A Query-Based Approach to Ontologies Using the Theory of Institutions

Thesis submitted in accordance with the requirements of the University of Liverpool
for the degree of Doctor in Philosophy by

Daniel Pokrywczyński

August 2010

Department of Computer Science
University of Liverpool

Primary Supervisor:	**Professor Frank Wolter** Department of Computer Science University of Liverpool
Secondary Supervisor:	**Doctor Grant Malcolm** Department of Computer Science University of Liverpool
Adviser:	**Doctor Boris Konev** Department of Computer Science University of Liverpool
Internal Examiner:	**Doctor Boris Konev** Department of Computer Science University of Liverpool
External Examiner:	**Professor Carlos Caleiro** Departamento de Matemática Instituto Superior Técnico Lisboa

Abstract

In recent years we can observe an increasing interest in using ontologies in different branches of science and commerce. This includes disciplines like medicine, bio-informatics, the semantic web, artificial intelligence, and software engineering, to name a few.

The need to use ontologies in new and evolving applications requires ontologies to evolve. Typical modifications of ontologies include extending an ontology with new axioms, extracting a module (by which we mean a self-sufficient part), and merging two ontologies together. While performing these operations one usually wants to know whether the semantics of the ontologies are, in some sense, preserved.

As the number of ontology applications grew, so did the number of formalisms for ontology formulation. But this increasing number of ontology languages, while helping to develop ontologies and answering the various needs of users, turned out to be a potential source of problems as well. This becomes evident when one is working with multiple ontologies. For instance, when merging two ontologies one not only has to make sure that unwanted consequences are not entailed as a result of this operation but one may also have to solve the problem of these ontologies being given in different formalisms. Even within one formal language, different ontologies may use different vocabularies. Again, different vocabularies make ontologies difficult to use together. Similar problems arise when one wants to compare two ontologies or use an ontology to answer a query that may be given in different formalisms, or that may use different vocabularies.

In the literature, modularity of ontologies, extending, merging and comparing ontologies have received a lot of attention, but usually these problems are considered within one formalism only. On the other hand the problem of comparing and combining ontologies formulated in distinct formalisms has not yet been deeply analyzed.

In our work we consider the issues of querying, merging and comparing ontologies in a more general way. In particular, we investigate how one can query an ontology if the query and the ontology are formulated in different formalisms and possibly different vocabularies. We research how to compare and how to merge ontologies if they are formulated in distinct formalisms and vocabularies. To make this possible we start by presenting an abstract view on ontologies; instead of focusing on the axioms inside the ontology, in our approach we look at its consequences within certain query languages. Then we use the theory of institutions to define the consequence relation in a way that does not depend on a particular formal language. Thanks to that ontologies and queries do not have to be formulated in the same formal language anymore; moreover, the ontology and the query may be formulated with the use of different vocabularies. This provides the first steps towards a formalism that allows us to compare and combine arbitrary ontologies. As the next step we introduce a structure which allows us to work with multiple ontologies, and we formulate the notions of entailment and inseparability of ontologies relative to a signature of interest in a way that does not depend on a particular formalism. This structure allows us to compare and combine arbitrary ontologies.

Furthermore, we show how an abstract description logic can be extended to a description logic with individuals in a systematic and uniform way. We also investigate the relations between description logics and their counterparts with individuals. Thanks to that we are

able to use ontologies together with sets of assertion (ABoxes) to answer queries about individuals. Again, we provide a structure allowing for answering queries about individuals originally formulated in a different formal language than the ontology and the ABox, we assume that ABoxes and ontologies are formulated in the same language. We also present a formulation of entailment and inseparability of ontologies based on instance checking as the one based on subsumption is not strong enough if we consider ontologies together with ABoxes. This formulation is also presented in a way that does not depend on a particular formal language.

Finally, we investigate the problem of entailment with respect to some vocabulary Σ formulated in the lightweight description logic $\mathcal{ELSH}$ and prove that the corresponding decision problem is ExpTime-complete. This extends the result presented by Lutz and Wolter [61] for description logic $\mathcal{EL}$.

Acknowledgements

When I was about to start my PhD, I knew that it is a lot of effort, what I did not know was that it is a lot of effort to many people. I would like to express my gratitude to all of them.

First and foremost, I would like to thank Frank Wolter and Grant Malcolm. I find no words to express how grateful I am for their great supervision and invaluable support during my PhD studies. When I first came here, I hardly knew anything about description logics or category theory, but Frank and Grant patiently explained me everything. And whenever I confused something they explained it again, and again... During the course of my studies I was lost on many occasions, but Frank and Grant always helped me to get back on track. With their wisdom and experience they could foretell and avoid difficulties that were on the way to the final point. I would like to thank them for their constructive criticism, it helped me a lot, and what is very important it was always presented in a gentle way. Frank and Grant always kept their doors open for me and always did support me. They kept their trust in me, even when I had doubts myself. I cannot imagine better supervisors. Thank you.

I would like to thank Boris Konev, my thesis adviser, for all the invaluable feedback and the encouragement over all these years. His great attitude, friendliness and all the good words I heard from him helped me a lot to keep going.

I would like to thank my family and friends for their support, and finally the members of staff and my fellow PhD colleagues at the Department of Computer Science, who contributed to creating an enjoyable working place.

Thank you all,
Daniel

Contents

Chapter 1

Introduction

1.1 Ontologies

The term **ontology** (from Greek *ον*, genitive *οντος* – of being and *λογια* – theory, study) was introduced in the seventeenth century as the name of the field of philosophy which is the study of being. The field itself originates in ancient Greece and is a part of metaphysics. In philosophy ontology aims to answer questions like what entities exist or can be said to exist, it aims also at explaining the very nature of existence. Finally, it tries to determine how these entities can be grouped.

In computer science the aim is less ambitious. In computer science we do not want to study what exists and how it exists, we want to provide a model describing some aspects of the world. For that reason an ontology is often defined as a communicative, essentially syntactic artefact. As it was introduced in Gruber's definition [45], 'an ontology is an explicit specification of a conceptualization', it is often elaborated to a definition capturing also semantic use of ontologies in sharing information. For example, Studer et al. [75] combine Gruber's definition with that of Borst [17], giving what has become a standard definition: 'an ontology is a formal, explicit specification of a shared conceptualization'. This emphasis on sharing information is also present, for example, in Uschold and Grüninger's [83] statement that an ontology is 'used to refer to the shared understanding of some domain of interest'.

In these definitions 'formal' means that ontologies are supposed to be presented by means of a language with an unambiguous syntax and semantics. The requirement to be explicit means that the ontology has to define in an unequivocal way the types of concepts that are used, and the restrictions on their use. 'Shared' in the above definition means that the understanding of the vocabulary used for creating an ontology, as well as the represented knowledge is the result of an agreement and is accepted by a group of users interested in the domain which is described by the ontology. On the other hand, conceptualization is an abstract and generalized representation of the aspects of the world that are in the scope of our interest. In other words, having set the vocabulary, the ontology provides us with a representation of the objects, concepts and other entities present in the domain of interest, together with the relations between them.

From the above we can conclude that an ontology in computer science is an engineered creation based on a social agreement and representing a domain of interest by describing the concepts and relationships between these concepts. We will refer to this understanding of ontologies as the **standard approach**.

1.1.1 Ways of presenting ontologies

Recently, many languages have been developed for ontology specification. We present only a short overview on ways of ontology representation. We will present a few formal languages and point out how lack of expressivity in one language is overcome by providing another, more expressive language. Using a simple example we will also show what kinds of problems one may face while translating ontologies between various languages.

In general, an ontology may be represented in graphical or textual form. The choice of mode of representation usually depends on the purpose of the ontology representation. If we aim at presenting an ontology in a human readable form then, especially for small ontologies, it might be convenient to choose a graphical mode of representation (as graphs are easy to read and understand for humans). Nevertheless, large ontologies would be better presented in textual form, due to the fact that graphs with large numbers of elements and relations may become hard to read. Also, if we need an ontology to be machine-processable we will use the textual form.

The "Handbook on Ontologies" [73] is a good source of information about ways of representation of ontologies, both graphical and textual.

We start our overview by introducing UML which is a well known language used for graphical representation of ontologies.

UML

The Unified Modeling Language (UML) is a standardized general-purpose modeling language used to specify, visualize, construct and document the artifacts of an object-oriented software system and for object-oriented programming. UML is also used for business modeling, system engineering and for representing organizational structures. It is commonly used with its graphical representation, where its elements are represented as symbols related to each other in diagrams. UML diagrams are commonly used for graphical representation of ontologies. Due to its versatility we cannot talk about one type of UML diagrams, in fact we have several types of diagrams (cf. [68]).

The creators of UML aimed to present a language that is as simple as possible yet expressive enough to model any practical system under development, a language that can be used at many different levels and stages of the software development life cycle. But, as is easy to predict, the requirement to be very expressive made UML a large and varied language.

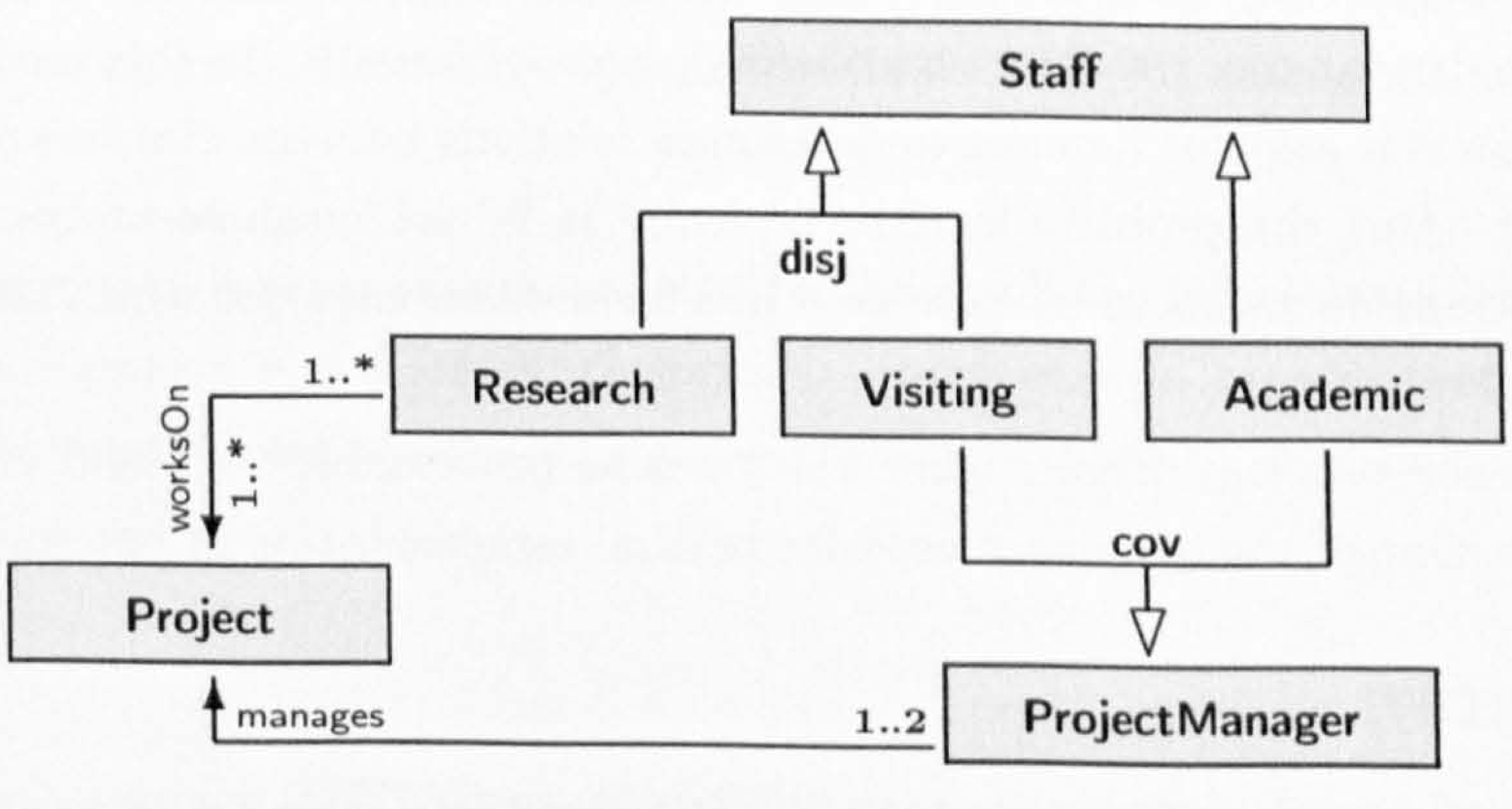


Figure 1.1: An ontology represented as an UML graph

Figure 1.1 presents an example of a simple ontology represented as a UML graph. The ontology uses six concepts: Staff, Research, Visiting, Academic, Project and ProjectManager and two relations: manages and worksOn, together with some restrictions on their use. These concepts and relations form the vocabulary used in the ontology, their understanding should be a result of a consensus among the members of the group interested in the domain. Each concept describes a class of objects from the domain, i.e. in this example objects present in the domain are distributed between six classes. Relation 'IS_A', which is also known as "Supertype-Subtype relationship" or "Parent-Child relationship", is an inheritance relationship providing us with the information about inclusions between the concepts. IS_A relationships are the basis of the hierarchies of concepts. In this example classes Research, Visiting and Academic are subclasses of Staff (we also say that Staff is their superclass). To state that two classes have no common element, we use restriction disj (disjoint), in our example classes Research and Visiting are disjoint. The covering constraint (cov) used in this example tells us that every ProjectManager belongs to Visting or Academic. We also learn that at least one and at most two members of the class ProjectManager manage an element from the class Project. Moreover, every member of Research class works on at least one element from Project class, and at least one member of Research class works on an element from Project class.

Entity-Relationship Diagram

Another method used for graphical representations of ontologies is Entity-Relationship Diagram. It is a graphical representation of entities and their relationships to each other. Entity-relationship modeling was originally developed and is still used as a database modeling method. ER-Modeling was introduced in [25], see [14,80–82] for further information.

The graphical form of ontology representation is very convenient for humans to read. Psychologists have shown that human are able to grasp and remember visual information much faster than the same information presented in textual form [46,66]. A graphical form of representation is very convenient for supporting ontology planning and development, as well as for understanding the structure of the domain of interest. In this mode of ontology representation it is easy for humans to determine relations between classes and restrictions on their use. But the problem is that typically it is not machine-processable. To be machine-processable means to be expressed in a formal language, for which efficient parsers are available.

To make the ontology from Figure 1.1 machine-processable we could convert it into textual form. One of the languages used for textual representation of ontologies is RDF.

RDF and RDFS

The Resource Description Framework (RDF) [47,49,63] is a member of a family of World Wide Web Consortium (W3C) specifications used for supporting mechanisms for metadata schema representation and resource descriptions, in particular Web resources like Web pages. It is often deemed to be the basic representation format used for developing the Semantic Web [47].

Objects of interest are described by simple statements of the type 'subject - predicate - object' by means of vocabularies with named properties. Subjects and objects in these statements are entities and predicates indicate relationships between these entities. However, RDF cannot provide descriptions for these subjects, predicates and objects. In order to specify the information about the terms used in the vocabulary we have to use RDF Schema (RDFS) [22], which is a simple declarative language of restricted expressive power. It provides generic language constructors allowing for semantic characterization of the vocabulary. It introduces a notion of class and property to describe information in the domain and uses semantic relationships for structuring this information. For instance these notions are used for describing hierarchies of concepts using the relationship of subclass and hierarchies of properties using the relationship of subproperty.

To be able to process and store RDF(S) documents we need syntax. Among early propositions of syntax for RDF we can distinguish N3, N-Triples and Turtle. The problem with those was that many programming languages do not offer standard libraries for processing them. In contrast to that, basically every programming language provides libraries for processing XML (Extensible Markup Language) code. For this reason XML is the syntax commonly used for creating RDF(S) documents. RDF/XML is an XML format for representing RDF structures. The specification of RDF/XML can be found in [15].

It is important to notice that contrary to many ontology languages RDFS does not support logical concepts like **equivalence**, and **cardinality constraints** (using cardinality constraints we can for instance declare that every member of Research class works on at least one Project and at least one member of Research class works on a Project). In RDFS we cannot declare characteristics of properties like **transitivity** (like "greater than"), **uniqueness** (like "is father of") or **inverse** of another property (like "ancestor" and "descendant"). Moreover, RDFS has no notion of negation and thus no notion of contradiction and it does not support **disjointness of classes**. RDFS defines the range of a property for all classes and cannot express that certain property ranges apply to some classes only, i.e. we cannot use **properties with local scope**. Due to the lack of negation we cannot declare **Boolean combination of classes**. These concepts are available in OWL, which in a sense extends RDFS.

