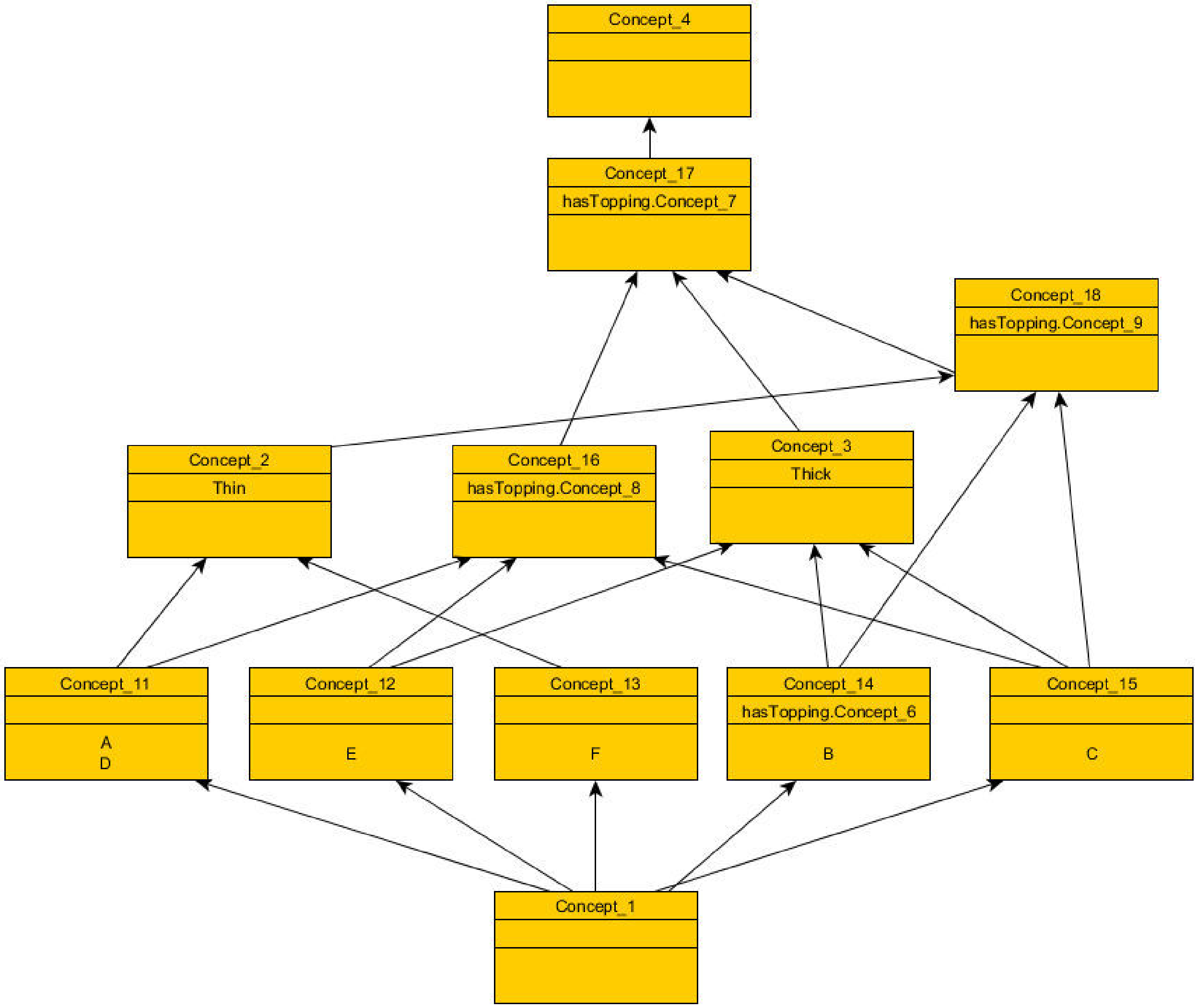

Figure 5-6. Lattice of Pizza (RCA)

Even though lattices used in RCA approach contain both concepts and relations and some complex artifacts but still these lattices may not have all the inferences of theory because based on closed world assumption. Therefore, correspondences of ontology mapping may contain some logical formulas of theory that are not part of lattice based on FCA or RCA. This procedure becomes hard due to involvement of the objects/instances and volatile because based on sample of objects.

We shall use these lattices in our definition of compatible mappings as *Representation function R* maps artifacts/inferences of ontology to one of the lattice described in this section.

5.3 Relating Ontology Mappings and Lattices

Lattices provide more or less direct way to relate mappings between ontologies and mapping between these lattices. In Table 5-1, we describe the ontology mapping formalism (described in Chapter 2) and whether these formats permit symbol to symbol or formula to formula mappings and what lattices described in Section 5.2 is appropriate for that formalism and comments about the choice.

Table 5-1. Corresponding Lattices for given Ontology Mapping

Ontology mapping	**Mapping**	**Corresponding Lattice(s) for given ontology mapping**	**Comments**
DDL	Symbol to Symbol, Formula to Formula, Symbols to Formula, Formula to Symbol	Lattice based on single Taxonomy, Lattice containing complex concepts, Lattice of theory, Lattice of Language, Lattice of logical formulas, Lattice of theories, FCA/RCA lattices	Appropriate lattice should be selected that covers both domain and range of mapped inferences
DFOL	Formula to Formula	Lattice of Language, Lattice of theory, Lattice of logical formulas	Lattice of inferences can be used under some conditions described in Chapter 6.
Enderton	Symbols to Formula	Lattice of Language, Lattice of theory, Lattice of logical formulas	Lattice of inferences can be used under some conditions described in Chapter 6.
Morphism	Formula to Formula	Lattice of Language, Lattice of theory, Lattice of logical formulas	Lattice of inferences can be used under some conditions described in Chapter 6.
Inferences are mapped	Formula to Formula	Lattice of Language, Lattice of theory, Lattice of logical formulas	Lattice of inferences can be used under some conditions described in Chapter 6.

Lattice of language covers all kinds of ontology mapping, however, the main problem is that available ontology mappings are partial and therefore it is not possible in such a situation that we have complete mapping for lattice of language. Indeed, for instance, from a simple mapping $O_i: A \cup B \subseteq O_j: C$, if we use a lattice of language then we have to define a mapping for A and for B; however, this mapping is not available in the original ontology mapping so it is required to complete it in some ways. This is not the case with a lattice of inferences where only $A \cup B$ is defined (except if A, B can be inferred independently by using other axioms). The same applies for lattice of taxonomies where artifacts are part of the structure.

The sufficient condition to check the existence of defects in ontology mappings is to have mappings of inference of given ontologies. In Chapter 6, we have identified the conditions by which given mappings can be used for checking the existence of defects by means of lattice of inferences and mapping between lattice of inferences.

If mapping between ontologies is formalized in DL, under extension of the taxonomy, a mapping between taxonomy can be possible. However, if ontology is not arranged in a taxonomy, i.e., inferences are more than taxonomical one, and it is needed to map symbols of the language of theory, then simple taxonomy will not work and it is needed to move to lattice containing all possible inferences as we've done in Chapter 6.

Remark: If there exists a Galois connection between lattices of taxonomies of ontologies then it also exists a Galois connection between lattices of inferences of ontologies $s.t.\ \sigma$ inferred from lattice of taxonomy iff Lattice of inference has inference σ.

The reason is that they are equivalent lattices in terms of inferences, however, they differ in terms of order relation. Order relation in taxonomy is generally implication while in lattice of inferences it is subset $\subseteq$ relation.

5.4 Compatible and Incompatible ontology mappings

Galois connection is a natural choice when dealing with preservation of orders between two posets. In the context of Ontology mappings, Galois connection is interesting whenever dealing with mappings between ontologies because at least it:

1. is Independent of the kind of formalization (such as the kind of logics) used to represent ontologies;
2. Introduces a kind of unified treatment of defects (by its definition and it is proved in Chapter 6)
3. Introduces a kind of unified syntax for mappings (even $\sqsubseteq, \sqsupseteq$ can be ambiguous)

5.4.1 Key points

Defects caused by symptoms such as inconsistency, incoherence and violation of principle of conservativity can be obtained by logical theory as described in Chapter 4. However, finding out defects using notions presented in Chapter 4 requires ontologies and mappings with specific features otherwise defects do not occur. Some of these features are:

1. Ontologies should not contain implicit axioms (e.g., any $A \cap B = \emptyset$ should be explicitly stated)

2. Mappings should be almost complete, for instance, if $O_1{:}A \equiv O_2{:}C$, if $O_1{:}B$ is somehow related to $O_2{:}C$, this should be formally stated with appropriate axioms, otherwise, some inconsistences or incoherence or whatever do not occur.
3. Galois connections can first be perceived as a methodological tool for dealing with problems of ontologies with implicit axioms and mappings which may be not complete. Indeed, Galois connection even in the case where some implicit axioms should be part of the ontology, Galois connection may warn unsuitable situations. Mappings are forced to be completed because they are always presented in a "function style" and not as constraints.
4. Galois connections can also be perceived as a tool for representing a *"relative semantics"* for mappings. A relative semantics of a mapping means that the meaning of $m{:}O_1 \rightarrow O_2$ is defined in term of the meaning of $m'{:}O_2 \rightarrow O_1$ and vice-versa. This corresponds to a natural approach where each ontology provides a specific perspective on a domain and mappings state correspondences from one ontology to another and vice-versa, following the single ontology perspectives. More formally a meaning of an element X can be defined in term of a lfp for mappings compositions both $m \circ m'$ or $m' \circ m$.

Concrete relation such as $\sqsubseteq, \sqsupseteq$ can be sometimes ambiguous, as they implicitly covers the case of equality. We introduce a kind of unified syntax for mappings by treating all relations as abstract mapping relation and Galois connection covers such relation implicitly. For instance, considering ontology O_j is more general than ontology O_i and artifact A of ontology O_i is mapped to artifact B of ontology O_j in α mapping, then to have a Galois connection, in γ mapping B should be mapped to the least upper bound of A. This is shown in Figure.

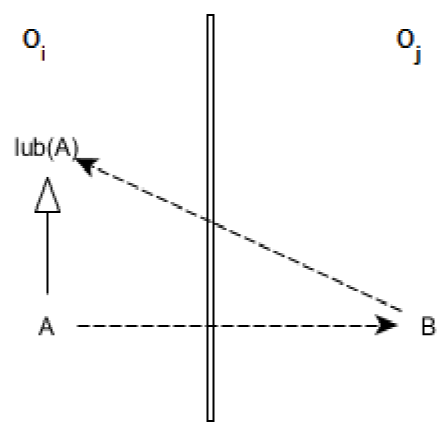

Figure 5-7. Unified treatment of mapping relations

5.4.2 Compatible and Incompatible ontology mappings

Galois connection is defined on ordered set, which in our case is a lattice. We have presented various lattices in Section 5.2, however, we are interested in defining notions of compatibile and incompatible ontology mappings on a lattice which is generic one and could be used in all possible situations for checking the existence of defects caused by ontology mappings. We find that this lattice is a Lattice of theory. This Lattice of theory consists of ordered set of subsets of inferences of a theory.

We can relate other lattices presented in Section 5.2 with Lattice of theory.

Lattice based on single taxonomy and Lattice containing complex concepts, Lattice of logical formulas (implicitly or explicitly) possess some inferences of the theory and these inferences are part of Lattice of theory.

Lattice of theories consists of those elements, which are sub-theories of that theory for which Lattice of theory was built. Theories are represented by logical axioms, which can be treated as inference, as inferences of the theory are derived from these axioms. These axioms are part of lattice of theory.

Lattice of language is most expressive and it not only contains inferences of the theory but also all other possible logical formulas which can be expressed in the signature of the language. However, this most expressive lattice is of no use for us in our problem of defining compatibility of ontology mappings. The reason is that if Galois connection is established between lattices of language then we cannot recover mapping f and g between ontologies as we have done in Chapter 6.

Lattices other than power set lattice of theory presented in Section 5.2 are either subset of power set lattice of theory or it (Lattice of language) is not useful in the context of defining compatibility of ontology mappings.

Mappings are treated, here, either as an effect of mapping $f(O_i, O_j, M_{ij})$, defined in Chapter 4, as

$$\left\{f\left(O_i, M_{ij}\right) \middle| x \leftrightarrow y, x \in I(O_i), y \in I\left(O_j\right)\right\}$$

OR

a theory interpretation according to Enderton mapping as

$$\langle \sigma, Syt^m(\sigma)\rangle \Big| \sigma \in O_i, Syt^m(\sigma) \in O_j\Big\}$$

To provide a universal formulation of compatibility and incompatibility of ontology mappings, we need to add some constraints.

The appropriate necessary additional constraints are

- For representation of an ontology O by a lattice L, there is a function $R: O \to L$ which is injective and O is the ontology representation such that L embeds O.

  Representation function R maps artifacts/inferences of ontology to one of the lattice described in Section 5.2.

- As a consequence, if m is an ontology mapping between O_i and O_j then the considered Galois connection should be limited to $m(x) = y$ (or a mapping triplet $\langle m; x; y\rangle$) and by procedure P

  $P(m)R(x) \geq R(y)$, i.e., the $P(m)$ mapping should correspond to the M mapping.

The contents of ontology O_i and O_j may be just simple taxonomy, restricted or extended taxonomy or all the inferences of that ontology or some subset of inferences. Depending on the contents of ontology, mapping space is defined/built accordingly.

Here, we are just using ontology and not mentioning its content explicitly.

Mapping space m has all the couples of ontologies O_i and O_j, i.e.,

$\langle x, y\rangle$ when $\langle x, y\rangle \in m$; where $x \in O_i$ and $y \in O_j$

$\langle x, *\rangle$ when $\nexists z\langle x, z\rangle \in m$;where $x \in O_i$ and $z \in O_j$

$\langle *, y\rangle$ when $\nexists z\langle z, y\rangle \in m$; where $z \in O_i$ and $y \in O_j$

$D^*: (m: O_i \mapsto O_j) \to O_i)$ and $C^*: (m: O_i \mapsto O_j) \to O_j)$ relate mapping space (all couples of O_i/O_j) to ontologies, i.e.,

$$D^*(\langle x, y\rangle) = x$$

$$D^*(\langle x, *\rangle) = \perp$$

$$C^*(\langle x, y\rangle) = y$$

$$C^*(\langle *, y\rangle) = \perp$$

Function Π is defined from mapping space to lattices, as mapping space can be extended for more expressive Lattices such as lattice of inferences to cover mapping of all the elements of such lattices.

The definition of compatibility/incompatibility is as follows:

Definition 5-17 Compatible and Incompatible ontology mappings: Two ontology mappings $m_1: O_i \mapsto O_j$ and $m_2: O_j \mapsto O_i$ are compatible, given a representation function R (extended to any inference in single ontologies), iff for some procedures P, such that

$$\forall \langle x, y\rangle \in m_1: O_i \mapsto O_j \quad R(D^*(x) = \Pi^1(\langle x, y\rangle)$$

$$P(m_1)\Pi^1(\langle x,y\rangle) \geq R\big(C^*(y)\big)$$

and

$$\forall\langle y,x\rangle \in m_2: O_i \mapsto O_j \quad R(D^*(x) = \Pi^1(\langle x,y\rangle)$$

$$P(m_2)\Pi^2(\langle y,x\rangle) \geq R\big(D^*(x)\big)$$

There is a Galois connection between L_i and L_j.

The condition expressed in definition above provides a kind of commutativity saying that by performing D^* and then R_i is the same as applying Π_i and analogously for $C^*, R\ and\ \Pi_j$. Also another kind of commutativity exists in the condition, which is performing D^* and then R_i or just performing Π_i and then $P(m_i)$ or just perform Π_j. This is depicted in the Figure 5-8 and this figure illustrates the procedure of transforming ontologies into lattices and then applying functional mapping between lattices. Similarly, mapping $m: O_j \mapsto O_i$ is incorporated with source ontologies and lattices by the same procedure.

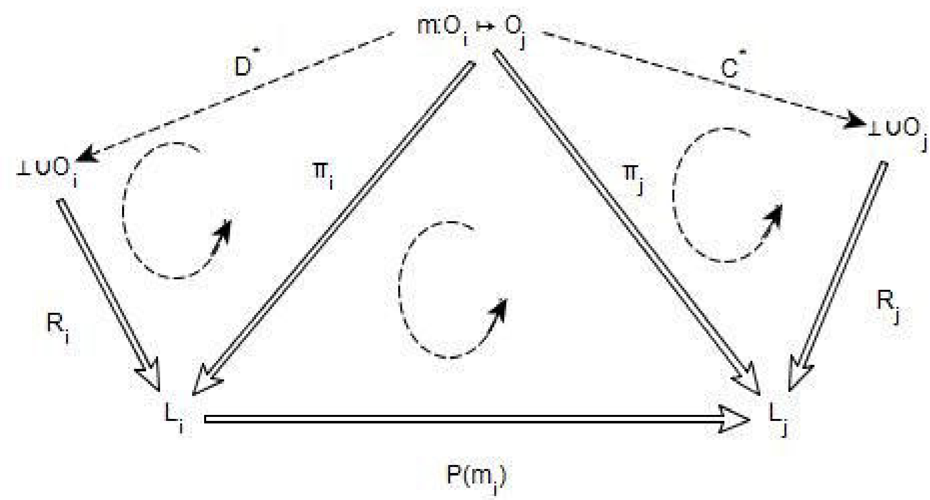

Figure 5-8. Illustration of transforming ontologies into Lattices

Above condition is easily applicable to ontology mappings which are in the form of function but it is not case for DDL like mappings. However, in this specific case, the convention $m_1: O_i \mapsto O_j$ really means $f(O_i, O_j, m_1)$ as defined in Chapter 4 and the condition above should be read as

$$\{f\big(O_i, M_{ij}\big) | x \leftrightarrow y, x \in I(O_i), y \in I\big(O_j\big)\}$$

For some procedure P, such that when $x = y$ (i.e., x and y are equivalent artifacts/interferences) is in $f(O_i, O_j, m_1)$ then if $R(x)$ and $R(y)$ are in L_i and L_j respectively, then $P(m_1)R(x) \geq R(y)$.

Note that the procedure P is mapping independent to be sure to use the same procedure for getting lattice mappings from ontology mappings. Also, note that by using order relationships on lattices, it seems to be possible to formally express a connection between ontology mappings and lattice mappings. However, when a mapping involves subsumption (as in the case of DDL) it remains unclear which order relationship should be used. Nevertheless, the key point is that for any "mapping triple" there is only one way to codify it. For instance, if a mapping triple is ⟨*sub, A, B*⟩ (meaning A sub B) then this always corresponds to $R(A) \leq P(m1)(R(B)$, i.e., sub corresponds to $\leq$.

This is possible because definition above takes into account 2 ontology mappings. In this case, it is possible to use the same convention for both mappings, i.e., for each mapping, only subsumption in one direction can be specified. Consequently, ontology mappings can be codified by using only <=, by assuming that a mapping can be rearrange in 2 mappings.

Under the constrained definition, universality of incompatibility remains an objective because it requires to proof that *for each procedure P* satisfying the given constraints, resulting connections between lattices are not Galois connection. We prove that if this is the case then no procedure P under given constraints can establish Galois connection between lattices L_i and L_j. However, the definition provides a universal formulation of incompatibility, which is one of the thesis objectives.

Compatibilities and incompatibilities can also be stated at the level of ontology artifacts according to the following definition.

Definition 5-18 Compatible and Incompatible ontology artifacts: Given ontology mappings m_1 and m_2, functions α and γ built by respecting all the constraints mentioned in the definition of compatible and incompatible ontology mappings, an ontology artifact A is compatible with ontology artifacts $\alpha(\{A\})$ iff Galois connection conditions are respected between A and $\alpha(\{A\})$. Symmetrically, an ontology artifact B is compatible with ontology artifacts $\gamma(\{B\})$ iff Galois connections conditions are respected between B and $\gamma(\{B\})$. Otherwise, involved artifacts are incompatible.

5.5 Sketch of Detecting ontology mappings defects with the notions of Compatible and Incompatible ontology mappings

In this section, we use description logic syntax for presenting ontology axioms.

In this section, when only atomic artifacts that are part of the respective taxonomy of source ontologies are used in correspondences of ontology mappings then lattice of taxonomy of source ontology is used. While, when complex artifacts are used in ontology mappings, then we use lattice of extended taxonomy of source ontology.

We treat Ontology mappings as we are using DDL formalism and the defects listed below are those that occur in DDL.

In this section, we show that our Galois connection-based notion of compatibility detects defects and covers patterns used for debugging ontology mappings (except redundancy and non-standard user-based defects) as presented in Chapter 4. We are not using complete mappings, instead of this we use only some correspondences of mappings that cause defects in DDL and then check whether Galois connection can exist with these mappings between lattices of ontologies.

Below, we are using the ontologies described in **Example 4-27** of chapter 4, however, mappings used in this example are in one direction, so here we describe mappings explictly as bi-directional.

	$M_{ij}: O_i \mapsto O_j$
m_1	O_i: Admin Staff $\mapsto$ O_j:Director
m_2	O_i:Director Admin $\mapsto$ O_j:Administrative Staff
m_3	O_i:Computer Scientist$\mapsto$ O_j:Researcher
m_4	O_i:Computer Scientist$\mapsto$ O_j:Teaching Faculty
m_5	O_i: Computer Scientist $\mapsto$ O_j:Researcher, Teaching Faculty
m_6	O_i:Social Scientist$\mapsto$ O_j:Researcher
m_7	O_i:Research organiztion$\mapsto$ O_j:University
m_8	O_i: Computer Scientist, Social Scientist $\mapsto$ O_j:Researcher

	$M_{ji}: O_j \mapsto O_i$
m'_1	O_j: Director $\mapsto$ O_i: Research officer
m'_2	O_j: Administrative Staff $\mapsto$ O_i: Director Admin
m'_3	O_j: Director $\mapsto$ O_i: Admin Staff
m'_4	O_j:Researcher, Teaching Faculty $\mapsto$ O_i: Computer Scientist
m'_5	O_j:Researcher $\mapsto$ O_i:Social Scientist
m'_6	O_j:Researcher $\mapsto$ O_i:Computer Scientist, Social Scientist
m_7	O_j:University $\mapsto$ O_i:Research organiztion

Incoherence.

Example 5-1: When correspondences are m_1, m'_1.

(m_1) O_i: Admin Staff $\mapsto$ O_j:Director

(m'_1) O_j: Director $\mapsto$ O_i: Research officer

Considering these mappings cause incoherence in DDL formalism.

Since mappings are $concept - concept$ and remain in domain and range of taxonomies of source ontologies, we are using lattices based on respective taxonomies of source ontologies to check the existence of Galois connection.

If mapping m_3 is in mapping α and m_6 is in γ while mapping of remaining unmapped artifacts are not used here. Where α and γ are mappings in reverse direction to each other; $\alpha: O_i \mapsto O_j$ and $\gamma: O_j: \mapsto O_i$.

$$\alpha(Admin\ Staff) = Director$$

$$\gamma(Director) = Research\ Officer$$

$$\gamma \circ \alpha(Admin\ Staff) = Research\ Officer \preccurlyeq\!\!\!\!/\ Admin\ Staff$$

i.e.,

$$Admin\ Staff\ \perp Research\ officer$$

Therefore, with these mappings Galois connection could not exist, thus they are incompatible.

Example 5-1 also shows that our compatible mapping procedure also detects the defect which *Subsumption–disjointness contradiction* pattern detects.

Abnormal mappings. (semantic inconsistencies)

Assume that there are no disjointness axioms, then still case of Incoherence presented above (incoherence) remains a defect in our case, since it falls to semantic inconsistency, i.e., concepts in hierarchy are mapped to artifacts that are not semantically consistent with each other. The reason is that these mappings do not form Galois connection.

Example 5-2: when correspondences are

(m_1) O_i: Computer Scientist $\mapsto$ O_j:Researcher

(m_2) O_i: Social Scientist Admin $\mapsto$ O_j: Researcher

(m'_5) O_j:Researcher $\mapsto$ O_i:Social Scientist

These mappings do not cause defect in DDL formalism.

If $m_1$and m_2 are part of α mapping and m'_5 is in γ mapping then

$\gamma \circ \alpha(Computer\ Scientist) = Social\ Scientist \preccurlyeq\!\!\!\!/\ Computer\ Scientist$

Hence, there cannot be a Galois connection with these mappings.

Example 5-2 also shows that our compatible mapping procedure also detects the defect which *Criss-Cross (Bow-tie)* pattern detects.

Violation of principle of conservativity.

Example 5-3: when correspondences are

(m_1) O_i: Admin Staff $\mapsto$ O_j:Director

(m_2) O_i:Director Admin $\mapsto$ O_j:Administrative Staff

(m_7) O_i:Research organiztion$\mapsto$ O_j:University

(m'_2) O_j: Administrative Staff $\mapsto$ O_i: Director Admin

(m'_3) O_j: Director $\mapsto$ O_i: Admin Staff

(m'_7) O_j:University $\mapsto$ O_i:Research organization

This is not an absolute defect and require some kind of computation of deductive difference to locate this defect in logical formalism.

Correspondences m_1, m_2 and m_7 are neither monotone nor antitone, so there does not exist any Galois connection with such mappings.

Moreover, m'_2, m'_3 and m'_7 are neither monotone nor antitone.

Example 5-3 also shows that our compatible mapping procedure also detects the defect which *Principle of Locality* detects.

5.6 Conclusions

In Chapter 5, we have described the basic definitions and properties related to Galois connection.

We have reported that given ontologies, it is always possible to build lattices from them and therefore it is always possible to establish mappings between these lattices that may or may not form Galois connection. Several lattices are possible and their usage depends on

- Ontology content, ranging from simple lightweight ontologies to full logical theories
- Ontology formalism, ranging from simple DL to FOL
- Ontology mappings, which can be explicit such as functions to hidden such as constraints
- Defects, which can be related to concepts, properties or both

We have found that Galois Connection is a natural choice for defining the notions of compatibility and incompatibility because it is independent of the kind of formalization (such as the kind of logics) used to represent ontologies, it introduce a kind of unified treatment of defects and it introduce a kind of unified syntax for mappings.

We have defined the notions of compatible and incompatible ontology mappings. Under the constrained definition, universality of incompatibility remains an objective because it requires proving that for each procedure P satisfying the given constraints, resulting connections between lattices are not Galois connection. We prove that if this is the case then no procedure P under given constraints can establish Galois connection between lattices L_i and L_j. However, the definition provides a universal formulation of incompatibility, which is one of the thesis objectives.

We have provided a sketch that our defined notions of compatibility and incompatibility can detect absolute and relative defects and in Chapter 6, we will formally prove it.

In Chapter 6, we will show that starting from lattices of theory and having Galois Connection between lattice of theory, it is possible in some cases, to recover ontology mappings (as defined in Chapter 2 and Chapter 3). Additionally, we will show that the recovered ontology mappings do not suffer of some defects identified in Chapter 4. At the same time, starting from mappings which do not suffer from some defects, it is possible to use lattices of theory and build mappings which are Galois Connection.

Chapter 6.

Relating Compatible Ontology Mappings to Correct Ontology Mappings

Galois connection based compatibility mapping definition provides a way of checking whether two ontology mappings are conflicting with each other. This definition provides independence from the languages in which ontologies and ontology mappings are formalized and from logics that is used to reason about ontology mappings. It is interesting to show that this definition relates and complement to existing approaches of ontology mappings. Approaches to define mappings can be naturally divided into two categories: mappings that are defined while adhering to some constraints and mappings that are defined in any manner. Enderton approach of defining mappings, theory interpretation, is an example of former approach. Enderton proposes to start with defining mappings of signature of language and then define mappings of formulas of the language. Using these mappings of formulas, interpretation of one theory into another theory can be judged. Mappings that are additional constraints on source ontologies are examples of later approach; this is the case in mappings defined in DDL and $\epsilon -$ connections. Correct ontology mappings refer, here, to ontology mapping that is free from any defect.

In this chapter, we relate our Galois connection based definition of compatibility with theory interpretation and correct ontology mappings. Theory interpretation provides a well-established way of comparing theories even when signatures of these theories are different. Correct ontology mappings (free from any defect) become a basic requirement for applications using these mappings particularly where consistency is a requirement. The results of this chapter are the main properties of compatible mappings and they highlight the relevance and importance of compatible mappings.

This chapter is organized as follows: Section 6.1 relates Galois connection based compatible mappings with theory interpretation by proving some theorems, while emphasis is on covering the first-order Logic (FOL) theories; Section 6.2 relates Galois connection based compatible mappings with correct ontology mappings in the context of conservative $f(O_i, O_j, M_{ij}, M_{ji})$, by presenting formal proofs; Section 0 describes the important properties of compatible and incompatible ontology mappings. Finally, Section 6.4 and 6.5 discuss how notion of compatibility and incompatibility ontology mappings relate with mapping defects and mapping acceptability, respectively.

6.1 Interpretation between theories and existence of Galois Connection

In this section, we present some proofs to relate Galois connection and interpretation between theories.

Here, a theory is treated as a set of all ground inferences entailed by its axioms through basic inference rules such as deduction, modus-ponens etc.

6.1.1 Relating Theory Interpretation with Existence of Galois Connection of theories

Theorem 6-1: *If a theory T_0 is an interpretation of another theory T_1 (according to Enderton interpretation of theories) such that α mapping $\alpha: \langle \wp(T_0), \subseteq \rangle \rightarrow \langle \wp(T_1), \subseteq \rangle$ defined as*

$$\alpha(s) = \{Syt^m(\sigma \mid \sigma \in s)\} \in \wp(T_1)$$

then there exists γ mapping $\gamma: \langle \wp(T_1), \subseteq \rangle \rightarrow \langle \wp(T_0), \subseteq \rangle$ such that $\langle \alpha, \gamma \rangle$ forms a Galois connection.

Proof :

Given theory T_0 is interpreted into theory T_1, Formally as

$$\forall \sigma \in T_0 \Rightarrow Syt^m(\sigma) \in T_1$$

Now, we prove that α mapping defined as $\alpha(s) = \{Syt^m(\sigma \mid \sigma \in s)\} \in \wp(T_1)$ is monotone and join-preserving.

Let $s \subset s' \Rightarrow \exists e \in s' \wedge e \notin s$

then $\alpha(s') = \alpha(s) \cup \alpha(\{e\})$, hence α is monotone.

Let $\alpha(s') = \alpha(s \vee \{e\})$

then $= \alpha(s) \cup \alpha(\{e\})$, hence α is join-preserving.