From the above it is clear that we cannot represent all the information carried by the ontology from Figure 1.1 in RDF/XML format. To be able to present that ontology in XML format we have to use a more expressive language. An example of such a language is OWL.

OWL

Web Ontology Language (OWL) is another W3C recommendation. It was introduced with the aim to provide a language which extends the expressiveness of RDFS but still admits efficient reasoning. During the development of OWL it had to be taken into account that since RDFS modeling primitives, like Class and Property, are very strong, extending RDFS might lead to undecidability of reasoning. To address this issue, three different sublanguages of OWL were introduced. Each sublanguage is an answer to different needs relative to the expressiveness of the language. Below we present a short description of the family of OWL languages based on [4]. It is also one of the sources which could be used for finding more information about OWL.

The most expressive language is called **OWL Full**, it uses all the primitives available in OWL languages. The advantage of OWL Full is that it is fully, syntactically and semantically, upward compatible with RDF. Thanks to that any legal RDF document is also legal in OWL Full, and any valid conclusion in RDF or RDFS is also valid in OWL Full. But its expressivity makes OWL Full undecidable.

OWL DL (an abbreviation of OWL Description Logic) is a restriction of OWL Full. It aims at providing maximum expressiveness together with decidability and the availability of practical reasoning algorithms. To achieve that OWL DL restricts the way that OWL language constructors can be used, for instance, cardinality restrictions[1] may not be applied to transitive properties.

The advantage of OWL DL is that it has efficient reasoning support, i.e., derivations about class membership, equivalence of classes, consistency, classification can be made mechanically. The price for that is that OWL DL is not completely compatible with RDF, even though any legal OWL DL document is a legal RDF document, an RDF document usually will have to be modified to be a legal OWL DL document.

OWL Lite is a further restriction of the language. It restricts the constructors available. For instance, among others, it excludes disjointness and arbitrary cardinality restrictions (it only permits cardinality values of 0 or 1).

Compared with the two other languages OWL Lite is much less expressive but is easier to grasp. It was also expected to be easier to implement, but development of OWL Lite tools turned out to be not much easier than development of tools for OWL DL.

OWL is expressive enough to present the ontology from Figure 1.1 as an XML document. This is shown in the following example:

Example 1.1.1.

In OWL, classes are defined as an `owl:Class` element, `owl:Class` is a subclass of `rdfs:Class`. For example we define a class `Staff` in the following way:

```
<owl:Class rdf:ID="Staff"/>
```

To define classes `Research`, `Visiting` and `Academic` as subclasses of `Staff` we use `rdfs:subClassOf`:

[1] Cardinality restrictions are used for specifying how many distinct values a property may or must take.

```
<owl:Class rdf:ID="Research">
 <rdfs:subClassOf rdf:resource="#Staff"/>
</owl:Class>

<owl:Class rdf:ID="Visiting">
 <rdfs:subClassOf rdf:resource="#Staff"/>
</owl:Class>

<owl:Class rdf:ID="Academic">
 <rdfs:subClassOf rdf:resource="#Staff"/>
</owl:Class>
```

In order to state that every member of class `ProjectManager` also belongs to class `VistingStaff` or `AcademicStaff` first we have to define a class which is a union of `VistingStaff` and `AcademicStaff`. To this end we use `owl:unionOf`, we also use the `rdf:parseType` attribute, which is an abbreviation of an explicit syntax used for building a list with tags `<rdf:first>` and `<rdf:rest>`. Lists of that type are required due to certain limitations of built-in containers of RDF, in particular, there is no way to close them. The reason for this is the fact that while one graph describes some of the members of a class, we cannot exclude the possibility that there is another graph which describes additional members of that class. The list syntax provides that function, but since it is lengthy, the `rdf:parseType` is a convenient abbreviation (cf. [73]).

```
<owl:Class rdf:ID="ProjectManager">
 <rdfs:subClassOf>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="\#Visiting"/>
    <owl:Class rdf:about="\#Academic"/>
   </owl:unionOf>
  </owl:Class>
 </rdfs:subClassOf>
</owl:Class>
```

The following entry declares that the classes `Visiting` and `Research` are disjoint, this is done by using element `owl:disjointWith`. This element can be included in the definition of the class, or can be added by referring to its ID using `rdf:about`, which is inherited from RDF.

```
<owl:Class rdf:about="Visiting">
 <owl:disjointWith rdf:resource="#Research"/>
</owl:Class>
```

The next entry describes an object property (object properties relate objects to other objects and can be understood as relations) `worksOn` with domain in class `Research` and codomain in class `Project`.

```
<owl:ObjectProperty rdf:ID="worksOn">
 <rdfs:domain rdf:resource="Research"/>
 <rdfs:range rdf:resource="Project"/>
</owl:ObjectProperty>
```

Now we extend the information about `Research` by adding the information that every member of the class `Research` is in the relation `worksOn` to at least one element from class `Project`. We use `owl:allValuesFrom` to set the class of the possible values the property determined by `owl:onProperty` can take. In our example we state that all the values of property `worksOn` come from `Project`. We also require that the minimal cardinality of the values the property can take is 1. We have to explicitly state that the literal "1" is to be interpreted as a `nonNegativeInteger`. We also use the `xsd` namespace declaration made in the header element to refer to the XML Schema document.

```
<owl:Class rdf:about="Research">
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#worksOn"/>
   <owl:allValuesFrom rdf:resource="#Project"/>
   <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">
    1
   </owl:minCardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>
```

OWL allows us to use inverse properties, here we use this feature to define `isInvestigatedBy` as the inverse of `worksOn` property.

```
<owl:ObjectProperty rdf:ID="isInvestigatedBy">
 <rdfs:domain rdf:resource="Project"/>
 <rdfs:range rdf:resource="Research"/>
 <owl:inverseOf rdf:resource="#worksOn">
</owl:ObjectProperty>
```

Similarly as above we define property `isInvestigatedBy` to have `owl:allValuesFrom` the class `Research` and we set the minimal cardinality to be 1. In this way we are able to say that for every `Project` there is at least one researcher working on it.

```
<owl:Class rdf:about="Project">
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#isInvestigatedBy"/>
   <owl:allValuesFrom rdf:resource="#Research"/>
   <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">
    1
   </owl:minCardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>
```

The next entry defines property manages in the same manner as was done for other properties above.

```
<owl:ObjectProperty rdf:ID="manages">
 <rdfs:domain rdf:resource="ProjectManager"/>
 <rdfs:range rdf:resource="Project"/>
</owl:ObjectProperty>
```

The final entry is an example of introducing at the same time both minimal and maximal cardinalities of the values the property can take. In this example we say that a `Project` is managed at least by one and at most by two members of `ProjectManager` class.

```
<owl:Class rdf:about="ProjectManager">
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#manages"/>
   <owl:allValuesFrom rdf:resource="#Project"/>
   <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">
    1
   </owl:minCardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#manages"/>
   <owl:allValuesFrom rdf:resource="#Project"/>
   <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">
    2
   </owl:minCardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>
```

An ontology presented in a formal language like OWL is machine-processable but usually it requires a human to make some effort to read and understand it. The reason for developing this language was that XML syntax is widely used for creating web documents, this makes OWL one of the standards used for writing description logic ontologies for the semantic web as it puts description logics into XML syntax

OWL 2

While OWL is very successful and applied in numerous contexts, users have identified some deficiencies in its design. These limitations encouraged the designers to work on a successor of OWL called OWL 2. Development of OWL 2 was also used as an opportunity to clean the language and its specification, this provided a more robust platform for future development.

Based on [44], below we discuss some of the limitations of OWL and how they were addressed in OWL 2. Here we discuss the following groups of problems:

- expressivity limitations,
- syntax issues,
- semantics,
- metamodeling,
- annotations.

Expressivity limitations. Users of OWL have found that its most expressive, but still decidable sublanguage OWL-DL lacks some constructors that are often needed for modeling complex domains. For instance, OWL does not allow for cardinality restrictions to be qualified with a class. That means that while one can provide a definition of a person with at least three children, it is impossible to define a person with at least three children who is male. This problem was solved by introducing qualified number restrictions, which had no impact on decidability and caused no problems with implementation. Another problem is that with OWL we are unable to describe properties in detail. For instance OWL does not allow for propagation along properties or introducing properties of properties. This was addressed in OWL 2 by allowing for complex property inclusion axioms. To avoid undecidability a regularity restriction is imposed on these axioms, which means that complex subproperty axioms should not define properties in a cyclic way. Another limitation is that with OWL-DL it is not possible to express key constraints on data properties, which are an important feature of datatype technologies. But adding keys to languages which are based on description logics leads to theoretical and practical problems. For that reason it was decided to add a restricted versions of keys (known as easy keys), which are useful but relatively easy to implement.

Syntax issues. In OWL we can distinguish two normative types of syntaxes; Abstract Syntax and OWL RDF. The problem is that both syntaxes are difficult to parse correctly and the relationship between them is quite complex, which leads to some difficulties when an ontology is transformed from one syntax to another. Another difficulty is that despite the fact that OWL is based on description logics, Abstract Syntax of OWL does not correspond exactly to the constructors used in DLs. This led to some confusion among developers of OWL APIs, who would rather follow DL structure. In addition, neither Abstract Syntax nor OWL RDF is fully context free, axioms containing URI often do not provide sufficient information to determine if they refer to a class, property, or an individual. Moreover, RDF syntax often proves to be difficult to use. This is because in RDF everything is represented with triples, but in OWL many constructors cannot be represented in that way without introducing new objects. As a result OWL RDF ontologies are difficult to read and process. In OWL 2 Abstract Syntax was replaced with Functional-Style Syntax. These two differ in many ways and OWL 2 Functional-Style Syntax is not backwards compatible with Abstract Syntax. The most important difference is that Functional-Style Syntax does not contain the frame-like syntactic constructors of OWL (which were causing some confusion among the developers), but describes ontology entities using axioms. OWL RDF, which was mainly used for publishing ontologies on the Web, was replaced in OWL 2 by XML Syntax. The main advantages of XML Syntax are its ease to parse and process, and the fact that it is a widely used format on the Web and is supported by a number of tools. In addition XML Syntax is well suited for use in protocols and APIs for accessing OWL 2 implementations.

Semantics. When OWL was under development, it was expected to be compatible with Semantic Web languages like RDF, whereas OWL-DL was originally designed as a notational variant of $\mathcal{SHOIN}(\mathbf{D})$. But the differences in semantics of RDF and $\mathcal{SHOIN}(\mathbf{D})$ were causing some difficulties. To overcome that, two coexisting semantics were introduced for OWL. But this solution caused other difficulties, as both semantics had their own problems and bringing them together was complicated as well. This was addressed in OWL 2 by introducing model-theoretic semantics that corresponds to $\mathcal{SROIQ}(\mathbf{D})$. Moreover it was defined for ontologies in Functional-Style Syntax and, as there is a one-to-one correspondence between this syntax and XML Syntax, the semantics can be directly used in the latter representation. At the moment, OWL 2 does not provide RDF-style semantics but the design of such a semantics is in progress. In the meanwhile the transformation of ontologies presented in Functional-Style Syntax into RDF graphs is a purely syntactic process.

Metamodeling. As practice shows the distinction between classes and individuals sometimes is not entirely clear. Sometimes the same concept name in one context plays the role of an individual and of a class in another context, this is called metamodeling. During the development of OWL the importance of metamodeling was not widely recognized yet and for that reason it is available only in OWL Full, but it was introduced in a way which leads to undecidability of standard reasoning problems. In the metamodeling semantics in OWL 2 the usage of a name as a class is unrelated to its usage as an individual (this is achieved by adding a prefix which tells in what context the name is used), consequently names of concepts and individuals do not interact even if they are the same. This type of metamodeling is often called punning.

Annotations. While the annotation system (extra-logical information describing ontologies or entities) does not restrict the use of annotations, the annotation system used in OWL-DL has been identified as insufficient. For example, OWL-DL does not allow axioms to be annoted, which sometimes is necessary, for instance, if one needs to indicate the origin of the information (i.e. who introduced a particular axiom). Users were also dissatisfied with the fact that they could not define domains and ranges of annotation properties. To answer those issues, in OWL 2 it is possible to annotate entities, axioms and ontologies. Annotations in OWL 2 do not carry formal semantics, thus they do not have any impact on the set of consequences derived from an ontology. On the other hand, annotations do affect structural equivalences of ontologies, i.e., for two ontologies (and similarly axioms) to be structurally equivalent they need to have structurally equivalent annotations on them.

Profiles in OWL 2. Although OWL was originally designed with three sublanguages in practice it is difficult to determine which sublanguage was used for ontology formulation. This is particularly difficult for ontologies formulated in RDF, as there is no direct mapping between RDF and Abstract Syntax. In addition it was found that even though OWL Lite is much simpler than OWL DL, the complexity was not reduced significantly, this is due to the fact that negation can be implicitly formulated in axioms. In OWL 2 these difficulties were addressed by designing sublanguages which give up some expressive power to gain efficiency of reasoning, these sublanguages are called OWL 2 profiles. The profiles are defined by placing restrictions on the Functional-Style Syntax of OWL 2 and were designed for different reasoning tasks. We distinguish three profiles:

- OWL 2 EL is based on the $\mathcal{EL}^{++}$ family of description logics. This profile was designed to achieve efficient reasoning (which in this case usually is classification of concepts) with large ontologies. The profile captures the expressive power used by many ontologies, and will be usually used for classification. The reasoning can be performed in time that is polynomial with respect to the size of the ontology.

- OWL 2 QL is based on the DL-Lite family of description logics, which was originally designed to provide efficient reasoning with large volumes of instance data. The profile is used for answering conjunctive queries, i.e., given an ontology $\mathcal{O}$ and a conjunctive query q, we want to compute all tuples of individuals that constitute an answer to q with respect to $\mathcal{O}$. With the use of a suitable reasoning technique, sound and complete conjunctive query answering can be done in LogSpace with respect to the size of the data. The expressive power of the profile is quite restricted, but it has most of the main features of conceptual models such as UML class diagrams and ER diagrams.

- OWL 2 RL was designed to support applications that require scalable reasoning without giving up too much expressive power. The design of OWL 2 RL allows for implementing reasoning tasks as a set of rules. While OWL 2 RL uses most of the constructors used in OWL 2, to allow for rule-based implementations of reasoning, the way they are used was restricted to ensure that a reasoner needs to take into account only the individuals that are explicitly used in the ontology. The typical reasoning problems like ontology consistency, class subsumption, instance checking, and conjunctive query answering can be solved in time that is polynomial with respect to the size of the ontology.

none of the profiles is a subset of another.

First Order Logic

In the 1970's it was suggested that First Order Logic (FOL) could be the source of precise semantics for knowledge representation, with unary predicates used to make assertions about individual objects and binary predicates denoting relations between objects. This approach was based on the idea that FOL is well known and could be used to express facts about the world in an unambiguous way.

For instance the ontology from Figure 1.1 can be represented with the following axioms:

Example 1.1.2.

$$\forall x(\exists y\ \mathit{manages}(x,y) \Rightarrow \mathit{ProjectManager}(x)),$$
$$\forall x(\exists y\ \mathit{manages}(y,x) \Rightarrow \mathit{Project}(x)),$$
$$\forall x(\mathit{Project}(x) \Rightarrow \exists y\ \mathit{manages}(y,x)),$$
$$\forall x,y_1,y_2,y_3((\mathit{manages}(y_1,x) \wedge \mathit{manages}(y_2,x) \wedge \mathit{manages}(y_3,x)) \Rightarrow (y_1 = y_2 \vee y_1 = y_3 \vee y_2 = y_3)),$$
$$\forall x(\exists y\ \mathit{worksOn}(x,y) \Rightarrow \mathit{Research}(x)),$$
$$\forall x(\exists y\ \mathit{worksOn}(y,x) \Rightarrow \mathit{Project}(x)),$$
$$\forall x(\mathit{Research}(x) \Rightarrow \exists y\ (\mathit{worksOn}(x,y))),$$
$$\forall x(\mathit{Project}(x) \Rightarrow \exists y\ (\mathit{worksOn}(y,x))),$$
$$\forall x(\mathit{Research}(x) \Rightarrow \mathit{Staff}(x)),$$
$$\forall x(\mathit{Visiting}(x) \Rightarrow \mathit{Staff}(x)),$$
$$\forall x(\mathit{Academic}(x) \Rightarrow \mathit{Staff}(x)),$$
$$\forall x(\neg \mathit{Visiting}(x) \vee \neg \mathit{Research}(x))$$
$$\forall x(\mathit{Visiting}(x) \Rightarrow \mathit{ProjectManager}(x)),$$
$$\forall x(\mathit{Academic}(x) \Rightarrow \mathit{ProjectManager}(x)),$$
$$\forall x(\mathit{ProjectManager}(x) \Rightarrow \mathit{Academic}(x) \vee \mathit{Visiting}(x)).$$

The problem with FOL is that its high expressivity leads to undecidability of the satisfiability problem and consequence relation.

This together with the fact that it was argued that often full expressivity of FOL is not needed for ontology formulation, was an incentive to take into consideration fragments of FOL as languages for ontology formulation. Using fragments of FOL makes the complexity of reasoning lower and allows for efficient use of ontologies. Of course different fragments of FOL differ in expressivity and complexity. Many description logics can be regarded as fragments of FOL.

Description Logics

Description Logics (DLs) are a family of knowledge representation languages, that provide formal foundations for ontology representation and for performing the tasks related to the use of ontologies [9]. The name itself points to two important aspects of DLs. First of all DLs use concept descriptions to describe the domain of interest. To this end expressions built from atomic concepts (corresponding to unary predicates in *FOL*) and atomic roles (corresponding to binary predicates in *FOL* and object properties in OWL) with use of concept and role constructors are introduced. The other important aspect of DLs is that they have a formal, logic-based semantics.

Members of the DL family differ in their expressivity and complexity. In this family we have expressive languages like $\mathcal{ALC}$, $\mathcal{ALCQI}$ or $\mathcal{ROIQ}$ but with high complexity (at least ExpTime-hard) of reasoning, and languages like $\mathcal{EL}$ or $\mathcal{EL}^+$ which are less expressive but in which reasoning is tractable. Interestingly, weak languages like $\mathcal{EL}$ and $\mathcal{EL}^+$ were found to be expressive enough to formulate large scale ontologies. In fact several ontologies used in medicine are formulated in lightweight description logics such as $\mathcal{EL}$ or mild extension thereof, e.g., $\mathcal{EL}$ with role inclusions. In particular, role inclusion axioms expressing role hierarchies, transitive roles, and right-identity are of practical importance, e.g. in medical terminologies [48, 72].

Here we briefly discuss some typical constructors of DLs and give an overview of commonly used DLs. The formal definitions of the systems of our interest will be provided in Section 2.2.

All the members of the DL family share the same type of alphabet. This alphabet consists of two disjoint sets: one is used to denote **atomic concepts,** and the other one is used to denote **atomic roles.** The former are used to express classes of objects of the universe, whereas the latter are used to represent the relations between these objects.

Concept expressions denote sets of all individuals satisfying a property described by the concept. Description logics use different types of connectives to express more complex concepts. For instance, **intersection of concepts,** denoted by $C \sqcap D$, is used to represent the individuals that belong both to C and D, the corresponding first order logic expression would be of the form $C(x) \wedge D(x)$. We also have **complement of concept** (denoted by '$\neg$') and **concept disjunction** (denoted by '$\sqcup$').

One of the most important aspects of description logics is their ability to describe relations between the concepts of the domain. This is achieved by using role names (from the set of atomic roles) together with constructors (role restrictions) establishing that relation. We distinguish four types of role restrictions: **value restriction, existential restriction, number restriction** and **qualified number restrictions.**

Value restriction allows us to describe concepts like "individuals all of whose pets are dogs", formally expressed as $\forall$hasPet.Dog. Existential restriction (existential quantification) allows us to describe concepts like "individuals having a dog as a pet", formally $\exists$hasPet.Dog. Number restriction enables us to express concepts like "individuals having at least 2 children" and "individuals having at most 3 brothers" written respectively as ≥ 2 hasChildren and ≤ 3 hasBrother. Finally, qualified number restrictions are used to formulate concepts like "individuals having at least 2 adult children" and "individuals of whose children at most 3 are male", and respectively these sentences are of the form ≥ 2 hasChild.Adult and ≤ 3 hasChild.Male.

Description logics allow for concept definitions $A \equiv C$ and general concept inclusion axioms (GCIs) $C_1 \sqsubseteq C_2$. For instance, we can build expressions like

$$\mathsf{Father} \equiv \mathsf{Male} \sqcap \mathsf{Parent}$$

telling us that being Father is equivalent to being Male and Parent. An instance of a GCI is the following:

$$\mathsf{Mammal} \sqsubseteq \mathsf{Animal}$$

which tells us that concept Mammal is more specific than concept Animal, i.e. every Mammal is also Animal.

A collection of GCIs is called a Tbox. In the DL context we use the terms "TBox" and "ontology" interchangeably.

Description logics allow also for role-forming connectives, for instance using **intersection of roles,** intuitively hasChild $\sqcap$ hasFemaleRelative yields the role hasDaughter.

As we can formulate concept hierarchies in description logics, we can also express conditions on roles such as $r \sqsubseteq s$, transitivity $r \circ r \sqsubseteq r$ and right-identity $s \circ r \sqsubseteq s$, where r and s are role names. Right-identity axioms have been proven useful for expressing "propagation" of one property along another one. For instance, we can formulate axioms of the form:

1. isPartOf $\circ$ isPartOf $\sqsubseteq$ isPartOf.

2. hasLocation $\circ$ isPartOf $\sqsubseteq$ hasLocation

If we use these axioms together with:

3. Toe $\sqsubseteq \exists$isPartOf.Foot

4. Foot $\sqsubseteq \exists$isPartOf.Leg

Then using (1), (3) and (4) we can infer that Toe $\sqsubseteq \exists$isPartOf.Leg, i.e. that a toe is a part of a leg. Whereas, given Injury $\sqcap \exists$hasLocation.Toe, we can use (2) to infer that Injury$\sqcap\exists$hasLocation.Foot i.e., if one suffers from an injury located in a toe then it also means that one suffers from an injury located in a foot, and further that Injury$\sqcap\exists$hasLocation.Leg, i.e., that it also means that one suffers from an injury located in a leg.

Having various connectives available and allowing for different types of axioms we can create diverse description logics by allowing various combinations thereof. Clearly these logics differ in expressivity and complexity.

As already mentioned, a very simple, yet very important, description logic is $\mathcal{EL}$ [8]. Its concept constructors are $\top$ (denoting the whole domain), conjunction and existential restriction.

As $\mathcal{EL}$ has only two constructors available, we cannot use it to express the UML diagram from Figure 1.1. An important extension of $\mathcal{EL}$ in which the UML diagram from Figure 1.1 can be expressed is $\mathcal{ALCQI}$, which in addition to $\mathcal{EL}$ has negation, inverse roles and qualified number restrictions. The ontology expressing the UML diagram from Figure 1.1 is now:

Example 1.1.3.

$$\begin{aligned}
\exists manages &\sqsubseteq ProjectManager,\\
\exists manages^- &\sqsubseteq Project,\\
Project &\sqsubseteq \exists manages^-,\\
\geq 3 manages^- &\sqsubseteq \bot,\\
\exists worksOn &\sqsubseteq Research,\\
\exists worksOn^- &\sqsubseteq Project,\\
Research &\sqsubseteq \exists worksOn,\\
Project &\sqsubseteq \exists worksOn^-,\\
Research &\sqsubseteq Staff,\\
Visiting &\sqsubseteq Staff,\\
Academic &\sqsubseteq Staff,\\
Visiting \sqcap Research &\sqsubseteq \bot,\\
Visiting &\sqsubseteq ProjectManager,\\
Academic &\sqsubseteq ProjectManager,\\
ProjectManager &\sqsubseteq Academic \sqcup Visiting.
\end{aligned}$$

Another group worth mentioning is the DL-Lite family of description logics. The DL-Lite logics were designed to provide efficient access to large data repositories, without giving up too much of expressive power. It is assumed that the data to be accessed are stored in a standard relational database and that the user is interested in formulating, with use of an ontology, queries that are more complex than asking for instances of single concepts and roles (instance checking), for example one could formulate conjunctive queries. DL-Lite family have polynomial time computational complexity with respect to standard reasoning tasks, and LOGSPACE data complexity with respect to complex query answering. These logics were first proposed by Calvanese et al. in [24]. They are also studied in [6], where DL-Lite is extended with full Booleans and number restrictions, resulting in DL-Lite$_{bool}$. The authors also introduce its two sublanguages DL-Lite$_{horn}$ and DL-Lite$_{krom}$. Some interesting properties of members of DL-Lite family are also studied in [52–54].

Here we will consider DL-Lite$_{bool}$ only. As other description logics, DL-Lite$_{bool}$ has concept names and role names in the vocabulary, it has $\top$ and $\bot$ concepts available, it also allows for use of existential restriction, conjunction, negation, number restriction and inverse role constructor. Having these constructors available, DL-Lite$_{bool}$ is expressive enough to cover all the axioms of the Example 1.1.3 as they were originally formulated, but the reasoning now is coNP-complete [6].

1.1.2 Formal semantics and reasoning

When an ontology is under development typically it may not be convenient, or even possible, to present all the required information explicitly. But not all the information is actually required to be explicit (this allows to reduce the size of the ontology, which is important from a practical point of view, as using smaller ontologies in applications is more time effective). Many of the languages used to write ontologies are *logics*, whose semantics provide a notion of consequence that provides implicit information. To obtain this information, it is necessary to query an ontology: to deduce or infer information that is implicit in the ontology's statement. We may wish to obtain the induced concept hierarchy, or to access instance data using the ontology (e.g., as in Ontology-Based Data Access [24]). The form of the queries corresponds to the applications of ontologies. Important types of queries include three which are of interest to us:

Classification. Classification of an ontology aims to compute all the subclasses of atomic classes present in the ontology. For instance, let Σ be a vocabulary of medicine, then the query

$$\mathsf{Pseudopseudohypoparathyroidism} \sqsubseteq \mathsf{Genetic\_Disorder}$$

asks if it follows from the ontology that pseudopseudohypoparathyroidism is a genetic disorder.

Answering subsumption queries. This application aims to compute subsumption between complex concepts. Again, in a vocabulary of medicine, a query

$$\mathsf{Cystic\_Fibrosis} \sqsubseteq \mathsf{Fibrosis} \sqcap \exists \mathsf{located\_In.Pancreas} \sqcap \mathsf{has\_Origin.Genetic\_Origin}$$

asks if it is the case that Cystic Fibrosis is always a fibrosis which is located in the pancreas and has genetic origin.

Answering instance data queries. Another important application of ontologies is their use when one is querying instance data. In this scenario we are interested in instance queries that are posted to a pair consisting of an ontology and an ABox, which stores instances of classes and relations but is not a part of an ontology. For example, if Σ is a medical vocabulary, then a query might consist of the ABox of the form:

$$\mathcal{A} = \{\mathsf{Patient(John)}, ..., \mathsf{Broken\_Leg(John)}\}$$

together with the conjunctive query of the form

$$\mathsf{Treated\_in\_Orthopedic\_Unit(John)}.$$

This query asks if the ABox $\mathcal{A}$ and the ontology entail that John is treated in an orthopedic unit. If the ontology states for instance that every patient with a leg broken is treated in orthopedic unit, then the answer is yes.

To be able to answer any query first we need formal semantics allowing us to define a notion of consequence which is essential in any logical system. Defining what it means to be a consequence enables us to answer queries and concept classifications. As already mentioned we have a number of logical systems used for ontology representation that provide us with formal semantics. This gives us a good choice for finding one that is expressive enough to describe the domain of our interest. But the expressivity of the language has impact on its complexity. For instance, we may use *FOL* for formulating ontologies, but this leads to undecidability of some interesting problems and full expressivity of *FOL* is not always needed. For that reason often we would rather choose a fragment of *FOL* for instance one of the description logics mentioned above. The complexity of many problems in various DLs is well studied, unfortunately often it is high. For instance, Σ-entailment[2] in $\mathcal{ALCQIO}$ is undecidable; while Σ-entailment in $\mathcal{ALC}$ and $\mathcal{ALCQI}$ is 2ExpTime-complete, as it was shown in [40,60]. But it has been shown in [61,62] that Σ-entailment in $\mathcal{EL}$ becomes ExpTime-complete. For that reason choosing the language for ontology representation often is about finding the right balance between expressivity and complexity. In recent years we can observe an increasing interest in lightweight description logics. Their popularity is caused by the fact that reasoning is tractable even w.r.t. general sets of concept inclusions. As described above, we can distinguish two groups of light weight DLs, the $\mathcal{EL}$ [8, 21, 61] family of tractable DLs and the family of DL-Lite tractable DLs.

Having chosen the formalism, it is possible to investigate many interesting problems, like concept subsumption, various notions of entailment, inseparability or the related logical difference problem, as introduced below in this thesis.

1.2 Problems

In the standard approach, the function of an ontology is to state, explicitly, a conceptualisation. However, as well as reading and writing ontologies, in practice one also wants to *use* existing ontologies, perhaps to browse the induced concept hierarchy, or to access instance data, or perhaps to create a new ontology that extends either an entire ontology or a manageably small fragment of one. Or perhaps one may want to test whether one ontology is in some way consistent with another, or provides the same information regarding some subset of concepts.

These applications have been found to be beneficial in various fields of science and commerce. This includes disciplines like medicine, bio-informatics, the Semantic Web, artificial intelligence, software engineering and others.

In bio-informatics, for instance, ontologies are used in order to get answers to biological questions. Researchers use mathematical and computational techniques to manage and analyze biological concepts. This is due to the fact that the experimental way of testing hypotheses in biology is rather expensive in time and resources. Thus it was found useful to compute information in order to test hypotheses of interest. A common way to use

[2] Σ-entailment is formally introduced in Definition 3.2.16. Roughly stated, if ontology $\mathcal{O}_1$ Σ-entails ontology $\mathcal{O}_2$ it means that every sentence over Σ that is a consequence of $\mathcal{O}_2$ is also a consequence of $\mathcal{O}_1$.

ontologies in bio-informatics is creating a common vocabulary used for managing database annotation like describing, linking, sharing and querying—for instance, MGED[3] (Microarray Gene Expression Data) is a project focussed on establishing standards for microarray data annotation and exchange in order to support the creation of microarray databases and implementing these standards in software.

An example of ontology used in medicine is the Systematized Nomenclature of Medicine (SNOMED CT). It is used in the health systems of the US, the UK, and other countries [72]. It comprises ~0.5 million concepts and describes many of the aspects of medicine and health care. Another example is the thesaurus of the US National Cancer Institute (NCI), which comprises ~45.000 concepts and is designed to become the reference terminology for cancer research [74]. SNOMED CT (similarly NCI) satisfies the need for unified clinical information exchange between different health care providers, researchers and others. It also helps to deal with the problem of differences in the ways of recording the medical information and differences in the terminology. It is used in applications like Electronic Medical Records, Genetic Databases or Cancer Reporting.

As a consequence of this interest in ontologies their number is continuously increasing. For that reason it is often the case that within one field there are several ontologies describing one domain of interest. Often they complement each other by focusing on different aspects of that domain, this usually happens if they were designed to satisfy diverse needs of varied groups of users. Alongside the growth of importance of ontologies we can observe increasing interest in multiple use of ontologies. But this may raise some problems. We distinguish different dimensions of the possible problems, but they can appear at the same time as well.

1. **Ontologies formulated in distinct formalisms.** One problem originates in the requirement that ontologies be formal. It obviously helps to meet this requirement if the ontology is written in a precise language, and as already mentioned a large number of languages have been developed specifically for writing ontologies. Thanks to that variety of ontology representation languages we can find a good balance between expressivity and complexity of the formalism used for ontology representation. It is important to bear in mind that it is not the case that one language or notation is 'better' than all others, but one may be found more appropriate than others for a particular domain or application. In any case, choosing a particular notation for an ontology restricts the use that may be made of it. Most likely, an ontology, or its component parts, written in one language may be incorporated only into other ontologies that are written in the same language; in order to combine two or more existing ontologies, it may be necessary to translate at least one of them into another language.

2. **Different formalisms used for ontology and queries.** Another problem stems from the requirement that ontologies be explicit. As already mentioned, it may not be convenient, or even possible, to present all the required information explicitly in

[3] http://mged.sourceforge.net/ontologies/index.php

an ontology. To obtain the desired information, it is necessary to query an ontology. Obviously, an answer to a query may best be obtained when the query and the ontology are written in the same logical language; if they are not then, again, the ontology or the query may need to be translated into another language.

3. **Different signatures.** But even if the formal language is fixed, different ontologies may use different vocabularies. For instance, this may happen if groups of primary ontology users use different natural languages and thus concept and role names are from different natural languages. Again, this makes ontologies difficult to use together. (Similarly we may have an ontology and a query using different signatures.) To overcome it we have to find a correspondence between the vocabularies, see e.g. [38].