We define γ mapping as $\gamma(t) = \cup \{s \in \wp(T_0) | \alpha(s) \subseteq_{T_1} t\}$. Now, we have to show that α and γ mappings respect the condition $\alpha(s) \subseteq_{T_1} t$ *iff* $s \subseteq_{T_0} \gamma(t)$ forms Galois connection.

Since we have already defined γ mapping in terms of α, so now we have to just prove that $\alpha(s) \subseteq_{T_1} t$.

When $\alpha(s) \subseteq_{T_1} t$

$\Rightarrow s \in \{u | \alpha(u) \subseteq_{T_1} t\}$

$\Rightarrow s \subseteq_{T_0} lub\{u | \alpha(u) \subseteq_{T_1} t\}$ by existence of lub

$\Rightarrow s \subseteq_{T_0} \gamma(t)$ by def. of γ

$\Rightarrow \alpha(s) \subseteq_{T_1} \alpha(lub\{u | \alpha(u) \subseteq_{T_1} t\})$ by monotonicity of α and γ

$\Rightarrow \alpha(s) \subseteq_{T_1} lub\{\alpha(u) | \alpha(u) \subseteq_{T_1} t\}$ α preserves existing lubs

$\Rightarrow \alpha(s) \subseteq_{T_1} t$ By def. of lub

Q.E.D.

Theorem 6-1 satisfies the condition of compatibility by following its procedure since Mapping in Enderton is a function. $m: O_i \rightarrow O_j = \{\langle \sigma, syt^m(\sigma)\rangle | \sigma \in I(O_i) \wedge syt^m(\sigma) \in I(O_j)\}$.

$$P(m_{ij}) \circ \Pi^i(\{\langle x, Syt^m(x)\rangle | x \in O_i, Syt^m(x) \in O_j\} = P(m_{ij}(x)) = Syt^m(x) = y$$

Hence, respect the condition of procedure of compatible mapping described in Chapter 5, as

$$P(m)\Pi(\langle x, y\rangle) = y \geq R(C^*(y))$$

Theorem 6-2: If $\langle \alpha, \gamma \rangle$ is a Galois connection $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$, such that $\exists a \neq \emptyset\ \alpha(a) = \emptyset$ then $\forall b \neq \emptyset\ \gamma(b) \neq \emptyset$.

Proof :

Proving by contradiction. Suppose that we have a Galois connection. Given $\exists a \neq \emptyset$ and $\alpha(a) = \emptyset$ and we assume that $\exists b \neq \emptyset\ \gamma(b) = \emptyset$.

Galois connection condition is $\alpha(a) \subseteq_{T_1} b\ iff\ a \subseteq_{T_0} \gamma(b)$

When $\alpha(a) = \emptyset$ and $\gamma(b) = \emptyset$ and $b \neq \emptyset$, putting these values in Galois connection condition we have

$$\emptyset \subseteq_{T_1} b\ iff\ a \subseteq_{T_0} \emptyset$$

i.e., $\emptyset \subseteq_{T_1} b$ and $a \subseteq_{T_0} \emptyset$, while we have $a \neq \emptyset$, a contradiction.

Q.E.D.

6.1.2 Relating Existence of Galois Connection of theories with Theory Interpretation

Here, we prove some theorems to show relationships between existence of Galois connection of theories and theory interpretation.

6.1.2.1 Theories using Quantifiers and only Conjunction operator

Theories only used Quantifiers and Conjunction operator are simple but less expressive theories.

Theorem 6-3: *If there exists a Galois connection between lattices of power set of theories T_0 and T_1 as $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$ where α is a mapping s.t. $\forall s\ \alpha(s) \neq \emptyset$ (by Theorem 6-2) and $\forall s | s\ is\ singleton\ \alpha(s)$ is a finite set, then one of the theory let's say T_0 that is not using disjunction $\vee$ and negation $\neg$ can be interpreted (according to Enderton's approach of interpretation between theories) into another theory T_1.*

Proof:

In this theorem, we are treating logical formulas as FOL formulas.

Our objective is to show that $\forall \sigma \in T_0 \Rightarrow Syt^m(\sigma) \in T_1$, where $Syst^m$ is the mapping recovered from α or γ mappings of a Galois connection between lattices $\wp(T_0)$ and $\wp(T_1)$,

Given $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$, however, we don't have explicit information about how signature of theory is mapped. Therefore, we have to specify this mapping explicitly.

As our basic assumption in this thesis is that theories (ontologies) are consistent, so there exists a non-empty set in the universe of T_1 that is to be used as the universe of structure of T_0 and it defines the mapping for parameter $\forall$ of L_0 (language of theory T_0).

For all n-place predicate $P(v_1, v_2, .., v_n)$ in the signature of L_0that are part of sentences σ of theory T_0, there exists a formula $\sigma(v_1, v_2, .., v_n)$ in the language of T_1 in which at most $v_1, v_2, .., v_n$ occurs free.

We made a logical formula from the element (which may not be atomic) of our power set lattice, by using conjunction.

Since negation $\neg$ and disjunction $\vee$ is not part of the theory, so we have to map atomic sentences and conjunction of atomic sentences.

Using α or γ mapping of the Galois connection $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$, mappings for signatures of language L_O of theory T_0 are defined as below.

For atomic sentences, $\boldsymbol{\sigma} \in \boldsymbol{L_0}$ and $\boldsymbol{\sigma} \in \boldsymbol{T_0}$

Since $\sigma \in T_0$, therefore $\sigma \in \langle \wp(T_0), \subseteq \rangle$

$$m(\sigma) = \wedge \alpha(\sigma) \in \langle \wp(T_1), \subseteq \rangle, \qquad \alpha(\sigma) \neq \emptyset$$

$\alpha(\sigma)$ may be a collection of elements while we need a final formula, we have used conjunction operator which is a natural choice to make $\alpha(\sigma \in \langle \wp(T_0), \subseteq \rangle)$ a logical formula.

For the case of $\alpha(\sigma) = \emptyset$, then according to Theorem 6-2 if one of the mapping involved maps an element of a theory T_0 other than $\emptyset$ to $\emptyset$ of another theory T_1, then other mapping does not map any element of theory T_1 other than $\emptyset$ to $\emptyset$ of theory T_0. For recovering mapping of symbols from mappings that form Galois connection, we would use the mapping that does not map an element other than $\emptyset$ of a lattice to $\emptyset$ of another lattice. In this theorem, we are assuming that this mapping is α mapping, so this case does not occur.

For sentences, $\boldsymbol{\sigma} \in \boldsymbol{L_0}$ and $\boldsymbol{\sigma} \notin \boldsymbol{T_0}$

Since $Sign(T_0) \subseteq Sign(L_0)$, there may exist some signatures of the language L_0 that are not part of the theory T_0.

Mappings for these signatures are defined as

$$m(\sigma) = p \in L_1$$

This mapping ensures that signature of language L_0 are mapped in the signature of language L_1 of theory T_1. Mapping of those signatures that are not part of the theory would not affect the interpretation between theories.

With the above mapping of signature of language, now we have to prove that

$$\forall\sigma\, T_0 \vDash \sigma \Rightarrow T_1 \vDash syt^m(\sigma)$$

We use syt^m for referring the mapping of non-atomic formulas of theory.

Mapping of non-atomic formulas

To complete mapping of all sentences of the theory, we use structural induction. Since we permit quantifiers and conjunction in theory, so mapping of

Non-atomic sentences involving conjunction $\wedge$ is

When $\sigma = \phi \wedge \psi$,

As we define mapping for each predicate in $\langle \wp(T_1), \subseteq \rangle$, i.e., they are ture in T_1, so mapping syt^m for σ is

$$syt^m(\sigma) = syt^m(\phi \wedge \psi) = syt^m(\phi) \wedge syt^m(\psi)$$

Because $m(\phi) \in \langle \wp(T_1), \subseteq \rangle$ and $syt^m(\psi) \in \langle \wp(T_1), \subseteq \rangle$, therefore $syt^m(\phi) \wedge syt^m(\psi) \in \langle \wp(T_1), \subseteq \rangle$.

For quantification $(\exists, \forall)$, please note that

$$\forall x\big(P(x) \wedge Q(x)\big) \Leftrightarrow \forall x P(x) \wedge \forall x Q(x)$$

$$\exists x\big(P(x) \wedge Q(x)\big) \Rightarrow \exists x P(x) \wedge \exists x Q(x)$$

As we have mapping of atomic predicates which is inductively applied to sentences, dealing with quantifiers when applied to conjunction of predicates is problematic in case of existential quantifiers $\exists$, since $\exists x\big(P(x) \wedge Q(x)\big) \Rightarrow \exists x P(x) \wedge \exists x Q(x)$ while converse is not always true. However, in this special case, P and Q are mapped to closed formulas, so the existential quantifier has no effect, as shown below.

When $\sigma = \exists x\big(P(x) \wedge Q(x)\big)$

$$syt^m(\sigma) = syt^m\left(\exists x\big(P(x) \wedge Q(x)\big)\right)$$

$$= \exists x\; syt^m\big(P(x) \wedge\; Q(x)\big)$$

$$= \exists x(m(P(x)) \wedge m(Q(x))$$

$$= m\big(P(x)\big) \wedge m\big(Q(x)\big)$$

$m\big(P(x)\big)$ and $m\big(Q(x)\big)$ are closed formulas (i.e., with no free variable) by definition.

Case of universal quantifier $\forall$ is already implicitly covered above, since universal quantifier is implicitly considered in $\sigma = \phi \wedge \psi$.

Using above mappings, T_1 is an interpretation of T_0 as for every sentence of T_0 there is a formula that is true in T_1.

Q.E.D.

A less expressive Description logic $EL^{\perp}$ is a language that allows formula with valid syntax of unlimited use of existential quantifiers and concept intersection and top $\top$ and bottom $\perp$. $EL^{\perp}$ is widely used in expressing ontologies. $EL^{\perp}$ is an example of the theory which does not allow disjunction and negation as desired in the above theorem.

Galois connection are bi-directional, it is interesting to know if mutual interpretation of theories (ontologies) is possible. Two theories T_0 and T_1 are *mutually interpretable* if they interpret each other. Following corollary states the condition on Galois connection that hold mutual interpretation of theories expressed by using quantifiers and conjunction operator.

Corollary 6-3: When theories T_0 and T_1 are expressed only by using quantifiers and conjunction operator and there exists a Galois connection between power set lattices of theories $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$ where $\alpha(s \neq \emptyset) \neq \emptyset$ and $\gamma(s \neq \emptyset) \neq \emptyset$ then both theories mutually interpret each other.

Proof:

In Theorem 6-3, we are using Theorem 6-2 to select one of the Galois connection mapping which is either α s.t. $\alpha(s \neq \emptyset) \neq \emptyset$ or γ s.t. $\gamma(s \neq \emptyset) \neq \emptyset$ to recover mapping of symbols of theory. Here, we have $\alpha(s \neq \emptyset) \neq \emptyset$ and by Theorem 6-3, we have theory T_0 is interpreted into theory T_1,

To prove that theory T_1 is interpreted into theory T_0, by definition of this theorem we also have $\gamma(s \neq \emptyset) = \emptyset$, and by Theorem 6-3 using γ mapping we have theory T_0 is interpreted into theory T_1,

Q.E.D.

Following theorem is a refinement of Corollary 6-3 and it provides less explicit condition for mutual interpretation.

Theorem 6-4: *Under conditions of Theorem 6-3 if there exists a monotone Galois connection* $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$ *such that* α *mapping is surjective and* $\nexists x \in \langle \wp(T_0), \subseteq \rangle \wedge x \neq \emptyset\ s.t.\ \alpha(x) = \emptyset$ *then theories* T_0 *and* T_1 *are mutually interpretable.*

Proof:

Given α is surjective $(\forall y \in \langle \wp(T_1), \subseteq \rangle : \exists x \in \langle \wp(T_0), \subseteq \rangle; \alpha(x) = y)$

Proof by contradiction.

Assume $\exists y \in \langle \wp(T_1), \subseteq \rangle \neq \emptyset; \gamma(y) = \emptyset$

So

$\alpha \circ \gamma(y) = \alpha(\emptyset)$ since $\gamma(y) = \emptyset$

$\Rightarrow \alpha(\emptyset) = \emptyset$ since when $\alpha(\emptyset) \neq \emptyset$ then there will be no Galois connection

$\Rightarrow \alpha \circ \gamma(y) = \emptyset \subseteq_{T_1} y$ since $y \supseteq \emptyset$

While

$$\exists x \in \langle \wp(T_0), \subseteq \rangle \exists y \in \langle \wp(T_1), \subseteq \rangle\ \alpha(x) = y \wedge \gamma(y) = \emptyset;\ \gamma \circ \alpha(x) = \gamma(y)$$

$\gamma \circ \alpha(x) = \gamma(y) = \emptyset$ since $\gamma(y) = \emptyset$

$\Rightarrow \gamma \circ \alpha(x) = \emptyset \subseteq_{T_0} x$

Both $\alpha \circ \gamma$ and $\gamma \circ \alpha$ is reductive for y and x, respectively, a contradiction while having monotone Galois connection.

Proving that $x \neq \emptyset$.

Proof by contradiction

Let $x = \emptyset$, In this case $\alpha(x) = \alpha(\emptyset) = y \neq \emptyset$

α is surjective, therefore $\exists z | \alpha(z) = \emptyset$

Since Galois connection is monotone, $z \supseteq \emptyset, \alpha(x = \emptyset) = y$ and $y \supseteq \emptyset$

$\alpha(z) \subseteq_{T_1} \alpha(x = \emptyset)$ α is anti-tone in this case, a contradiction.

Therefore, $x \neq \emptyset$.

But for $x \neq \emptyset$, we have $\gamma \circ \alpha(x) = \emptyset \subseteq_{T_0} x$, a contradiction.

By Theorem 6-3, and given conditions $\nexists x \in \langle \wp(T_0), \subseteq \rangle \wedge x \neq \emptyset; \alpha(x) = \emptyset$ *and* α *is surjective ensures that theory* T_0 *is interpretable in theory* T_1.

We can apply Theorem 6-3 in the reverse sense to prove that *theory* T_1 *is interpretable in theory* T_0.

Q.E.D.

6.1.2.2 Theories using Quantifiers and only Conjunction and atomic negation operators

Now, it is important to examine the case of negation and disjunction. We consider only atomic negation because in FOL negation can be applied to relevant predicate symbols, i.e., negation can be only applied to atomic formulas; for instance, using Negation Normal Form (NNF) in first-order logic.

If there exists a Galois connection between lattices of power set of theories T_0 and T_1 as $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$ where α is a mapping s.t. $\forall s\, \alpha(s) \neq \emptyset$ (by Theorem 6-2) and $\forall s | s\ is\ singleton\ \alpha(s)$ is a finite set, then one of the theory let's say T_0 that is using atomic negation $\neg$ can be interpreted (according to Enderton's approach of interpretation between theories) into another theory T_1 in most of the cases.

Extension (1) of Theorem 6-3: (when atomic negation is permitted)

If there exists a Galois connection between lattices of power set of theories T_0 and T_1 as $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$ where α is a mapping s.t. $\forall s\, \alpha(s) \neq \emptyset$ (by Theorem 6-2) and $\forall s | s\ is\ singleton\ \alpha(s)$ is a finite set, then one of the theory let's say T_0 that is using atomic negation $\neg$ and conjunction operators can be interpreted (according to Enderton's approach of interpretation between theories) into another theory T_1 in most of the cases.

Proof:

When a theory using only conjunctions and atomic negations as operators and quantifiers are applied to the inner most predicates, the following five cases can be identified.

(1) $\forall x \neg A(x)$ and $\exists x \neg A(x)$

(2) $\exists x \neg A(x)$ and $\exists x\, A(x)$

(3) $\forall x \neg A(x)$ and $\forall x\, A(x)$

(4) $\forall x\, A(x)$ and $\exists x\, \neg A(x)$

(5) $\forall x\, \neg A(x)$ and $\exists x\, A(x)$

Cases (3), (4) and (5) leads to inconsistency and the theory containing at least one of these inferences becomes an inconsistent theory. We assume that theories are consistent that's why we are not considering them.

In case (1), defining mapping only on the basis of universal quantifier $syt^m(\forall x \neg A(x)$ might not satisfy $syt^m(\exists x \neg A(x)$ in theory T_1, while defining mapping only on the basis of existential quantifier $syt^m(\exists x \neg A(x)$ might not satisfy $syt^m(\forall x \neg A(x)$ in theory T_1 (the reason is that there is no implication between $\forall x \neg A(x)$ and $\exists x \neg A(x)$). Therefore, we have to define specific mapping for this symbol $A(x)$ such that syntactical translation of both formulas $\forall x \neg A(x)$ and $\exists x \neg A(x)$ in is theory T_1. We define mapping of symbol $A(x)$ based on $syt^m(\neg A(x))$.

Syntactical translation of $\neg A(x)$ is defined, here, in the following two ways.

a)

$$syt^m(\neg A(x)) = (\wedge\, \alpha(\forall x \neg A(x))) \vee (\wedge\, \alpha(\exists x \neg A(x)))\} \in \langle \wp(T_1), \subseteq \rangle$$

b)

$$syt^m(\neg A(x)) = \left(\wedge\, \alpha(\forall x \neg A(x))\right) \wedge (\wedge\, \alpha(\exists x \neg A(x)))\} \in \langle \wp(T_1), \subseteq \rangle$$

$$m(A(x)) = \neg syt^m(\neg A(x))$$

For case (a), we verify that

if $\forall x \neg A(x) \in T_0$ then $syt^m(\forall x \neg A(x)) = \forall x \neg m(A(x)) = \forall x(\wedge\, \alpha(\forall x \neg A(x))) \vee (\wedge\, \alpha(\exists x \neg A(x)))\} \in \langle \wp(T_1), \subseteq \rangle$;

if $\exists x \neg A(x) \in T_0$ then $syt^m(\exists x \neg A(x)) = \exists x \neg m(A(x)) = \exists x(\wedge\, \alpha(\forall x \neg A(x))) \vee (\wedge\, \alpha(\exists x \neg A(x)))\} \in \langle \wp(T_1), \subseteq \rangle$; and

mapping of $\forall x \neg A(x)$ and $\exists x \neg A(x)$ as $syt^m(\forall x \neg A(x)) \wedge syt^m(\exists x \neg A(x)) \in \langle \wp(T_1), \subseteq \rangle$

For case (b), we verify that

if $\forall x \neg A(x) \in T_0$ then $syt^m(\forall x \neg A(x)) = \forall x \neg m(A(x)) = \forall x \left(\wedge\, \alpha(\forall x \neg A(x))\right) \wedge (\wedge\, \alpha(\exists x \neg A(x)))\} \in \langle \wp(T_1), \subseteq \rangle$;

if $\exists x \neg A(x) \in T_0$ then $syt^m(\exists x \neg A(x)) = \exists x \neg m(A(x)) = \exists x \left(\wedge\, \alpha(\forall x \neg A(x))\right) \wedge (\wedge\, \alpha(\exists x \neg A(x)))\} \in \langle \wp(T_1), \subseteq \rangle$; and

mapping of $\forall x \neg A(x)$ and $\exists x \neg A(x)$ as $syt^m(\forall x \neg A(x)) \wedge syt^m(\exists x \neg A(x)) \in \langle \wp(T_1), \subseteq \rangle$

In case (2), $A(x)$ is neither true nor false. If we define mapping for $A(x)$ a ground inference of theory T_1 then $\exists x \neg A(x)$ cannot be satisfiable in the theory and the same for $\neg A(x)$. One technique is to introduce free variable(s) (that may be later replaced by skolem functions).

By using as usual the α mappings, $\exists x A(x)$ is mapped to $\left(\wedge\, \alpha(\exists x A(x))\right)$ and $\exists x \neg A(x)$ is mapped to $\left(\wedge\, \alpha(\exists x \neg A(x))\right)$, if it is possible, we remove quantifiers from $\left(\wedge\, \alpha(\exists x A(x))\right)$ and

$\left(\wedge\, \alpha\left(\exists x \neg A(x)\right)\right)$ such that one of the variable becomes free. Here, formula $\left(\wedge\, \alpha\left(\exists x A(x)\right)\right)$ with removed quantifier and free variable is denoted by $\phi(x)$ and formula $\left(\wedge\, \alpha\left(\neg \exists x A(x)\right)\right)$ with at least one free variable is denoted by $\psi(x)$.

Mapping of $A(x)$ is defined as

$m(A(x)) = \phi(x) \wedge \neg\psi(x)$, under the hypothesis that $T_1 \vDash \exists x\left(\phi(x) \wedge \neg\psi(x)\right)$

Case of $\forall x A(x) \in T_0$ and $\forall x \neg A(x)$ causes inconsistency so it will not occur in theory T_0 as we assume theories are consistent.

While syntactical translation of $\neg A(x)$ is

$m(\neg A(x)) = \neg(\phi(x) \wedge \psi(x))$, where $T_1 \vDash \exists x(\neg\phi(x) \vee \psi(x))$.

So, syntactical translation of inference $\exists x A(x)$ of theory T_0 is

$$Syt^m\left(\exists x\, A(x)\right) = \exists x Syt^m\left(A(x)\right) = \exists x\left(\phi(x) \wedge \neg\psi(x)\right) \in \langle \wp(T_1), \subseteq \rangle$$

and syntactical translation of inference $\exists x \neg A(x)$ of theory T_0 is

$$Syt^m\left(\exists x\, \neg A(x)\right) = \exists x Syt^m\left(\neg A(x)\right) = \exists x\left(\neg\phi(x) \wedge \psi(x)\right) \in \langle \wp(T_1), \subseteq \rangle_1$$

Alternative mapping of $A(x)$ defined as

$m(A(x)) = \neg\phi(x) \wedge \psi(x)$, under the hypothesis that $T_1 \vDash \exists x\left(\neg\phi(x) \wedge \psi(x)\right)$ also works in case (2).

Putting an extra condition in case (2), as described above, theory T_0 using only atomic negation and conjunction operators along with quantifiers is interpretable in another theory T_1 when Galois connection $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$ exists.

However, in the case of generalizing things with a negation and disjunction, the existence of Galois Connection does not guarantee mutual interpretation.

> Note that if $T_1 \nvDash \exists x\left(\phi(x) \wedge \neg\psi(x)\right)$ then it means that mapping is not properly defined because two completely opposite things in theory T_0 are mapped to common thing in theory T_1 in an equivalence relation, i.e., $A(x) = \exists x\left(\phi(x) \wedge \neg\psi(x)\right)$ and $\neg A(x) = \exists x\left(\phi(x) \wedge \neg\psi(x)\right)$.
>
> With reference to Galois connection, if $T_1 \nvDash \exists x\left(\phi(x) \wedge \neg\psi(x)\right)$ then it means that existence of Galois connection does not provide sufficient condition for interpretability of theories.

6.1.2.3 Theories using Disjunction operator

Now, we prove that introducing disjunction in theories will result in losing good properties of Galois Connection since existence of Galois connection $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$ will not guarantee that theories are mutually interpretable in this case.

Theorem 6-5: *If there exists a Galois connection between lattices of power set of theories* $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$, *then it is not necessary that theories* T_0 *and* T_1 *expressed by using more expressive language that allows disjunction and complex negation are mutually interpretable.*

Proof:

Let suppose that T_0 describes a situation with two distinct things while T_1 decribes a situation where only one thing is available. In the remainder, we will show that T0 cannot be interpreted in T1 in any case, even if a Galois Connection can be easily defined between T_0 and T_1.

For instance, consider theory T_0 entails a set of inferences $\{\exists x Ax, \exists x Bx, \neg(\exists x A(x) \wedge \exists x B(x)\}$ and theory T_1 entails a set of inferences $\{\exists x C(x)\}$

In Figure 6-1, α and γ mappings form Galois connection, where $\alpha: \langle \wp(T_0, \subseteq \rangle \rightarrow \langle \wp(T_1), \subseteq \rangle$ and $\gamma: \langle \wp(T_1, \subseteq \rangle \rightarrow \langle \wp(T_0), \subseteq \rangle$, where $\alpha(s|s \neq \emptyset) = \exists x C(x)$ and $\alpha(s|s = \emptyset) = \emptyset$, where $s \in \wp(T_0)$ and $\gamma(s'|s' \neq \emptyset) = \{\exists x C(x)\}$, $s' \in \wp(T_1)$ and $\gamma(s'|s' = \emptyset) = \emptyset$, where $s' \in \wp(T_1)$.

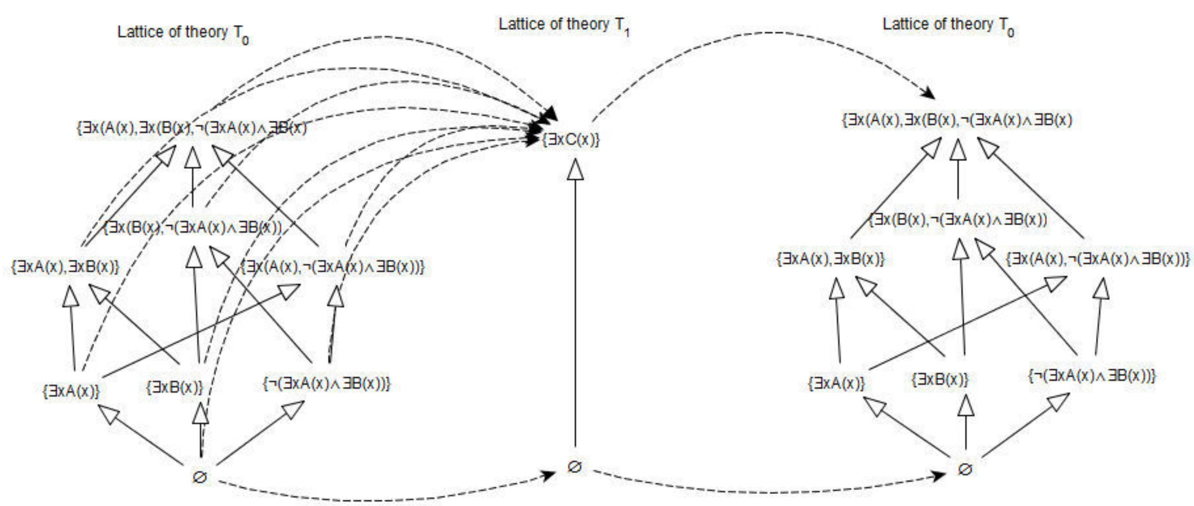

Figure 6-1. Existence of Galois connection between power set of theories does not mean that theories are mutually interpretable

However, theory T_0 is not interpreted in theory T_1 since syntactical translation of inferences of theory T_0 should be $\{\exists x C(x)\}$ in theory T_1 to fulfill the condition of theory interpretation $T_0 \vDash \sigma \Rightarrow T_1 \vDash syt^m(\sigma)$.

Now, we prove that syntactical translation of inferences of theory T_0 as $\{\exists x C(x)\}$ in theory T_1 does not fulfill the condition of theory interpretation for every possible mappings.

In this example, there are not so many possibilities of mapping of signature of language of T_0 into language of T_1 by just assuming that all the signature of respective language of each theory are used in given inferences. We list the possible mappings in Table 6-1.

Table 6-1 Possible mappings for signature of theory T_0 and their effects

	Possible mapping for $\boldsymbol{A(x)}$	**Possible mapping for $\boldsymbol{B(x)}$**	**Applying mapping to one of the inference of theory $\boldsymbol{T_0}$**
1	$\forall x C(x)$	$\forall x C(x)$	$Syt^m(\neg(\exists x A(x) \wedge \exists x B(x)) = \neg(\exists x C(x) \wedge \exists x C(x)) = \neg\exists x C(x) \notin T_1$
2	$\forall x C(x)$	$\forall x \neg C(x)$	$Syt^m(\neg(\exists x B(x)) = \neg\exists x C(x) \notin T_1$
3	$\forall x \neg C(x)$	$\forall x C(x)$	$Syt^m(\neg(\exists x A(x)) = \neg\exists x C(x) \notin T_1$
4	$\forall x \neg C(x)$	$\forall x \neg C(x)$	$Syt^m(\neg(\exists x A(x)) = \neg\exists x C(x) \notin T_1$
5	$\exists x C(x)$	$\exists x C(x)$	$Syt^m(\neg(\exists x A(x) \wedge \exists x B(x)) = \neg(\exists x C(x) \wedge \exists x C(x)) = \neg\exists x C(x) \notin T_1$
6	$\exists x C(x)$	$\exists x \neg C(x)$	$Syt^m(\neg(\exists x B(x)) = \neg\exists x C(x) \notin T_1$
7	$\exists x \neg C(x)$	$\exists x C(x)$	$Syt^m(\neg(\exists x A(x)) = \neg\exists x C(x) \notin T_1$
8	$\exists x \neg C(x)$	$\exists x \neg C(x)$	$Syt^m(\neg(\exists x A(x)) = \neg\exists x C(x) \notin T_1$
9	$\forall x C(x)$	$\exists x C(x)$	$Syt^m(\neg(\exists x A(x) \wedge \exists x B(x)) = \neg(\exists x C(x) \wedge \exists x C(x)) = \neg\exists x C(x) \notin T_1$
10	$\exists x C(x)$	$\forall x C(x)$	$Syt^m(\neg(\exists x A(x) \wedge \exists x B(x)) = \neg(\exists x C(x) \wedge \exists x C(x)) = \neg\exists x C(x) \notin T_1$
11	$\forall x \neg C(x)$	$\exists x \neg C(x)$	$Syt^m(\neg(\exists x A(x)) = \neg\exists x C(x) \notin T_1$
12	$\exists x \neg C(x)$	$\forall x \neg C(x)$	$Syt^m(\neg(\exists x A(x)) = \neg\exists x C(x) \notin T_1$
13	$\exists x C(x)$	$\forall x \neg C(x)$	$Syt^m(\neg(\exists x B(x)) = \neg\exists x C(x) \notin T_1$
14	$\forall x \neg C(x)$	$\exists x C(x)$	$Syt^m(\neg(\exists x A(x)) = \neg\exists x C(x) \notin T_1$
15	$\exists x \neg C(x)$	$\forall x C(x)$	$Syt^m(\neg(\exists x A(x)) = \neg\exists x C(x) \notin T_1$

16	$\forall x C(x)$	$\exists x \neg C(x)$	$Syt^m(\neg(\exists x B(x)) = \neg\exists x C(x) \notin T_1$

Table 6-1 shows that there is no way for theory T_0 being interpretable in theory T_1.

Q.E.D

Despite Theorem 6-5 that seems to exclude disjunction, the importance of disjunction in representation of ontologies is undisputable because an important aspect about disjunction operator is that it is an important operator to express even a simple taxonomy in logic, so we cannot exclude it.