The problems described in points (1) and (2) are similar and require bringing distinct formalisms together, which in practice may be a difficult task. The problem described in point (3) requires signature mapping. As an example of a problem of type (3), we may consider SNOMED CT. Originally it was created in English, but as it describes many useful aspects of medicine it was found to be useful to have it available also in other languages. To do that it was necessary to determine mappings between different signatures used in particular languages to describe medical terms. To this end SNOMED CT has a built-in framework to manage different languages and dialects. This made it possible to provide SNOMED CT in US English, UK English, German, Spanish and Danish [1], and translations into French, Swedish, Lithuanian, and several other languages are currently being undertaken. To make sure that these translations are accurate, usable and safe they have to be concept-based (term-to-term translations may return literal expressions that are often meaningless). Therefore before deciding on the translation each concept has to be analyzed by the translator relative to its description, position in the hierarchy and relations to other concepts.

Another problem that might appear is caused by synonyms and homonyms in natural languages, but as already mentioned, concept and role names used in ontologies are the result of a consensus among the users and this consensus aims at preventing from introducing synonyms and homonyms into the signature. Nevertheless it should be taken into account while merging two ontologies, especially if they were developed for different groups of users (for instance groups working in different areas of science). This problem may also appear during querying ontologies if the user for some reason is not aware of that agreement.

Of course a combination of the above problems may appear, so we may have two ontologies (or an ontology and a query) expressed in different formalisms, with their vocabularies also coming from different natural languages. In some sense, those who are developing and using ontologies may face problems similar to those that the builders of the Tower of Babel had. A great amount of work has been done gathering a great amount of knowledge in a vast number of ontologies. But these ontologies are often formulated in different formal languages and use different vocabularies and thus in many cases we are unable to use them together.

1.3 Solution

As the core of our solution to these problems we propose a fresh view on what ontologies are. Contrary to the standard approach our primary focus is not on the way ontologies are built or what formalisms are used to construct them, but on the way we can use them. We want to identify an ontology with its function. Thus we adopt an abstract view of an ontology as a black box providing answers to queries about some vocabulary of interest. We call this the **functional approach.** A similar approach has been proposed for knowledge representation by Levesque et al. [20] and [59]. In their work the functional view of a knowledge base is characterized by determining what can be told or asked, without taking into account the structures used to represent knowledge. This was the foundation for designing and implementing a knowledge-representation system called Krypton.

Identifying the function of ontologies is also important for modularity of ontologies. The significance of modularity may be observed in ontology development and application. In ontology development, it allows for reuse of existing ontologies and also for distributing work among independent groups of designers when developing a large scale ontology. In ontology application, modularity allows for reasoning over the relevant part of ontology only, and thus increases the efficiency of reasoning.

To say that some part of an ontology is a module means that this part may function independently. Therefore we have to set the context in which the ontology (and its part) are to function. That is, we have to fix a language that will be used to answer queries. This allows us to say that two ontologies are equivalent if in a fixed language they give the same answers to all queries over a fixed vocabulary. For a module of an ontology, we say that it functions independently if in a fixed language both the module and the entire ontology give the same answers to all queries over a fixed vocabulary.

These intuitions are reflected in the concept of **inseparability.** In the standard approach two ontologies are indistinguishable if and only if they have the same axioms. But as observed in [65] computing the syntactic difference between ontologies consisting of axioms is hardly useful. Whereas in the functional approach two ontologies are indistinguishable with respect to some vocabulary Σ if they are inseperable with respect to the fixed query language over Σ, regardless of what the axioms are and what formalisms are used in their formulation. One of the benefits of this approach is that instead of a big ontology we can freely use a part which has the same consequences relative to the vocabulary of interest: then we say that this part of an ontology can 'function independently'. Similarly, we can freely replace different ontologies with each other if they are indistinguishable in the functional approach.

Even though this change of view on ontologies allows us to compare ontologies with each other and determine if they give the same answers to queries over some signature, it does not solve all the problems. The functional approach is just the first step towards working with multiple ontologies; it allows us to tell what are the consequences of an ontology relative to some signature in the context of some formal language. But the functional approach still does not tell us how to bring two arbitrary ontologies together in order to compare or merge them. Moreover it does not solve the problem of different signatures, even if the ontologies are constructed with the use of the same formalism. For that reason we still need a bridge which allows us to bring together arbitrary ontologies, regardless of their signatures and the formalisms used for their construction. Similarly, we want it to allow us to bring together an ontology and a query and answer the query even if they use different formalisms and their signatures originate from different natural languages. In our work we propose a theoretical foundation of a comprehensive method which provides these bridges. We introduce a notion of **framework** that captures the situation of a 'global' language into which both an 'ontology' language and a 'query' language can be translated, in a more general and abstract way. To do this, we use **institutions**, which were introduced by Goguen and Burstall [41, 42] to treat logics with model-theoretic semantics in a systematic way. A great many logics can be represented as institutions [34], including many logics for specifying ontologies [57]. An advantage of this is that institutions also allow a systematic treatment of translation between languages, and recent research has applied this to problems such as alignment and integration [28, 55, 56, 71, 85]. Within a framework, it is possible to capture a general notion of consequence, whereby an ontology answers a query, when both are translated into the global language. This in turn gives rise to a language-independent notion of **inseparability**, i.e. we can compare two ontologies formulated in different formalisms (and possibly with different signatures). Figure 1.2 is a graphical representation of a framework having two ontologies expressed over two distinct languages $\mathcal{L}_1$ and $\mathcal{L}_2$, which are compared in the global language $\mathcal{G}$ relative to queries originally formulated in a query language $\mathcal{Q}$.

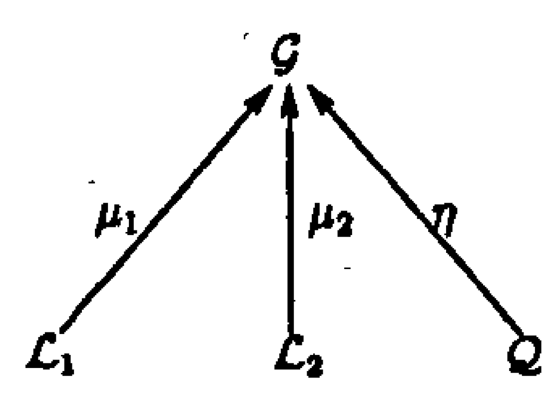


Figure 1.2: A framework

In our work we explore properties of that construct by considering various scenarios, in which we take into account various configurations of languages of different strength. We investigate robustness properties of frameworks and the inheritance of the Craig interpolation property.

We also investigate the problem of using ontologies together with ABoxes and determining Σ-entailment and Σ-inseparability of ontologies in the presence of ABoxes. But this requires extending the signature with individuals. In other words we have to show how a description logic can be extended to a description logic with individuals. To do that we use the theory of institutions. After we show how to introduce individuals into the signature we investigate the relations between institutions of description logics and their counterparts with individuals. We also show how to extend a framework to a framework allowing for expressions with individuals. We also present an institution independent formulation of

Σ-entailment and Σ-inseparability of ontologies based on instance checking which allows us to compare arbitrary ontologies in the presence of ABoxes.

1.4 Novel Contributions

The novel contributions that can be found in this thesis consist of the following.

- Introducing the functional approach to ontologies, which gives a new, general perspective on ontologies abstracting from the particular ontology language used.
- Introducing a general notion of framework which provides a language in which both the ontology and the query languages are translated by means of institution comorphisms. This new approach allows to combine ontologies independently of the formal language and signature used for their formulation.
- Presenting institution independent formulations of notions of inseparability, conservative extension and robustness. As well as presenting in an institution independent way results describing how robustness under vocabulary extension, robustness under joins and robustness under replacement in a framework are related to the Craig interpolation property.
- Approaching the problem of working with ontologies together with ABoxes in the same general manner. Showing how any description logic can be extended to a description logic with individuals and how a framework for description logics can be extended to allow for instance checking queries. This includes investigating the relations between institutions of description logics and their counterparts with individuals.
- Formulating an institution independent notions of query conservativity, query expansion and concept interpolation.
- Establishing that the Σ-entailment problem for ontologies formulated in the description logic $\mathcal{ELSH}$ can be solved in ExpTime. This extends the result presented in [61] for $\mathcal{EL}$.

1.5 Thesis Outline

The structure of the thesis is the following.

In Chapter 2 we shortly present description logics which are used in many examples of our work. This chapter introduces also the central notions of category theory and thus builds the foundations for introducing the theory of institutions which plays a central role in our research. The chapter also introduces the theory of institutions itself. As part of this introduction we discuss how particular logical systems can be viewed as institutions and how we can relate different institutions with each other.

In Chapter 3 we discuss how, with the use of the theory of institutions, we can build a construct (called a framework) which allows us to query an ontology even if the query and the ontology are given in different formalisms and possibly different vocabularies. We also show how frameworks allow us to compare and use together arbitrary ontologies. We define, in a way that does not depend on a particular formal language, the consequence relation and entailment between ontologies with respect to some vocabulary of interest (called Σ-entailment). This chapter also introduces different types of robustness and investigates relations between interpolation properties and robustness.

In Chapter 4 we investigate how to build frameworks with description logics and how to use them for answering queries about hierarchies of concepts. This involves providing a definition of what a description logic is in the institutional setting. We also investigate the problem of answering queries with individuals (instance checking). To make this type of reasoning feasible in frameworks, we first show how to formally introduce individuals into the signatures of institutions. After showing that given an institution of description logic we can construct a corresponding institution of description logic with individuals, we investigate basic properties of this construction. We also show that given a framework we can generate a framework which allows for queries about individuals. We investigate relations between these constructs, among other aspects we investigate how Σ-entailment is inherited between both types of frameworks.

In Chapter 5 we use a particular type of framework to investigate the problem of Σ-entailment for ontologies formulated in the description logic $\mathcal{ELSH}$ obeying some additional restrictions. The main result states that the Σ-entailment problem for such ontologies can be solved in ExpTime. Finally, in the appendix we present an alternative formulation of a framework which uses a dual concept to that used in Chapter 3.

Chapter 2

Preliminaries: Description Logics, Category Theory and Theory of Institutions

2.1 Introduction

In this chapter we introduce the logics considered in this thesis and provide a brief introduction to category theory and the theory of institutions.

The first section formally introduces some of the logical systems that were mentioned in Chapter 1 as formal languages used for ontology formulation. Here we present description logics $\mathcal{EL}$, $\mathcal{EL}^+$, $\mathcal{ELSH}$, $\mathcal{ALC}$, and first order logic. These logical systems will be used in many examples throughout the thesis. We also shortly mention various extensions of $\mathcal{ALC}$.

The second part of the chapter forms a short introduction to category theory. In our approach category theory is important as it provides the foundations for introducing the theory of institutions (the notion of institution itself strongly relies on concepts from category theory) and provides us with tools that will be used throughout the thesis allowing us to treat logical systems, and investigate their properties, in a systematic and general way.

In the final part we introduce the theory of institutions and its basic notions. We present the intuitions behind it, show how the logical systems introduced earlier form institutions and what are the relations (morphisms and comorphisms) between these institutions. The reason why in our work we look at the theory of institutions is the fact that it allows us to describe logical systems in an abstract and general way. It also allows to present and solve problems independently of any particular logical system. In the later chapters we will use the theory of institutions to construct structures allowing us to use an ontology for answering a query even if they are formulated in different formal languages and use different signatures. In the similar way we will present how these structures allow us to work with multiple ontologies even if they are formulated in distinct formalisms. With the use of the theory of institutions we will show how to introduce individuals into the signatures of description logics and how to construct assertions about individuals.

2.2 Description logics and first order logic

Even though most of our work is presented in a way that does not depend on a particular logical system, many examples use FOL, $\mathcal{EL}$, $\mathcal{EL}^+$, $\mathcal{ALC}$, and their versions with individuals. Therefore we find it useful to introduce formally these systems.

2.2.1 Description logic $\mathcal{EL}$

We begin with introducing description logic $\mathcal{EL}$. Concepts in $\mathcal{EL}$ are build according to the following syntax rule:

$$C ::= \top \mid A \mid C \sqcap D \mid \exists r.C,$$

where A ranges over the **concept names** taken from a countably infinite set P, r ranges over **role names** taken from a countably infinite set R, and C, D over $\mathcal{EL}$-concepts. An $\mathcal{EL}$ **terminology** (TBox) is a finite set of **concept inclusions** (CIs) $C \sqsubseteq D$, where C and D are $\mathcal{EL}$ concepts, an ABox is a finite set of **concept assertions** $C(a_i)$ and **role assertions** $r(a_i, a_j)$, where a_i, a_j range over a countably infinite set I of **individual names**. A **knowledge base** (KBox) $\mathcal{K}$ is a pair $(\mathcal{O}, \mathcal{A})$ with $\mathcal{O}$ a TBox and $\mathcal{A}$ an ABox.

The semantics of $\mathcal{EL}$ is defined by means of **interpretations** $\mathcal{M} = (\Delta^{\mathcal{M}}, \cdot^{\mathcal{M}})$, where the interpretation **domain** $\Delta^{\mathcal{M}}$ is a non-empty set, and $\cdot^{\mathcal{M}}$ is a function mapping each concept name A to a subset $A^{\mathcal{M}}$ of $\Delta^{\mathcal{M}}$, each role name $r^{\mathcal{M}}$ to a binary relation $r^{\mathcal{M}} \subseteq \Delta^{\mathcal{M}} \times \Delta^{\mathcal{M}}$ and each individual name a to an element $a^{\mathcal{M}} \in \Delta^{\mathcal{M}}$. The function $\cdot^{\mathcal{M}}$ is inductively extended to arbitrary concepts by setting

$$\begin{aligned} \top^{\mathcal{M}} &:= \Delta^{\mathcal{M}}, \\ (C \sqcap D)^{\mathcal{M}} &:= C^{\mathcal{M}} \cap D^{\mathcal{M}}, \\ (\exists r.C)^{\mathcal{M}} &:= \{d \in \Delta^{\mathcal{M}} \mid \exists e \in \Delta^{\mathcal{M}} \text{ such that } (d,e) \in r^{\mathcal{M}} \text{ and } e \in C^{\mathcal{M}}\}\ . \end{aligned}$$

An interpretation $\mathcal{M}$ satisfies a concept inclusion $C \sqsubseteq D$ (written $\mathcal{M} \models C \sqsubseteq D$) if $C^{\mathcal{M}} \subseteq D^{\mathcal{M}}$ concept assertion $C(a)$ (written $\mathcal{M} \models C(a)$) if $a^{\mathcal{M}} \in C^{\mathcal{M}}$, role assertion $r(a_i, a_j)$ (written $\mathcal{M} \models r(a_i, a_j)$) if $(a_i^{\mathcal{M}}, a_j^{\mathcal{M}}) \in r^{\mathcal{M}}$. $\mathcal{M}$ is a **model** of a TBox $\mathcal{O}$ if it satisfies all CIs in $\mathcal{O}$. We write $\mathcal{O} \models C \sqsubseteq D$ if every model of $\mathcal{O}$ satisfies $C \sqsubseteq D$. $\mathcal{M}$ is a model of an ABox $\mathcal{A}$ if it satisfies all assertions in $\mathcal{A}$. We write $\mathcal{A} \models C(a)$ and $\mathcal{A} \models r(a_i, a_j)$ if every model of $\mathcal{A}$ satisfies $C(a)$ and $r(a_i, a_j)$ respectively. $\mathcal{M}$ is a model of a knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{A})$ if it is a model of $\mathcal{O}$ and $\mathcal{A}$. For a concept inclusion or assertion φ, we write $\mathcal{K} \models \varphi$ if φ is satisfied in all models of $\mathcal{K}$. If empty, $\mathcal{A}$ is simply omitted.

A **signature** Σ is a finite subset of $P \uplus R \uplus I$; if empty, I is omitted. The signature $\mathsf{Sig}(C)$ ($\mathsf{Sig}(C(a))$, $\mathsf{Sig}(r(a_i, a_j))$, $\mathsf{Sig}(\mathcal{O})$, $\mathsf{Sig}(\mathcal{A})$) of a concept C (concept assertion $C(a)$, role assertion $r(a_i, a_j)$, TBox $\mathcal{O}$, ABox $\mathcal{A}$) is the set of concept, role and individual names which occur in C ($C(a)$, $r(a_i, a_j)$, $\mathcal{O}$, $\mathcal{A}$). If $\mathsf{Sig}(C) \subseteq \Sigma$, we also call C a Σ-concept.

2.2.2 Description logic $\mathcal{EL}^+$

The description logic $\mathcal{EL}^+$ [10,11] is an extension of $\mathcal{EL}$ with **role inclusions** (RIs). So we have $\mathcal{EL}^+$-concepts build following the syntax rule for $\mathcal{EL}$ together with role inclusions of the form $r \sqsubseteq s$, where $r = r_1 \circ \cdots \circ r_n$, for $n \geq 1$, is a sequence of role names and s a role name. ABoxes remain as defined for $\mathcal{EL}$. A finite set of RIs is called **a role box** (RBox). An interpretation $\mathcal{M}$ satisfies an RI $r_1 \circ \cdots \circ r_n \sqsubseteq r$, $n \geq 1$, (written $\mathcal{M} \models r_1 \circ \cdots \circ r_n \sqsubseteq r$) if $r_1^{\mathcal{M}} \circ \cdots \circ r_n^{\mathcal{M}} \subseteq r^{\mathcal{M}}$, where '$\circ$' is interpreted as the composition of binary relations (i.e., we consider '$\circ$' to be defined as $R \circ S = \{(d, d'') \mid dRd', d'Sd'', \text{ for some } d'\}$). $\mathcal{M}$ is a model of an RBox $\mathcal{R}$ if it satisfies all RIs in $\mathcal{R}$. We write $\mathcal{R} \models r \sqsubseteq s$ if every model of $\mathcal{R}$ satisfies $r \sqsubseteq s$. A **constraint box** (CBox) $\mathcal{C} = (\mathcal{O}, \mathcal{R})$ consists of a TBox $\mathcal{O}$ and an RBox $\mathcal{R}$. An interpretation $\mathcal{M}$ is a model of a CBox $\mathcal{C} = (\mathcal{O}, \mathcal{R})$ if $\mathcal{M}$ is a model of both $\mathcal{O}$ and $\mathcal{R}$. We write $\mathcal{C} \models C \sqsubseteq D$ if every model of $\mathcal{C}$ satisfies $C \sqsubseteq D$.

The signature $\mathsf{Sig}(\mathcal{C})$ of a CBox $\mathcal{C}$ is the set of concept and role names which occur in $\mathcal{C}$.

2.2.3 Description logic $\mathcal{ELSH}$

The logic $\mathcal{ELSH}$ is situated between $\mathcal{EL}$ and $\mathcal{EL}^+$. Its concepts coincide with those of $\mathcal{EL}$ and it allows for a restricted form of role boxes. Namely, a $\mathcal{ELSH}$ role box $\mathcal{R}$ consists of inclusions

$$r \sqsubseteq s\ ,$$

where r and s are role names and transitivity axioms

$$r \circ r \sqsubseteq r$$

stating that r is transitive such that whenever there exists s with $r \sqsubseteq s \in \mathcal{R}$ then $r \circ r \sqsubseteq r \notin \mathcal{R}$. Thus, in $\mathcal{ELSH}$ we cannot declare a role to be transitive if it is included in another role. The language OWL DL of the OWL standard [4] is based on the description logic $\mathcal{SHIQ}$ that has exactly the same role boxes as $\mathcal{ELSH}$. The notation used for $\mathcal{ELSH}$ constraint boxes is exactly the same as for $\mathcal{EL}^+$.

2.2.4 Description logic $\mathcal{ALC}$

The description logic $\mathcal{ALC}$ is an extension of $\mathcal{EL}$ with negation ($\neg$). Formally, $\mathcal{ALC}$-concepts are build according to the following syntax rule:

$$C ::= \top \mid A \mid \neg C \mid C \sqcap D \mid \exists r.C,$$

where A ranges over the concept names in P, r over role names in R, and C, D over $\mathcal{ALC}$-concepts. It is clear that $\mathcal{ALC}$ is propositionally complete, that is all boolean set operations can be expressed, $\mathcal{ALC}$ is strong enough to express universal quantifier along a role. As mentioned above we use standard abbreviations: $\bot$ is an abbreviation of $\neg\top$, $C \sqcup D$ abbreviates $\neg(\neg C \sqcap \neg D)$ and $\forall r.C$ is short for $\neg(\exists r.\neg C)$. ABoxes in $\mathcal{ALC}$ extend $\mathcal{EL}$ ABoxes with concept assertions over $\mathcal{ALC}$ concepts.

$\mathcal{ALC}$ models are similar to $\mathcal{EL}$ models but the function $\cdot^{\mathcal{M}}$ is extended further by setting

$$\begin{aligned}
\top^{\mathcal{M}} &:= \Delta^{\mathcal{M}}, \\
(\neg C)^{\mathcal{M}} &:= \Delta^{\mathcal{M}} \setminus C^{\mathcal{M}}, \\
(C \sqcap D)^{\mathcal{M}} &:= C^{\mathcal{M}} \cap D^{\mathcal{M}}, \\
(\exists r.C)^{\mathcal{M}} &:= \{d \in \Delta^{\mathcal{M}} \mid \exists e \in \Delta^{\mathcal{M}} \text{ such that } (d,e) \in r^{\mathcal{M}} \text{ and } e \in C^{\mathcal{M}}\}.
\end{aligned}$$

As already discussed in Section 1.1.1, having various connectives and different types of axioms available, by allowing combinations thereof we can create various description logics extending $\mathcal{ALC}$, here we shortly present these extensions.

A simple extension of $\mathcal{ALC}$ is description logic $\mathcal{S}$, which allows for axioms expressing transitive roles. Description logic $\mathcal{S}$ can be extended further by adding inverse roles r^- (we indicate that by adding a letter $\mathcal{I}$ to the name), role hierarchies $r_1 \sqsubseteq r_2$ (indicated by a letter $\mathcal{H}$). Number restrictions of the form $\geqslant n\ r$ and $\leqslant n\ r$ (we append a letter $\mathcal{N}$ to the name). Qualified number restrictions (represented by adding a letter $\mathcal{Q}$ to the name) are of the form $\geqslant n\ r.C$ and $\leqslant n\ r.C$. Finally, nominals $\{i\}$ (we append a letter $\mathcal{O}$), using them it is possible to construct a concept representing a singleton set $\{i\}$ (a nominal concept) from an individual i. We can use different combinations of these extensions, for instance $\mathcal{ALCO}$ extends $\mathcal{ALC}$ with nominals; $\mathcal{SHIQ}$ is a well known extension of S with role hierarchies, inverse roles and qualified number restrictions; and $\mathcal{SHOIQ}$ uses all constructors and axioms presented above. We refer the reader to [9] for details.

2.2.5 First Order Logic (FOL)

Signatures of FOL are pairs (F, Π), where F and Π are families of sets of function $(F_n)_{n\in\omega}$ and predicate names $(\Pi_n)_{n\in\omega}$ respectively, where n is an arity. The following logical symbols are in use in FOL: we have truth constant '$\top$', existential quantifier '$\exists$', conjunction '$\wedge$' and negation '$\neg$'. It is customary to introduce following abbreviations: $\bot$ which is short for $\neg\top$, $\forall x P(x)$ abbreviates $\neg\exists x\neg P(x)$, $\varphi \vee \psi$ is short for $\neg(\neg\varphi \wedge \neg\psi)$, $\varphi \Rightarrow \psi$ stands for $\neg\varphi \vee \psi$ and $\varphi \Leftrightarrow \psi$ is short for $(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$.

The formation rules define the terms and formulae of FOL.

The set Term of terms is inductively defined by the following rules:

1. Any variable is a term.

2. Any expression $f(t_1, ..., t_n)$ of n arguments (where each argument t_i is a term and f is a function symbol of valence n) is a term.

Only expressions which can be obtained by finitely many applications of rules (1) and (2) are terms. For example, no expression involving a predicate symbol is a term.

If P is an n-ary predicate and $t_1, ...t_n$ are terms, then $P(t_1, .., t_n)$ is an **atomic formula.** The set of formulas (also called well-formed formulas) is inductively defined by the following rules:

1. If t_1 and t_2 are terms, then $t_1 = t_2$ is a formula.

2. Every atomic formula is a well formed formula.

3. If φ is a formula, then $\neg\varphi$ is a formula.

4. If φ and ψ are formulas, then $\varphi \wedge \psi$ is a formula.

5. If φ is a formula and x is a variable, then $(\exists x)\varphi$ is a formula.

Only expressions which can be obtained by finitely many applications of rules (1) to (5) are formulas. The formulas obtained from rules (1) and (2) are said to be atomic formulas. By rules (3) and (4) formulas created using remaining binary logical connectives are well formed. Using rule (3) and (5) we can show that $(\forall x)\varphi$ is a well formed formula.

A FOL-model $\mathcal{M}$ is a triple $\mathcal{M} = (\Delta^{\mathcal{M}}, f^{\mathcal{M}}_{f\in F}, \Pi^{\mathcal{M}}_{\Pi\in\Pi_n})$, where the interpretation domain Δ is not empty, for n-ary function $f \in F$, $f^{\mathcal{M}} : (\Delta^{\mathcal{M}})^n \to \Delta^{\mathcal{M}}$, for n-ary predicate $\Pi \in \Pi_n$, $\Pi^{\mathcal{M}}$ is an n-ary relation on Δ, i.e., $\Pi^{\mathcal{M}} \subseteq (\Delta^{\mathcal{M}})^n$.

A valuation in model $\mathcal{M}$ is a function $v : \{x_1, ..., x_2\} \to \Delta^{\mathcal{M}}$ and $\bar{v} : \mathtt{Term} \to \Delta^{\mathcal{M}}$, such that for variable x, n-ary function f and terms $t_1, .., t_n$ the following holds:

$$\begin{aligned} \bar{v}(x) &= v(x), \\ \bar{v}(f(t_1, .., t_n)) &= f^{\mathcal{M}}(\bar{v}(t_1), .., \bar{v}(t_n)). \end{aligned}$$

Let φ be a well formed formula and $\mathcal{M}$ a model. We have that $\mathcal{M} \models \varphi$ iff $v \models \varphi$ for every v in $\mathcal{M}$. A valuation v is said to satisfy φ if it can be show inductively to do so under following conditions:

(1)	$v \models t_1 \equiv t_2$	iff	$\bar{v}(t_1) = \bar{v}(t_2)$,
(2)	$v \models \Pi(\bar{t})$	iff	$\bar{v}(t) \in \Pi^{\mathcal{M}}$,
(3)	$v \models \neg\psi$	iff	$v \not\models \psi$,
(4)	$v \models \psi_1 \wedge \psi_2$	iff	$v \models \psi_1$ and $v \models \psi_2$,
(5)	$v \models (\exists x)\varphi$	iff	there is v' (same as v, except on x) and $v' \models \varphi$.

2.3 Introduction to Category Theory

In this part we present some basic concepts of category theory. We follow the outlines presented in [13] and [2], the latter can be treated as a textbook for beginners in category theory. We also fix notation to make it easier to read our results presented later.

2.3.1 Background

In 1945 Eilenberg and Mac Lane in their paper "General theory of natural equivalences" [37] first formulated what we know as Category Theory. This approach was in the spirit of Felix Klein's *Erlanger Programm.* It was designed to provide general concepts applicable to all branches of abstract mathematics and allowing for uniform treatment of different mathematical disciplines, cf. [37]. In the late 1940s it was mainly used in the fields of abstract algebra and algebraic topology. Later, in the 1950s, it was also applied to geometry, this was started by Grothendieck, who used category theory for solving classical problems of geometry and number theory. In the 1960s Lawvere used category theory for investigating properties of logical systems. In the 1970s category theory has proven to be useful in disciplines like computer science, cognitive science, philosophy, linguistics etc., cf. [7].

In mathematics category theory can be described as an abstract study of mathematical structures. The main idea behind category theory lies in the observation that it is possible to represent a number of properties of mathematical systems by means of diagrams of arrows. To build some intuitions behind category theory we could compare it to set theory, then in a diagram each arrow $f : X \to Y$ would represent a function; that is, a set X, a set Y, and a rule $x \mapsto f(x)$ which assigns to each element $x \in X$ an element $f(x) \in Y$, cf. [58]. Nevertheless it is important to keep in mind that sets are just an illustration and in category theory we talk about objects, and arrows can represent more (but also less) than functions.

In our work category theory is important as it provides tools for the theory of institutions. The theory of institutions was first introduced by Goguen and Burstall in their article "Introducing Institutions" [41] published in 1984. Institutions and some ideas about the history of the concept and intuitions behind it are presented in Section 2.4.

2.3.2 Graphs

We start by presenting a notion of graph, which is useful to understand the notion of category. Another reason for introducing graphs is that they will be useful below where we introduce commutative diagrams and limits.

Here we present directed multigraphs with loops.

Definition 2.3.1. *A graph is a collection of objects connected by arrows. Each arrow has a source and a target, which do not have to be different. A graph with no arrows is called a* **discrete graph**. *A graph is* **finite** *if the number of objects and arrows is finite. A graph that has a set of objects and arrows is a* **small graph**, *otherwise, it is a* **large graph**.

There is no restriction on the number of arrows with given objects as source and target.

Notation 2.3.2. *We will be using notation $f : a \to b$ to mean that f is an arrow where object a is the source of f and b is its target. Sometimes we find it more convenient to use $a \xrightarrow{f} b$ to express the same fact.*

A loop, i.e. an arrow with the same object as source and target, is called an identity arrow or an endoarrow.

Now we define a homomorphism of graphs which is a transformation preserving the abstract shape of the graph.

Definition 2.3.3. *A homomorphism $\phi = (\phi_1, \phi_2)$ from a graph $\mathcal{G}$ to a graph $\mathcal{H}$, denoted $\phi : \mathcal{G} \to \mathcal{H}$, is a pair of functions ϕ_1 and ϕ_2, such that ϕ_1 takes objects in $\mathcal{G}$ and returns objects in $\mathcal{H}$, whereas ϕ_2 takes arrows in $\mathcal{G}$ and returns arrows in $\mathcal{H}$, with the property that if $v : m \to n$ is an arrow of $\mathcal{G}$, then $\phi_2(v) : \phi_1(m) \to \phi_1(n)$ is an arrow of $\mathcal{H}$.*

It is important to note that notation of the form $f : A \to B$ is overloaded, it denotes set theoretic functions, an arrow in a graph and a graph homomorphism. On the other hand, when $\phi : \mathcal{G} \to \mathcal{H}$ is a graph homomorphism, ϕ in fact is a pair of functions ϕ_1 and ϕ_2, which also overloads the notation. In practice though, it does not lead to any confusion as it is clear from context how the notation is used.

2.3.3 Categories

In general, we can say that a category is a graph with additional requirements. Namely, it is required that each object has an identity arrow and that this graph has a rule for composition of arrows.

Definition 2.3.4. *A category $\mathcal{A}$ consists of:*

- *a collection of $\mathcal{A}$-objects,*
- *for objects A, B, a collection of morphisms (arrows) $f : A \to B$,*

such that following conditions are satisfied:

- *for every $\mathcal{A}$-object A there is an identity arrow $1_A : A \to A$,*
- *for $\mathcal{A}$-objects A, B, C, and $f : A \to B$, $g : B \to C$, there is the composition arrow $f;g : A \to C$,*
- *for every $f : A \to B$*

- $1_A; f = f$,
- $f; 1_B = f$

- *for every* $f : A \to B$, $g : B \to C$ *and* $h : C \to D$ *in* $\mathcal{A}$ *the following condition is satisfied* $(f; g); h = f; (g; h)$.

As a simple and intuitive example of a category consider the following:

Example 2.3.5. Set *is the category of sets, with:*

- *objects: sets*
- *arrows: an arrow* $A \xrightarrow{f} B$ *is a triple* (A, f, B) *where* f *is a function that takes arguments from* A *and gives results in* B.

Example 2.3.6. *Any directed graph generates a category of the same cardinality. The objects are the nodes of the graph, and the morphisms are the paths in the graph where composition of morphisms is concatenation of paths. Such a category is called the free category generated by the graph. Note that every category* $\mathcal{C}$ *has an* ***underlying graph*** *which has the arrows of* $\mathcal{C}$ *as edges, and objects of* $\mathcal{C}$ *as objects.*

Reversing the arrows in any categorical definition gives its **dual**, which often is named by appending the prefix 'co-'. Note that reversing the arrows in the category axioms gives exactly these axioms back. For that reason reversing the arrows in any theorem gives a dual theorem.

Using Definition 2.3.4 and the notion of duality we get that every category $\mathcal{C}$ has an **opposite (or dual)** category $\mathcal{C}^{op}$. The objects of $\mathcal{C}^{op}$ are the objects of $\mathcal{C}$, and the arrows of $\mathcal{C}^{op}$ are the arrows of $\mathcal{C}$ but with reversed source and target. Thus every arrow $f : C \to C'$ of $\mathcal{C}$ appears as an arrow $f^{op} : C' \to C$ of $\mathcal{C}^{op}$. As arrows

$$A \xrightarrow{f} B \xrightarrow{g} C$$

of $\mathcal{C}$ appear as arrows

$$C \xrightarrow{g^{op}} B \xrightarrow{f^{op}} A$$

of $\mathcal{C}^{op}$, the order of composition is reversed. Thus we define composition in $\mathcal{C}^{op}$ as $g^{op}; f^{op} = (f; g)^{op}$, where $f; g$ is a composition in $\mathcal{C}$. Every category is the dual of its dual: $\mathcal{C} = (\mathcal{C}^{op})^{op}$.