Extension (2) of Theorem 6-3: (when disjunction is permitted)

If there exists a Galois connection between lattices of power set of theories T_0 and T_1 as $\langle\wp(T_0),\subseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle\wp(T_1),\subseteq\rangle$ where α is a mapping s.t. $\forall s\ \alpha(s) \neq \emptyset$ (by Theorem 6-2) and $\forall s | s\ is\ singleton\ \alpha(s)$ is a finite set, then one of the theory let's say T_0 that is using disjunction ($\vee$) operator can be interpreted (according to Enderton's approach of interpretation between theories) into another theory T_1 in most of the cases.

Proof:

Let consider that $\exists xAx \vee \exists xBx$ is in T_0.

We have 5 cases that can be considered whenever disjunction is permitted and we are not in the trivial case in which both $\exists xAx$ and $\exists x\mathrm{B}x$ are true. This means that for instance

$\exists xAx$ is neither true nor false so you have to find a "new" mapping for A, which is not defined according to any of the mappings introduced in earlier parts of Theorem 6-3.

1) $\exists xAx \vee \exists xBx$ and $\exists xBx$ are in T_0,
2) $\exists xAx \vee \exists xBx$ and $\exists x\neg Bx$ are in T_0
3) $\exists xAx \vee \exists xBx$ and $\exists x\neg Ax \vee \exists xBx$ are in T_0, and $\exists xBx$ is in T_0;
4) $\exists xAx \vee \exists xBx$ and $\exists x\neg Ax \vee \exists xBx$ are in T_0, and $\exists x\neg Bx$ is in T_0;
5) $\exists xAx \vee \exists xBx$ is in T_0 but also $\exists xBx$ is neither true nor false;

In case of universal quantifier $\forall$, only case (1) and case (5) are relevant. While when we have universal quantifier $\forall$ for Ax as $\forall xAx$ and existential quantifier $\exists$ for Bx as $\exists xBx$, only case (1), (2) and (5) are applicable.

Note that, here we are giving the proof of this theorem with sentences of the theory using existential quantifier $\exists$ in sentences of theory. Proof of theory involving universal quantifier $\forall$ in disjunction formulas (relevant cases as mentioned above) can be done analogously.

Mapping defined in **Theorem 6-3** as $m(\sigma) = \alpha(\sigma \in \langle\wp(T_0),\subseteq\rangle)\} \in \langle\wp(T_1),\subseteq\rangle$ will not always work in case of disjunction as σ may be neither true nor false in the theory but used in a theory since mapping of such symbols are mapped to an element of language p and it is not necessary that $p \in T_0$.

Recovering mappings from Galois connection mappings between lattices of theories when theories involve disjunction is different from the situation when theories involve conjunction operator. The reason is that in the case of conjunction $Ax \wedge Bx$ means that both Ax and Bx are true in the theory, while in the case of disjunction $Ax \vee Bx$ there can be several possibilities either Ax and Bx are true or one of Ax and Bx can be true or none of Ax and Bx can be true.

We have identified the following mappings as candidate mappings that we use to show that at least one of these mappings always work in the given situation such that using this mapping given theory T_0 is interpretable in another theory T_1.

1.
$$m_1(Ax) = \bigwedge\big(a\ non - empty\ subset\ of\ \alpha(\exists xAx \vee \exists xBx)\big)$$
$$m_1(Bx) = \bigwedge\big(a\ non - empty\ subset\ of\ \alpha(\exists xAx \vee \exists xBx)\big)$$

2.
$$m_2(Ax) = \bigwedge\big(\alpha\ (\exists xAx \vee \exists xBx)\big)$$
$$m_2(Bx) = \neg\left(\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big)\right)$$

3.
$$m_3(Ax) = \neg\left(\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big)\right)$$
$$m_3(Bx) = \bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big)$$

4. When $T_0 \vDash (\exists x \neg A(x) \vee ..)$ and $T_0 \vDash (\exists x\, A(x) \vee ..)$ and $T_0 \nvDash \exists x \neg A(x)$ and $T_0 \nvDash \exists x\, A(x)$ then mapping is defined by removing quantifiers as below.

   If α mappings of Galois connection $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$, $\exists x \neg A(x)$ has mappings for $\left(\wedge\, \alpha\big((\exists xA(x) \vee ..)\big)\right)$ and $\left(\wedge\, \alpha\big((\exists x \neg A(x) \vee ..)\big)\right)$, where $\left(\wedge\, \alpha\big((\exists x \neg A(x) \vee ..)\big)\right) \neq \left(\wedge\, \alpha\big((\exists xA(x) \vee ..)\big)\right)$ and if it is possible, we remove quantifiers from $a\ non - empty\ subset\ of \big(\wedge\, \alpha(\exists xA(x) \vee ..)\big)$ and $a\ non - empty\ subset\ of \big(\wedge\, \alpha(\exists x \neg A(x) \vee ..)\big)$ such that one of the variable becomes free. Here, formula $a\ non - empty\ subset\ of \left(\wedge\, \alpha\big((\exists xA(x) \vee ..)\big)\right)$ with removed quantifier and free variable is denoted by $\phi^{\vee}(x)$ and formula $a\ non - empty\ subset\ of \left(\wedge\, \alpha\big((\exists x \neg A(x) \vee ..)\big)\right)$with at least one free variable is denoted by $\psi^{\vee}(x)$.

$$m_4(A(x)) = \phi^{\vee}(x) \wedge \neg\psi^{\vee}(x), \text{ where } T_1 \vDash \exists x\big(\phi^{\vee}(x) \wedge \neg\psi^{\vee}(x)\big)$$

$$m_4(\neg A(x)) = \neg\phi^{\vee}(x) \vee \psi^{\vee}(x), \text{ where } T_1 \vDash \exists x\big(\neg\phi^{\vee}(x) \vee \psi^{\vee}(x)\big)$$

$$\exists xAx \vee \exists x\chi x$$

5.

$$m_5(Ax) = p_A \in L_1, \text{ no matter whether } T_1 \vDash p_A \text{ or not}$$

$$m_5(Bx) = p_B \in L_1, \text{ no matter whether } T_1 \vDash p_B \text{ or not}$$

We use m_i for referring to a mapping m_i both for Ax and Bx and we use $m_i(Ax)$ exclusively to highlight the fact that $m_i(Bx)$ is not used.

Case 1 $\exists xAx \vee \exists xBx$ and $\exists xBx$ are in T_0,

Mapping	$\boldsymbol{Syt^m}()$	**Syntactical translation**
$m_1(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\bigwedge\big(a\ non-empty\ subset\ of\ \alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
$m_2(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
$m_3(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\neg\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
m_4		N/A
$m_5(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists xp_A \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$

Case 2 $\exists xAx \vee \exists xBx$ and $\exists x\neg Bx$ are in T_0

Mapping	$\boldsymbol{Syt^m}()$	**Syntactical translation**
$m_1(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\bigwedge\big(a\ non-empty\ subset\ of\ \alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\neg\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Bx)$	$\exists x\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$

$m_2(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\neg\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Bx)$	$\exists x\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$
$m_3(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\neg\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\neg\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Bx)$	$\exists x\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$
m_4		N/A
$m_5(Ax)$ where p_A is not in T_1	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists xp_A \vee \exists x\neg\bigwedge\big(\alpha(\exists x\neg Bx)\big) \notin \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Bx)$	$\exists x\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$

Case 3: $\exists xAx \vee \exists xBx$ and $\exists x\neg Ax \vee \exists xBx$ are in T_0, and $\exists xBx$ is in T_0;

Mapping	$\boldsymbol{Syt^m}()$	**Syntactical translation**
$m_1(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\bigwedge\big(a\ non-empty\ subset\ of\ \alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Ax \vee \exists xBx)$	$\exists x\neg\bigwedge\big(a\ non-empty\ subset\ of\ \alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
$m_2(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Ax \vee \exists xBx)$	$\exists x\neg\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
$m_3(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\neg\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Ax \vee \exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists xBx)$	$\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
$m_4(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\big(\phi^{\vee}(x) \wedge \neg\psi^{\vee}(x)\big) \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Ax \vee \exists xBx)$	$\exists x\big(\neg\phi^{\vee}(x) \vee \psi^{\vee}(x)\big) \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists xBx)$	$\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$

$m_5(Ax)$ where p_A is not in T_1	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists xp_A \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Ax \vee \exists xBx)$	$\exists x\neg p_A \vee \exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xBx)\big) \in \langle\wp(T_1, \subseteq\rangle$

Case 4: $\exists xAx \vee \exists xBx$ and $\exists x\neg Ax \vee \exists xBx$ are in T_0, and $\exists x\neg Bx$ is in T_0;

Mapping	$\boldsymbol{Syt^m}()$	**Syntactical translation**
$m_1(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\bigwedge\big(a\ non-empty\ subset\ of\ \alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\neg\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Ax \vee \exists xBx)$	$\exists x\neg\bigwedge\big(a\ non-empty\ subset\ of\ \alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\neg\bigwedge\big(\alpha(\exists x\neg Bx)\big) \notin \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Bx)$	$\exists x\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$
$m_2(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\neg\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Ax \vee \exists xBx)$	$\exists x\neg\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\neg\bigwedge\big(\alpha(\exists x\neg Bx)\big) \notin \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Bx)$	$\exists x\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$
$m_3(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\neg\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\neg\bigwedge\big(\alpha(\exists x\neg Bx)\big) \notin \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Ax \vee \exists xBx)$	$\exists x\bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x\neg\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Bx)$	$\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$
$m_4(Ax)$	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x\big(\phi^{\vee}(x) \wedge \neg\psi^{\vee}(x)\big) \vee \exists x\neg\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$
	$Syt^m(\exists x\neg Ax \vee \exists xBx)$	$\exists x\big(\neg\phi^{\vee}(x) \vee \psi^{\vee}(x)\big) \vee \exists x\neg\bigwedge\big(\alpha(\exists x\neg Bx)\big) \in \langle\wp(T_1, \subseteq\rangle$

	$Syt^m(\exists x \neg Bx)$	$\exists x \bigwedge\big(\alpha(\exists x \neg Bx)\big) \in \langle \wp(T_1, \subseteq \rangle$
$m_5(Ax)$ where p_A not in T_1	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists xp_A \vee \exists x \neg \bigwedge\big(\alpha(\exists x \neg Bx)\big) \notin \langle \wp(T_1, \subseteq \rangle$
	$Syt^m(\exists x \neg Ax \vee \exists xBx)$	$\exists x \neg p_A \vee \exists x \neg \bigwedge\big(\alpha(\exists x \neg Bx)\big) \in \langle \wp(T_1, \subseteq \rangle$
	$Syt^m(\exists x \neg Bx)$	$\exists x \bigwedge\big(\alpha(\exists x \neg Bx)\big) \in \langle \wp(T_1, \subseteq \rangle$

Case 5: $\exists xAx \vee \exists xBx$ is in T_0 but also $\exists xBx$ is neither true nor false;

Mapping	$\boldsymbol{Syt^m}()$	**Syntactical translation**
m_1	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x \bigwedge\big(a\ non-empty\ subset\ of\ \alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x \bigwedge\big(a\ non - empty\ subset\ of\ \alpha(\exists xAx \vee \exists xBx)\big) \in \langle \wp(T_1, \subseteq \rangle$
m_2	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x \bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x \neg \bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \in \langle \wp(T_1, \subseteq \rangle$
m_3	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists x \neg \bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \vee \exists x \bigwedge\big(\alpha(\exists xAx \vee \exists xBx)\big) \in \langle \wp(T_1, \subseteq \rangle$
m_4	$Syt^m(\exists xAx \vee \exists xBx)$	N/A
m_5 where p_A, p_B not in T_1	$Syt^m(\exists xAx \vee \exists xBx)$	$\exists xp_A \vee \exists xp_B \notin \langle \wp(T_1, \subseteq \rangle$

We have shown that in each case, at least one of the potential mappings $m_1 - m_5$ syntactically translate theory T_0 into theory T_1.

Q.E.D.

6.1.3 α and γ mappings for Galois connection defined independently on theory interpretations

If α and γ mappings are defined independently, based on ontology mappings such that with these mappings one theory is interpreted into another, then they do not necessarily form a Galois Connection. This means that there is no counterpart Galois connection of given ontology mapping. In other words, from one ontology mapping, we may build a Galois connection. However, it does not make sense to start from 2 ontology mappings and building a Galois connection from the 2 former ontology mappings. This is confirmed by Theorem 6-6.

α and γ mappings for Galois connection can be defined on the mappings that interpret one theory into another theory as

Definition 6-1 (α and γ mappings for Galois connection based on theory interpretation):

Let Δ be a mapping as $\Delta: T_0 \rightarrow T_1$ such that T_1 interprets T_0 and Π be a mapping as $\Pi: T_1 \rightarrow T_0$, such that T_0 interprets T_1. α and γ mappings are defined as

$$\alpha(s) = \{\Delta(\sigma) | \sigma \in s\}$$

and

$$\gamma(t) = \{\Pi(\tau) | \tau \in t\}$$

Theorem 6-6:

If theories T_0 and T_1 are mutually interpretable s.t. $\Delta: T_0 \rightarrow T_1$ and $\Pi: T_1 \rightarrow T_0$ *then it is not necessary that α and γ mappings based on* Definition 6-1 *form Galois connection* $\langle \wp(T_0), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(T_1), \subseteq \rangle$.

Proof:

We prove this theorem by counter example and it is depicted in Figure 6-2(a) and (b). Theories are mutually interpretable all the formulas of theory T_O is interpreted as $\exists zXz$ in theory T_1, while all the formulas of theory T_1 are interpreted as $\exists xAx$ in theory T_0.

$$\Delta: \{m(Ax) = \exists zXz, m(Bx) = p_1 \in L_1\}$$

$$\Pi: \{m(Xz) = \exists xAx, m(Yz) = p_0 \in L_0\}$$

$$Syt^m\left(\exists x\big(A(x) \vee B(x)\big)\right) = \exists z(Xz \vee p_1)$$

$$Syt^m\left(\exists x\big(X(z) \vee Y(z)\big)\right) = \exists x(A(x) \vee p_0)$$

When checking the existence of Galois connection based on α and γ mappings that are defined according to Definition 6-1 , then we do not have Galois connection.

Since we have $\alpha\left(\exists x\big(A(x) \vee B(x)\big)\right) = \exists zXz$ and $\gamma(\exists zXz) = \exists xAx$, so when checking condition of Galois connection we have

$\gamma \circ \alpha(\exists x\big(A(x) \vee B(x)\big)) \nleq \exists x\big(A(x) \vee B(x)\big)$.

As $\gamma \circ \alpha$ is not ordered, hence there does not exist Galois connection.

Q.E.D.

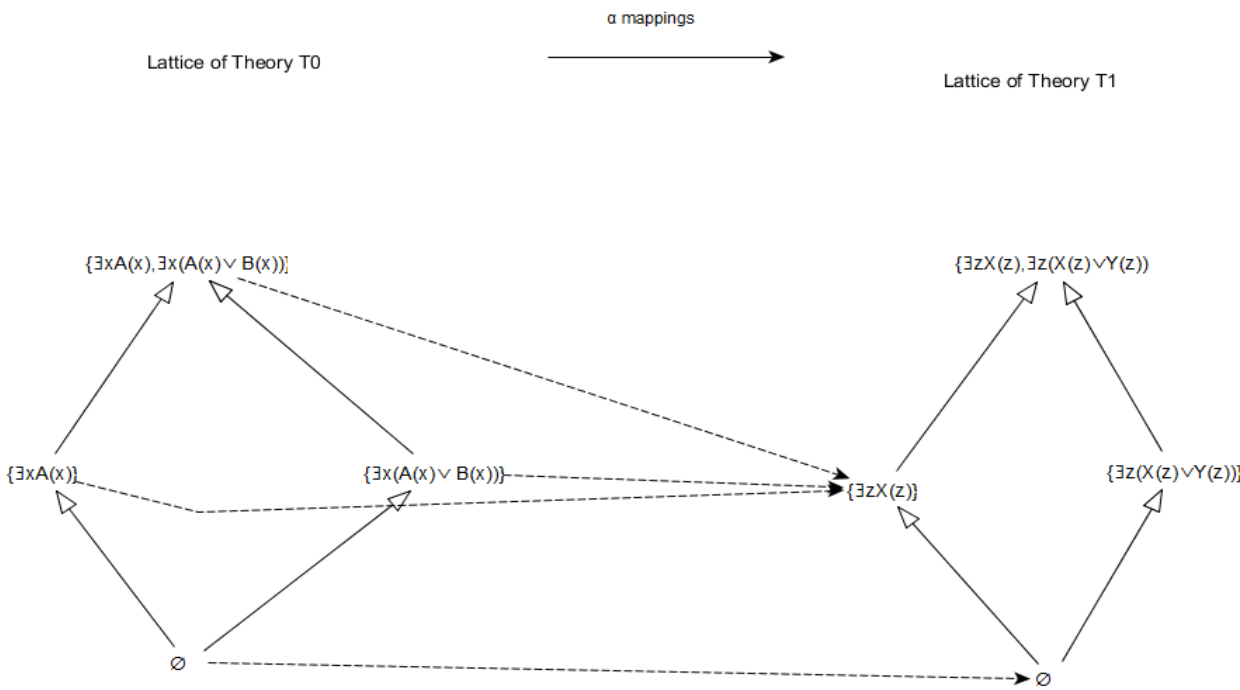

Figure 6-2 (a). α mappings between power set lattice of theory T_0 and power set lattice of theory T_1

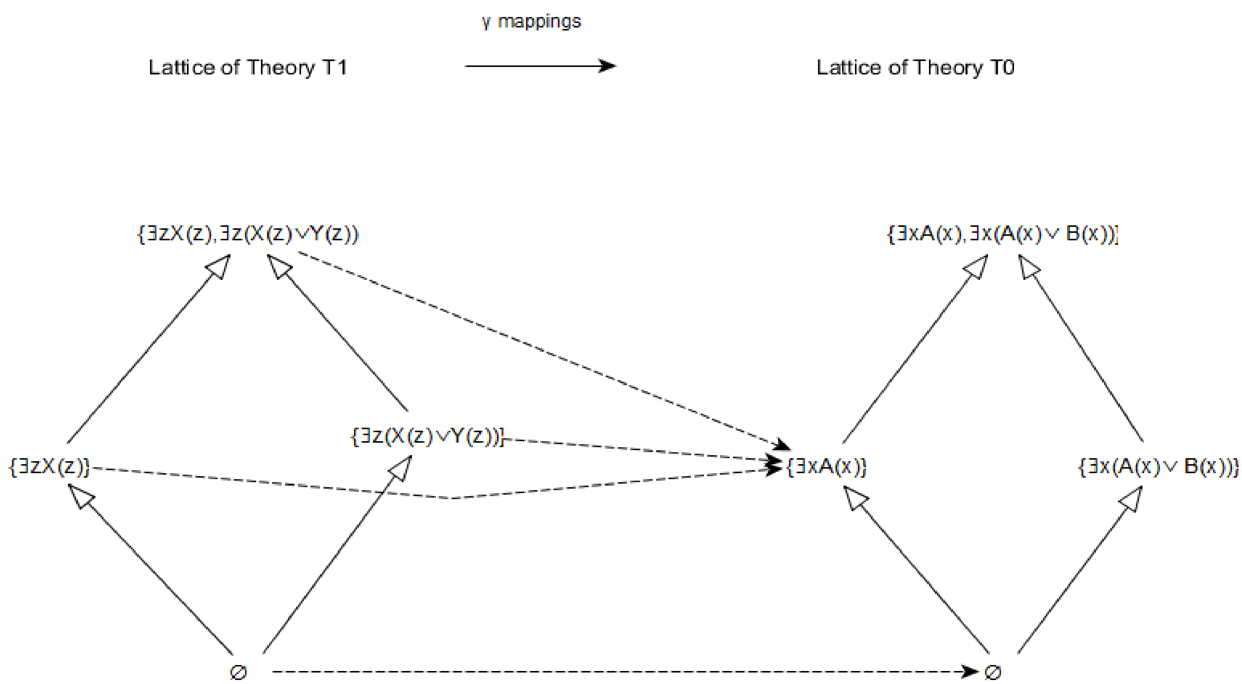

Figure 6-2 (b). γ mappings between power set lattice of theory T_1 and power set lattice of theory T_0.

6.2 Combination of source ontologies and mappings $f(O_i, O_j, M_{ij}, M_{ji})$ and Galois connection

In the case of $f(O_i, O_j, M_{ij}, M_{ji})$, which is defined in Chapter 4, ontology mappings are much less structured and do not have any sense (can be read in any direction). Additionally, mappings are not precise — different from the case of Enderton. Therefore, there is need of analyzing properties of correct mappings such as inconsistency and conservativity related to $f(O_i, O_j, M_{ij}, M_{ji})$.

We mention here the fact that $f(O_i, O_j, M_{ij}, M_{ji})$ ontology which is a combination of source ontologies O_i and O_j and mappings M_{ij} and M_{ji} do not remove the existing knowledge of source ontologies. All the mappings that we have discussed in Chapter 2 and Chapter 3 may add more knowledge but they cannot remove the existing knowledge of source ontologies involved. $f(O_i, O_j, M_{ij}, M_{ji})$ ontology might be consistent or inconsistent or might respect or violate principle of conservativity, however, knowledge of source ontologies will remain part of the $f(O_i, O_j, M_{ij}, M_{ji})$ ontology.

Content of $f(O_i, O_j, M_{ij}, M_{ji})$ ontology are inferences of ontology O_i, inferences of ontology O_j and inferences about relatedness of inferences of ontology O_i and inferences of ontology O_j.

We build the lattice of inferences $\langle \wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq\rangle$ for ontology $f(O_i, O_j, M_{ij}, M_{ji})$. We abstract the elements of $\langle \wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq\rangle$ and they are labeled on the basis of inferences they are referring.

$I(O_i)$ refers to inferences of ontology O_i;

$I(O_j)$ refers to inferences of ontology O_j; and

$I(O_i, O_j)$ refers to inferences that involve both ontology $I(O_i, O_j)$.

An example of $I(O_i, O_j)$ is. $O_i: \exists x A(x) \equiv O_j: \exists x P(x)$; if the 2 artifacts are inferences in respected ontologies (i.e., A and P are satisfiable in respected ontologies). $I(O_i, O_j)$ can be interpreted as a cross product of inferences of ontology O_i and inferences of ontology O_j. However, here for checking the compatibility of ontology mappings, we do not require this complete set of cross product. Note that a subset of inferences $I(O_i, O_j)$ can be represented by using couples; a relation to express clear relationship between members of the couple can be added in this couple, as is the case in DDL. For instance, $O_i: A \sqsubseteq B \equiv O_j: C \sqsubseteq D$. Each Couple is comprised of artifacts of ontology O_i and artifacts of ontology O_j and a relation is used to relate members of the couple. Reason for this restricted set of inferences is that we are only interested in establishing Galois connection between two ontologies. We represent this restricted set as $I(O_i, O_j)^R$

However, following two issues arises in this case.

(1) Such inferences may not be considered as inference in a particular logic. For instance, $A \sqsubseteq B \equiv C \sqsubseteq D$ is not an inference in DL.

(2) Such subset may be a void set.

For case (1), this issue can be resolved by translating these theories/ontologies in first-order logic syntax where such formulas are well-formed formulas.

For case (2), i.e., when $I(O_i, O_j)^R = \emptyset$, we don't have sufficient information to reason about ontology mappings. Therefore, we explicitly mention that this should not be the case.

Abstract contents of $\langle \wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq\rangle$ are subsets of $\{I(O_i), I(O_j), I(O_i, O_j)\}$. And they are

$$\left\{\begin{matrix} \emptyset, \{I(O_i)\}, \{I(O_j)\}, \left\{I(O_i, O_j)^R\right\}, \{I(O_i), I(O_j)\}, \left\{I(O_i), I(O_i, O_j)^R\right\}, \\ \left\{I(O_j), I(O_i, O_j)^R\right\}, \left\{I(O_i), I(O_j), I(O_i, O_j)^R\right\} \end{matrix}\right\}$$

While the abstract content of $\langle \wp(O_i), \subseteq\rangle$ are subsets of $\{I(O_i)\}$ and abstract contents of $\langle \wp(O_j), \subseteq\rangle$ are subsets of $\{I(O_j)\}$.

Theorem 6-7: *If $f(O_i, O_j, M_{ij}, M_{ji})$ is coherent, i.e., no symbol in $f(O_i, O_j, M_{ij}, M_{ji})$ in all models is interpreted as empty set, respects principle of conservativity and $I(O_i, O_j)^R \neq \emptyset$ then there exists a Galois connection $\langle \wp(O_i), \subseteq \rangle \underset{\alpha_i \circ \gamma_j}{\overset{\alpha_j \circ \gamma_i}{\leftrightarrows}} \langle \wp(O_j), \subseteq \rangle$ where $\langle \wp(f(O_i, O_j, M_{ij}, M_{ji})), \subseteq \rangle \underset{\alpha_i}{\overset{\gamma_i}{\leftrightarrows}} \langle \wp(O_i), \subseteq \rangle$ and $\langle \wp(f(O_i, O_j, M_{ij}, M_{ji})), \subseteq \rangle \underset{\alpha_j}{\overset{\gamma_j}{\leftrightarrows}} \langle \wp(O_j), \subseteq \rangle$.*

Proof:

We are using the property of Galois connection that states α and γ mappings that form Galois connection uniquely determines each other. If the γ mappings (both γ_i and γ_j) is not mapped to $\{I(O_i), I(O_j), I(O_i, O_j)^R\}$ then we don't have Galois connection between $\langle \wp(f(O_i, O_j, M_{ij}, M_{ji})), \subseteq \rangle \underset{\alpha_i}{\overset{\gamma_i}{\leftrightarrows}} \langle \wp(O_i), \subseteq \rangle$ and $\langle \wp(f(O_i, O_j, M_{ij}, M_{ji})), \subseteq \rangle \underset{\alpha_j}{\overset{\gamma_j}{\leftrightarrows}} \langle \wp(O_j), \subseteq \rangle$. The reason is that $\gamma_i \circ \alpha_i(x) \nsubseteq x$ *and* $\gamma_i \circ \alpha_i(x) \nsupseteq x$, for instance, for *if* $\gamma_i(\{I(O_i)\}) = \{I(O_i), I(O_j)\}$ while $\alpha_i(\{I(O_i), I(O_j)\}) = \{I(O_i)\}$ *and then* $\alpha_i\left(\{I(O_i), I(O_j), I(O_i, O_j)^R\}\right) = \{I(O_i)\}$ *and* $\alpha_i(\{I(O_i)\}) = \{I(O_i)\}$. *So* $\gamma_i \circ \alpha_i(\{I(O_i)\}) = \{I(O_i), I(O_j)\} \supseteq \{I(O_i)\}$ *and*

$\gamma_i \circ \alpha_i\left(\{I(O_i), I(O_j), I(O_i, O_j)^R\}\right) = \{I(O_i), I(O_j)\} \subseteq \{I(O_i), I(O_j), I(O_i, O_j)^R\}$ and there is no Galois connection in this case.

Therefore, Galois connection between $\langle \wp(O_i), \subseteq \rangle$ and $\langle \wp(O_j), \subseteq \rangle$ that respects the following condition

$$\langle \wp(f(O_i, O_j, M_{ij}, M_{ji})), \subseteq \rangle \underset{\alpha_i}{\overset{\gamma_i}{\leftrightarrows}} \langle \wp(O_i), \subseteq \rangle$$

$$\langle \wp(f(O_i, O_j, M_{ij}, M_{ji})), \subseteq \rangle \underset{\alpha_j}{\overset{\gamma_j}{\leftrightarrows}} \langle \wp(O_j), \subseteq \rangle$$

$$\langle \wp(O_i), \subseteq \rangle \underset{\alpha_i \circ \gamma_j}{\overset{\alpha_j \circ \gamma_i}{\leftrightarrows}} \langle \wp(O_j), \subseteq \rangle$$

has the following mappings

$$\alpha_i(\emptyset) = \emptyset$$

$$\alpha_i(\{I(O_i)\}) = \{I(O_i)\}$$

$$\alpha_i(\{I(O_j)\}) = \{I(O_i)\}$$

$$\alpha_i(\{I(O_i, O_j)^R\} = \{I(O_i)\}$$

$$\alpha_i(\{I(O_i), I(O_j)\}) = \{I(O_i)\}$$

$$\alpha_i\left(\left\{I(O_i),I\left(O_i,O_j\right)^R\right\}\right)=\{I(O_i)\}$$

$$\alpha_i\left(\left\{I\left(O_j\right),I\left(O_i,O_j\right)^R\right\}\right)=\{I(O_i)\}$$

$$\alpha_i\left(\left\{I(O_i),I\left(O_j\right),I\left(O_i,O_j\right)^R\right\}\right)=\{I(O_i)\}$$

$$\gamma_i(\emptyset)=\emptyset$$

$$\gamma_i(\{I(O_i)\})=\left\{I(O_i),I\left(O_j\right),I\left(O_i,O_j\right)^R\right\}$$

From these mappings, we have $\langle\wp(f\left(O_i,O_j,M_{ij},M_{ji}\right)),\subseteq\rangle \underset{\alpha_i}{\overset{\gamma_i}{\leftrightarrows}} \langle\wp(O_i),\subseteq\rangle$

$$\alpha_j(\emptyset)=\emptyset$$

$$\alpha_j(\{I(O_i)\})=\left\{I\left(O_j\right)\right\}$$

$$\alpha_j\left(\left\{I\left(O_j\right)\right\}\right)=\left\{I\left(O_j\right)\right\}$$

$$\alpha_j\left(\left\{I\left(O_i,O_j\right)^R\right\}\right)=\left\{I\left(O_j\right)\right\}$$

$$\alpha_j\left(\left\{I(O_i),I\left(O_j\right)\right\}\right)=\left\{I\left(O_j\right)\right\}$$

$$\alpha_j\left(\left\{I(O_i),I\left(O_i,O_j\right)^R\right\}\right)=\left\{I\left(O_j\right)\right\}$$

$$\alpha_j\left(\left\{I\left(O_j\right),I\left(O_i,O_j\right)^R\right\}\right)=\left\{I\left(O_j\right)\right\}$$

$$\alpha_j\left(\left\{I(O_i),I\left(O_j\right),I\left(O_i,O_j\right)^R\right\}\right)=\left\{I\left(O_j\right)\right\}$$

$$\gamma_j(\emptyset)=\emptyset$$

$$\gamma_j\left(\left\{I\left(O_j\right)\right\}\right)=\left\{I(O_i),I\left(O_j\right),I\left(O_i,O_j\right)^R\right\}$$

From these mappings, we have $\langle\wp(f\left(O_i,O_j,M_{ij},M_{ji}\right)),\subseteq\rangle \underset{\alpha_j}{\overset{\gamma_j}{\leftrightarrows}} \langle\wp\left(O_j\right),\subseteq\rangle$.