As in category theory one focus on morphisms rather than objects, properties are often formulated in terms of arrows instead of objects. One of the important types of morphisms is isomorphism.

Definition 2.3.7. *A morphism $f : A \to B$ in a category is called an* ***isomorphism*** *provided that there exists a morphism $g : B \to A$ with $f;g = 1_A$ and $g;f = 1_B$. Such a morphism g is called an* ***inverse*** *of f.*

If there is an isomorphism from A to B, we say that A is isomorphic to B and write $A \cong B$. There may be more than one isomorphism between two objects.

Note that if f is an isomorphism in category C, then it is also an isomorphism in category C^{op}.

Definition 2.3.8. *A category in which every arrow is an isomorphism is a* ***groupoid***. *A category in which every arrow is an identity arrow is called* ***discrete***.

Another important distinction between categories takes into consideration the class of objects of a category. This is formulated in the following definition.

Definition 2.3.9. *A category C is said to be* ***small*** *provided that its class of objects is a set, otherwise it is called* ***large***.

The smallest category is the one with no objects nor arrows. The next smallest one has one object and one arrow. Category Set of example 2.3.5 is a large category, but this distinction is not crucial in our work and more details may be found in [58].

2.3.4 Slice categories

Another notion which is important for us is the notion of **slice category**. It allows to view arrows in a different way, instead of presenting them as relations between objects of a category, arrows are now described as objects of a category. We will use that to systematize description logics.

Definition 2.3.10 (Slice categories). *If $\mathscr{C}$ is a category and A is an object of $\mathscr{C}$, the slice category $\mathscr{C}/A$ is described in the following way:*

SC-1 An object of $\mathscr{C}/A$ is an arrow $f : C \to A$ of $\mathscr{C}$ for some object C.

SC-2 An arrow of $\mathscr{C}/A$ from $f : C \to A$ to $f' : C' \to A$ is an arrow $h : C \to C'$ with the property that $f = h; f'$

SC-3 The composite of $h : f \to f'$ and $h' : f' \to f''$ is $h; h'$.

We have to show that $h;h'$, as defined in SC-2, satisfies the requirements of being an arrow from f to f'. Let $h : f \to f'$ and $h' : f' \to f''$. By definition this means that $h; f' = f$ and $h'; f'' = f'$. To show that $h;h' : f \to f''$ is an arrow of $\mathscr{C}/A$, we have to show that $(h;h'); f'' = f$. But this is implied by the following calculation:

$$(h;h'); f'' = h; (h'; f'') = h; f' = f.$$

Note that the usual notation for arrows in $\mathscr{C}/A$ is deficient: an arrow h can satisfy $f = h; f'$ and $g = h; g'$ with $g \neq f$ or $g' \neq f'$ (or both). Then $h : f \to f'$ and $h : g \to g'$ are different arrows of $\mathscr{C}/A$.

2.3.5 Functors

As we have homomorphisms for graphs we also have structure-preserving maps for categories, we call them functors. We can view functors as graphs homomorphisms preserving identities and composition.

Notation 2.3.11. *Since any category C consists of two types of collections, we will distinguish them by writing $|C|$ whenever referring to the collection of objects and $\langle C \rangle$ when referring to the collection of arrows in category C.*

Definition 2.3.12. *For any two categories $\mathcal{A}$ and $\mathcal{B}$ a functor $F : \mathcal{A} \to \mathcal{B}$ is a pair of functions $F_1 : |\mathcal{A}| \to |\mathcal{B}|$ and $F_2 : \langle \mathcal{A} \rangle \to \langle \mathcal{B} \rangle$ for which:*

- *if $f : C \to D$ in $\mathcal{A}$, then $F_2(f) : F_1(C) \to F_1(D)$ in $\mathcal{B}$,*
- *for any object C in $\mathcal{A}$, $F_2(1_C) = 1_{F_1(C)}$,*
- *if $f;g$ is defined in $\mathcal{A}$, then $F_2(f); F_2(g)$ is defined in $\mathcal{B}$, and $F_2(f); F_2(g) = F_2(f;g)$.*

Note that both F_1 and F_2 are usually just written F.

Functors can be composed in the following way:

Definition 2.3.13 (Composition of functors). *If $F : \mathcal{A} \to \mathcal{B}$ and $G : \mathcal{B} \to \mathcal{C}$ are functors, then the composite $G \circ F : \mathcal{A} \to \mathcal{C}$ defined by*

$$(G \circ F)(A \xrightarrow{f} A') = G(FA) \xrightarrow{G(Ff)} G(FA')$$

is a functor.

2.3.6 Categories of categories

As functors behave just as arrows in a category it is natural to ask if we can construct the "category of all categories".

Definition 2.3.14 (The category of categories). *The category Cat has all small categories as objects and all functors from A to B as morphisms, as identities the identity functors, and composition of functors in the standard way.*

In some cases we will refer to $\mathcal{CAT}$, which has all small categories and ordinary large categories as objects and all functors between them as morphisms.

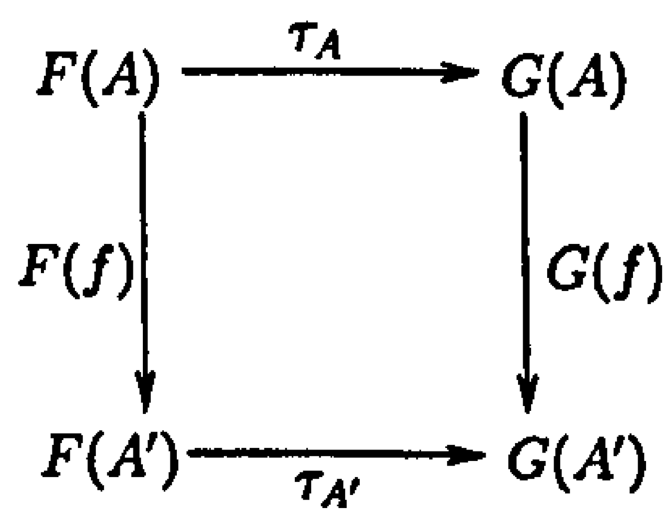
To be able to form entities like a category of all categories the notion of a quasicategory was introduced, which frees the concept of category from its set-theoretical restrictions, for instance, it is not required that objects form a class. Such foundational issues are beyond the scope of this thesis, but the interested reader is referred to [2].

Definition 2.3.15. *The quasicategory $\mathcal{CAT}$ (often called proper quasicategory) has as objects all categories (small and large) and as morphisms from A to B all functors from A to B, identities and composition have their usual meaning.*

2.3.7 Natural Transformation

A natural transformation is a map between functors and often is called a morphism of functors.

Definition 2.3.16. *Let $F, G : \mathcal{A} \to \mathcal{B}$ be functors. A* **natural transformation** *τ from F to G (denoted by $\tau : F \to G$ or $F \xrightarrow{\tau} G$) is a family of arrows in $\mathcal{B}$ ($\tau_A : F(A) \to G(A)$) indexed by objects A of $\mathcal{A}$, such that for any $f : A \to A'$ in $\mathcal{A}$ the square*

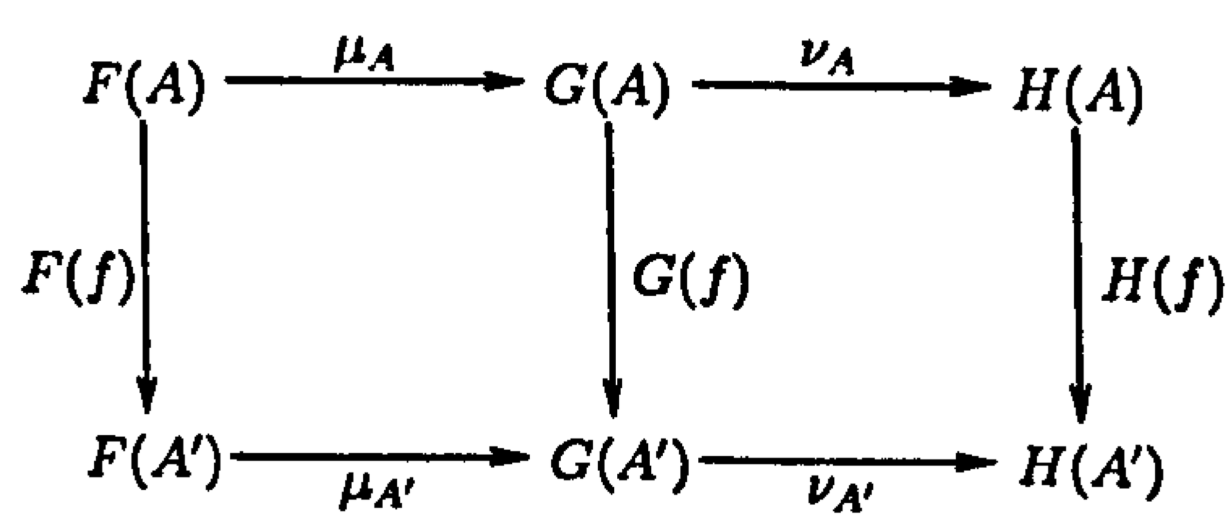


commutes, i.e., $F(f);\tau_{A'} = \tau_A; G(f)$.

We call $\tau_A, \tau_{A'}, \ldots$ the **components** of the natural transformation τ.

For functors $F, G : \mathcal{A} \to \mathcal{B}$ a natural transformation $\tau : F \to G$ whose components τ_A are isomorphisms is called a **natural isomorphism** from F to G (sometimes also called **natural equivalence**), and denoted by $\tau : F \cong G$. Then the inverses $(\tau_A)^{-1}$ in $\mathcal{B}$ are the components of a natural natural isomorphism $\tau^{-1} : G \to F$.

Natural transformations compose, so given functors F, G and H from $\mathcal{A}$ to $\mathcal{B}$, and natural transformations $\mu : F \to G$ and $\nu : G \to H$, there is a natural transformation $\mu;\nu : F \to H$ defined by composing components $\mu_A; \nu_A$:



Naturality is clear and this composition is associative. There is also an identity natural transformation $1_F : F \to F$, with components $1_F(A)$. This gives rise to a category called the functor category.

Definition 2.3.17. *Given categories $\mathcal{A}$ and $\mathcal{B}$, the* **functor category**, *denoted $\mathcal{B}^{\mathcal{A}}$ is the category whose objects are functors $F : \mathcal{A} \to \mathcal{B}$ and morphisms are natural transformations between these functors.*

2.3.8 Diagrams

A closely related notion is that of diagrams. In fact, functors and diagrams are just different aspects of the very same idea; they are types of graph homomorphisms.

In category theory a very important notion is that of **commutative diagram** as it is used for expressing equations.

Definition 2.3.18. *Let $\mathcal{I}$ and $\mathcal{G}$ be graphs. A* ***diagram*** *in $\mathcal{G}$ of shape $\mathcal{I}$ is a homomorphism $D : \mathcal{I} \to \mathcal{G}$ of graphs. $\mathcal{I}$ is called the* ***shape graph*** *of the diagram D.*

We could say that a diagram is just a graph homomorphism viewed from a different perspective.

The following two definitions shows the close relation between functors and diagrams.

Definition 2.3.19. *If $\mathcal{J}$ is a category, then a diagram in category $\mathcal{C}$ of shape $\mathcal{J}$ is a functor $D : \mathcal{J} \to \mathcal{C}$.*

The following definition uses the fact that every directed graph generates a category and that every category has an underlying graph, recall Example 2.3.6.

Definition 2.3.20. *Let $\mathcal{C}$ be a category, $U\mathcal{C}$ the underlying graph of $\mathcal{C}$, and $\mathcal{G}$ any graph. Then a diagram in $\mathcal{C}$ of shape $\mathcal{G}$ is a morphism $D : \mathcal{G} \to U\mathcal{C}$ of graphs. Equivalently, this is a functor $F\mathcal{G} \to \mathcal{C}$, where $F\mathcal{G}$ is the free category generated by $\mathcal{G}$.*

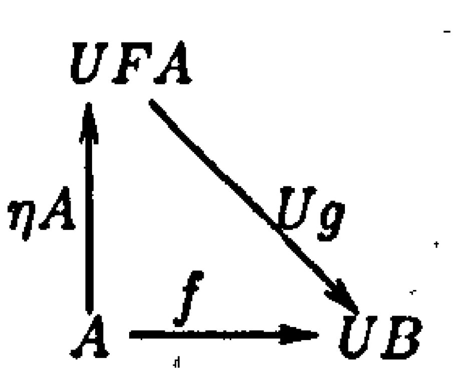
In the literature it is common to refer to the category of diagrams in $\mathcal{C}$ of shape $\mathcal{J}$ as the functor category $\mathcal{C}^{\mathcal{J}}$.

2.3.9 Adjoints

One of the most important notions of category theory is that of adjoint functors. This type of relation between functors is very common in mathematics, their ubiquitousness was expressed in [58]: "The slogan is 'Adjoint functors arise everywhere'."

Notation 2.3.21. *Here we follow notation which is standard in the literature on adjoints. For example, we write UF as an abbreviation of $U \circ F$ which is the same as $F;U$.*

Definition 2.3.22. *Let $\mathcal{A}$ and $\mathcal{B}$ be categories. If $F : \mathcal{A} \to \mathcal{B}$ and $U : \mathcal{B} \to \mathcal{A}$ are functors, we say that F is* ***left adjoint*** *to U and U is* ***right adjoint*** *to F provided there is a natural transformation $\eta : id \to UF$ such that for any objects A of $\mathcal{A}$ and B of $\mathcal{B}$ and any arrow $f : A \to UB$, there is a unique arrow $g : FA \to B$ such that*

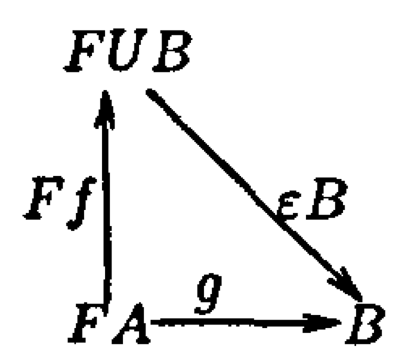


commutes.

The property of η is called the **universal mapping property.** It is customary to write $F \dashv U$ to denote the situation described in the above definition. The triple (F, U, η) constitutes an **adjunction**, the transformation η is called the **unit** of the adjunction.

As pointed in [13] this definition is asymmetric in F and U, but it is also mentioned there that the following proposition is a remedy for that.

Proposition 2.3.23. *Let $F : \mathcal{A} \to \mathcal{B}$ and $U : \mathcal{B} \to \mathcal{A}$ be functors such that $F \dashv U$. Then there is a natural transformation $\varepsilon : FU \to 1_{\mathcal{B}}$ such that for any $g : FA \to B$, there is a unique arrow $f : A \to UB$ such that*



The transformation ε is called the **counit** of the adjunction.

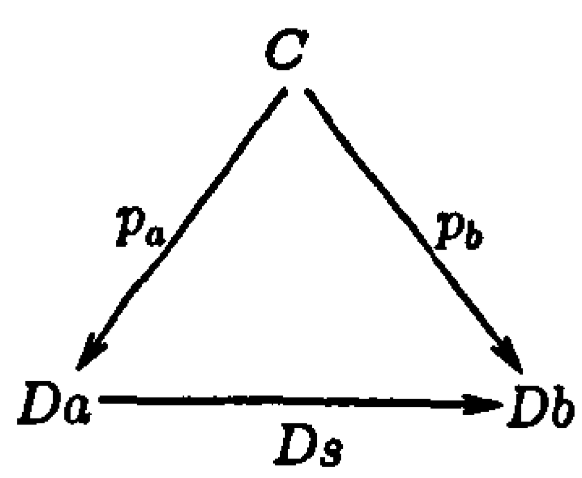
Adjoints are studied in more detail in [2,12,13,58], here we will use them to recall some properties of institutions.

2.3.10 Cones, limits and products

In this part we introduce the notion of (co)cone, which is used for defining the notion of (co)limit. Then we show the relation between (co)limit and (co)products.