Here, we are not using composition of Galois connection which is composition of α mappings and composition of γ mappings, however, we use mappings of Galois connections $\langle\wp(f\left(O_i,O_j,M_{ij},M_{ji}\right)),\subseteq\rangle \underset{\alpha_i}{\overset{\gamma_i}{\leftrightarrows}} \langle\wp(O_i),\subseteq\rangle$ and $\langle\wp(f\left(O_i,O_j,M_{ij},M_{ji}\right)),\subseteq\rangle \underset{\alpha_i}{\overset{\gamma_i}{\leftrightarrows}} \langle\wp(O_i),\subseteq\rangle$ for forming another Galois connection $\langle\wp(O_i),\subseteq\rangle \underset{\alpha_i\circ\gamma_j}{\overset{\alpha_j\circ\gamma_i}{\leftrightarrows}} \langle\wp\left(O_j\right),\subseteq\rangle$.

α and γ Mappings and power set lattices are shown in Figure 6-3 and Figure 6-4.

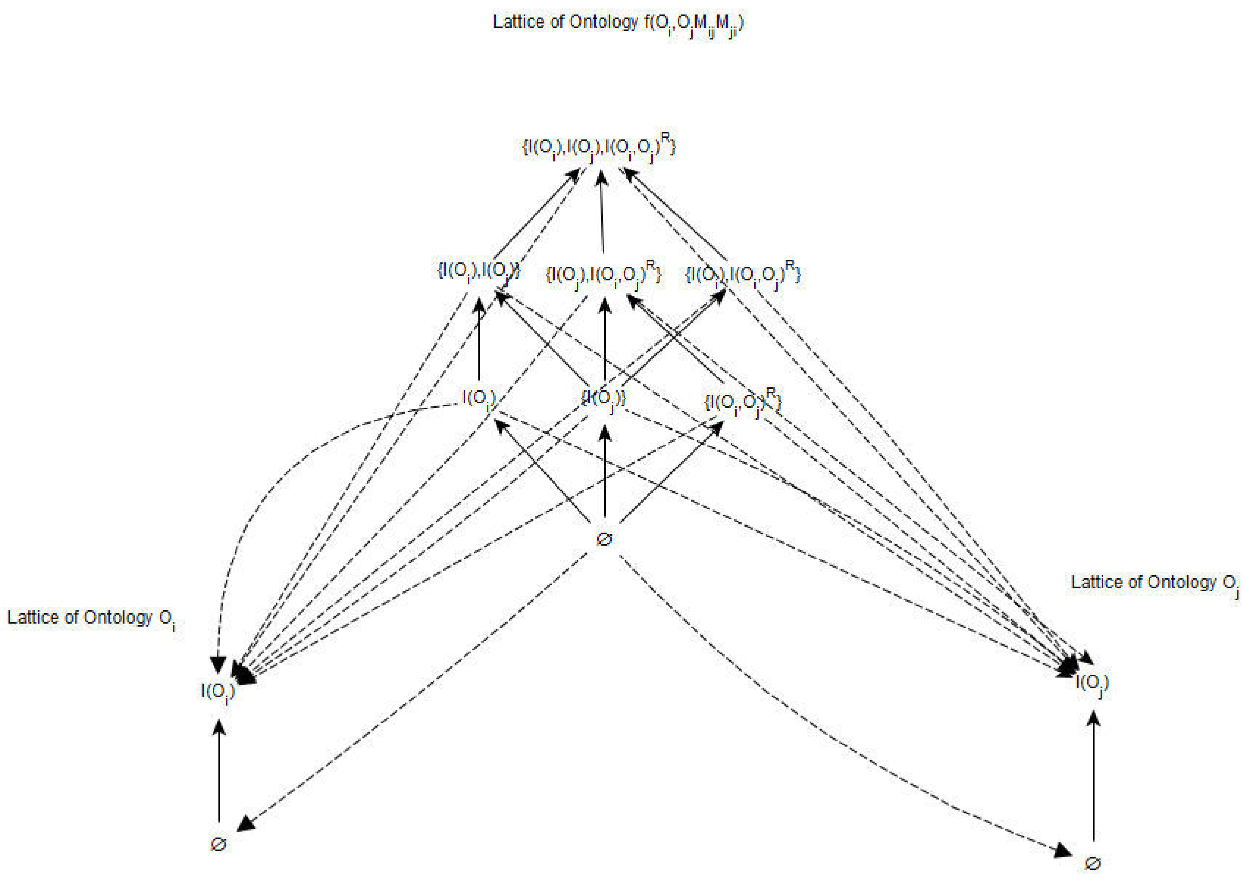

Figure 6-3. α mappings (α_i: $\langle \wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq\rangle \rightarrow \langle \wp(O_i), \subseteq\rangle$ and

α_j: $\langle \wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq\rangle \rightarrow \langle \wp(O_j), \subseteq\rangle$

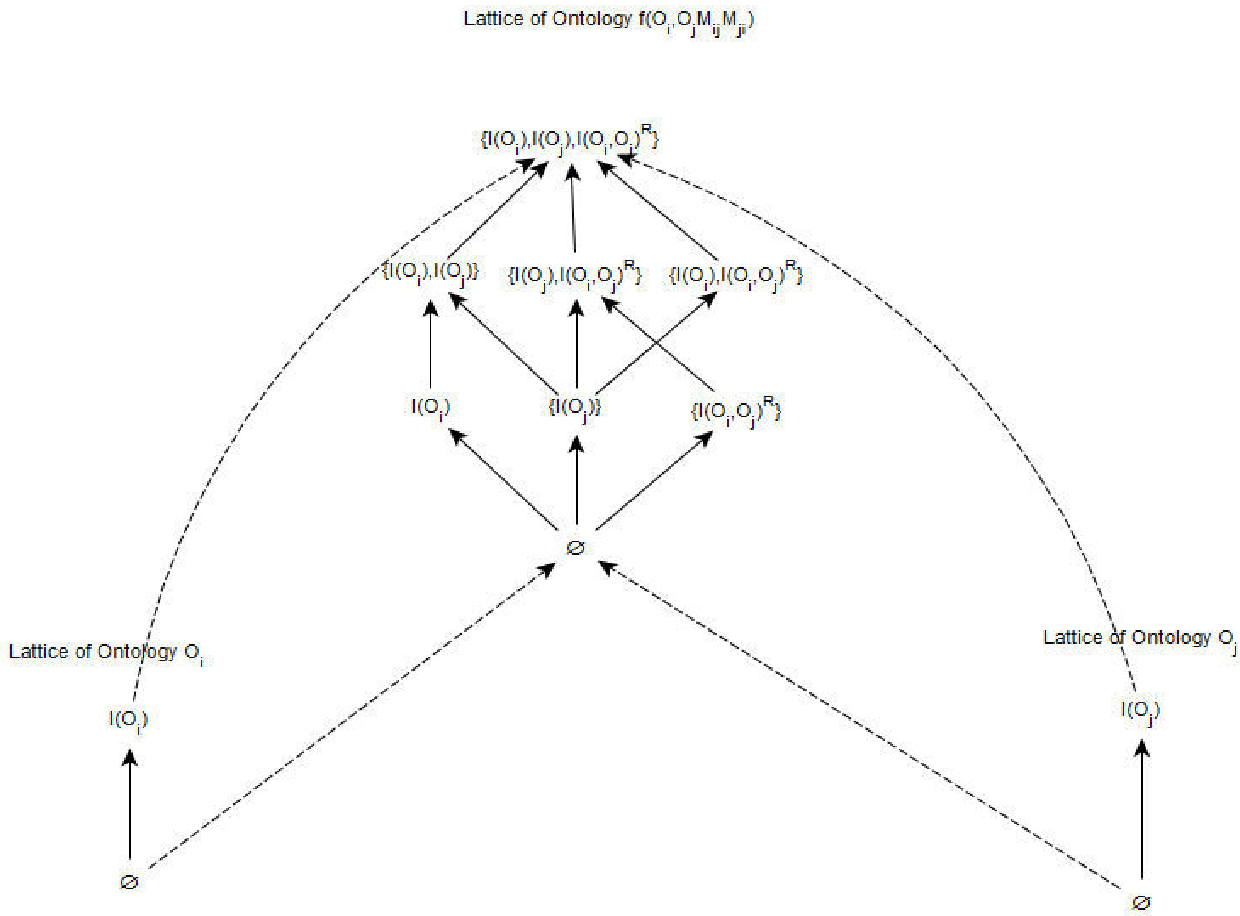

Figure 6-4. γ mappings ($\gamma_i: \langle \wp(O_i), \subseteq \rangle \rightarrow \langle \wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq \rangle$ and $\gamma_j: \langle \wp(O_j), \subseteq \rangle \rightarrow$
$\langle \wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq \rangle$

When we concretize the content of abstract lattices, it is clear that elements of $\left\{I(O_i), I\left(O_j\right), I\left(O_i, O_j\right)^R\right\}$ are inference of ontology O_i and inferences of ontology O_j and inferences that relate ontology O_i and O_j.

$\alpha_i: \langle \wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq \rangle \rightarrow \langle \wp(O_i), \subseteq \rangle$ and α_i mapping is defined as below. *Concretization of mapping* $\alpha_i(\{I(O_i)\}) = \{I(O_i)\}$ *is carried out as*

$I(O_i)$ inferences in $\langle \wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq \rangle$ are the same inferences as in $\langle \wp(O_i), \subseteq \rangle$ and their mappings are defined as

$$\forall x \in I(O_i) \alpha_i(x) = x$$

Concretization of mapping $\alpha_i(\{I\left(O_i, O_j\right)^R\} = \{I(O_i)\}$ is carried out as

$I(O_i, O_j)^R$ inferences in $\langle \wp\left(f(O_i, O_j, M_{ij}, M_{ji})\right), \subseteq\rangle$ are the inferences that relate inferences of ontology O_i with inferences of ontology O_j and their mappings are defined as

$$\forall (x, y) \in I(O_i, O_j)^R \mid x \in I(O_i), y \in I(O_j) \alpha_i(x, y) = x$$

Note that x is an inference and should be read as $x \rightarrow true$ in first-order logic.

Concretization of mapping $\alpha_i(\{I(O_j)\}) = \{I(O_i)\}$ *is carried out as*

$I(O_j)$ inferences in $\langle \wp\left(f(O_i, O_j, M_{ij}, M_{ji})\right), \subseteq\rangle$ are the same inferences as in $\langle \wp(O_j), \subseteq\rangle$ and they are mapped in α_i mappings as

$$\forall y \in I(O_j) \exists \hat{x} \in \langle \wp(O_i), \subseteq\rangle \ \alpha_i(y) = \left\{x \middle| x \rightarrow \hat{x} \ or \ \hat{x} \rightarrow x, \langle \hat{x}, y\rangle \in I(O_i, O_j)^R\right\}$$

However, it may be the case that mappings are *partial*, i.e., $\langle \hat{x}, y\rangle \notin I(O_i, O_j)^R$. This situation can be treated either by completing the missing mappings of $I(O_i, O_j)^R$ or by reducing the contents of source ontologies here, source ontologies representing as $I(O_i)$ and $I(O_j)$. Reducing the content should be done in such a way that only those artifacts/inferences should be part of the ontology which are used in ontology mapping. In this case, our condition for mapping of $\forall y \in I(O_j) \ \alpha_i(y)$ remains the same but only the contents of source ontologies are different.

$\emptyset$ in $\langle \wp\left(f(O_i, O_j, M_{ij}, M_{ji})\right), \subseteq\rangle$ is mapped in α_i mapping as

$$\alpha_i(\emptyset) = \emptyset$$

Remaining mappings are completed monotonically.

Similarly $\alpha_j: \langle \wp\left(f(O_i, O_j, M_{ij}, M_{ji})\right), \subseteq\rangle \rightarrow \langle \wp(O_i), \subseteq\rangle$ and α_j mapping is defined as below.

Concretization of mapping $\alpha_j(\{I(O_j)\}) = \{I(O_j)\}$ *is carried out as*

$I(O_j)$ inferences in $\langle \wp\left(f(O_i, O_j, M_{ij}, M_{ji})\right), \subseteq\rangle$ are the same inferences as in $\langle \wp(O_j), \subseteq\rangle$ and their mappings are defined as

$$\forall y \in I(O_j) \alpha_j(y) = y$$

Concretization of mapping $\alpha_j(\{I(O_i, O_j)^R\} = \{I(O_j)\}$ is carried out as

$I(O_i, O_j)^R$ inferences in $\langle \wp\left(f(O_i, O_j, M_{ij}, M_{ji})\right), \subseteq\rangle$ are the inferences that relate inference of ontology O_i with inferences of ontology O_j and their mappings are defined as

$$\forall (x, y) \in I(O_i, O_j)^R \mid x \in I(O_i), y \in I(O_j) \alpha_j(x, y) = y$$

Note that y is an inference and should be read as $y \to true$ in first-order logic. In Description logics, it should be read as $y \sqsubseteq \top$.

Concretization of mapping $\alpha_j(\{I(O_i)\}) = \{I(O_j)\}$ *is carried out as*

$I(O_i)$ inferences in $\langle \wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq\rangle$ are the same inferences as in $\langle \wp(O_i), \subseteq\rangle$ and they are mapped in α_j mappings as

$$\forall x \in I(O_i) \exists \hat{y} \in \langle \wp(O_j), \subseteq\rangle \; \alpha_j(x) = \left\{y \middle| y \to \hat{y} \; or \; \hat{y} \to y, \langle x, \hat{y}\rangle \in I(O_i, O_j)^R\right\}$$

However, it may be the case that mappings are *partial*, i.e., $\langle x, \hat{y}\rangle \notin I(O_i, O_j)^R$. This situation can be treated either by completing the missing mappings of $I(O_i, O_j)^R$ or by reducing the contents of source ontologies here, source ontologies representing as $I(O_i)$ and $I(O_j)$. Reducing the content should be done in such a way that only those artifacts/inferences should be part of the ontology which are used in ontology mapping. In this case, our condition for mapping of $\forall x \in I(O_i) \; \alpha_i(x)$ remains the same but only the contents of source ontologies are different.

$\emptyset$ in $\langle \wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq\rangle$ is mapped in α_j mapping as

$$\alpha_j(\emptyset) = \emptyset$$

Remaining mappings are completed monotonically.

α mappings that are defined in this way are monotone, γ mappings are defined in terms of α mappings as

$$\gamma_i(s) = lub\{x | \alpha_i(x) \subseteq s\}$$

$$\gamma_j(s) = lub\{x | \alpha_j(x) \subseteq s\}$$

These mappings form Galois connections $\langle \wp(O_i), \subseteq\rangle \underset{\alpha_i \circ \gamma_j}{\overset{\alpha_j \circ \gamma_i}{\leftrightarrows}} \langle \wp(O_j), \subseteq\rangle$ *where* $\langle \wp(f(O_i, O_j, M_{ij}, M_{ji})), \subseteq\rangle \underset{\alpha_i}{\overset{\gamma_i}{\leftrightarrows}} \langle \wp(O_i), \subseteq\rangle$ *and* $\langle \wp(f(O_i, O_j, M_{ij}, M_{ji})), \subseteq\rangle \underset{\alpha_j}{\overset{\gamma_j}{\leftrightarrows}} \langle \wp(O_j), \subseteq\rangle$.

Note that $\alpha_i \circ \gamma_i = 1_{O_i}$ and $\alpha_j \circ \gamma_j = 1_{O_j}$ and $\alpha_i \circ \gamma_j \circ \alpha_j \circ \gamma_i(x) = 1_{O_i}$ and $\alpha_j \circ \gamma_i \circ \alpha_i \circ \gamma_j(y) = 1_{O_j}$.

Q.E.D.

Theorem 6-7 satisfies the condition of compatibility by following its procedure since mappings from $\langle \wp(O_i), \subseteq\rangle$ to $\langle \wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq\rangle$ and then from

$\langle\wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq\rangle$ to $\langle\wp(O_j), \subseteq\rangle$ respect the procedure of compatible mapping described in Chapter 5, as

$$P(m_{ij}) \circ \Pi^i(\left\{\langle x, y\rangle \middle| \langle x, y\rangle \in I\left(O_i, O_j\right)^R\right\} = P\left(m_{ij}(x)\right) = \{y\} \geq y$$

Hence, respect the condition of procedure of compatible mapping described in Chapter 5, as $P(m_1)\Pi^1(\langle x, y\rangle) = y \geq R\left(C^*(y)\right)$.

Same is the case with the mapping from $\langle\wp(O_j), \subseteq\rangle$ to $\langle\wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq\rangle$ and then from $\langle\wp\left(f\left(O_i, O_j, M_{ij}, M_{ji}\right)\right), \subseteq\rangle$ to $\langle\wp(O_i), \subseteq\rangle$.

Now, we provide some foundations for the following theorem, which is a converse of Theorem 6-7.

Lemma 1: *When mutual interpretations of theories $M_{ij}: T_0 \to T_1$ and $M_{ji} : T_1 \to T_0$ are used in combination with theories T_0 and T_1, represented as $f\left(O_i, O_j, M_{ij}, M_{ji}\right)$, then it is possible that that $f(O_i, O_j, M_{ij}, M_{ji})$ is consistent.*

Proof:

Theories are represented in terms of inferences below.

Theory T_i		**Theory T_j**
$\exists x A x$		$\exists x \phi x$
$\exists x \neg A x$		$\exists x \neg \phi x$
$\exists x A x \perp \exists x \neg A x$		$\exists x \phi x \perp \exists x \neg \phi x$

Mappings are represented below.

Mapping M_{ij}			**Mapping M_{ji}**	
Ax	ϕx		ϕx	$\neg Ax$

$M_{ij}: T_i \to T_j$ is theory Interpretation of T_i into T_j is shown below.

Theory T_i		**Syntactical Translation of Theory T_i**
$\exists x A x$		$\exists x \phi x \in T_j$
$\exists x \neg A x$		$\exists x \neg \phi x \in T_j$

$\exists x Ax \perp \exists x \neg Ax$		$\exists x \phi x \perp \exists x \neg \phi x \in T_j$

$M_{ji}: T_j \rightarrow T_i$ is theory Interpretation of T_j into T_i is shown below.

Theory T_j		**Syntactical Translation of Theory T_i**
$\exists x \phi x$		$\exists x \neg Ax \in T_i$
$\exists x \neg \phi x$		$\exists x Ax \in T_i$
$\exists x \phi x \perp \exists x \neg \phi x$		$\exists x Ax \perp \exists x \neg Ax \in T_i$

Thus, M_{ij} and M_{ji} are mutual interpretations.

$$f\left(O_i, O_j, M_{ij}, M_{ji}\right) = O_i \cup O_j \cup M_{ij} \cup M_{ji}$$

$$\Rightarrow O_i \cup O_j \cup (Ax \rightarrow \phi x) \cup (\phi x \rightarrow \neg Ax)$$

From above,

$$f\left(O_i, O_j, M_{ij}, M_{ji}\right) \vDash Ax \rightarrow \neg Ax$$

$$f\left(O_i, O_j, M_{ij}, M_{ji}\right) \vDash \exists x Ax$$

Therefore, $f\left(O_i, O_j, M_{ij}, M_{ji}\right)$ is inconsistent.

Q.E.D.

Lemma 2:

Let consider a mapping $M: O_i \rightarrow O_j$ an interpretation of O_i. If O_j is consistent, then the theory $f(O_i, O_j, M)$ containing (all inferences from below):

1. O_j;
2. $\forall x A(x) \rightarrow F(x)$, for each symbol A in the language of O_i whenever M maps A on F where F is a formula in the language of O_j;
3. O_i

is also consistent.

Proof:

Indeed, a model of $f(O_i, O_j, M)$ can be built as follows:

Let consider a model $\mathcal{M}_2$ a model of O_j. We can define an interpretation $\mathcal{I}$ of $f(O_i, O_j, M)$ as

$\mathcal{J}(A) = \mathcal{M}_2(A)$ if A is part of the language L_2 of O_j

$\mathcal{J}(B) = \mathcal{M}_2(F)$ if B is part of O_i, F is part of O_j and we have $B(x) \rightarrow F(x)$

Now, $\mathcal{J}$ is an and $f(O_i, O_j, M)$ has a model $\mathcal{M}_2$. The reason is that if s is an element of O_i, s is satisfied by $\mathcal{J}$ because any s is in $\{s|s\ is\ ture\ in\ every\ \mathcal{M}_2\}$. Additionally, every $\forall x A(x) \rightarrow F(x)$, is trivially true in $\mathcal{J}$ and each s in O_j is true in $\mathcal{M}_2$.

Q.E.D.

Lemma 3:

Let suppose as above that $M: O_i \rightarrow O_j$ is a theory interpretation. If O_i and O_j are coherent, then $f(O_i, O_j, M)$ is coherent.

Proof:

If O_i is coherent then $\exists x B(x)$ is part of O_i. As a consequence, $\exists x B(x)$ is also part of $f(O_i, O_j, M)$. Now, let consider $\exists x A(x)$ is part of O_j, in this case, because $\mathcal{M}_2$ is a model of O_j, then $\exists x B(x)$ is also satisfied.

Q.E.D.

Theorem 6-8:

If ontologies O_i and O_j are consistent and coherent and the mapping $M_{ij} = Syt^{m_{ij}}$ between source ontologies O_i and O_j provides an interpretation of ontology O_i in ontology O_j and the mapping $M_{ji} = Syt^{m_{ji}}$ between source ontologies O_j and O_i provides an interpretation of ontology O_j in ontology O_i, and α and γ mapping between lattices of logical formulas of ontologies O_i and O_j are extensions of mapping M_{ij} and M_{ji} as $\forall s \in \mathcal{L}_i M_{ij}(s) = \alpha(s)$ and $\forall t \in \mathcal{L}_j M_{ji}(t) = \gamma(t)$ such that there exists an isotone Galois connection between Lattice of logical formulas of source ontologies $\langle L(O_i) \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle L(O_j) \rangle$ then $f(O_i, O_j, M_{ij}, M_{ji})$ is coherent, consistent, and conservant.

Proof:

First, we prove that there exists a model $\mathcal{M}$ of $f(O_i, O_j, M_{ij}, M_{ji})$. Then, we prove that theory obtained by reasoning of syntactical translations of theories by using mutual interpretations corresponds to the model $\mathcal{M}$ of $f(O_i, O_j, M_{ij}, M_{ji})$.

We apply mappings M_{ij} and M_{ji}, respectively, iteratively and show that there exists a model in each iteration. We prove this by induction.

In the iteration 0, when mapping M_{ij} is applied to source ontologies, then model M^0 is

$$M^0 = \begin{cases} M^0(B) = M^j(B), B \in O_j \\ M^0(A) = M^j(\phi), A \in O_i, and\ A \to \phi \in M_{ij} \end{cases}$$

$$M^0 \vDash O_i \cup O_j$$

$$M^0 \vDash A \to \phi$$

In the iteration 1, when mapping M_{ji} is applied to source ontologies, then model $M^1$is

$$M^1 = \begin{cases} M^1(A) = M^0(A) = M^j(\phi), A \in O_j \\ M^1(B) = M^0(\psi), B \in O_j, and\ B \to \psi \in M_{ji} \end{cases}$$

$$M^1 \vDash O_i \cup O_j$$

$$M^1 \vDash B \to \psi$$

In the iteration 2, when mapping M_{ij} is applied to source ontologies, then model $M^2$is

$$M^2 = \begin{cases} M^2(A) = M^1(\phi) \\ M^2(B) = M^1(B) = M^0(\psi) \end{cases}$$

$$M^2 \vDash O_i \cup O_j$$

In the iteration k, when mapping is applied to source ontologies, then model $M^k$is

$$M^k = \begin{cases} M^k(A) = M^{k-1}(\phi) \\ M^k(B) = M^{k-1}(B) \end{cases}$$

$$M^k \vDash O_i \cup O_j$$

In the iteration $k + 1$, when mapping is applied to source ontologies, then model $M^k$is

$$M^{k+1} = \begin{cases} M^{k+1}(A) = M^k(A) \\ M^{k+1}(B) = M^k(\psi) \end{cases}$$

$$M^{k+1} \vDash O_i \cup O_j$$

Next, we prove that theory $f(O_i, O_j, M_{ij}, M_{ji})$ obtained by reasoning on syntactical translations based on mutual interpretations has counterpart in model side.

Iteration 0 (corresponding to M^0)

$$\left[O_i[A\backslash\phi] \cup\ O_j \cup (A \to \phi)\right]$$

Because of being an interpretation, we have $O_i[A\backslash\phi] \subseteq O_j$, therefore,

$$\left[O_i[A\backslash\phi] \cup\ O_j \cup (A \to \phi)\right] \Rightarrow \left[O_j \cup (A \to \phi)\right]$$

Iteration 1 (corresponding to M^1)

$$[O_j[B\backslash\psi] \cup (A \to \phi[B\backslash\psi]) \cup O_i \cup (B \to \psi)]$$

Because of being an interpretation, we have $\left(O_j[B\backslash\psi]\right) \subseteq O_i$, therefore,

$$[O_j[B\backslash\psi] \cup (A \rightarrow \phi[B\backslash\psi]) \cup O_i \cup (B \rightarrow \psi)] \Rightarrow O_i \cup (B \rightarrow \psi) \cup (A \rightarrow \phi)[B\backslash\psi]$$

$B \rightarrow \psi$ can be anything, $\boldsymbol{A \rightarrow \phi[B\backslash\psi]}$ **should be satisfied by a model of $\boldsymbol{O_i}$.**

The condition is same as provided by Galois connection. Indeed, Implication ($\rightarrow$) corresponds to the order relation ($\leq$) in the lattice. Condition is respected because it can be represented as

$$\boldsymbol{A \leq \phi[B\backslash\psi]}$$

$\phi[B\backslash\psi] = \gamma \circ \alpha(A)$, because of the assumptions in the theorem.

Iteration 2 (corresponding to $\boldsymbol{M^2}$)

$$\left[O_i[A\backslash\phi] \cup (B \rightarrow \psi[A\backslash\phi]) \cup (A \rightarrow \phi)[B\backslash\psi][A\backslash\phi] \cup O_j \cup (A \rightarrow \phi)\right]$$

Because of being an interpretation, we have $O_i[A\backslash\phi] \subseteq O_j$, therefore,

$$\left[O_i[A\backslash\phi] \cup (B \rightarrow \psi[A\backslash\phi]) \cup (A \rightarrow \phi[B\backslash\psi])[A\backslash\phi] \cup O_j \cup (A \rightarrow \phi)\right]$$

$$\Rightarrow O_j \cup (B \rightarrow \psi[A\backslash\phi]) \cup (A \rightarrow \phi[B\backslash\psi])[A\backslash\phi] \cup (A \rightarrow \phi)$$

$A \rightarrow \phi$ can be anything and $(A \rightarrow \phi[B\backslash\psi])[A\backslash\phi]$, $B \rightarrow \psi[A\backslash\phi]$ **should be satisfied by a model of $\boldsymbol{O_j}$.** $(A \rightarrow \phi[B\backslash\psi])[A\backslash\phi]$ is the same as $\phi \rightarrow \phi[B\backslash\psi][A\phi]$

The condition is same as provided by Galois connection. Indeed, Implication ($\rightarrow$) corresponds to the order relation ($\leq$) in the lattice. Condition is respected because it can be represented as

$$\boldsymbol{(B \leq \psi)[A\backslash\phi]}$$

$\psi[A\backslash\phi] = \alpha \circ \gamma(B)$, $\phi[B\backslash\psi][A\phi] = \alpha \circ \gamma(\psi)$because of the assumptions in the theorem.

Iteration 3 (corresponding to $\boldsymbol{M^3}$)

$$O_j[B\backslash\psi] \cup (B \rightarrow \psi[A\backslash\phi])[B\backslash\psi] \cup (A \rightarrow \phi[B\backslash\psi]) \cup (\phi \rightarrow \phi[B\backslash\psi][A\phi])[B\backslash\psi] \cup O_i \cup (B \rightarrow \psi)$$

Because of being an interpretation, we have $O_j[B\backslash\psi] \subseteq O_i$, therefore,

$$O_j[B\backslash\psi] \cup (B \rightarrow \psi[A\backslash\phi])[B\backslash\psi] \cup (A \rightarrow \phi[B\backslash\psi]) \cup (\phi \rightarrow \phi[B\backslash\psi][A\phi])[B\backslash\psi] \cup O_i \cup (B \rightarrow \psi)$$

$$\Rightarrow O_i \cup (B \rightarrow \psi[A\backslash\phi])[B\backslash\psi] \cup (A \rightarrow \phi[B\backslash\psi]) \cup (\phi \rightarrow \phi)[B\backslash\psi][A\backslash\phi] \cup (B \rightarrow \psi)$$

$B \rightarrow \psi$ can be anything, $(A \rightarrow \phi[B\backslash\psi])$ is already satisfied in iteration 1, $(A \rightarrow \phi[B\backslash\psi])[A\backslash\phi] \in O_i$, $B \rightarrow \psi[A\backslash\phi][B\backslash\psi]$ is same as $\psi \rightarrow \psi[A\backslash\phi][B\backslash\psi]$ **should be satisfied by a model of**

$\boldsymbol{O_i}$. While $(\phi[B\backslash\psi] \rightarrow \phi[B\backslash\psi][A\phi][B\backslash\psi])$ is $\gamma(\phi) \leq \gamma \circ \alpha \circ \gamma(\phi)$ and it is proved by monotonicity.

The condition is same as provided by Galois connection. Indeed, Implication ($\rightarrow$) corresponds to the order relation ($\leq$) in the lattice. Condition is respected because it can be represented as

$$\boldsymbol{\psi \rightarrow \psi[A\backslash\phi][B\backslash\psi]}$$

Iteration 3 shows that in successive iterations, because of monotonicity condition of Galois connection, and implications proved earlier, successive iterations will be satisfied.

Q.E.D

6.3 Properties of Compatible and incompatible ontology mappings

In this section, we describe the properties of compatible mappings. Properties of compatible ontology mappings will also highlight the strength of our proposed approach for solving the problem of identifying compatible and incompatible mappings.