Definition 2.3.24. *Let $\mathcal{G}$ be a graph and $\mathcal{C}$ be a category. Let $D : \mathcal{G} \to \mathcal{C}$ be a diagram in $\mathcal{C}$ with shape $\mathcal{G}$. A* **cone with base** *D is an object C of $\mathcal{C}$ together with a family $\{p_a\}$ of arrows of $\mathcal{C}$ indexed by the nodes of $\mathcal{G}$, such that $p_a : C \to Da$ for each node a of $\mathcal{G}$, the arrow p_a is the* **component** *of the cone at a.*

The cone is **commutative** *if for any arrow $s : a \to b$ of $\mathcal{G}$, the diagram*

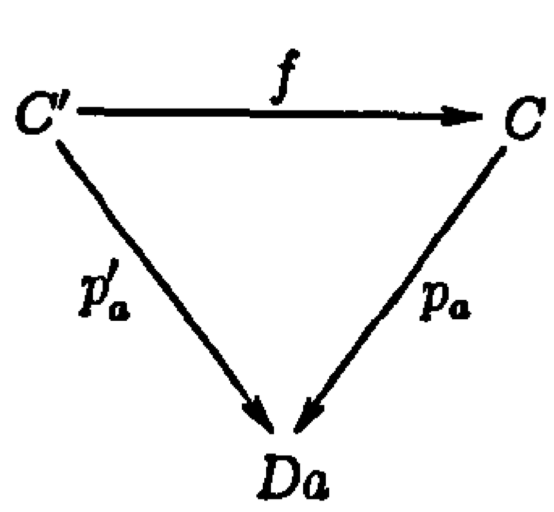


commutes.

A **cocone** *is a cone in the dual graph.*

The following definition show us how we can relate with each other cones with the same base.

Definition 2.3.25. *If $p' : C' \to D$ and $p : C \to D$ are cones, an* **arrow** *from the former to the latter cone is an arrow $f : C' \to C$ such that for each node a of $\mathcal{G}$, the diagram*



commutes.

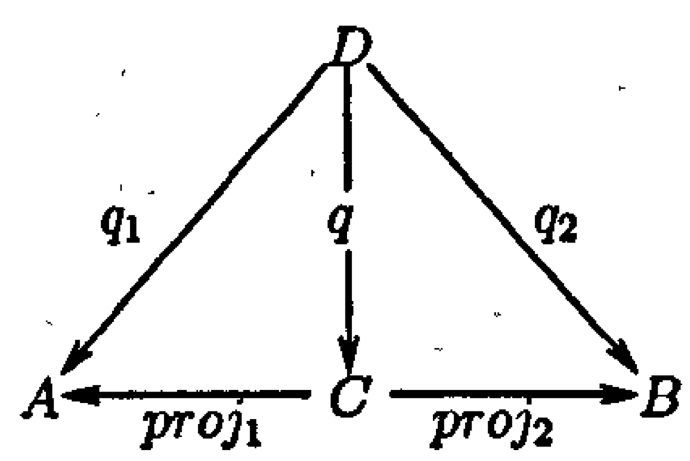
Now we consider the situation where we have a cone C which is a target for all the arrows from the cones with the same base as C.

Definition 2.3.26. *A commutative cone over the diagram D is called* **universal** *if every other commutative cone over the same diagram has a unique arrow to it. If there is a universal cone, then it is called a* **limit** *of the diagram D. The commutative cocone is called a* **colimit** *if it has a unique arrow to every other commutative cocone over the same diagram.*

Special cases of limits and colimits are products and coproducts respectively. With the use of products we are able to define operations of n-ary arity. Coproducts, on the other hand, are used for the specification of alternatives. In Set, the products are cartesian products and the coproducts are disjoint unions. This is presented in the definitions below.

Definition 2.3.27. *If S and T are sets, the* **cartesian product** *$S \times T$ is the set of all ordered pairs with first coordinate in S and second coordinate in T, i.e. $S \times T = \{(s,t) \mid s \in S \text{ and } t \in T\}$. The coordinates are functions $proj_1 : S \times T \to S$ and $proj_2 : S \times T \to T$, called the* **coordinate projections**, *or simply* **projections**.

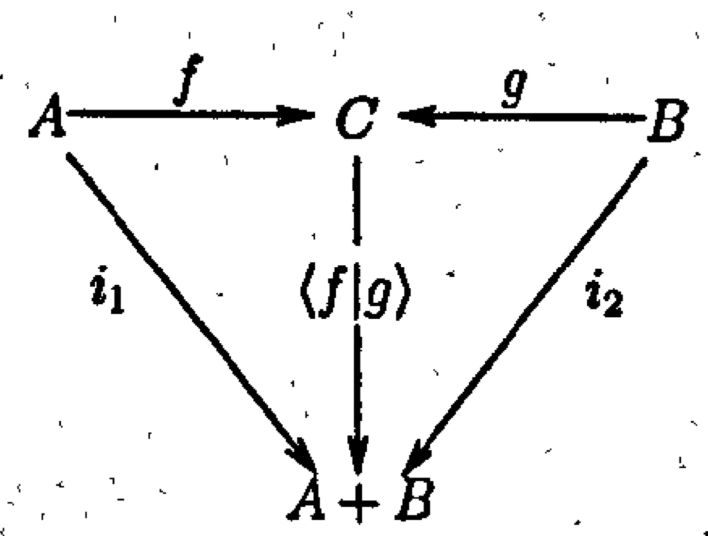
Definition 2.3.28. *[The product of two objects] Let A and B be two objects in a category $\mathcal{C}$. By a* **product** *of A and B, we mean an object C together with arrows $proj_1 : C \to A$ and $proj_2 : C \to B$ that satisfy the following condition:*
For any object D and arrows $q_1 : D \to A$ and $q_2 : D \to B$, there is a unique arrow $q : D \to C$:



(2.1)

such that $q; proj_1 = q_1$ and $q; proj_2 = q_2$.

Definition 2.3.29. *The* **sum**, *also called the* **coproduct**, *$A+B$ of two objects in a category consists of an object called $A+B$ together with arrows $i_1 : A \to A+B$ and $i_2 : B \to A+B$ such that given any arrows $f : A \to C$ and $g : B \to C$, there is a unique arrow $\langle f|g \rangle : A+B \to C$ for which*



(2.2)

commutes.

Products and sums of two objects, as discussed in Definition 2.3.28 and Definition 2.3.29 are called **binary products** and **binary sums** respectively. We can define products and sums of more than just two objects by an obvious modification of the definition.

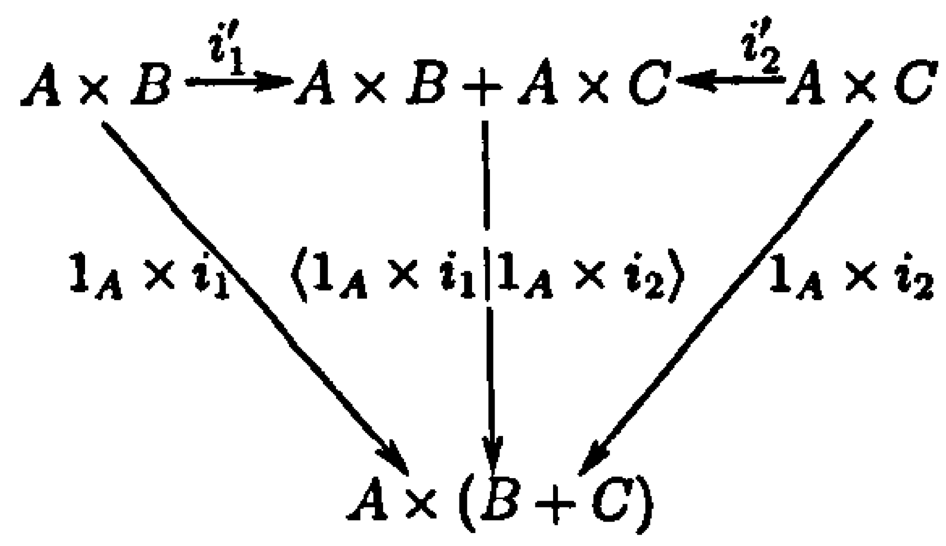
Let $\mathbb{C}$ be a category with binary products and binary sums. Then for any objects A, B and C we have sum cocones

$$B \xrightarrow{i_1} B+C \xleftarrow{i_2} C$$

and

$$A\times B \xrightarrow{i'_1} A\times B + A\times C \xleftarrow{i'_2} A\times C$$

There is a unique arrow $\langle 1_A \times i_1 \mid 1_A \times i_2\rangle : A\times B + A\times C \to A\times(B+C)$ making



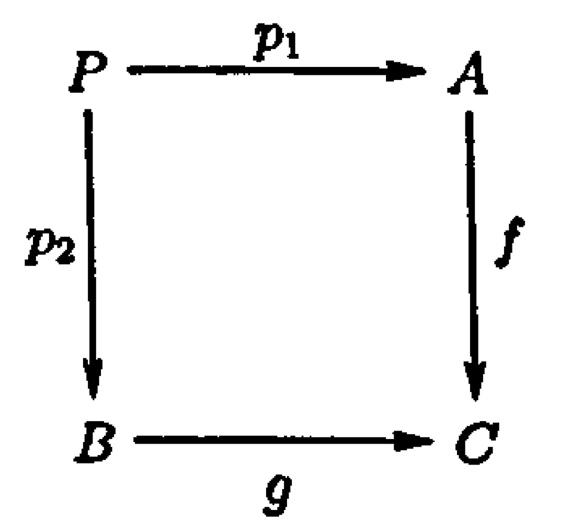
(2.3)

commute.

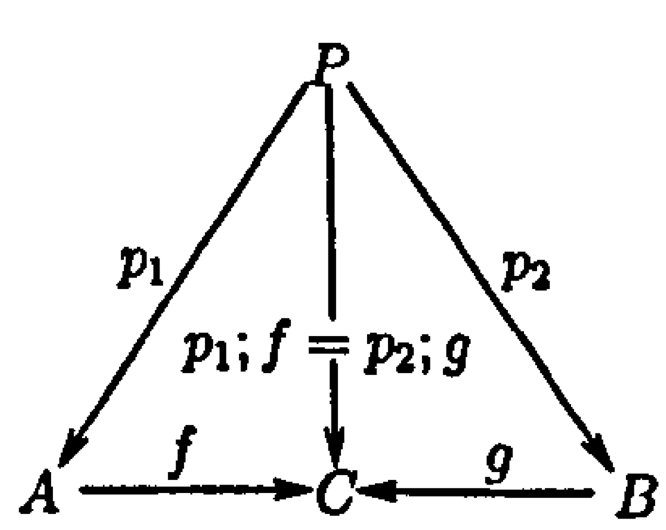
2.3.11 Pullbacks and Pushouts

Another type of limits and colimits are that of pullback and pushout, which are introduced below. Contrary to products and coproducts the importance of pullbacks and pushouts was recognized only after category theory was formulated.

Given $f : A \to C$ and $g : B \to C$ consider an object P together with arrows $A \xleftarrow{p_1} P \xrightarrow{p_2} B$ such that the diagram

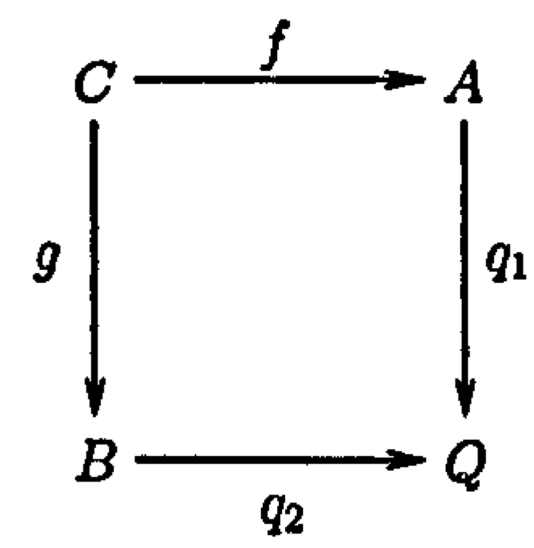


commutes. To see that this is indeed a cone think of compositions of arrows $p_1; f$ and $p_2; g$ as an arrow from P to C and redraw the diagram in the following way:



If this is a universal cone, then we say that P together with arrows p_1 and p_2 is a **pullback** of the pair. We can also say that p_2 is a **pullback of** f **along** g, and that the above forms a **pullback diagram.**

The dual notion to the notion of pullback is pushout. More precisely, a commutative square



is called a **pushout** if for any object R and any pair of arrows $r_1 : A \to R$ and $r_2 : B \to R$, for which $f; r_1 = g; r_2$ there is a unique arrow $r : Q \to R$, such that $q_i; r = r_i$, for $i = 1, 2$

2.3.12 Distributive categories

Categories in which products distribute over sums are called distributive categories. Detailed description of the theory of distributive categories was presented in [27] and many of their applications can be found in [84].

Definition 2.3.30 (Distributive category). *A* ***distributive category*** *is a category with finite sums and finite products in which for all objects A, B and C, the arrow d defined by the diagram 2.3 is an isomorphism.*

As an example we show that category of $\mathcal{EL}$ signatures is distributive. Here we introduce category of $\mathcal{EL}$ signatures only, $\mathcal{EL}$ is properly introduced in Example 2.4.9.

Example 2.3.31. *A category of $\mathcal{EL}$-signatures is defined in the following way:*

- *objects are pairs (P, R), where P is a set of 'concept names' and R is a set of 'role names',*
- *arrows, signature morphisms, $\sigma : (P, R) \to (P', R')$ consist of two functions between the sets of concept names and sets of role names respectively, i.e., $\sigma = (f, g)$ with $f : P \to P'$ and $g : R \to R'$. Composition is defined pairwise $(f, g); (f', g') = (f; f', g; g')$.*

Example 2.3.32 (Category $\mathsf{Sig}^{\mathcal{EL}}$ is distributive). *To show that the category of $\mathcal{EL}$ signatures is distributive we have to show that for any $\mathcal{EL}$-signatures (P_1, R_1), (P_2, R_2) and (P_3, R_3) we have that $((P_1, R_1) \times (P_2, R_2)) + ((P_1, R_1) \times (P_3, R_3))$ is isomorphic to $(P_1, R_1) \times ((P_2, R_2) + (P_3, R_3))$.*

First note that sums and products in $Sig^{\mathcal{EL}}$ are taken pointwise (i.e., $(P_1, R_1) \times (P_2, R_2) = (P_1 \times P_2, R_1 \times R_2$ and $(P_1, R_1) + (P_2, R_2) = (P_1 + P_2, R_1 + R_2)$. Then the required property is a straightforward calculation.

This property can also be inferred from the fact that the category of $\mathcal{EL}$ signatures is the category of sets of pairs, and sets and tuples of sets are distributive categories.

2.3.13 Inclusions and Inclusive Categories

In many cases it is convenient to consider simple morphisms within a category as inclusions. This is due to the fact that the notion of inclusion is very natural and thus very useful when we study the syntactical part of a logical system. In particular, even though signatures are not sets, it is important for us that they behave like sets, thus we prefer to use inclusions while considering subsignatures. For instance consider signatures of $\mathcal{EL}$. An $\mathcal{EL}$-signature is a pair of sets (P, R). Therefore we cannot strictly talk about inclusion, understood as set inclusion, between $\mathcal{EL}$-signatures $\Sigma \subseteq \Sigma'$, it is rather a composition of two functions $f : P \to P'$ and $g : R \to R'$, which are inclusions, i.e., we say $(P, R) \subseteq (P', R')$ iff $P \subseteq P'$ and $R \subseteq R'$.

It is well known that certain small categories correspond to partially ordered sets (posets). These categories satisfy the following conditions: they have at most one morphism between any two objects, there is a morphism from A to B if and only $A \leqslant B$, they also satisfy anti-symmetry, i.e. if there is a morphism from A to B and another from B to A then $A = B$. In what follows we will identify posets with their corresponding categories. Sums and products in these categories correspond to greatest lower bounds and least upper bounds, respectively. A poset with finite sums and products is a lattice, with the usual properties. Things generalize from sets to classes, which are called **poclasses**. We set $\hookrightarrow$ to denote the poclass morphisms (cf. [43]).