6.3.1 Symmetry

If an ontology mapping M_{AB} is compatible with other ontology mapping M_{BA}, then M_{BA} is also compatible with M_{AB}.

However, as per theorem 5-13, Galois connection are not necessarily symmetric. In the case of compatible mappings, it is sufficient to establish a Galois Connection between adequate lattice of ontology O_A and adequate lattice of ontology O_B by projecting mapping M_{AB} and M_{BA}; no matter whether adequate lattice of ontology O_A is upper adjoint or adequate lattice of ontology O_B. Therefore, compatible ontology mappings are interpreted as symmetric and we can say that ontology mapping M_{AB} is compatible with ontology mapping M_{BA} and also mapping M_{BA} is compatible with ontology mapping M_{AB} (Symmetric).

In the same manner, incompatible mappings are interpreted as symmetric.

6.3.2 Composition

According to 5-14, composition of monotone Galois connection is a Galois connection.

Given compatible mappings M_{AB} and M_{BA} between ontology O_A and ontology O_B and M_{BC} and M_{CB} between ontology O_B and ontology O_C, then by composition there is also a compatible mapping $M_{AB} \circ M_{BC}$ and $M_{CB} \circ M_{BA}$ between ontology O_A and ontology O_C given that same procedure P is used for establishing compatible mappings M_{AB} and M_{BA} between ontology O_A and ontology O_B and M_{BC} and M_{CB} between ontology O_B and ontology O_C.

If procedure is not same, then we may have not complete mapping to have a Galois connection.

6.3.3 Subset of a mapping

Having a subset of mapping means that mapping is partial (not complete), we can check compatibility of this partial mapping in two ways:

Completing partial mapping by mapping artifacts/symbols of an ontology which are not mapped are mapped to top $\top$ of the appropriate lattice of another ontology. Then, using some procedure P, we can check whether these mappings are compatible. Based on existence of Galois connection, these mappings are classified as *weak compatible* or *weak incompatible* mappings.

By using appropriate procedure P and Theorem 6-7, we can check compatibility of partial mapping without completing them.

If we extract a subset of mapping from compatible mapping, then they are also compatible. The reason is that we can extend these subset of mappings to at least two mappings (from where these subset of mappings are obtained) which are compatible.

A subset or module of ontology mapping can be extracted from existing ontology mapping. However, the best use of ontology mapping is in having maximum ontology mapping correspondences. What may be desirable in some scenarios is extracting compatible ontology mapping from incompatible ontology mapping and our approach can be used in this regard. We can either extract compatible ontology mapping correspondences from both ontology mappings or extract compatible ontology mapping correspondence by considering that one of the ontology mappings is correct and remove incompatible ontology mapping correspondences from other ontology mapping. However, we do not commit to any preference about which ontology mapping is considered as correct, indeed, we leave it to the users for handling them according to their application scenario.

6.3.4 Dealing with Heterogeneity

Here, heterogeneity refers to different kinds of artifacts are mapped to each other; particularly when concept is mapped to property and vice-versa and when we mix two different kinds of formalism, for instance, DL and FOL, in formalizing ontologies.

Mix of two different kinds of formalism can be easily handled in our approach by using lattice of inferences when possible and translating different formalism to same formalism.

While, dealing Concept to property (C-P) and Property to Concept (P-C) mappings depends on type of formalism (logic).

If first-order logic (FOL) is used for formalizing ontologies then

(C-P) is represented as $C(x) \to P(y, z)$ we need to project variable x to y and z.

(P-C) is represented as $P(y, z) \to C(x)$, we need to introduce a new variable x which is realted to y and z.

For checking compatibility, depending on situation, considering mappings are complete, we can use either Theorem 6-1 or Theorem 6-3.

If Description Logic (DL) is used for formalizing ontologies then

(C-P) is represented as $C \sqsubseteq \exists r.P$

(P-C) is presented as $\exists r.P \sqsubseteq C$,

They are dealt analogously as it has been described in the case of FOL.

For checking compatibility, if we have complete mappings we can use either Theorem 6-1 or Theorem 6-3, otherwise Theorem 6-7.

6.4 Ontology mappings defects and compatibility/incompatibility

According to Theorem 6-1 and Theorem 6-3, given an ontology mapping M_{ij} is correct (i.e., it conveys a theory interpretation), a Galois connection can be established between specific inference lattices (not only between language lattices).

Let now consider two ontology mappings m_1, m_2 and build a connection according to the procedure provided in Theorem 6-1. If m_1 is correct, the procedure results in a Galois connection.

Now, let suppose that we get a Galois connection (but without knowing if m_1 is correct) by applying the procedure given in Theorem 6-1. By applying Theorem 6-3, from this Galois Connection is possible to recover a mapping $m_1{}'$ which is a theory interpretation. If $m_1 = m_1{}'$, then m_1 is correct. This equivalence seems to be verified in several cases, the cases in which $m_1 = m_1{}'$, is not verified then it might be possible that m_1 was not correct and m_1' represents better correct alternative of m_1. Then the procedure of Theorem 6-1 and extraction of Theorem 6-3 are one the inverse of the other one. This means that we can check the correctness of mapping M_{ij} when M_{ji} is not given as in the case of $f(O_i, O_j, M_{ij})$.

Additionally, when mapping M is defined between two ontologies and it is not given that when it is from O_i to O_j or from O_j to O_i then we can treat M in either or both direction.

If the equivalence above is verified, having proved that m_1, m_2 is a Galois Connection, then m_1 is being correct. This is particularly important to talk about "*relative semantics*". In this case, in fact, m1 is proved to be correct by using m2.

However, compatible ontology mappings is still different from ontology mapping correctness: two correct mappings may not be compatible just because any mapping is not the relative semantics of the any other one.

The interest of compatible ontology mappings is to generalize the notion of ontology mapping correctness so that independently of format, logics and so on, compatibility provides a reference concept for defining when 1 mapping is correct.

This interest has been proved in Theorem 6-1 to Theorem 6-6.

Theorem 6-7 shows that from an ontology mapping with good properties (such as conservativity), a Galois Connection can be extracted between inferences. **Theorem 6-8** shows that mappings M_{ij} and M_{ji} are interpretation of ontologies and α and γ containing M_{ij} and M_{ji} forms a Galois connection between lattice of logical formulas then mappings M_{ij} and M_{ji} are when added with consistent and coherent source ontologies as $f(O_i, O_j, M_{ij}, M_{ji})$, then $f(O_i, O_j, M_{ij}, M_{ji})$ remains consistent and coherent.

6.5 Compatible and Incompatible ontology mappings and Mapping Acceptability

We have found in Chapter 4 that consensus (accepted) mappings may be erroneous and we need to check the correctness of accepted mappings. However, our work helps in verifying whether consensus mappings are correct or not. From incompatible mappings we can identify incompatible correspondences. These incompatible correspondences should not be simultaneously part of any preferred extension in Argumentation Framework.

Chapter 7.

Two Applications of Compatible and Incompatible Ontology mappings

In this chapter we show that how notion of compatible and incompatible ontology mappings is applicable to upper ontologies and their mappings and show that incompatible ontology mappings leads to incorrect inferences by employing mappings in the context of ontology merging. We present a short review on upper ontologies and propose a method for dealing with mappings of upper ontology mappings in Section 7.1. We present a case study of DOLCE and GFO mappings for evaluating their compatibility in Section 7.4. In Section 7.5, we present and discuss our findings about existing ontology mappings between upper ontologies. In Section 7.6, we compare our approach with category theory approach of ontology merging and generic approach of ontology merging.

7.1 Introduction

Our focus in this section is on upper ontologies, even though we have also reviewed core ontologies and WordNet in this work. The reason for looking at core ontologies is because a) they are presented in formal way; b) they also describe general concepts, for instance, in PSL (Grüninger & Menzel, 2003) where concepts like activity, activity occurrence, time points and objects can be used to describe (enterprise) processes; c) they provide good insight to understand upper ontologies. We also considered WordNet because even not a pure ontology a) it is a widely used linguistic resource and even used in ontology mapping field beside upper ontologies b) and it is mapped to various upper ontologies.

Upper ontologies are interesting because they can be used to work with several (domain, task, application, role, core) ontologies and other digital resources by guaranteeing an increased consistency. However, upper ontologies are often complex artifacts preventing an effective usage. On the other hand, various, while not many upper ontologies are available, consequently deciding to use one upper ontology over another one remains a challenging task. Therefore, researchers spent some efforts to compare upper ontologies by identifying possible relationships (mappings) between those upper ontologies. These efforts have been often based on some informal analysis so that distinct authors have identified distinct relationships between the same ontologies: unfortunately, some of these relationships are fundamentally different as we discuss in the remainder of the chapter so that some of them cannot be considered correct or cannot be used consistently together in one single application.

7.1.1 Short review of Upper and Core Ontologies

In this section, we provide some basic features of the upper and core ontologies that we have considered. These basic features are:

- *Overview*: A basic description about the ontology that includes foundation theories followed in the design of ontology, key points about them.
- *Dimensions and modularity:* Dimensions of upper ontologies is expressed by number of concepts and axioms used by them. In addition, information about modules of upper ontologies is described.
- *Languages:* Which language is used to represent these upper ontologies.
- *Applications*: Applications developed by using these upper ontologies and the fields in which they are used.
- *Alignment with WordNet:* Whether these upper ontology is aligned to WordNet
- *Licensing:* Which type of license is needed to use these upper ontologies.
- *Documentation:* What kind of documentation is available to use these upper ontologies.
- *Alignment with upper and core ontologies*: Whether these upper ontologies are aligned to other upper or core ontologies.

A. BFO (Basic Formal Ontology):

Home page: http://www.ifomis.org/bfo. *Overview:* BFO consists of series of sub-ontologies; most important of which are: *SNAP* - a series of snapshot ontologies (O_{ti}), indexed by times, *SPAN* - a single videoscopic ontology (O_v). Each O_{ti} is an inventory of all entities existing at a time. O_v is an inventory of all processes unfolding through time. BFO has been developed in accordance with principle of realism (naïve-realism), adequatism and perspectivalism. SNAP/SPAN concepts are used to avoid confusion between continuants and occurrents. BFO has eternalist view. *Dimensions and modularity:* In version 1.1 BFO contains 1 top connecting class ("Entity"), 21 SNAP classes, and 17 SPAN classes for a total of 39 classes. BFO is divided into the SNAP and SPAN modules. *Language(s):* OWL, Isabelle, OBO, CLIF version of BFO merged with RO. *Applications:* BFO has been applied to the biomedical and environment domain. A detail list of application is available at http://www.ifomis.org/bfo/users. *Alignment with WordNet*: Not available. *Licensing:* BFO is freely available. *Documentation:* A manual is available for detail description of BFO. *Alignment with upper and core ontologies*: DOLCE

B. Cyc:

Home page: http://www.cyc.com/. *Overview:* The Cyc knowledge base (KB) is a formalized representation of a vast quantity of fundamental human knowledge for reasoning about the objects and events of everyday life. The Cyc KB focused on a particular domain of knowledge, a particular level of detail, a particular interval in time, etc. *Dimensions and modularity:* The Cyc KB contains nearly five hundred thousand terms, including about fifteen thousand types of relations, and about five million facts (assertions) relating these terms. The "microtheory" approach supports modularity. Cyc has thousands of micro-theories. *Language(s):* Cyc is represented in the CycL formal language. Cyc Ontology Exporter allows exporting specified portions of Cyc to OWL files. *Applications*: Cyc has been used in the domains of machine learning, natural language processing, decision support systems, network risk assessment, and terrorism management. *Alignment with WordNet*: Cyc-to-WordNet mapping that includes some 8,000 WordNet noun synsets. *Licensing:* Cyc is a commercial product. Cycorp offers a no-cost license to the research community (ResearchCyc). Additionally, it has placed the core Cyc ontology (OpenCyc) into the public domain. *Documentation:* Tutorials are available and Cycorp offers training. *Alignment with upper and core ontologies:* DOLCE; SUMO, SOWA, and GFO.

C. DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering):

Home page: http://www.loa-cnr.it/DOLCE.html. *Overview:* DOLCE is the module of the WonderWeb Foundational Ontologies Library. DOLCE has a clear cognitive bias, in the sense that it aims at capturing the ontological categories underlying natural language and human common sense. DOLCE is the ontology of particulars, in the sense that its domain of discourse is restricted to them. DOLCE follows multiplicative approach, different entities can be co-located in the same space-time. *Dimensions and Modularity:* DOLCE-Lite version 2.2 has 80 classes and 24 axioms. DOLCE version 2.1 in KIF format has 82 axioms and 18 theorems. It is not currently divided into modules. *Language(s):* first-order Logic, KIF, OWL. *Applications:* DOLCE is used in applications related to evaluation of ontologies, interoperability, creating foundation ontologies like general ontology of programs, used in domain like biomedical, linguistic, legal, manufacturing and others. *Alignment with WordNet:* DOLCE is aligned with WordNet 1.6; only top level of WordNet is mapped to DOLCE. *Licensing:* DOLCE can be freely downloaded from http://www.loa-cnr.it/ontologies/DLP3971.zip. *Documentation:* DOLCE documentation includes KIF, FOL, and OWL versions of Dolce. WonderWeb Deliverable D18 provides good understanding of Dolce. *Alignment with upper and core ontologies:* GFO, BFO, UFO, BWW, Sowa, Cyc, SUMO, PROTON.

D. GFO (General Formal Ontology):

Home page: http://www.onto-med.de/ontologies/gfo.html. *Overview:* GFO includes levels of reality. GFO includes objects (3D objects) as well as processes (4D entities) and both are integrated into one coherent framework. GFO presents a multi-categorical approach by admitting universals, concepts, and symbol structures and their interrelations. *Dimensions and Modularity:* The OWL version of GFO consists of 79 classes and 67 properties. GFO has a three-layered meta-ontological architecture consisting of an abstract top level, an abstract core level, and a basic level. *Language(s):* OWL, first-order Logic (not yet made public). *Applications:* One of the aims of GFO is used it in the field of biomedical science. GFO is also used in the domain of conceptual modelling, creating domain ontologies, and interoperability of *Applications: Alignment with WordNet:* Not supported. *Licensing:* The OWL version of GFO is released under the modified BSD License and can be downloaded from http://www.onto-med.de/ontologies/gfo.owl. *Documentation:* A detailed information about GFO ontology is available in (Herre et al., 2006). *Alignment with upper and core ontologies:* DOLCE, SOWA, Cyc, SUMO, BWW, UFO

E. PROTON (PROTo ONtology):

Home page: http://proton.semanticweb.org/. *Overview:* PROTON (PROTo ONtology) is a basic upper-level ontology providing coverage of the general concepts necessary for a wide range of tasks, including for semantic annotation, indexing, and retrieval of documents. Extensional semantics approach is followed, it provides minimal and general concepts, and it follows the DOLCE approach of Endurants and Perdurants categorization. *Dimensions and Modularity:* PROTON contains about 300 classes and 100 properties. PROTON is organized in three levels (including four modules). PROTON ontology modules are System, Top, Upper, and Knowledge Management. *Language(s):* A fragment of OWL Lite. *Applications:* PROTON has been used for semantic annotation, and knowledge management systems in legal and telecommunications domain, creating domain ontologies, and Semantic Web Services *Applications: Alignment with WordNet:* Not supported. *Licensing:* The four modules that compose PROTON are freely accessible at http://proton.semanticweb.org/. *Alignment with upper and core ontologies*: DOLCE

F. Sowa's Ontology:

Home page: http://www.jfsowa.com/ontology/. *Overview:* Sowa's ontology's basic categories and distinctions have been derived from a variety of sources in logic, linguistics, philosophy, and artificial intelligence. Sowa presents top-level ontology in the form of a polytree. This ontology was developed in pursuit of a combinatorial approach based on orthogonal distinctions. *Dimensions and Modularity:* The KIF encoding of Sowa's ontology contains about 27 classes, and 33 axioms. Sowa's ontology is not explicitly divided into modules. *Applications:* Sowa's ontology inspired many existing upper ontologies, and is used for evaluation of upper ontologies. *Language(s):* Sowa's ontology uses first-order language with the modal operators "necessary" and "possible". A version written in KIF also exists. *Alignment with*

WordNet: Not supported. *Licensing:* The KIF encoding of Sowa's upper ontology can be freely downloaded from http://suo.ieee.org/SUO/ontologies/Sowa.txt. *Documentation:* A detail description of Sowa ontology is available at (Sowa, 2000). *Alignment with upper and core ontologies:* GFO, DOLCE, SUMO, Cyc.

G. SUMO (Suggested Upper Merged Ontology):

Home page: http://www.ontologyportal.org/. *Overview:* SUMO and its domain ontologies form one of the largest formal public ontology in existence today. SUMO is free and owned by the IEEE. *Dimensions and Modularity:* SUMO contains about 1000 terms, 4000 axioms, 750 rules. These consists of SUMO itself, MILO, and domain ontologies. *Language(s):* SUMO is written in SUO-KIF, OWL. *Applications:* SUMO is used in Semantic web, creating domain ontologies, linguistics, pure representation and reasoning, industrial applications, and research. *Alignment with WordNet:* SUMO has been mapped to all of WordNet 3.0 by hand. *Licensing:* SUMO is free and owned by the IEEE. Ontologies that extend SUMO are available under GNU General Public License. *Documentation:* An introductory tutorial about SUMO is available, and more information is available in (Niles & Pease, 2001). *Alignment with upper and core ontologies:* DOLCE, Sowa, GFO, PSI, and Cyc.

H. UFO (Unified Foundation Ontology):

Home page: UFO is not exclusively available on internet. But it can be found at http://www.inf.ufes.br/~gguizzardi/. *Overview:* UFO is derived from a synthesis of two other foundational ontologies, GFO/GOL and OntoClean/DOLCE. Synthesis is obtained by selecting categories from the union of both category sets, renaming certain terms in order to create a more 'natural' language, and adding some additional categories and corresponding theories. *Dimensions and Modularity:* About 85 terms and 30 relational types. UFO is divided into three incrementally layered compliance sets: UFO-A defines the core of UFO, UFO-B defines terms related to perdurants; and UFO-C defines terms related to the spheres of intentional and social things, including linguistic things. *Language(s):* UFO is described in UML. *Applications:* Conceptual Modelling languages, Agent oriented Engineering Language, Discrete Event Simulation Ontology (DESO), derived from the Unified Foundational. *Alignment with WordNet:* Not supported. *Licensing:* – *Documentation:* A detail description of UFO is available in (Guizzardi & Wagner, 2004). *Alignment with upper and core ontologies:* GFO, DOLCE, BWW.

I. BWW (Bunge–Wand–Weber Ontology)

Home page: BWW (Wand & Weber, 1990) is not exclusively available for download. But it is available in the publications at http://mis.commerce.ubc.ca/members/wand/publications.htm. *Overview:* Scientific realism approach is used, No clear categorization on the basis of Endurants and Perdurants. (BWW in described in set-theoretic language, Text definition). *Dimensions and Modularity:* About 82 classes, and 75 properties. BWW is not divided into modules. *Language(s):* UML, OWL. *Applications:* For creating other Core ontologies, and evaluation of ontologies. *Alignment with WordNet:* Not available. *Licensing:* Freely available. *Alignment with upper and core ontologies:* DOLCE, GFO, UEMO, PSL, UFO

J. OCHRE (Object-Centered High-level Reference):

Home page: OCHERE is not exclusively available for download. However, it is available in (Masolo et al., 2003). *Overview:* Revisionary approach. Ontology of particulars and there is no Universals in it, ontology of perdurants and objects are based on Tropes. *Dimensions and Modularity:* About 41 concept definitions, 52 axioms. OCHRE is not divided into modules. *Language(s):* FOL, KIF. *Applications:* To compare the formal complexity of upper ontologies, creating other core ontologies. *Alignment with WordNet:* Not available. *Licensing:* Freely available. *Alignment with upper and core ontologies:* -

K. PSI (Performance Simulation Initiative):

Home page: https://psi.vcad-vlab.net/edit/documents/PSI-Ontologies/v2.2/. Overview: commits to the ontological choices of DOLCE therefore it is descriptive, possibilistic, multiplicative, and perduranistic, Dimensions and Modularity: PSI upper level ontology has 42 concepts. PSI ontology is divided into PSI upper level ontology and PSI Domain ontologies. Language(s): UML, OWL. Applications: The major use of PSI ontology is knowledge representation and schema for the assessment of industrial engineering design processes. Alignment with WordNet: PSI ontology is aligned with WordNet. Licensing*:* Not freely available. *Alignment with upper and core ontologies:* DOLCE, SUMO.

L. UEMO (Unified Enterprise Modelling Ontology):

Home page: http://www.uemlwiki.org/. *Overview:* UEMO (Anaya et al., 2010) is an on-going attempt to develop theories, technologies and tools for integrated use of enterprise and IS models expressed using different languages. UEMO is based on BWW ontology. *Dimensions and Modularity:* Class taxonomy comprises 35 concepts, property taxonomy comprises 56 concepts, and State and Transformation taxonomies comprise 9 concepts each. Divided in four taxonomies; Class, Property, State, and Transformation. *Language(s):* UML, OWL. *Applications:* Interoperability among enterprise ontologies, UEMO incorporates constructs of UML, Petri nets, GRL, and

KAOS languages. *Alignment with WordNet:* Not supported. *Licensing:* Freely available. *Alignment with upper and core ontologies:* SUMO, BWW, PSL.

M. Chisholm Ontology:

Home page. Chisholm ontology is not exclusively available for download. But can be found in at (Chisholm, 1996). *Overview:* Chisholm ontology is based on common sense realism, primacy of intentional, mereologically essentialism theories. *Dimensions and Modularity:* About 12 categories and not divided into modules. *Language(s):* Text definitions. *Applications:* To evaluate data modelling language. *Alignment with WordNet:* Not available. *Licensing:* Freely available. *Alignment with upper and core ontologies:* BWW.

N. Penman Upper model:

Home page: http://www.fb10.uni-bremen.de/anglistik/langpro/kpml/um89/um89-root.htm. *Overview:* Bound to linguistic form, multiplicative view. *Dimensions and Modularity:* About 225 concepts. Penman upper model is not divided into modules. *Language(s):* NIKL, LOOM. *Applications:* Natural language understanding, creating upper model and ontologies of natural language. *Alignment with WordNet:* not available. *Licensing:* Freely available. *Alignment with upper and core ontologies:* -.

O. PSL (Process Specification Language):

Home page: http://www.mel.nist.gov/psl/. *Overview:* PSL has been designed to facilitate correct and complete exchange of process information among manufacturing systems. PSL has been published as the International Standard ISO 18629. *Dimensions and Modularity:* PSL has four kinds of entities -activities, activity occurrences, time points, and objects. PSL-core is a module which captures the high-level, primitive concepts inherent to process specification. All other modules are built on PSL-core. Implementation *Language(s):* Common Logic. *Applications:* PSL is used in the applications of scheduling, process modelling, process planning, production planning, simulation, project management, workflow, and business process reengineering. *Alignment with WordNet:* Not available. *Licensing:* Freely available. *Alignment with upper and core ontologies:* BWW, UEMO.

7.1.2 Existing upper Ontologies Mappings

Comparing upper ontologies is usually performed by establishing (explicitly or implicitly) some mappings between concepts belonging to distinct ontologies. We define a methodology for collecting existing mappings (Table 1) and analyzing them according to Galois connection. In Sections 7.2 and 7.3, we present in detail these two methodological steps.

Table 7-1. Collected Existing Mappings of Upper ontologies

	WordNet	DOLCE	GFO	SUMO	SOWA	Cyc	BFO	UFO	BWW	UEMO	PDL	Chisholm	PROTON	PSI
(Herre et al., 2006)		×	×											
(Herre et al., 2006)			×		×									
(Masolo et al., 2003)	×	×												
(Bens, 2011)		×	×	×	×	×								
(Opdahl, 2010)				×						×				
(Guizzardi & Wagner, 2004)		×	×					×	×					
(Grenon, 2003)		×					×							
(Colomb, 2002)		×							×					
(Lin, 2008)									×	×	×			
(Davies et al., 2003)									×			×		
(Terziev et al., 2005)		×											×	
(Keberle et al., 2007)	×	×		×										×

7.2 Proposed method for Dealing with Upper Ontology Mappings

For check the compatibility of existing upper ontology mappings, we have proposed the methods consist of following main steps.

Collecting existing upper ontology mappings

We collect existing upper ontology mappings from research reports, research papers and websites. Table 7-1 shows the collected existing mappings of Upper ontologies. First Column of the table describes the source from where ontology mappings between upper ontologies or WordNet is collected, while each upper ontology contained in the corresponding source is marked by cross sign at the intersection of that upper ontology and corresponding source.

Harmonizing Collected Upper Ontology mappings

Upper ontology mapping aims at establishing some relationships between concepts belonging to distinct ontologies. In most of the cases found in the literature, these relationships are not qualified, i.e., it remains unclear if the authors consider them as equivalence, subsumption, similarity and so on. Our proposal based on Galois connections does not require any information about the type of mapping relation such as equivalence, similarity and so on.

In some cases, authors do not specify any mapping for some concepts. In our proposal, we consider that the authors have tried to map all concepts, except if otherwise stated. This means that if one concept is not mapped to another concept, we consider that the authors have not been able to find a suitable mapping for those concepts.

In our proposal, we have also taken care of the "format" that the authors use for representing mappings. Indeed, mappings are implicitly explained in text, or provided by two columns tables when involving only two ontologies, or provided by multicolumn tables when involving several ontologies. Furthermore, in some cases involving only two ontologies, authors explicitly provide a two distinct mappings between the couple of ontologies for being more precise. For mappings implicitly explained in text, we have explicitly built a two columns table for each couple of ontologies. For mappings directly provided as two column tables, we have just considered the same tables.

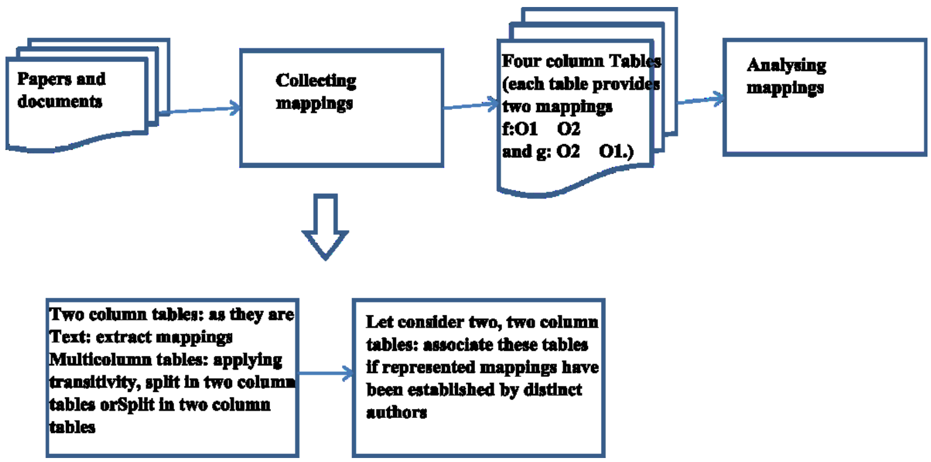

Figure 7-1. Proposed approach of Analyzing mappings between Upper ontologies for evaluating their compatibility

When dealing with a multicolumn table involving more than two ontologies, we have applied transitivity by assuming that, without specific assumptions provided by authors, the authors have used the same types of mapping (i.e., the same equivalence, subsumption or similarity across the table) for all ontologies. This means that if x is a concept in Ontology A, y is a concept in Ontology B, z is a concept in Ontology C, and x, y, z are shown on the same table row, a mapping exists between x and y, a mapping exists between y and z, and by transitivity a mapping exists between x and z.

However, in some cases, the authors explicitly state that multicolumn table is used for convenience for representing a set of mappings between two ontologies. An extract of such a multicolumn table is presented in Figure 7-1. In this specific case, authors mapp "Instance" to "Particular" in DOLCE, "Individual" in GFO, "Physical" in Sowa's ontology, "Physical" in SUMO. However, this table can also be interpreted as "Particular" in DOLCE is mapped to

"Individual" in GFO etc. The table therefore interpreted as the set of mappings: Ontology4 to DOLCE, Ontology4 to GFO, DOCLE to GFO, GFO to Sowa etc. by using transitivity.

Ontology4	DOLCE	GFO	Sowa	SUMO	Cyc
^Instance	^Particular	^Individual	^Physical	^Physical	

Figure 7-2. Mappings represented in (Bens, 2011)

In (Herre et al., 2006) mappings between GFO and DOLCE, and between GFO and Sowa's ontology are presented distinctly. We therefore do not apply transitivity to build a mapping between DOLCE and Sowa's ontology. We restrict ourselves by assuming that authors explicitly separate these mappings and the two distinct mappings cannot be assumed established according to the same relationship (of equivalence, subsumption, similarity). Figure 7-2 shows the one of the format used for mapping in the sources from where the existing upper ontology mappings are collected.

Finally, we have found some mappings referring to concepts that do not really exist in the involved ontologies: we have therefore removed any mapping concerning concepts that do not exist in one of the involved ontologies (non-existing concepts may raise because of usage of distinct versions of ontologies or because to not up-to-date publications).

Summarizing Collected mappings in tabular Form

To arrange the collected mappings, we have used several four column tables providing one mapping from some authors and an inverse mapping from some other authors. Extract of those tables is shown in Figure 7-3. Each of these four column tables provides possibly distinct mappings between distinct ontologies, and can be used for the further analysis step presented in Section 7.3 below.

DOLCE mapping by (Colomb, 2002)	BWW mapping by (Colomb, 2002)	DOLCE mapping by (Guizzardi & Wagner, 2004)	BWW mapping by (Guizzardi & Wagner, 2004)
Entity	Thing	Entity	Thing
		System	Endurant

Figure 7-3. Arrange Collected mappings in tabular form

7.3 Checking the Compatibility of Ontology Mappings

For checking the compatibility of Ontology Mappings, we need to select one of the appropriate lattice listed in the Chapter 5. The choice of lattice depends on the contents (taxonomy, Extended taxonomy containing complex artifacts, inferences and others) of the ontology we have. Mapping space is also built based on the contents of ontology.

After that, we apply procedure of checking compatibility of ontology mappings on Lattices. Based on the existence of Galois connection after performing all of the above steps, we declare whether given ontology mappings are compatible or not.

In the next section, we conduct a case study of checking compatibility of ontology mappings.

7.4 Case Study of DOLCE and GFO ontology mappings

The purpose of this case study is to demonstrate the notion of compatibility of ontology mappings. We systematically check the compatibility of upper ontology mappings. To achieve this objective, We perform following three main tasks.

1. First, we select existing upper ontology mappings. We also propose an approach of dealing upper ontology mappings. The process is conducted into three steps. First, by collecting existing upper ontology mappings. Second, by harmonizing these mappings which are available in different formats. Lastly, we summarize existing mappings in tabular form.
2. Second, we deal with partial mappings and complex artifacts involved in mappings.
3. Third, we check whether or not there exists a Galois connection between mappings.

We demonstrate this on mappings of DOLCE and GFO upper ontologies.

The objective of this case study is to evaluate how our notion of compatibility of ontology mapping identifies compatible mappings or incompatible mappings. In particular, we want to know that when given ontology mappings are incompatible then which correspondences cause incompatibility.

7.4.1 Case Study Design

We design a Case study of checking compatibility of upper ontology mappings and for that we present here the case of DOLCE and GFO ontology mappings in detail, while we summarize the result of other ontology mappings in tabular form.

Harmonizing and Completing mappings

We select the case of DOLCE and GFO to highlight how to deal with when we have a *directional semantic mapping* and a *simple mapping*. We selected a mapping from GFO to DOLCE which we have extracted from (Herre et al., 2006) and it is a *directional semantic mapping* since there is a mapping from DOLCE to GFO (Herre et al., 2006) and it is different from GFO to DOLCE. While other mapping we selected from (Bens, 2011) and it is a *simple mapping,* i.e., it can be treated as mapping from DOLCE to GFO and/or GFO to DOLCE. Here we are treating this mapping as from DOLCE to GFO.

In both these mappings, mapping relation is not explicitly shown. For mapping form GFO to DOLCE established by (Herre et al., 2006), authors differentiate between equivalence relation and subset relation while establishing mappings. In these mappings, for correspondences having equivalence relation an artifact or complex artifact (artifact made from artifacts of source ontology by using union and/or intersection operator) is presented in the cell of second column. While for correspondences which are not completely mapped it is clear that they are not in equivalence relation but it is not clear that which artifact is more general and which is more specific such correspondences are presented in parenthesis (). Artifacts presented in parenthesis refers that artifacts involved in DOLCE ontology are not completely mapped and this means that artifacts of GFO ontology are more general than their counterpart in DOCLE ontology presented in parenthesis. Our approach is independent of any relation between mapped artifacts, i.e., any relation such as equality, subsumption, similarity, etc. are abstracted.

Ontology correspondences consisting of artifacts that are not direct or indirect part of source ontology(ies) are not considered for checking compatibility and they are pruned in this work.

For mapping from DOLCE to GFO established by (Bens, 2011) have equivalence relation between mapped artifacts.

For artifacts of source ontologies which do not have any mapped artifacts they are presented by (Herre et al., 2006) as – and by (Bens, 2011) as empty cell. We complete these mappings by mapping such artifacts to Top ⊤.

Choice of Lattice

In the given ontology mappings, we do not have mappings of all signature (artifacts) of source ontologies and some of these artifacts are mapped to complex artifacts of other ontology. We need to make complex artifacts (artifacts that built by using combining artifacts of source ontologies with the help of union and/or intersection operator,) that are not part of original taxonomy of source ontology a part of our lattice to check existence of Galois connection. Therefore, our appropriate choice of lattice is *Lattice of Extended taxonomy* containing basic taxonomy and only those complex artifacts which are used in the mapping.

Here, we present taxonomies of DOLCE and GFO ontologies in Figure 7-4 and Figure 7-5 respectively. While lattices of extended taxonomy are shown in Figure 7-6 and Figure 7-7 without showing top (⊤) and bottom (⊥).

Incompatible and Weak Incompatible mappings

We distinguish two kinds of incompatible mappings and they are

(a) Incompatibility arises due to the original correspondences established by authors

(b) Incompatibility arises due to those correspondences completed by us such as mapping to ⊤

We named (a) as *weak incompatible* mappings and while (b) as incompatible mapping. The reason of making this distinction is that if someone can find proper mappings as compared to our choice of mapping to ⊤ then these weak incompatible mappings may not remain incompatible, however we have to check compatibility or incompatibility again in this case.

7.4.2 Compatibility of DOLCE and GFO ontology mappings

The taxonomy of DOLCE ontology is presented in Figure 7-4 and taxonomy of GFO ontology is presented in Figure 7-5. It is important to note that these taxonomies only present basic categories of these ontologies and do not cover all the axioms. Mappings are presented in Table 7-2 and Table 7-3.

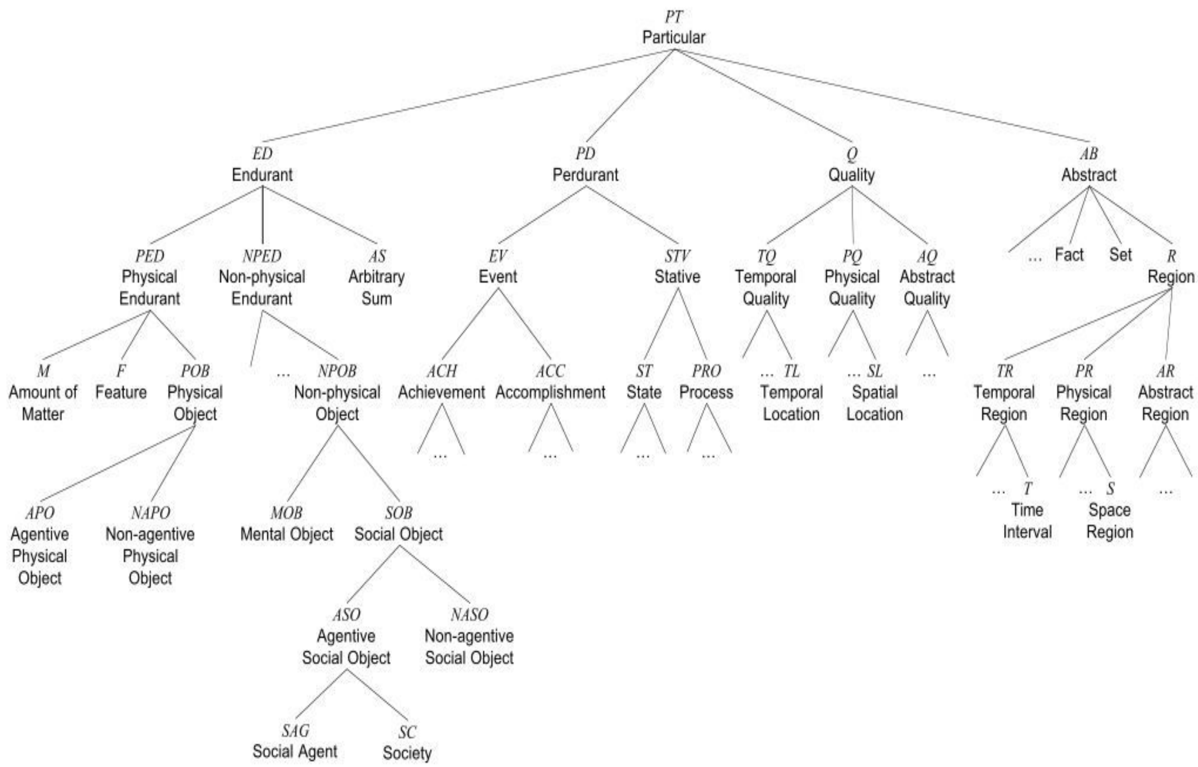

Figure 7-4. DOLCE Ontology

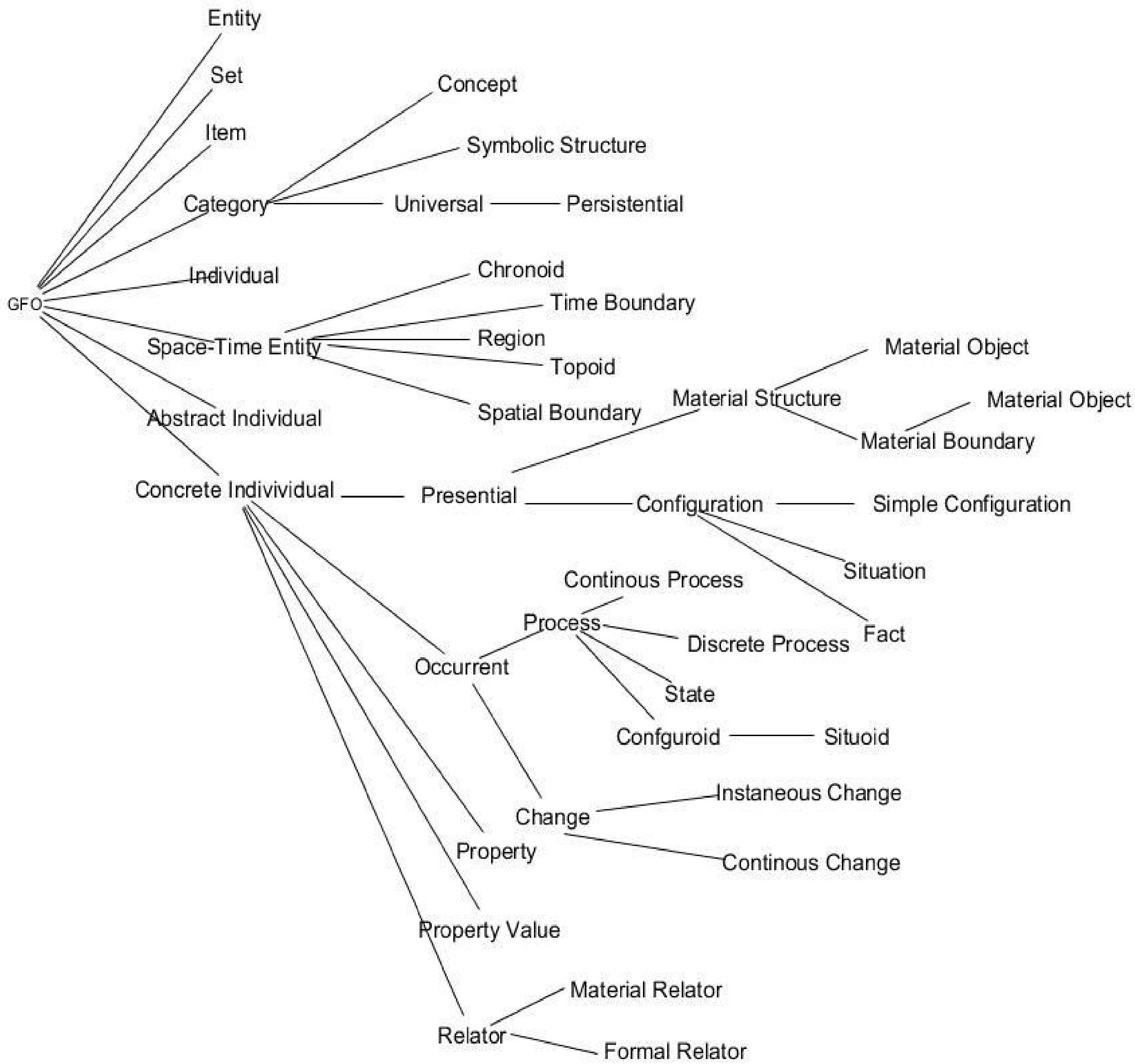

Figure 7-5. GFO Ontology

Available mappings

Table 7-2. Mapping from GFO to DOLCE (Herre et al., 2006)

GFO	DOLCE
Entity	(Entity)
Set	(Set)
Item	–
Category	–
Universal	–

GFO	DOLCE
Persistant	(Endurant)
Concept	–
Symbolic Structure	–
Individual	Particular
Space-Time Entity	Temporal Region ∪ Space Region
Chronoid	Time Interval
Time Boundary	–
Region	Space Region
Topoid	–
Spatial Boundary	–
Abstract Individual	Abstract
Concrete Individual	Endurant ∪ Perdurant ∪ Quality
Presential	(Endurant)
Material Structure	Physical Endurant
Material Object	Physical Object
Material Boundary	(Feature)
Configuration	–
Simple Configuration	–
Situation	–
Fact	Fact
Occurrent	(Perdurant)
Process	Stative
Continuous Process	–
Discrete Process	–
State	(State)
Configuroid	–
Situoid	–
Change	(Event)
Instantaneous Change –	–
Continuous Change	–
Property	Quality
Property Value	Quale
Relator	–
Material Relator	–
Formal Relator	–

Table 7-3. Mapping between DOLCE and GFO (Bens, 2011)

DOLCE	GFO
^Accomplishment	
	^MatStructureOrProcess
	^PropertyValue
	^Category
	^Configuration
	^PresentialOrPersistant
^Entity	
^Event	^DiscreteProcess
^Particular	^Individual
^SpacialLocation	^Region
^Perdurant	^Occurent
^TimeInterval	
^SocialAgent	
^Process	^Process
	^Instantiation
^Quality	
^AmountOfMatter	
	^Mediating
	>ProcessCategory
	^MaterialStructure
	^State

Checking Compatibility and Incompatibility of ontology mappings

We name these mappings as $M_{GFO \to DOLCE}$ for mapping presented in Table 7-2 and $M_{DOLCE \to GFO}$ for mapping presented in Table 7-3.

As there are two kinds of Galois connections: isotone and antitone, so we need to check whether these mappings are isotone or antitone.

For $M_{GFO \to DOLCE}$, we found that these mappings are neither isotone nor antitone as

In GFO, $Fact \preccurlyeq Presential$ while in $M_{GFO \to DOLCE}$ $Fact$ is mapped to $Fact$ and $Presential$ is mapped to $Endurant$, whereas, in DOLCE $Fact \not\preccurlyeq Endurant$ and $Fact \not\succcurlyeq Endurant$

While $M_{DOLCE \to GFO}$ is also neither isotone nor antitone mapping. As but they are mapped to $Occurent \perp Individual$.

In DOLCE, $Perdurant \preccurlyeq Particular$ while in $M_{GFO \to DOLCE}$ $Perdurant$ is mapped to $Occurent$ while $Particular$ is mapped to $Individual$, whereas, in GFO $Occurent \not\preccurlyeq Individual$and $Occurent \not\succcurlyeq Individual$.

However, we ignore that mappings are neither isotone nor anti-tone for both mapping $M_{DOLCE \to GFO}$ and $M_{DOLCE \to GFO}$ to find out that which ontology mapping correspondences are not respecting the conditions of Galois connection.

In mapping $M_{DOLCE \to GFO}$ and $M_{GFO \to DOLCE}$, only DOLCE ontology has complex artifact: Temporal Region ∪ Space Region and Endurant ∪ Perdurant ∪ Quality.

We chose lattice of extended taxonomy to check the compatibility of ontology mappings. We arrange the complex artifacts at proper order in the lattice and lattice of DOLCE ontology and and lattice of GFO ontology are respectivley shown in Figure 7-6 and Figure 7-7. Added complex artifacts are shown in blue font color. While other complex artifacts are not shown explicitly in the lattice. It should be noted that we have not shown top (⊤) and bottom (⊥).

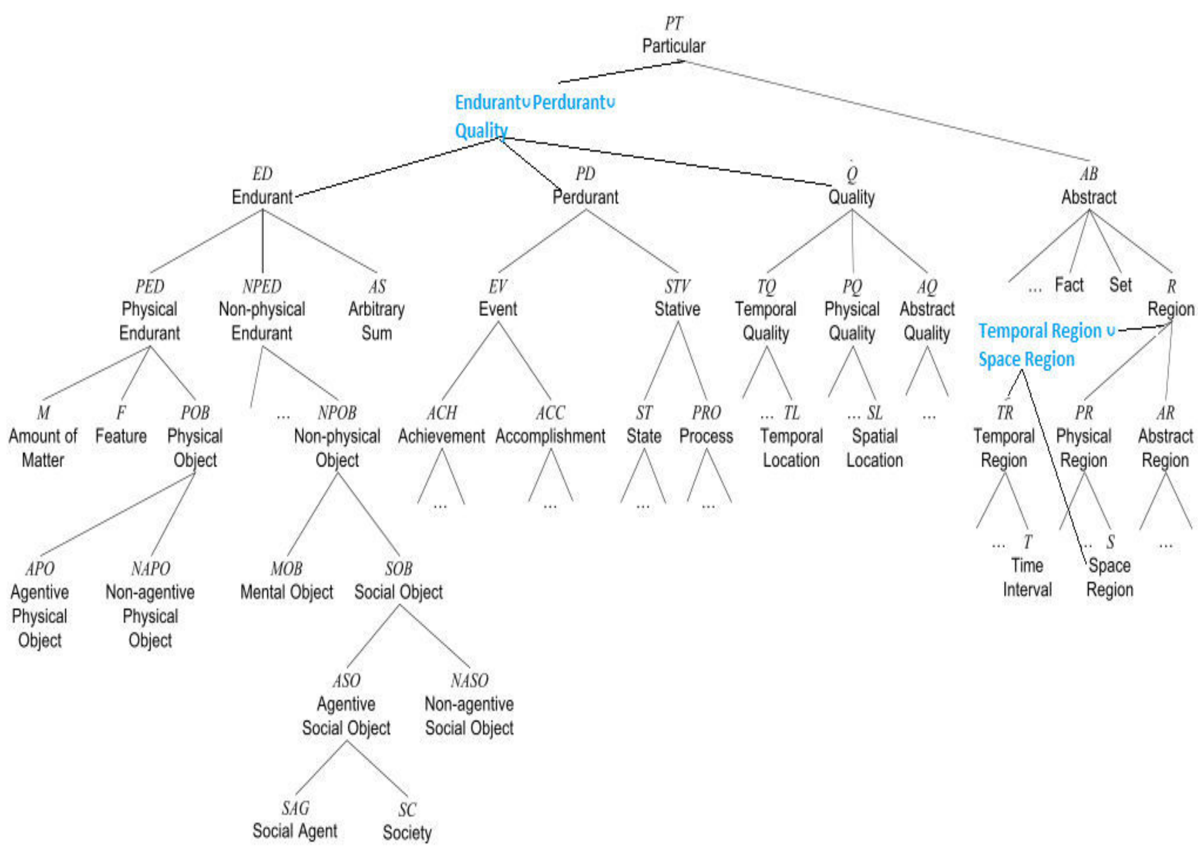

Figure 7-6. Lattice of DOLCE ontology with complex artifacts

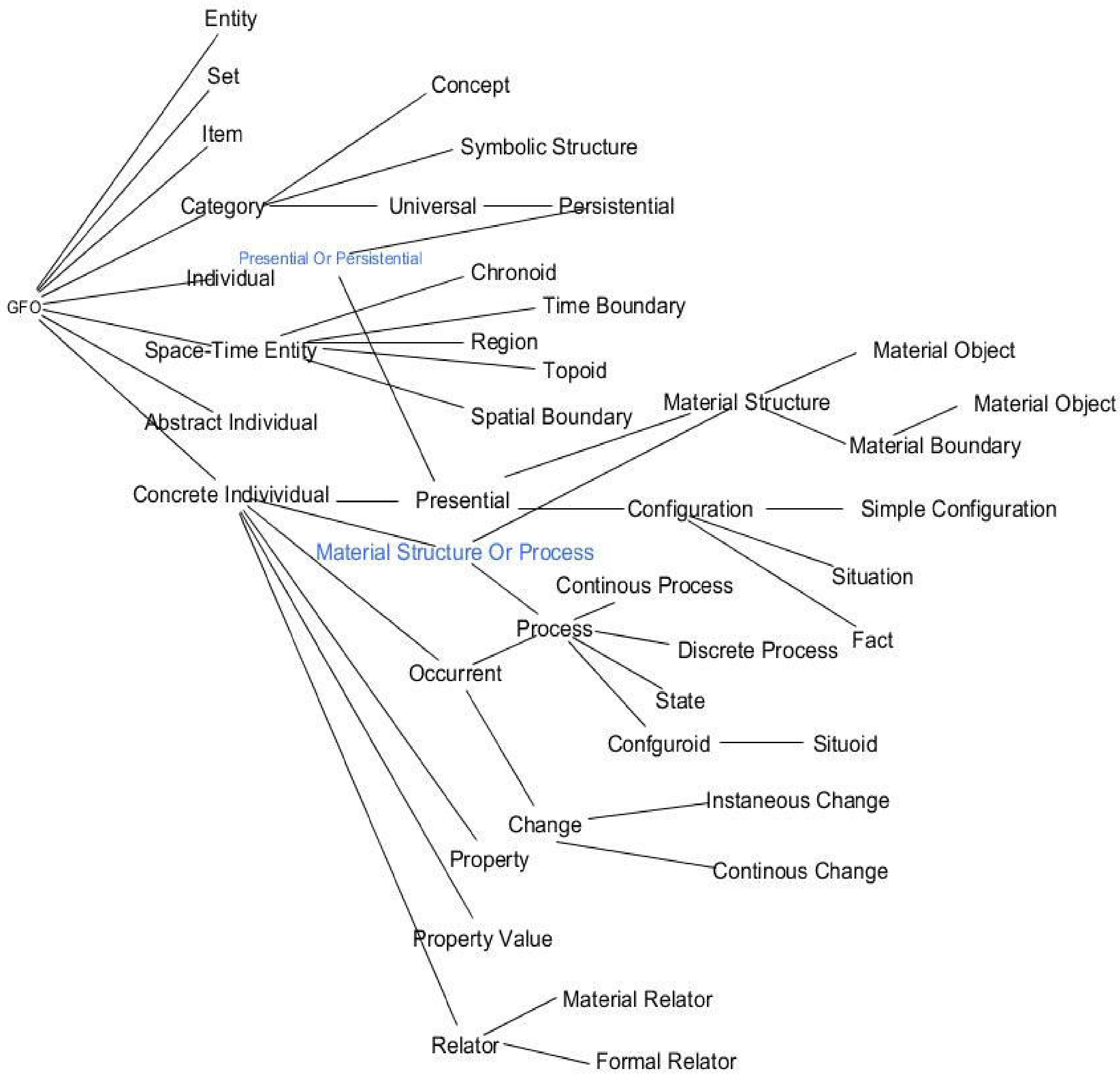

Figure 7-7. Lattice of GFO ontology with complex artifacts

Now we are interested in evaluating the compatibility and incompatibility of ontology mappings.

For that we check whether or not $\alpha \circ \gamma$ and/or $\gamma \circ \alpha$ respect, reverse or violate order in each lattice for each correspondence. We will check whether $\alpha \circ \gamma \leq x$ and $x \leq \gamma \circ \alpha$. We use α $M_{GFO \to DOLCE}$ for and γ for $M_{DOLCE \to GFO}$

$$\alpha(\{Individual\})_{GFO} = (\{Particular\})_{DOLCE}$$

$$\gamma(\{Particular\})_{DOLCE} = (\{Individual\})_{GFO}$$

$$\alpha \circ \gamma(\{Particular\})_{DOLCE} = a(\{Individual\})_{GFO} = (\{Particular)\}_{DOLCE} \preccurlyeq (\{Particular\})_{DOLCE}$$

$$\gamma \circ \alpha(\{Individual\})_{GFO} = \gamma(\{Particular\})_{DOLCE} = (\{Particular\})_{GFO} \succcurlyeq (\{Particular\})_{GFO}$$

$$\alpha(\{Occurrent\})_{GFO} = (\{Perdurant\})_{DOLCE}$$

$$\gamma(Perdurant)_{DOLCE} = (\{Occurrent\})_{GFO}$$

$$\alpha \circ \gamma(Perdurant)_{DOLCE} = a(Occurrent)_{GFO} = Perdurant_{DOLCE} \preccurlyeq Perdurant_{DOLCE}$$

$$\alpha \circ \gamma(\{Perdurant\})_{DOLCE} = a(\{Occurrent\})_{GFO} = (\{Perdurant)\}_{DOLCE} \preccurlyeq (\{Perdurant\})_{DOLCE}$$

$$\gamma \circ \alpha(\{Occurrent\})_{GFO} = \gamma(\{Perdurant\})_{DOLCE} = (\{Occurrentr\})_{GFO} \succcurlyeq (\{Occurrentr\})_{GFO}$$

$$\alpha(\{DiscreteProcess\})_{GFO} = (\{\top\})_{DOLCE}$$

$$\gamma(\{Event\})_{DOLCE} = (\{DiscreteProcess\})_{GFO}$$

$$\gamma\left(\{\overset{*}{\top}\}\right)_{DOLCE} = (\{\top\})_{GFO}$$

$\alpha \circ \gamma(\{Event\})_{DOLCE} = a(\{DiscreteProcess\})_{GFO} = (\{\overset{*}{\top}\})_{DOLCE} \succcurlyeq (\{Event\})$. Not respecting the condition of Galois connection so it is weakly incompatible case.

$$\gamma \circ \alpha(\{DiscreteProces\})_{GFO} = \gamma(\{\overset{*}{\top}\})_{DOLCE} = (\{\top\})_{GFO} \preccurlyeq (\{DiscreteProcess\})_{GFO}$$

Not respecting the condition of Galois connection So it is weakly incompatible case.

$$\alpha(\{Change\})_{GFO} = (\{Event\})_{DOLCE}$$

$$\gamma(\{Event\})_{DOLCE} = (\{DiscreteProces\})_{GFO}$$

$\alpha \circ \gamma(\{Event\})_{DOLCE} = a(\{DiscreteProcess\})_{GFO} = (\{\top\})_{DOLCE} \succcurlyeq (\{Event\})_{DOLCE}$.

so contradict Galois connection condition $\alpha \circ \gamma(x) \leq x$.

$$\gamma \circ \alpha(\{Change\})_{GFO} = \gamma(\{Event\})_{DOLCE} = (\{DiscreteProcess\})_{GFO} \perp (\{Change\})_{GFO}$$

Hence, this shows that these correspondences are incompatible since they are in unordered relation.

$$\alpha(\{Process\})_{GFO} = (\{Stative\})_{DOLCE}$$

$$\gamma(\{Process\})_{DOLCE} = (\{Process\})_{GFO}$$

$$\gamma(\{Stative\})_{DOLCE} = (\{\top\})_{GFO}$$

$$\gamma \circ \alpha(\{Process\})_{GFO} = \gamma(\{Stative\})_{DOLCE} = \left(\left\{\overset{*}{\top}\right\}\right)_{GFO} \geq (\{Process\})_{GFO}$$

$\alpha \circ \gamma(Process)_{DOLCE} = a(\{Process\})_{GFO} = (\{Stative\})_{DOLCE} \nleq (\{Process\})_{DOLCE}$; and this is not true in DOLCE ontology where $Stative_{DOLCE} \geq Process_{DOLCE}$. Therefore, this violates monotone Galois connection condition and hence, mappings are incompatible.

$$\alpha(\{Region\})_{GFO} = (\{Space\ Region\})_{DOLCE}$$

$$\gamma(\{Spatial\ Location\})_{DOLCE} = (\{Region\})_{GFO}$$

$$\alpha \circ \gamma(\{Spatial\ Location\})_{DOLCE} = a(\{Region\})_{GFO} = (\{Space\ Region\})_{DOLCE} \\ \perp (\{Spatial\ Location\})_{DOLCE}$$

$$\gamma \circ \alpha(\{Region\})_{GFO} = \gamma(\{Space\ Region\})_{DOLCE} = (\{\overset{*}{\top}\})_{GFO} \geq (\{Process\})_{GFO}$$

This is also incompatible correspondences due to $\alpha \circ \gamma(Spatial\ Location)_{DOLCE}$.

Summary:

We identify incompatible correspondences as

$(Change_{GFO}, Event_{DOLCE})$ and $(Event_{DOLCE}, Discrete\ Process_{GFO})$

$(Region_{GFO}, SpaceRegion_{DOLCE})$ and $(Spatial\ Location_{DOLCE}, Region_{GFO})$

and

$(Process_{DOLCE}, Process_{GFO})$ and $(Process_{GFO}, Stative_{DOLCE})$.

We have shown that *(DiscreteProcess, ⊤), (Event, DiscreteProcess) is weakly incompatible.*

Other, Weak incompatible mapping correspondences are

(Quality, ⊤) and (Property, Quality)

(Material Structure, Physical Endurant) and (Physical Endurant, ⊤)

(Presential, Endurant) and (Endurant, ⊤)

(Persistant, Endurant) and (Endurant, ⊤)

(Chronoid , Time interval) and (Time interval , ⊤)

(State, State) and (State,⊤)

7.5 Synthesis

Although we have correspondences in which some of the artifacts of one source ontology are unmapped, but it is possible to map such artifacts and for that our choice of mapping for such artifacts is to map to top ⊤. However, if this makes ontology mappings incompatible then we treat them as *weakly incompatible*. The situation is different when we have correspondences in which some of the artifacts of one source ontology are unmapped, then in such cases mappings are compatible. The reason is that we assume that there is no contradiction and our choice is right one. It is also possible that instead of mapping to top ⊤, someone can find right mapping for unmapped artifacts other than ⊤, and mappings are compatible, then this choice in this situation is considered as right one.

We have shown in Figure 7-6 how to add complex artifact in lattice. This is simple task just add such complex artifacts in correct position that respect the existing axioms of source ontology.

After finding ontology mappings are incompatible even just finding this fact after checking the first correspondence of each ontology mapping, we continue to check the remaining correspondences and check whether or not these correspondences also cause contradiction. After that, we list all correspondences that cause contradiction. This list tells that they are responsible in incompatibility of ontology mappings. It should be noted that even one of the correspondence of each ontology mapping is sufficient for making both ontology mappings as incompatible, the complete list is collected so that if users are interested in making them compatible she has to modify some or all these correspondences that are causing incompatibility.

From this case study and done similar case studies on other available ontology mappings, we have identified 4 cases of compatible and incompatible ontology mappings.

The objective of this step is to verify if the couples of collected mappings respect Galois connection conditions. Compatible mappings have been further distinguished in weak compatible mappings and compatible mappings. The former rises whenever Galois connection conditions are trivially respected because some concepts are mapped to ⊥.

Hereinafter, the reader can find in several situations, how compatibilities and incompatibilities have been established and this is described in terms of Compatibilities and incompatibilities of ontology artifacts.

1. **Trivial compatibility case**.

$$\alpha\left((\{Category\})\right)_{GFO} = (\{Abstract\})_{Sowa}$$

$$\gamma(\{Abstract\})_{Sowa} = (\{Category\})_{GFO}$$

$$\alpha \circ \gamma\ (\{Abstract\})_{Sowa} = \alpha\ (\{Category\})_{GFO} = (\{Abstract\})_{Sowa}$$

$$\gamma \circ \alpha(\{Category\})_{GFO} = \gamma(\{Abstract\})_{Sowa} = (\{Category\})_{GFO}$$

The situation above corresponds to (one type of) Galois connection for specific concepts.

2. Compatibility case

A concept X in one ontology is mapped to some concept Y in other ontology, but in another mapping performed by some other author(s), X is mapped to Z which is subsumed by Y, i.e., Z ⊆ Y. For instance,

$$\alpha(\{Process\})_{GFO} = (\{Stative\})_{DOLCE}$$

$$\gamma(\{Process\})_{DOLCE} = (\{Process\})_{GFO}$$

$$\alpha \circ \gamma\ (\{Process\})_{DOLCE} = \alpha(\{Process\})_{GFO} = (\{Stative\})_{DOLCE} \subseteq (\{Process\})_{DOLCE}$$

Stative is more general than Process, and Process is immediate descendant of Stative.

$$\gamma \circ \alpha(\{Process\})_{GFO} = \gamma(\{Stative\})_{DOLCE} = (\{Process\})_{GFO}$$

which corresponds to a reverse ordering Galois connection.

3. Incompatibility case.

A concept X in one ontology is mapped to some concept Y in other ontology, but X is mapped to Z, while Y and Z are not ordered.

$$\alpha(\{Region\})_{GFO} = (\{Space\ Region\})_{DOLCE}$$

$$\gamma(\{Spatial\ Location\})_{DOLCE} = (\{Region\})_{GFO}$$

$$\begin{aligned}\alpha \circ \gamma\ (\{Spatial\ Location\})_{DOLCE} &= \alpha(Region)_{GFO} \\ &= (\{Space\ Region\})_{DOLCE} \perp (\{Spatial\ Location\})_{DOLCE}\end{aligned}$$

$$\gamma \circ \alpha(\{Region\})_{GFO} = \gamma(\{Space\ Region\})_{DOLCE} = (\perp)_{GFO} = \perp \leq (\{Region\})_{GFO}$$

Spatial Location subsumes Physical Quality and Space Region subsumes Abstract Region respectively and Physical Quality and Abstract Region are not ordered. This situation therefore corresponds to neither order reversing not order preserving Galois connection, rising in incompatibility.

This case makes ontology mappings incompatible, user may stop here if she just want to know about the compatibility and incompatibility of ontology mappings and is not interested in finding which correspondence(s) is(are) causing incompatibility.

As, here ontology mappings are embedded in lattice mappings, indeed they are the same, so this is incompatibility for every procedure according to the definition of incompatibility.

4. Weak incompatibility case

$$\alpha(\{MaterialStructure\})_{GFO} = (\perp)_{DOLCE}$$

$$\gamma(\{PhysicalEndurant\})_{DOLCE} = (\{MaterialStructure\})_{GFO}$$

$$\alpha \circ \gamma(\{PhysicalEndurant\})_{DOLCE} = \alpha(\{MaterialStructure\})_{GFO} = (\perp)_{DOLCE} \sqsubseteq (PhysicalEndurant)_{DOLCE}$$

$$\gamma \circ \alpha(\{MaterialStructure\})_{GFO} = \gamma(\perp)_{DOLCE} = (\perp)GFO \sqsubseteq (\{MaterialStructure\})_{GFO}$$

However, $\gamma \circ \alpha$ should be extensive $\gamma \circ \alpha(x) \geq x$. Therefore, conditions of Galois connection are not respected.

It should be noted that the situation does not much change if instead of ⊥, the ontology root, ⊤, would have been used. Indeed:

$$\alpha(\{MaterialStructure\})_{GFO} = (\top)_{DOLCE}$$

$$\gamma(\{MaterialStructure\})_{GFO} = (\{PhysicalEndurant\})_{DOLCE}$$

$$\alpha \circ \gamma\ (\{PhysicalEndurant\})_{DOLCE} = \alpha(\{MaterialStructure\})_{GFO} = (\top)_{DOLCE} \geq (\{PhysicalEndurant\})_{DOLCE}$$

$$\gamma \circ \alpha(\{MaterialStructure\})_{GFO} = \gamma\ (\top)_{DOLCE} = \top \geq (\{PhysicalEndurant\})_{DOLCE}$$

$\alpha \circ \gamma$ should be reductive. Therefore, conditions of Galois connection are not respected.

This means that the two mappings are incompatible. Since, this incompatibility comes due to ⊤ or ⊥, we differentiate it with other Incompatibilities and we name it as weak incompatibility.

In this section, we present a synthesis of the result that we have obtained. We check the compatibility of available mappings (shown in Table 7-1) whenever distinct authors establish the two required mappings α and γ (however, it is possible to apply the methodology to mappings supplied by same authors).

In the remainder, we only focus on the following remarkable couples of ontologies:

DOLCE vs GFO; GFO vs Sowa's ontology; DOLCE vs SUMO; DOLCE vs WordNET; DOLCE vs BWW.

The detailed results are provided in Table 7-4, Table 7-5, Table 7-6, Table 7-7 and Table 7-8. Incompatibilities found due to mapping to ⊥ or ⊤ are provided in italic (because they are "weak incompatibilities" due to our interpretation of partial mappings). Table is interpreted as a compatible ontology mapping couples and incompatible ontology mapping couples. A compatible mapping couple observes the properties of Galois connection and Incompatible mapping couples does not respect the properties of Galois connection. $M_{GFO \rightarrow DOLCE}$ for and γ for $M_{DOLCE \rightarrow GFO}$

Table 7-4. Artifacts-based Compatibilities and Incompatibilities of DOLCE and GFO

Mappings couple α (Herre et al., 2006), γ (Bens, 2011)			
Compatibilities		**Incompatibilities**	
$\boldsymbol{M_{GFO\rightarrow DOLCE}}$ $\boldsymbol{(\alpha)}$	$\boldsymbol{M_{DOLCE\rightarrow GFO}}$ $\boldsymbol{(\gamma)}$	$\boldsymbol{M_{GFO\rightarrow DOLCE}}$ $\boldsymbol{(\alpha)}$	$\boldsymbol{M_{DOLCE\rightarrow GFO}}$ $\boldsymbol{(\gamma)}$
(Individual, Particular)	(Particular, Individual)	(Change, Event)	(Event, DiscreteProcess)
(Occurrent, Perdurant)	(Perdurant , Occurrent)	(Process, Stative)	(Process,Process), (Stative, ⊤)
		(Region, SpaceRegion)	(SpacialLocation, Region)
		(DiscreteProcess, ⊤)	*(Event, DiscreteProcess)*
		(Quality, ⊤)	*(Property, Quality)*
		(Material Structure, Physical Endurant)	*(Physical Endurant, ⊤)*
		(Presential, Endurant)	*(Endurant, ⊤)*
		(Persistant, Endurant)	*(Endurant, ⊤)*
		(Chronoid , Time interval)	*(Time interval , Chronoid)*
		(State, State)	*(State,⊤)*
Mappings couple α (Guizzardi & Wagner, 2004), γ (Bens, 2011)			
Compatibilities		**Incompatibilities**	
$\boldsymbol{M_{GFO\rightarrow DOLCE}}$ $\boldsymbol{(\alpha)}$	$\boldsymbol{M_{DOLCE\rightarrow GFO}}$ $\boldsymbol{(\gamma)}$	$\boldsymbol{M_{GFO\rightarrow DOLCE}}$ $\boldsymbol{(\alpha)}$	$\boldsymbol{M_{DOLCE\rightarrow GFO}}$ $\boldsymbol{(\gamma)}$
(Individual, Particular)	(Particular, Individual)	*(Region, ⊤)*	*(SpacialLocation, Region)*
(Property, Quality)	(Quality, Quality)	*(DiscreteProcess, ⊤)*	*(Event, DiscreteProcess)*
(Entity, Entity)	(Entity, Entity)	*(Chronoid , ⊤)*	*(Time interval , Chronoid)*

Table 7-5. Artifacts-based Compatibilities and Incompatibilities of GFO and Sowa's ontology

Mappings couple α (Herre et al., 2006), γ (Bens, 2011)			
Compatibilities		**Incompatibilities**	
$\boldsymbol{M_{SOWA \to GFO}}$ $\boldsymbol{(\alpha)}$	$\boldsymbol{M_{GFO \to SOWA}}$ $\boldsymbol{(\gamma)}$	$\boldsymbol{M_{SOWA \to GFO}}$ $\boldsymbol{(\alpha)}$	$\boldsymbol{M_{GFO \to SOWA}}$ $\boldsymbol{(\gamma)}$
(Physical, Individual)	(Individual, Physical)	(Entity, Entity)	(Entity, ⊤)
(Continuant, {Presential ∪ Persitant})	({PresentialOrPersistent}, Continuant)	*(Abstract, Category)*	*(Abstract, Abstract)*
(Occurrent, Occurrent)	(Occurrent, Occurrent)	*(Nexus, Mediating)*	*(Mediating,* ⊤*)*
(Process, Process)	(Process, Process)		
(Actuality, {Material Structure ∪ Process})	({MaterialStructureOrProcess}, Actuality)		
(Proposition, Instantiation)	(Instantiation, Proposition)		
(Script, Category of Process)	(Category of Process, Script)		
(Situation, MaterialStructure)	(MaterialStructure, Situation)		

Table 7-6. Artifacts-based Compatibilities and Incompatibilities of BWW and DOLCE

Mappings couple α (Colomb, 2002), γ (Guizzardi & Wagner, 2004)			
Compatibilities		**Incompatibilities**	
$\boldsymbol{M_{BWW \to DOLCE}}$ $\boldsymbol{(\alpha)}$	$\boldsymbol{M_{DOLCE \to BWW}}$ $\boldsymbol{(\gamma)}$	$\boldsymbol{M_{BWW \to DOLCE}}$ $\boldsymbol{(\alpha)}$	$\boldsymbol{M_{DOLCE \to BWW}}$ $\boldsymbol{(\gamma)}$
(Thing, Entity)	(Entity, Thing)	*(System, Endurant)*	*(Endurant,* ⊤*)*
(Intrinsic property, Quality) Manual Property}	(Quality, Property)		
(Manual Property, Quality)	(Quality, Property)		

Table 7-7. Artifacts-based Compatibilities and Incompatibilities of SUMO and DOLCE

Mappings couple α (Keberle et al., 2007), γ (Bens, 2011)			
Compatibilities		**Incompatibilities**	
$M_{SUMO\rightarrow DOLCE}$ (α)	$M_{DOLCE\rightarrow SUMO}$ (γ)	$M_{SUMO\rightarrow DOLCE}$ (α)	$M_{DOLCE\rightarrow SUMO}$ (γ)
(Cooperation, IntentionalProcess)	(Process, Process)	Intentional Process, Parameter},	(Process, Process)
(Attribute, Quality)	{Quality, Physical Quality}	(Intentional Process, pla:elementary task)	(Process, Process)
		(Cooperation, mod:commitment)	(Process, Process)
		(LegalAction, Pla:action-task)	(Process, Process}
		(Attribute, common: measurement)	*(common: measurement,* ⊤*)*
		(Plan, mod:commitment)	*(mod:commitment; mod:*⊤*)*
		(Organization, soc:organization)	soc:organization, ⊤)
		(Proposition, Non Physical Object)	*(Non Physical Object,* ⊤*)*
		(InheritableRelation, ends:cognitive modal-description)	*(ends:cognitive modal-description,* ⊤*)*
		(State, Event)	*(Event,*⊤*)*

Table 7-8. Artifacts-based Compatibilities and Incompatibilities of WordNet and DOLCE

Mappings couple α (Keberle et al., 2007), γ (Masolo et al., 2003)			
Compatibilities		**Incompatibilities**	
$M_{WordNet \rightarrow DOLCE}$ (α)	$M_{DOLCE \rightarrow WordNet}$ (γ)	$M_{WordNet \rightarrow DOLCE}$ (α)	$M_{DOLCE \rightarrow WordNet}$ (γ)
(Quality, Quality)	(Quality, Quality)	{Event, Event}	(edns:parameter, Event)
(Event, Accomplishment)	(Accomplishmen t, Event)	(Artifact, sys:system-as-description)	(Non-agentive Physical Object, Artifact)
		(Set, Set)	(coll: non-physical-collection, Set)
		(State, State)	(sys:system-as-situation, State)

From the Table 7-4, Table 7-5, Table 7-6, Table 7-7 and Table 7-8, it is quite clear that in all the cases, mappings established by distinct authors are incompatible because in each case, at least one incompatibility arises.

This case study highlights some important points and we discuss them below.

One could argue that approaches used for Incoherence can also solve the problem consider in this case study. From this case study of ontology mappings of DOLCE and GFO however, it is clear that the available mapping do not tell exactly whether mapped artifacts are mapped in equivalence or subset relation. So it is possible that logical approaches sometimes find contradiction when they treat such mappings has correspondences established by equivalence relation while sometimes do not find contradiction when they treat such mappings has correspondences established by subset relation. Therefore, this method is not reliable in this situation.

It is clear from this case study that we do not have complete mappings of one ontology into another ontology. We have two choices: one is to complete missing mappings and other is to deal with partial mappings. We prefer to deal with partial mappings. Moreover, it is clear that completion of mapping should be done in such a way that it does not contradict existing mapping.

We analyze incompatible mappings and this clearly leads to two ways: either the involved artifacts in correspondence causing incompatibility are not properly defined or mapping is established wrongly.

In the Section 7.6 , we show that incompatible ontology mapping always leads to incorrect inferences.

From this case study, it is clear that Galois connection based definition of compatibility is simple and easy to use and it can easily handle with different kinds of mapping relations, partial mappings and complex artifacts involved in correspondences.

7.6 Ontology merging and importance of compatibility of ontology mappings

Since many application fields use ontologies these days, the demand of matching system also increases. In addition, these matching systems are currently used in combination to produce better matching results. When there are more than one matching systems, they may give different matching results. When these matching results are combined and used with source ontologies, they may cause inconsistencies in ontologies. P. Shvaiko and J. Euzenat describe future challenges for ontology mapping and one of them is finding novel ways for combining ontology matching systems (Shvaiko & Euzenat, 2013).

Instead of creating ontologies from scratch, existing ontologies can be reused and merged to create new ontologies. In this section, we investigate the creation of ontologies by merging mapped ontologies. However, several mappings can have been established for performing the merging operation, by using matching tools or by hand.

We therefore propose a method that identifies which ontology mappings couples of two ontology mappings can be used together and which cannot. Combining several mappings is naturally useful for getting benefits of each mapping. A merged ontology O can be abstractly defined as $\mathrm{O} = \mathrm{merge}(\mathrm{O_A} \cup \mathrm{O_B} \cup \mathrm{M_{Ab}})$, where $\mathrm{O_A}$ is the first ontology and $\mathrm{O_B}$ is the second ontology, $\mathrm{M_{AB}}$ is the ontology mapping between $\mathrm{O_A}$and $\mathrm{O_B}$, and merge is the merge operation. In (Abbas & Berio, 2013), we present a method for identifying incompatible ontology mappings, which are symmetric $(\mathrm{M_{AB}} = \mathrm{M_{BA}})$, in creating ontologies can cause unsatisfiability. In this section, we assume $\mathrm{M_{AB}}$ is supposed to be asymmetric, i.e., $\mathrm{M_{AB}} \neq \mathrm{M_{BA}}$. Since it is not necessary that ontology mappings are symmetric (Borgida & Serafini, 2003), (Kutz et al., 2004). For instance, two ontologies describe the same domains with overlapping information, in this scenario, mappings are usually symmetric and they have a single universal interpretation domain. However, when there are two ontologies describing a domain from different point of views, in this scenario, mappings are not symmetric and ontologies have their own interpretation domain.

If two distinct mappings $\mathrm{M_{AB}}$ and $\mathrm{M'_{BA}}$ are obtained by using different matching algorithms or matching tools or even manually by distinct experts, we define merged ontology as $\mathrm{O'} = \mathrm{merge}\{\mathrm{O_A} \cup \mathrm{O_B} \cup \mathrm{M_{AB}} \cup \mathrm{M'_{BA}}\}$. However, not all couples of mapping correspondences of $\mathrm{M_{AB}}$ and $\mathrm{M'_{BA}}$ can be used together. Some of them are deeply incompatible and are sources of further problems. For instance, if mappings are expressed as axioms in some Description Logics, saying that $\mathrm{M_{AB}}$ =A⊑B, $\mathrm{M'_{BA}}$=A⊑ C , being B and C disjoint in $\mathrm{O_B}$ leads to a problem if

A is satisfiable in O_A and the merge operation simply adds those axioms to the union of these two ontologies

In this section, we develop a method based on Galois connections for deciding when two mappings can be combined (used together) in the context of a merge operation. Galois connections are interesting because work on ordered structures (taxonomies are the basic structure for organizing, reasoning and analyzing ontologies). Therefore, Galois connections work independently if a mapping is concretely a logical relation (such as subsumption) or a non-logical relation (such as similarity). Galois connections work also if the ordered structure is just an order, a subsumption or a simple taxonomy. Finally, Galois connections may be used to map simple artifacts but also more complex artifacts (such as a set of axioms). It should be noted that in literature two mapping correspondences like $m_{AB} = A \sqsubseteq B$, $m'_{AB} = A \sqsubseteq C$ can be combined as union and the resulting mapping can be judged erroneous (Meilicke & Stuckenschmidt, 2009), (Qi et al., 2009), (Jiménez-Ruiz et al., 2011). However, you can achieve the same result by using Galois connections which are however independently formulated and work well both if ontology is represented in logics but also if ontology is just a taxonomy or category. We therefore argue that Galois connections are the most adapted and theoretically sound and generic approach for evaluating when distinct mappings can be used together in the context of merge operation.

7.6.1 A Motivating Scenario: Merging Ontologies

A new ontology can be created from other source ontologies by establishing mappings between source ontologies and then combining source ontologies and mappings.

Let consider two source ontologies O_A and O_B and their concepts as shown in Ontology O_A Figure 7-8 and Figure 7-9 respectively. Links between ontology concepts are hierarchical relation. When there is no link between concepts, it means that these concepts are disjoint (or there is not order relation between them). Let also consider ontology mappings M_{AB}= {(University, Research Org.), (Director, Director Admin), (Administrative Staff, Research Officer)}, and M'_{BA}= {(Research Org., University), (Research Staff, Teaching Faculty), (Admin. Staff, Director), (Research Officer, Researcher), (Director Admin, Administrative Staff)}. Both M_{AB} and M'_{BA} can be used separately for creating a new ontology by using, for instance, categorical merge operation (Hitzler et al., 2005), (Zimmermann et al., 2006)). As M_{AB} and M'_{BA} are not same, users may want to get maximum advantage from both of them. There will be a risk that the combination operation may result in inconsistent ontology or there will be no resultant ontology.

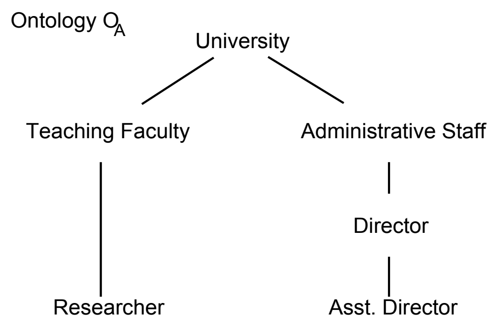

Figure 7-8. Ontology O_A

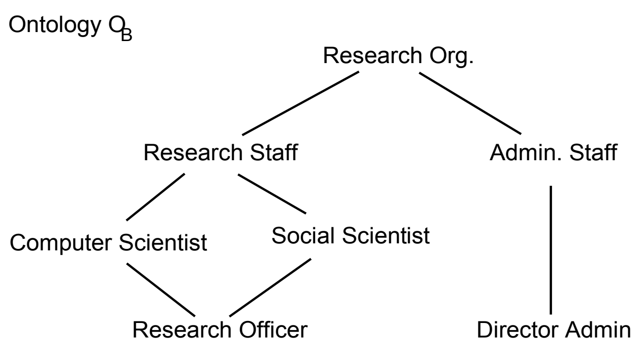

Figure 7-9. Ontology O_B

Indeed, if mapping M_{AB} is rewritten in Distributed Description Logic (Borgida & Serafini, 2003) it may become (by using equivalence)

O_A: University $\xrightarrow{\equiv}$ O_B: Research Org.
O_A: Director $\xrightarrow{\equiv}$ O_B: Director Admin
O_A: Administrative Staff $\xrightarrow{\equiv}$ O_B: Research Officer
and M'_{BA} as
O_B: Research Org. $\xrightarrow{\equiv}$ O_A: University
O_B: Research Staff $\xrightarrow{\equiv}$ O_A: Teaching Faculty
O_B: Admin. Staff $\xrightarrow{\equiv}$ O_A: Director
O_B: Research Officer $\xrightarrow{\equiv}$ O_A: Researcher
O_B: Director Admin $\xrightarrow{\equiv}$ O_A: Administrative Staff

Using Category theory approach for ontology merging and mapping $M_{AB} \cup M'_{BA}$ results is undefined; using a Full-Merge approach (Raunich & Rahm, 2012) for ontology merging results in an incoherent ontology, and using a logical approach will also make the merged ontology incoherent. Therefore, in both cases, we can say that the two mappings cannot be combined. The underlying idea of Galois connections is to provide generic conditions, without referring to one specific theory (for instance, logics).

7.6.2 Compatible and incompatible ontology mappings in the context of ontology merging

We show how our proposed method can be used for finding compatible and incompatible ontology mappings. We apply this method to the example described in Section 7.6.1.

We will check the Galois connection condition between ontology mapping couples by using ontology structure for identifying compatibility and incompatibility. We use definition 4 for compatibilities and incompatibilities at artifact level. We present here one case of compatible and incompatible mappings from our example.

$\alpha(\{\text{Director}\}) = \{\text{Director Admin}\}$ according to M_{AB}

$\gamma(\{\text{Director Admin}\}) = \{\text{Administrative Staff}\}$ according to M'_{BA}

They form order preserving Galois connection as Director $\preccurlyeq$ Administrative Staff and Director Admin $\preccurlyeq$ Director Admin.

While for mapping couple

$\alpha(\text{Administrative Staff}) = \text{Research Office}$ according to M_{AB}.

$\gamma(\text{Research Staff}) =$ Teaching Faculty according to M'_{BA}.

They neither form order preserving nor order reversing Galois connection as Research Officer $\preccurlyeq$ Research Staff and Administrative Staff $\npreccurlyeq$ Teaching Facultyand Teaching Faculty $\npreccurlyeq$ Administrative Staff.

We will show the creation of new merged ontology with the help of mappings $M_{AB} \cup M'_{BA}$ by category theory approach, by Full-Merge operation (Raunich & Rahm, 2012), and by logical approach.

Categorical merge operation requires two properties to be fulfilled; a) The source ontologies must embed into merged ontology, b) merged ontology must not identify anything unnecessarily which are not present in the source ontologies. Categorical Pushout operation is shown in Figure 7-10. These two properties are satisfied by the following two properties for categorical pushout operation.

(i) $e_A \circ p_A = e_B \circ p_B$.

(ii) For every other object O''_{AB} and morphisms $f_A: O_A \to O''_{AB}$, and $f_B: O_B \to O''_B$, with $f_A \circ p_A = f_B \circ p_B$ there is a unique morphism $m: O'_{AB} \to O''_{AB}$ such that $f_A = e_A \circ m$.and $f_B = e_B \circ m$.

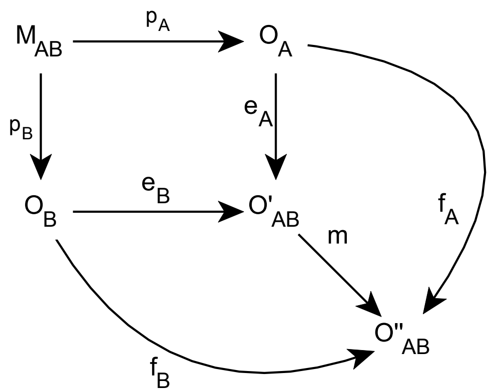

Figure 7-10. Pushout operation for Categorical merge

Resulting ontology by using categorical merge operation with the help of pushout operation (Hitzler et al., 2005), (Zimmermann et al., 2006) is shown in Figure 7-11 . Cross on Research Officer shows that due to this concept merge operation is not possible. Mapping correspondences in $\mathrm{M_{AB}} \cup \mathrm{M'_{BA}}$ are incompatible such as {m_{AB}:(Administrative Staff, Research Officer), m'_{BA}:(Research Officer, Researcher)}, as these ontology mapping couples do not establish Galois connection. These incompatible mappings are also verified from categorical merge operation, since artifact 'Research Officer' violate the partial order relationship exist in the source ontology, and there is no pushout operation which will respect the partial order by using these mappings. Therefore, it is not possible to obtain new merged ontology by using categorical merge operation with the help of categorical pushout operation.

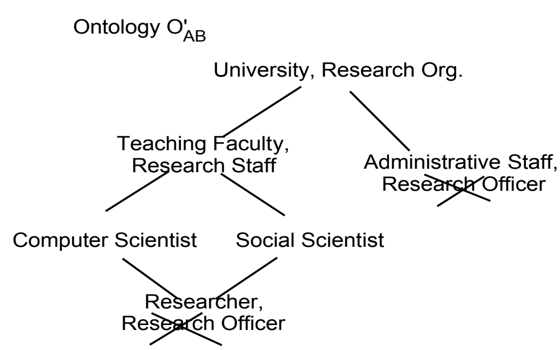

Figure 7-11. Ontology obtained by merging $\mathrm{O_A}$, $\mathrm{O_B}$, and $\mathrm{M_{AB}} \cup \mathrm{M'_{AB}}$using Category Theory approach

We focus on Full Merge approach because it is the proper ontology merging according to the definition of merging by (Euzenat & Shvaiko, 2007). Source driven and Target driven merge solution falls in the category of ontology integration according to (Euzenat & Shvaiko, 2007). Figure 7-12 shows the working of Full Merge approach (Raunich & Rahm, 2012). Full-Merge approach produce merged ontology by taking union of input ontologies and combining equivalent concepts.

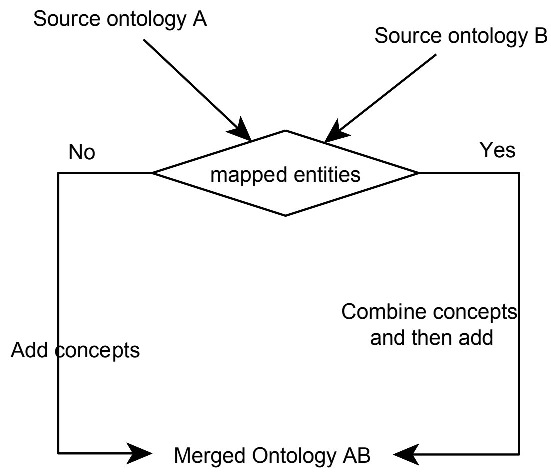

Figure 7-12. Full Merge approach (Raunich & Rahm, 2012)

We create a new ontology merging, as shown in Figure 7-12, by using Full-Merge operation (Raunich & Rahm, 2012). We are using DDL logical equivalence as the equivalence required to apply full-merge. But we can see that ontology $\mathrm{O}'_{\mathrm{AB}}$ obtained by using mapping $\mathrm{M}_{\mathrm{AB}} \cup \mathrm{M}'_{\mathrm{AB}}$ is incoherent. We present here only one incompatible case to show that resulting ontology is incoherent.

$\alpha(A\text{dministrative Staff}) = \text{Research Officer}$, according to M_{AB}

$\gamma(\text{Research Officer}) = \text{Researcher}$, according to $\mathrm{M}'_{\mathrm{BA}}$.

This mapping couple does not form GC, so the new ontology is incoherent as $\text{Researcher} \sqcap \text{Administrative Staff} \sqsubseteq \bot$ in ontology O_{A}but in resulting ontology $\text{Researcher} \equiv \text{Administrative Staff}$. Concepts marked with * in Figure 7-13 represents that these concepts are redundant concepts and they should appear only one time in the merged ontology.

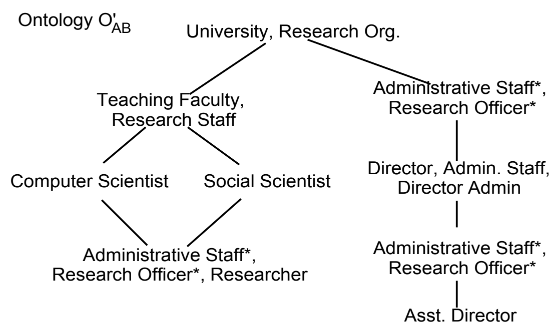

Figure 7-13. Resulting Ontology obtained by merging $\mathrm{O_A}$, $\mathrm{O_B}$, and $\mathrm{M_{AB}} \cup \mathrm{M'_{AB}}$ using Full-Merge (Raunich & Rahm, 2012) approach

In logical approach, the new merged ontology is not coherent, $\mathrm{O_1} \cup \mathrm{O_2} \cup \mathrm{M_{AB}} \cup \mathrm{M'_{BA}}$ is considered as merged ontology and it provides the same inferences which merged ontology provides. The main property of mappings is that it should not contradict the axioms of original ontologies.

We can see that there are unsatisfiable concepts in $\mathrm{O_1} \cup \mathrm{O_2} \cup \mathrm{M_{AB}} \cup \mathrm{M_{BA}}$., such as in $\mathrm{O_A}$ has axiom Researcher ⊓ Administrative Staff ⊑⊥, while in $\mathrm{O_1} \cup \mathrm{O_2} \cup \mathrm{M_{AB}} \cup \mathrm{M_{BA}}$, we can derive

$\mathrm{O_A}$: Administrative Staff $\xrightarrow{\equiv}$ $\mathrm{O_B}$: Research Officer, and

$$\mathrm{O_B}\text{: Research Officer} \xrightarrow{\equiv} \mathrm{O_A}\text{: Researcher}$$

and we can infer from above

$$\mathrm{O_A}\text{: Researcher} \equiv \mathrm{O_A}\text{: Administrative Staff}$$

So by above inference, $\mathrm{O_A}$: Researcher and $\mathrm{O_A}$: Administrative Staff become unsatisfiable concepts, and creating any instances of these concepts create inconsistency and we can derive $\top \sqsubseteq \bot$, so $\mathrm{O_1} \cup \mathrm{O_2} \cup \mathrm{M_{AB}} \cup \mathrm{M_{BA}}$ is incoherent. such as {m_{AB}:(Administrative Staff, Research Officer), m'_{BA}:(Research Officer, Researcher)}, as these ontology mapping couples do not establish Galois connection.

We have shown only the case when correspondences in one ontology mappings are incompatible. It should be noted that when GC conditions are not satisfied if the mapping is reinterpreted in some theories (such as Category, logics) where a merge operation is denied, the resulting ontologies may be incoherent/inconsistent or not existing. We argue that the GC conditions are common to all theories defining merging operation.

7.6.3 Discussion

The method we have proposed is based on Galois connections and on the definition of “mapping compatibility and incompatibility". We have applied the method to the example introduced in Section 7.6.2; the main contributions conveyed by the proposed method are (i) identifying incompatible ontology mappings, which give a-priori information to the user whenever merge operation by using couple of mappings is not possible, (ii) applicability to ontologies that may expressed by different formalisms, (iii) applicability to domain ontologies but also to upper ontologies which are mostly hierarchically organized.

Chapter 8.

Conclusion and Future work

In this chapter, we discuss our conclusions about the work presented in this thesis. We discuss how objectives of this thesis have been achieved. We review our work presented in the previous chapters and assess to what extent we have achieved the objectives of this thesis.

In Section 8.1, we present an overview of the thesis. We access how objectives of this thesis have been fulfilled in Section 8.2. In Section 8.3, we describe the major results of this thesis. Finally, in Section 8.4, we suggest some directions for future work.

8.1 Thesis overview

The objective of this thesis was to propose an approach for dealing with ontology mappings and their defects, by naturally covering heterogeneity and other problems in ontologies (such as omitted axioms).

The objective has been achieved by proposing a unified approach based on lattices and Galois connections.

Chapter 2 and Chapter 3 presented logic-based ontology mapping languages and ontology mappings in algebraic approaches, respectively. We have found that heterogeneity of ontology mapping is poorly taken into account in state of the art approaches.

Chapter 4 presented ontology mapping correctness (absence of defects) and acceptability (agreement on mappings). We have found that the same syntax of mapping behaves differently in different logic-based systems, i.e. mapping that causes a defect in one logic may not cause that defect in another logic.

Chapter 5 presented short introduction of Galois connection. Lattices that can be built for representing ontologies are described, and then the notion of compatible and incompatible ontology mappings has been formally introduced. Chapter 5 presented sketches to detect defects by using proposed approach.

Chapter 6 presented formal proofs to highlight the key characteristics of compatible and incompatible ontology mappings, especially how compatible ontology mappings are related to correct ontology mappings (as presented in Chapter 4).

Chapter 7 presented how notion of compatible ontology mappings can be applied for checking the existence of defects and how incompatible ontology mappings prevent the success of algebraically defined ontology merge operation.

8.2 Claimed Objectives and Achieved Results

Hereinafter, we want to discuss in detail how claimed objectives listed in Section 1.3 have been achieved through the work reported in this thesis. First, we remember that the objectives that we set and how they have been achieved through the performed work.

(O 1) To show the Importance of compatibility and incompatibility of ontology mappings;

(O 2) To define compatibility and incompatibility of ontology mappings;

(O 3) To check whether existing state of the art work will provide a solution for proposing unified approach of identifying compatibility and incompatibility of ontology mappings;

(O 4) To develop a unified approach for dealing with ontology mappings and their defects that is applicable to various formalisms of ontology and ontology mappings;

(O 5) To relate notions of compatible and incompatible ontology mappings with mapping correctness;

(O 6) To deal with upper ontology mappings.

With respect to objective (O1), we have presented scenarios in Section 1.2 where the need for identification of compatible and incompatible ontology mappings both at run time and design time has been shown. We have also shown that combination of ontology mappings is widely used.

We achieved objective (O2) in Chapter 5 by introducing the notion of compatible and incompatible ontology mappings. We have also shown in Chapter 5 that compatible mappings are different from logical correctness and agreed ontology mappings.

Objective (O3) has been achieved by analyzing the state of the art of ontology mapping and checking the adequacy of the state of the art for defining the notions of compatible and incompatible ontology mappings. We have found that logical approaches for debugging ontology mappings can be used for checking the existence of defects, however, these approaches required ontologies are well-formalized (i.e. all necessary axioms should be part of the ontology), otherwise, defects cannot be detected. Additionally, these approaches required that ontologies and ontology mappings are expressed in the same formalism.

Objective (O4) has been achieved by developing an approach based on Galois connection for identifying compatible and incompatible ontology mappings. Galois connection provides an 'independence from any kind of formalization' for representing ontologies and ontology mappings. We have showed that our proposed approach is applicable to different kinds of ontology mapping formalisms and it detects defects even when some axioms are omitted in ontologies.

For achieving objective (O5), we have proved in Chapter 6 the important characteristics of compatible ontology mappings. We have proved that if mappings respect the principle of conservativity and source ontologies are coherent and consistent then there exist a Galois connection between the lattices of inferences of source ontologies (or other lattices selected based on the contents of source ontologies and mappings). Our notions of compatibility fundamentally differs with correctness as two correct mappings may not be compatible just because any mapping is not the relative semantics of the other one.

For achieving objective (O6), we have proposed an approach for dealing upper ontology mappings. We have collected various existing upper ontology mappings and applied our approach manually on these mappings for verifying the compatibility of these mappings. We have presented the results in tabular form in Chapter 7.

8.3 Major Results

This thesis introduces the notion of compatible and incompatible ontology mappings (Chapter 5). Compatible and incompatible mappings provides a relatively new approach to correctness and acceptability discussed in Chapter 4. We say "relatively new" because a similar technique is used in the context of upper ontologies (Abbas & Berio, 2013) but limited to taxonomies and limited to "main concepts". This thesis extends the basic idea and provides a complete view on how known notions of correctness (i.e. absence of defects) and acceptability (i.e. mapping agreement) are covered (Chapter 6). Additionally, it has been shown to what the informal idea of "compatible mappings" i.e. two ontology mappings are compatible is they can be used safely within the same application (see Chapter 1) corresponds from the logical point of view (Theorem 6-8).

The major results of this thesis are:

- A synthesis and comparison of features of logic-based mapping formalisms, it is listed in Table 2-2.
- We explicitly show that the same syntax of ontology mappings does not have the same meaning in different logic-based ontology mapping formalism. We explicitly show this by giving a concrete example of mappings behaving in different formalisms in Section 4.7.
- Classification of ontology mapping formalisms in terms of dependence of contents and logic, shown in Figure 3-20.
- We introduce a new classification of ontology mappings defects, described in Section 4.3.2.
- We introduce lattices for representing ontologies, as a unified abstract structure and based on that we introduce the notion of compatible and incompatible ontology mappings. At the same time, a uniform semantics for ontology mappings is introduced as an ontology mapping is what the mapping produces (e.g. as logical consequence, as a merged ontology and so on).

- Heterogeneity is naturally approached because ontologies, independently on how they are represented, are then naturally represented as lattices (with no need of complex theories of equivalence); additionally depending on the used lattice, incompleteness or other problems in ontology axioms (leading to impossibility to proof logically the existence of some defects), are implicitly solved (the most trivial example is any omission of disjontness axioms but other cases have also been identified e.g. violation of principle of conservativity, mapping is not an interpretation and others).
- In Chapter 6, we provided formal proofs, showing that compatible/incompatible ontology mappings cover logical defects (reported in Chapter 4). Specifically:
    - compatible ontology mappings lead necessarily to ontology (theory) interpretations; this means that having compatible mappings prevents the defects of defining ontology mappings which are not ontology interpretation;
    - compatible mappings, when combined within a same logical theory, do not lead to inconsistency (nor to incoherence if the mapped ontologies are coherent); to some extent, compatible ontology mappings preserve the principle of conservativity (reported in Section 6.2).

  However, as said above, compatible mappings cover also defects which cannot be discovered in logics (by the implicit construction of a specific lattice) and also other defects (such as the ones empirically defined, or the ones preventing the success of algebraically defined operations, such as categorical merge, see Chapter 7)
- Because the original idea was found in the domain of upper ontologies, in Chapter 7 we show how compatible and incompatible mappings can be applied to upper ontologies. This is an important application because provides a base for highlighting the key differences between upper ontologies. Indeed, it was expected and effectively shown that several incompatibilities raise between upper ontology mappings. The important result is that people doing such as kind of mapping, implicitly, do not agree on how mapping upper concepts or highlight that mapping upper concepts is not necessarily performed with sufficient care.

8.4 Future Work

The idea of identifying compatible and incompatible ontology mappings opens for further research:

- Developing an automated tool for the effective use of proposed unified approach. At the same time, this may require some additional research, for instance, in Theorem 6-3 the lattice is not used fully as only few elements of lattice are used. However, we are able to identify the elements that are required in our case. So, the matter of building partial lattices instead of complete lattices, need to be investigated. In our view, this approach should be used during the establishment of mappings i.e., incrementally to avoid the complexities of lattices.

- Abstract constructions used in theorems in Chapter 6 can be generalized; this should enable to define in a complete abstract way what an ontology mapping defect is, without referring to concrete symptoms such as inconsistency, incoherence and others, which are heavily dependent of the used formalisms
- Comparing the performance of logical automated tool of debugging ontology mappings with our unified approach for dealing with ontology mappings and their defects.

References

Abbas, M.A. & Berio, G., 2013. Compatible and Incompatible Ontology Mappings. Valencia, 2013. CAiSE (Doctoral Consortium).

Abbas, M.A. & Berio, G., 2013. Creating Ontologies Using Ontology Mappings - Compatible and Incompatible Ontology Mappings. In *2013 IAT Workshop on Knowledge Discovery in Ontologies*. Atlanta, GA USA, 2013. IEEE Computer Society.

Adámek, J., Horst, H. & George, S., 1990. *Abstract and concrete categories*. New York, USA: Wiley-Interscience.

Amgoud, L. & Cayrol, C., 1998. On the acceptability of arguments in preference-based argumentation. In G. Cooper & S. Moral, eds. *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. San Francisco, CA: Morgan Kaufmann Publishers Inc. pp.1-7.

Anaya, V. et al., 2010. The Unified Enterprise Modelling Language -Overview and further work. *Computers in Industry*, 61(2), pp.99-111.

Anon., 2001. *Dynamically debugging user-defined functions and stored procedures*. USA: US Patent.

Atencia, M. et al., 2012. A formal semantics for weighted ontology mappings. *The Semantic Web-ISWC 2012*, pp.17-33.

Baader, F. & Nutt, W., 2003. Basic description logics. In F. Baader et al., eds. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge: Cambridge University Press. pp.43–95.

Baclawski, K. et al., 2004. Towards a symptom ontology for semantic web applications. In *The Semantic Web–ISWC*. Berlin Heidelberg: Springer. pp.650-67.

Bao, J., Caragea, D. & Honavar, V., 2006. On the semantics of linking and importing in modular ontologies. *The Semantic Web-ISWC 2006*, pp.72-86.

Baumeister, J. & Seipel., D., 2006. Verification and refactoring of ontologies with rules. In *Managing Knowledge in a World of Networks*. Berlin Heidelberg,: Springer-Verlag. pp.82-95.

Bench-Capon, T.J.M., 2003. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3), pp.429-48.

Bench-Capon, T.J.M. & Malcolm, G., 1999. Formalising Ontologies and Their Relations. *Database and Expert Systems Applications*, 1677, pp.250-59.

Bens, H., 2011. *Ontology4*. [Online] Available at: Ontology4.us.

Birkhoff, G., 1967. *Lattice theory*. 1st ed. Providence, RI: AMS Colloq.

Borgida, A. & Serafini, L., 2003. Distributed description logics: Assimilating information from peer sources. In *Journal on Data Semantics I*. Berlin Heidelberg: Springer. pp.153-84.

Bouquet, P. et al., 2003. C-owl: Contextualizing ontologies. In K. Sycara & J. Mylopoulos, eds. *The Semantic Web-ISWC*. Berlin Heidelberg: Springer. pp.164-79.

Caballero, R., García-Ruiz, Y. & Sáenz-Pérez, F., 2012. Declarative debugging of wrong and missing answers for SQL views. In *Functional and Logic Programming*. Berlin Heidelberg: Springer. pp.73-87.

Cafezeiro, I. & Haeusler, E.H., 2007. Semantic interoperability via category theory. In *Tutorials, posters, panels and industrial contributions at the 26th international conference on Conceptual modeling*. Australian Computer Society, Inc. pp.197-202.

Calvanese, D., De Giacomo, G. & Lenzerini, M., 2002. A framework for ontology integration. In I. Cruz, S. Decker, J. Euzenat & D. McGuinness, eds. *The Emerging Semantic Web - Selected Papers from the First Semantic Web Working Symposium*. pp.201-14.

Carnap, R., 2002. *The logical syntax of language*. Open Court Publishing.

Chaudhri, V.K. et al., 1998. OKBC: A programmatic foundation for knowledge base interoperability. In. In *AAAI/IAAI*. pp.600-07.

Chen, P.P.-S., 1976. The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems (TODS)* , 1(1), pp. 9-36.

Chisholm, R.M., 1996. *A Realistic Theory of Categories: An Essay onOntology*. Cambridge University Press.

Chiticariu, L. & Tan, W.-C., 2006. Debugging schema mappings with routes. In *Proceedings of the 32nd international conference on Very large data bases*., 2006. VLDB Endowment.

Colomb, R.M., 2002. *Use of upper ontologies for interoperation of information systems: a tutorial*. ISIB-CNR, Padua (Italy).

Corcho, O., Roussey, C. & Vilches-Blázquez, L.M., 2009. Catalogue of Anti-Patterns for formal Ontology debugging. In *Atelier Constructiond'ontologies: vers un guide des bonnes pratiques*. AFIA.

Cousot, P., Radhia, C. & and Laurent, M., 2013. Theories, solvers and static analysis by abstract interpretation. *Journal of the ACM (JACM)*, 59(6), p.31.

Crubézy, M. & Musen, M.A., 2004. Ontologies in support of problem solving. In *Handbook on ontologies*. Springer. pp.321-41.

Davey, B.A. & Priestley, H.A., 2002. *Introduction to lattices and order*. Cambridge university press.

Davies, I., Green, P., Milton, S. & Rosemann, M., 2003. Using meta models for the comparison of ontologies., 2003.

Dou, D., McDermott, D. & Qi, P., 2005. Ontology translation on the semantic web. In *Journal on data semantics II*. Springer. pp.35-57.

Dung, P.M., 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and person games. *Artificial intelligence*, 77(2), pp.321-57.

Eilenberg, S. & Lane., S.M., 1945. General theory of natural equivalences. *Transactions of the American Mathematical Society*, 58, pp.231–94.

Enderton, H.B., 2001. *A Mathematical Introduction to Logic*. 2nd ed. Harcourt Academic Press.

Euzenat, J., 2008. Algebras of ontology alignment relations. *The Semantic Web-ISWC 2008*, pp.387-402.

Euzenat, J. & Shvaiko, P., 2007. *Ontology matching*. Berlin Heidelberg: Springer-Verlag.

Fahad, M., Nejib, M. & Boura, A., 2012. Detection and resolution of semantic inconsistency and redundancy in an automatic ontology merging system. *Journal of Intelligent Information Systems*, pp.1-23.

Farquhar, A., Richard, F. & James, R., 1997. The ontolingua server: A tool for collaborative ontology construction. *International journal of human-computer studies*, 46(6), pp.707-27.

Ferrara, A., Nikolov, A., Noessner, J. & Scharffe, F., 2013. Evaluation of instance matching tools: The experience of OAEI. *Web Semantics: Science, Services and Agents on the World Wide Web*.

Flouris, G. et al., 2006. Inconsistencies, negations and changes in ontologies. In *In Proceedings of the 21th National Conference on ArtificialIntelligence*., 2006. AAAI Publications.

Flouris, G., Plexousakis, D. & Grigoris, A., 2006. Evolving ontology evolution. In *International Conference on Current Trends in Theory and Practice of Computer Science*. Merin, Czech Republic, 2006. Springer Berlin Heidelberg.

Gaerdenfors, P., 1992. *Belief revision: An introduction. In Belief Revision*. Cambridge University Press.

Gangemi, A., Catenacci, C., Ciaramita, M. & Lehmann, J., 2006. Modelling ontology evaluation and validation. In *The Semantic Web: Research and Applications*. Springer Berlin Heidelberg. pp.140-54.

Ganter, B., 2007. Relational Galois Connections. In S. Kuznetsov & S. Schmidt, eds. *Formal Concept Analysis*. Berlin: Springer. pp.1-17.

Ganter, B., Wille, R. & Franzke, C., 1997. *Formal concept analysis: mathematical foundations*. Springer-Verlag New York, Inc.

Gärdenfors, P., 2003. Belief Revision: An Introduction. In *Belief Revision*. Cambridge University Press.

Gehlert, A. & Esswein, W., 2007. Toward a formal research framework for ontological analyses. *Advanced Engineering Informatics*, 21(2), pp.119–31.

Genesereth, M.R. & Fikes, R.E., 1992. *Knowledge interchange format-version 3.0: reference manual*.

Ghidini, C. & Serafini, L., 1998. Distributed First Order Logics. In *Frontiers of Combining Systems Second International Symposium*., 1998.

Goguen, J.A.a.B.R.M., 1984. Introducing institutions. In *Logics of Programs*. Springer. pp.221-56.

Goguen, J., 1991. A Categorical Manifesto. *Mathematical structures in computer science*, 1(01), pp.49-67.

Goguen, J. & Burstall, R., 1992. Institutions: abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1), pp.95-146.

Goguen, J. & Roşu, G., 2002. Institution morphisms. *Formal aspects of computing*, 13(3-5), pp.274-307.

Gómez-Pérez, A., 2001. Evaluation of ontologies. *International Journal of intelligent systems*, 16(3), pp.391-409.

Grau, B.C., Parsia, B. & Sirin, E., 2004. Working with multiple ontologies on the semantic web. In *The Semantic Web–ISWC 2004. Springer Berlin Heidelberg*., 2004.

Grenon, P., 2003. *BFO in a Nutshell: A Bi-categorial Axiomatization of BFO and Comparison with DOLCE*. IFOMIS, Faculty of Medicine, University of Leipzig.

Gruber, T.R., 1993. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2), pp.199-220.

Gruber, T.R., 1995. Toward principles for the design of ontologies used for knowledge sharing. *International journal of human-computer studies*, 43(5), pp.907-28.

Gruninger, M. & Aameri, B., 2014. Preservation of Modules., 2014.

Grüninger, M., Hahmann, T., Hashemi, A. & Ong, D., 2010. Ontology Verification with Repositories. In *FOIS*. IOS Press. pp.317-30.

Grüninger, M. et al., 2012. Modular first-order ontologies via repositories. *Applied Ontology*, 7(2), pp.169-209.

Grüninger, M. & Menzel, C., 2003. The Process Specification Language (PSL) Theory and Applications. *AI Magazine*, 24, pp.63-74.

Guarino, N., 1999. The role of identity conditions in ontology design. *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science*, pp.221-34.

Guizzardi, G. & Wagner, G., 2004. A Unified Foundational Ontology and some Applications of it in Business Modeling., 2004.

Haase, P. & Qi., G., 2007. An analysis of approaches to resolvinginconsistencies in DL-based ontologies., 2007. Proceedings of the International Workshop on Ontology Dynamics.

Hacene, M.R., Huchard, M., Napoli, A. & Valtchev, P., 2013. Relational concept analysis:mining concept lattices from multi-relational data. *Ann. Math. Artif. Intell*, pp.81-108.

Hansson, S.O., 1999. *A textbook of belief dynamics: Theory change and database updating*. 11th ed. Springer.

Herre, H. et al., 2006. *General Formal Ontology (GFO): A Foundational Ontology Integrating Objects and Processes.* Onto-Med Report. Research Group Ontologies in Medicine (Onto-Med), University of Leipzig.

Hitzler, P. et al., 2006. *D2. 2.5 Integrated view and comparison of alignment semantics.*

Hitzler, P., Krötzsch, M., Ehrig, M. & Sure, Y., 2005. What Is Ontology Merging? In *American Association for Artificial Intelligence*., 2005.

Hitzler, P. et al., 2009. *OWL 2 Web Ontology Language: Primer. W3C Recommendation*. [Online] Available at: http://www.w3.org/TR/owl2-primer/.

Homola, M., 2007. Distributed Description Logics Revisited. *Description logics*.

Horridge, M., Parsia, B. & Sattler, U., 2008. Laconic and precise justifications in OWL. In *The Semantic Web-ISWC 2008*. Springer. pp.323-38.

Hovy, E., 1998. Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses. *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC)*.

Isaac, A., dos Santos, C.T., Wang, S. & Quaresma, P., 2008. Using Quantitative Aspects of Alignment Generation for Argumentation on Mappings. In Shvaiko, P., Euzenat, J., Giunchiglia, F. & Stuckenschmidt, H., eds. *Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) Collocated with the 7th International Semantic Web Conference (ISWC-2008)*. Karlsruhe, Germany, 2008. CEUR-WS.org.

Jannink, J., Pichai, S., Verheijen, D. & Wiederhold, G., 1998. Encapsulation and Composition of Ontologies. In *Proceedings of AAAI Workshop on AI & Information Integration.*., may 1998.

Jean-Mary, Y.R., Shironoshita, E.P. & Kabuka, M.R., 2009. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3), pp.235-51.

Jean-Mary, Y.R., Shironoshita, E.P. & Kabuka, M.R., 2010. Asmov: Results for OAEI 2010. *Ontology Matching*, 126.

Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I. & Berlanga, R., 2011. Logic-based assessment of the compatibility of UMLS ontology sources. *Journal of biomedical semantics*, 2.

Jon Barwise, K., 1974. Axioms for abstract model theory. *Annals of Mathematical Logic*, 7(2), pp.221-65.

Kalyanpur, A., Parsia, B., Sirin, E. & Cuenca-Grau, B., 2006. *Repairing unsatisfiable concepts in OWL ontologies*. Springer Berlin Heidelberg.

Kalyanpur, A., Parsia, B., Sirin, E. & Hendler, J., 2005. Debugging unsatisfiable classes in OWL ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(4), pp.268-93.

Keberle, N., Ermolayev, V. & Matzke, W.-E., 2007. Evaluating PSI Ontologies by mapping to the Common Sense., 2007. GI.

Kent., R., 2001. A KIF formalization of the IFF category theory ontology. In *IJCAI 2001 Workshop of the IEEE Standard Upper Ontology*. Seattle, Washington, USA.

Krötzsch, M., Hitzler, P., Ehrig, M. & Sure, Y., 2005. Category theory in ontology research: Concrete gain from an abstract approach. *Karlsruhe, Germany, Institut AIFB, University of Karlsruhe*, 6.

Kutz, O. et al., 2015. E Pluribus Unum. Formalisation, Use-Cases, and Computational Support. In *Computational Creativity Research: Towards Creative Machines*. Atlantis Press. pp.167-96.

Kutz, O., Lutz, C. & Wolte, F., 2004. E-connections of abstract description systems. *Artificial Intelligence*, 156(1), pp.1 - 73.

Kutz, O., Lutz, C., Wolter, F. & Zakharyaschev, M., 2003. E-connections of description logics. In *Description Logics Workshop, CEUR-WS Vol.*, 2003.

Kutz, O., Mossakowski, T. & Lücke, D., 2010. Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design. *Logica Universalis*, 4(2), pp.255-333.

Kutz, O., Normann, I., Mossakowski, T. & Walther, D., 2010. Chinese whispers and connected alignments. *Ontology Matching*, p.25.

Laera, L. et al., 2006. Reaching agreement over ontology alignments. *The Semantic Web-ISWC 2006*, pp.371-84.

Lange, C. et al., 2012. The Distributed Ontology Language (DOL): Use Cases, Syntax, and Extensibility. *CoRR*, abs/1208.0293.

Lenat, D.B., 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM,* 38(11), pp.33-38.

Lenzerini, M., 2002. Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM. pp.233-46.

Lin, Y., 2008. *Semantic Annotation for Process Models : Facilitating Process Knowledge Management via Semantic Interoperability*. Master's Thesis.

Lutz, C., Walther, D. & Frank, W., 2007. Conservative Extensions in Expressive Description Logics. *IJCAI*, 7, pp.453-58.

Maedche, A., Motik, B., Silva, N. & Volz, R., 2002. Mafra—a mapping framework for distributed ontologies. In *Knowledge engineering and knowledge management: ontologies and the semantic web*. Springer. pp.235-50.

Mascardi, V., Cordì, V. & Rosso., P., 2007. A Comparison of Upper Ontologies. In *WOA*. pp.55-64.

Masolo, C. et al., 2003. *"WonderWeb deliverable D18 ontology," IST WonderWeb project*. Trento: Laboratory For Applied Ontology - ISTC-CNR.

McGuinness, D.L., 2003. Ontologies for information fusion. In *Proceedings of the Sixth International Conference on Information Fusion*. Cairns, Queensland, Australia, 2003. IEEE.

McGuinness, D.L. & Pinheiro da Silva, P., 2004. Explaining answers from the semantic web: The inference web approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(4), pp.397-413.

Meilicke, C., 2011. *Alignment incoherence in ontology matching*. PhD thesis, University Mannheim.

Meilicke, C. & Stuckenschmidt, H., 2009. An Efficient Method for Computing Alignment Diagnoses. In *Web Reasoning and Rule Systems*. Springer. pp.182-96.

Michael, J. & Dampney, C.N.G., 2001. On category theory as a (meta) ontology for information systems research. *Proceedings of the international conference on Formal Ontology in Information Systems*, pp.59-69.

Mossakowski, T. & Kutz, O., 2011. The Onto-Logical Translation Graph. In *WoMO*. pp.94-109.

Niles, I. & Pease, A., 2001. Towards a standard upper ontology. In *Second International conference on Formal Ontology in Information System*. Ogunquit, Maine, USA, 2001.

Noy, N.F., 2009. Ontology mapping. In *In Handbook on ontologies*. Springer. pp.573-90.

OMG, 2016. *Unified Modeling Language (UML), [online] Available:*. [Online] Available at: www.omg.org/spec/UML/.

Opdahl, A.L., 2010. Aligning the UEML Ontology with SUMO. In *Enterprise Interoperability IV*. Springer London. pp.409-18.

Ore, O., 1944. Galois connexions. *Transactions of the American Mathematical Society*, 55(3), pp.493-513.

Poveda Villalon, M., Suárez-Figueroa, M.C. & Gómez-Pérez, A., 2010. A double classification of common pitfalls in ontologies. In *Workshop on Ontology Quality*.

Qi, G., Ji, Q. & Haase, P., 2009. A Conflict-Based Operator for Mapping Revision. In *In The Semantic Web-ISWC 2009*. Berlin, 2009. Springer.

Raunich, S. & Rahm, E., 2012. Towards a Benchmark for Ontology Merging. In *On the Move to Meaningful Internet Systems: OTM 2012 Workshops*. Springer. pp.124-33.

RAUNICH, S. & RAHM, E., 2012. Towards a benchmark for ontology merging. *On the Move to Meaningful Internet Systems: OTM 2012 Workshops*, pp.124-33.

Reiter, R., 1987. A theory of diagnosis from first principles. *Artificial intelligence*, 32(11), pp.57-95.

Saunders, M.L., 1998. *Categories for the working mathematician*. Springer.

Schlenoff, C. et al., 2000. *The process specification language (psl): Overview and version 1.0 specification*.

Schlobach, S., 2005. Debugging and semantic clarification by pinpointing. In *The Semantic Web: Research and Applications*. Springer. pp.226-40.

Schlobach, S., 2005. Diagnosing terminologies. *AAAI*, pp.670-75.

Schlobach, S. & Cornet, R., 2003. Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. *IJCAI*, 3, pp.355-62.

Schorlemmer, M. & Kalfoglou, Y., 2008. Institutionalising ontology-based semantic integration. *Applied Ontology*, 3(3), pp.131-50.

Serafini, L. & Tamilin, A., 2006. Distributed instance retrieval in heterogeneous ontologies. In *Applications and Perspectives (SWAP'05)', in 'Proc. of the 2nd Italian Semantic Web Workshop Semantic Web*. Trento, Italy, 2006. Springer.

Shvaiko, P. & Euzenat, J., 2013. Ontology Matching: State of the Art and Future Challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1), pp.158-76.

Smith, P., 2010. *The Galois connection between syntax and semantics*. University of Cambridge.

Solimando, A., Jimenez-Ruiz, E. & Guerrini, G., 2014. *Detecting and Correcting Conservativity Principle Violations in Ontology-to-Ontology Mappings.*

Sowa, J.F., 2000. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, California: Brooks Cole Publishing Co.

Sowa, J.F., 2014. *Principles of Semantic Networks: Explorations in the representation of knowledge*. Morgan Kaufmann.

Studer, R., Benjamins, V.R. & Fensel, D., 1998. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1), pp.161-97.

Terziev, I., Kiryakov, A. & Manov, D., 2005. *Base Upper-level Ontology (BULO) Guidance*. Deliverable 1.8.1, SEKT project.

Trojahn dos Santos, C. & Euzenat, J., 2010. Consistency-driven argumentation for alignment agreement. In *Proc. 5th ISWC workshop on ontology matching (OM)*. Springer. pp.37-48.

Trojahn, C., Moraes, M., Quaresma, P. & Vieira, R., 2008. A cooperative approach for composite ontology mapping. *Journal on data semantics X*, pp.237-63.

W3C, 2016. *RDF Schema*. [Online] Available at: https://www.w3.org/TR/rdf-schema/.

W3C, 2016. *Web Ontology Language (OWL)*. [Online] Available at: https://www.w3.org/OWL/.

W3C, 2016. *XML Standard*. [Online] Available at: https://www.w3.org/standards/xml/.

Wand, Y. & Weber, R., 1990. Mario Bunge's Ontology as a formalfoundation for information systems concepts. *in Studies on Mario Bunge's Treatise*, pp.123-49.

Wang, Z.a.Z.X. et al., 2010. RiMOM results for OAEI 2010. *Ontology Matching*, 195.

Wang, P. & Xu, B., 2012. Debugging ontology mappings: A static approach. *Computing and Informatics*, 27(1), pp.21-36.

Zimmermann, A., 2007. Integrated distributed description logics. In *Proceedings of the 20th International Workshop on Description Logics DL.*, 2007.

Zimmermann, A., Krotzsch, M., Euzenat, J. & Hitzler, P., 2006. Formalizing ontology alignment and its operations with category theory. *Frontiers in Artificial Intelligence and Applications*, 150, p.277.

Zimmermann, A., Krötzsch, M., Euzenat, J. & Hitzler, P., 2006. Formalizing ontology alignment and its operations with category theory. In *Proceeding of the 2006 conference on Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*., 2006.