Here we present a definition of an inclusive category, which was introduced by Goguen and Rosu in [43]. As they suggested, this notion of inclusion is similar to that of (weak) inclusion systems present in the literature [29,30,35,67]

Definition 2.3.33 (Inclusive category). *An* ***inclusive category*** $\mathbb{C}$ *is a category with a broad subcategory*[1] $\mathbf{I}$ *which is a poclass, called its* ***subcategory of inclusions****, having finite products and coproducts (which we shall call intersections and unions), such that for every pair of objects A, B, their union in $\mathbf{I}$ is a pushout in $\mathbb{C}$ of their intersection in $\mathbf{I}$. A functor between two inclusive categories is an* ***inclusive functor*** *(or preserves inclusions) iff it takes inclusions in the source category to inclusions in the target category.*

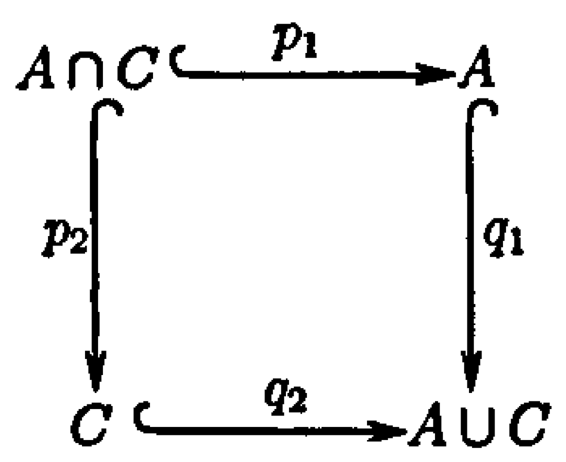
The following lemma demonstrates one way in which inclusive categories have set-like properties for union and intersection. It presents a correlation of inclusion, intersection and union, which is also true in set theory. We use this property later to show how inclusive categories have properties that are similar to sets. While considering this lemma it might be useful to keep in mind that the category of $\mathcal{EL}$ signatures is an instance of an inclusive category.

Notation 2.3.34. *As already mentioned products and coproducts correspond to intersections and unions in the set theory, in the remainder of the text we will write $A \cap B$ for $A \times B$ and $A \cup B$ for $A + B$, where A and B are objects of an inclusive category.*

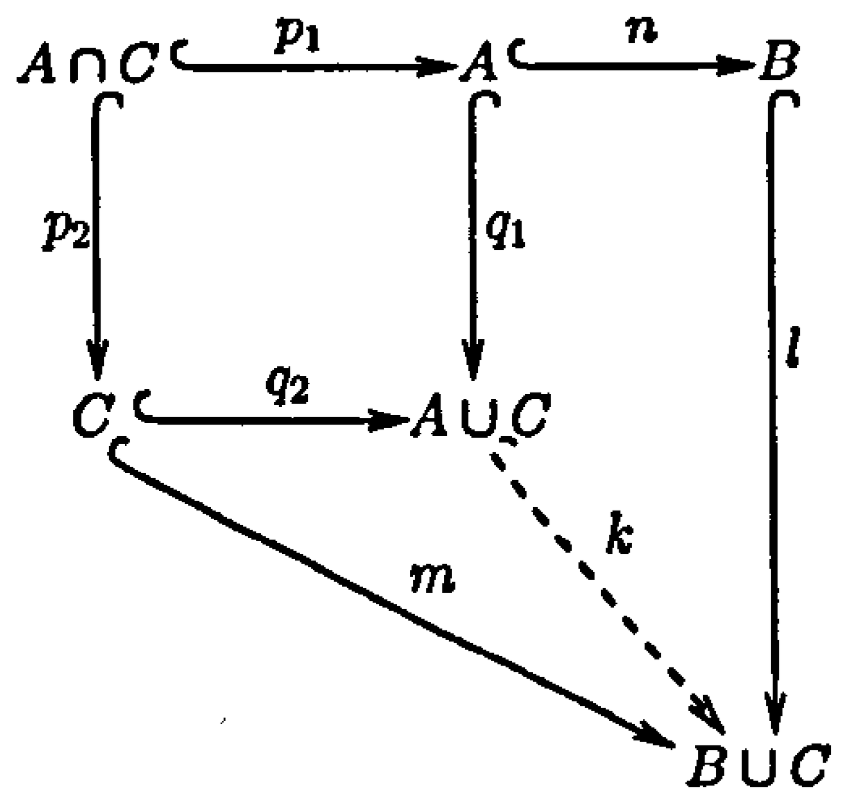
[1] In the sense that it has the same objects as $\mathbb{C}$.

Lemma 2.3.35. *Let A, B, C be objects in an inclusive category $\mathcal{C}$; if $A \hookrightarrow B$, then $A \cup C \hookrightarrow B \cup C$.*

Proof: The following is a pushout square for objects A and C of the form:



Since $A \hookrightarrow B$ and thanks to the fact that $\mathcal{C}$ is inclusive, we can construct the following diagram:



As $p_1; n; l$ and $p_2; m$ are both inclusions $A \cap C \hookrightarrow B \cup C$, they are equal. Therefore there is $k : A \cup C \hookrightarrow B \cup C$ in **I** (as q_1, q_2 is a pushout in **I**). The fact that k is an inclusion follows from the fact that q_1, q_2 is a pushout in **I**. □

2.4 Introduction to institutions

In this section we focus our attention on the notion of institution [42], which is central in our approach to ontologies. Before introducing institutions formally we briefly present some intuitions behind the theory of institutions, which might be found useful by the reader. For a more in-depth presentation of institution theory and its philosophical background one might refer to [33].

The notion of institution was first introduced by Joseph Goguen and Rod Burstall in the late 1970's as an answer to the increasing number of logical systems, originally presented in [23] and later also in [41]. But it took many years for a very important paper [42] to be eventually published, c.f. [33]. The original aim of institutions was to treat logics with model-theoretic semantics in a systematic way. Thanks to that it is possible to describe logical systems in an abstract and general way, this also allows to present and solve problems independently of any particular logical system. The theory of institutions formalizes the notion of logical system by presenting syntax (signatures, sentences), semantics (models) and the satisfaction relation between them. The theory of institutions appears as an important part of universal model theory and thus as a part of the universal logic project advocated by Béziau [16]. On the other hand, as suggested in [33], this abstract approach was the reason why some logicians regarded it as 'weakly informative' and rejected it.

In our work we use the theory of institutions to present relations between logical systems with particular attention to description logics and to approach some problems related with the use of ontologies such as entailment and inseparability.

The notion of institution strongly relies on concepts from category theory. An institution consists of a category of signatures; with each signature we associate sentences, models and a satisfaction relation. The core of the idea of institution is that change of the signature (by means of a signature morphism) leads to coherent changes in sentences and models and thus the satisfaction relation is not affected by the change of signature. This reflects the intuition that the truth of a sentence does not depend on the signature used in it, which is expressed in the slogan '*truth is invariant under change of notation*'. This is made explicit in the definition below. It is important to note that sentences are translated in the same direction as the signature, whereas models are translated in the opposite direction.

Formally, an institution is defined in the following way:

Definition 2.4.1. *An institution $\mathcal{I}$ consists of:*

1. *a category Sig of signatures,*
2. *a functor $Sen : Sig \to \mathsf{Set}$ giving, for each signature Σ, the set of sentences $Sen(\Sigma)$, and for each signature morphism $\sigma : \Sigma \to \Sigma'$, the sentence translation map $Sen(\sigma) : Sen(\Sigma) \to Sen(\Sigma')$,*
3. *a functor $Mod : Sig^{op} \to \mathcal{CAT}$ giving, for each signature Σ, the category $Mod(\Sigma)$, whose objects are called Σ-models, and whose arrows are called Σ-model homomorphisms, and for each signature morphism $\sigma : \Sigma \to \Sigma'$, the reduct functor $Mod(\sigma) : Mod(\Sigma') \to Mod(\Sigma)$,*
4. *a satisfaction relation $\models_\Sigma \subseteq |Mod(\Sigma)| \times Sen(\Sigma)$ for each $\Sigma \in |Sig|$, such that for each $\sigma : \Sigma \to \Sigma' \in Sig$ the following* **satisfaction condition** *holds:*

$$M' \models_{\Sigma'} Sen(\sigma)(\varphi) \text{ iff } Mod(\sigma)(M') \models_\Sigma \varphi$$

The generality of this definition, is the key point of theory of institutions as it allows to treat logical systems in an abstract way and captures the essence of the notion of logical system.

Before we continue we introduce some notations. First of all we simplify the notation for translation map for functors Sen and the reduct functor for functor Mod.

Notation 2.4.2. *For the sake of simplicity often we will write only σ instead of $Sen(\sigma)$, and $\_\!\upharpoonright_\sigma$ instead of $Mod(\sigma)$. The functor $\_\!\upharpoonright_\sigma$ is called the* **reduct** *functor associated to σ.*

Using this notation the satisfaction condition is of the form:

$$M' \models_{\Sigma'} \sigma(\varphi) \quad \text{iff} \quad M'\restriction_\sigma \models_\Sigma \varphi\,.$$

Now we introduce the notation for the satisfaction relation.

Notation 2.4.3. *When working with an institution, we will use the standard logical terminology. For instance, for institution $\mathcal{I}$, signature $\Sigma \in Sig^{\mathcal{I}}$, a sentence $\varphi \in Sen^{\mathcal{I}}(\Sigma)$, a finite set of sentences $\Gamma \subseteq Sen^{\mathcal{I}}(\Sigma)$ and a model $\mathcal{M} \in |Mod^{\mathcal{I}}(\Sigma)|$, we say that:*

- *$\mathcal{M}$ satisfies φ or that φ holds in $\mathcal{M}$, whenever $\mathcal{M} \models^{\mathcal{I}}_{\Sigma} \varphi$ (when clear from context we will omit the superscript $\mathcal{I}$),*
- *$\mathcal{M}$ is a model of Γ if it satisfies all the sentences in Γ.*

We write $\Gamma \models \varphi$ if every model $\mathcal{M}$ of Γ satisfies φ.

Notation 2.4.4. *In many cases, given an institution $\mathcal{I}$, we find it useful to use its name as a supper script of its components. So we have $\mathcal{I} = (Sig^{\mathcal{I}}, Sen^{\mathcal{I}}, Mod^{\mathcal{I}}, \models^{\mathcal{I}})$.*

This becomes very helpful when we start moving between different institutions.

Now, following [42], we present the basic properties of theories over an arbitrary institution. A theory consists of a signature Σ and a "closed" collection of Σ-sentences.

Definition 2.4.5. *Let $\mathcal{I}$ be an institution and Σ a signature.*

1. *Let $E \subseteq Sen(\Sigma)$, then a pair $\langle\Sigma, E\rangle$ is a* ***Σ-presentation.***
2. *Let $\mathcal{M} \in |Mod(\Sigma)|$. We say that $\mathcal{M}$* ***satisfies*** *a presentation $\langle\Sigma, E\rangle$ if it satisfies every sentence in E, for short $\mathcal{M} \models E$.*
3. *Given $E \subseteq Sen(\Sigma)$, let E° be the collection of all Σ-models that satisfy every sentence in E.*
4. *Given a collection $\mathcal{M}$ of Σ-models, let $\mathcal{M}^\circ$ be the collection of all Σ-sentences that are satisfied by each model in $\mathcal{M}$. We call $\mathcal{M}^\circ$ the theory of $\mathcal{M}$.*
5. *The* ***closure*** *of a collection E of Σ-sentences is $E^{\circ\circ}$, denoted E^*.*
6. *A collection E of Σ-sentences is* ***closed*** *iff $E = E^*$.*
7. *A* ***Σ-theory*** *is a presentation $\langle\Sigma, E\rangle$ such that E is closed. We denote a category of theories of $\mathcal{I}$ by $TH_{\mathcal{I}}$.*
8. *The Σ-theory* ***presented*** *by a presentation $\langle\Sigma, E\rangle$ is $\langle\Sigma, E^*\rangle$.*
9. *Let $\varphi \in Sen(\Sigma)$ and $E \subseteq Sen(\Sigma)$. We say that φ is* ***semantically entailed*** *by E, for short $E \models \varphi$, iff $\varphi \in E^*$.*

For description logics a presentation $\langle\Sigma, E\rangle$ is called a Σ-ontology, i.e. a Σ-ontology is a set of sentences over signature Σ.

Many examples of institutions can be found in the literature (e.g. [3,42,79]). To list only few examples of institutions we have: Propositional Logic (PL), unsorted First Order Logic (FOL), $\mathcal{EL}$, $\mathcal{EL}^+$, $\mathcal{ALC}$. Here we also consider Conceptual Hierarchy ($\mathcal{CH}$) as it may be considered a very simple institution allowing us to express hierarchies of concepts. Later we introduce its variant $\overline{\mathcal{CH}}$, which allows for formulation of hierarchies of roles. In Section 4.2 we use $\overline{\mathcal{CH}}$ to define description logics and for constructing expressions with individuals.

Example 2.4.6. ***Propositional Logic (PL).*** *Signatures and signature morphisms are sets of propositional variables and functions between them respectively. Given a signature Σ, the set of Σ-sentences is the least set of sentences finitely built over propositional variables in Σ and Boolean connectives in $\{\neg, \vee\}$. Given a signature morphism $\sigma : \Sigma \to \Sigma'$, $Sen^{PL}(\sigma)$ translates Σ-formulae to Σ'-formulae by renaming propositional variables according to σ. Given a signature Σ, the category of Σ-models is the category of mappings $v : \Sigma \to \{0,1\}$ (where $\{0,1\}$ are the usual truth-values) with identities as morphisms, i.e. it is a discrete category. Given a signature morphism $\sigma : \Sigma \to \Sigma'$, the reduct functor $\_\!\restriction_\sigma$ maps a Σ'-model v' to the Σ-model $v = \sigma; v'$. Satisfaction is the usual propositional satisfaction. A model v satisfies a formula φ (written $v \models \varphi$) iff $v(\varphi) = 1$.*

Given a signature Σ, signature morphism $\sigma : \Sigma \to \Sigma'$, a formula $\varphi \in Sen^{PL}(\Sigma)$ and a model $v \in |Mod^{PL}(\Sigma')|$ the following holds:

$$v \restriction_\sigma \models_\Sigma \varphi \quad \textit{iff} \quad v \models_{\Sigma'} \sigma(\varphi)\ .$$

Proof: *The proof is by induction. We distinguish three cases:*

- $\varphi = p$, *with* $p \in Sen^{PL}(\Sigma)$, *then:*

$$\begin{array}{ll} & v\restriction_\sigma \models_\Sigma p \\ \textit{iff} & \\ & \sigma; v \models_\Sigma p \\ \textit{iff} & \\ & v(\sigma(p)) = 1 \\ \textit{iff} & \\ & v \models_{\Sigma'} \sigma(p) \end{array}$$

- $\varphi = \neg p$, *with* $p \in Sen^{PL}(\Sigma)$, *then:*

$$\begin{array}{ll} & v\restriction_\sigma \models_\Sigma \neg p \\ \textit{iff} & \\ & \sigma; v \models_\Sigma \neg p \\ \textit{iff} & \\ & v(\sigma(p)) = 0 \\ \textit{iff} & \\ & v \models_{\Sigma'} \sigma(\neg p) \end{array}$$

Let $\varphi = c \vee d$, *such that* $c, d \in Sen^{PL}(\Sigma)$, *then:*

$$v{\restriction_\sigma} \models_\Sigma c \vee d$$
iff
$$\sigma; v \models_\Sigma c \vee d$$
iff
$$v(\sigma(c)) = 1 \text{ or } v(\sigma(d)) = 1$$
iff
$$v \models_{\Sigma'} \sigma(c \vee d)$$

□

Example 2.4.7. *FOL. A signature is a family of sets of predicate names* $(\Pi_n)_{n\in\omega}$, *where* n *is an arity.*

Signature morphisms $\sigma : \Pi \to \Pi'$ *are families of arity respecting functions between sets of predicates, i.e.* $\sigma_n : \Pi_n \to \Pi'_n$.

We assume the presence of a denumerable set of variables. Formulae are first-order formulae. Sentences are the first-order sentences. Sentence translation means replacement of the translated symbols. Given a signature Π, *FOL models are unsorted first-order structures of the form* $(\Delta^{\mathcal{M}}, (\pi^{\mathcal{M}})_{\pi\in\Pi_n})$, *with an n-ary relation* $\pi^{\mathcal{M}} \subseteq (\Delta^{\mathcal{M}})^n$ *for every* $\pi \in \Pi_n$, *which extend to formulae.*

Model reduct means reassembling the predicates according to the signature morphism, i.e. $\pi^{\mathcal{M}\restriction_\sigma} = (\sigma(\pi))^{\mathcal{M}}$, *the domain remains the same* $\Delta^{\mathcal{M}\restriction_\sigma} = \Delta^{\mathcal{M}}$. *Satisfaction is the usual satisfaction of a first-order logic.*

A further example we use in subsequent sections is a very basic description logic that we call $\mathcal{CH}$, which allows the specification of concept hierarchies. Here we present a variant of $\mathcal{CH}$ which does not capture role inclusion axioms but is very intuitive. Later, in Section 4.2 we present its variant $\overline{\mathcal{CH}}$, which captures also role inclusion axioms.

Example 2.4.8. ***Institution of Conceptual Hierarchy*** $\mathcal{CH}$.

A $\mathcal{CH}$*-signature is a set of concepts. (Later we consider institutions with institution morphisms to* $\mathcal{CH}$, *then for each such an institution* $\mathcal{I}$, *signature in* $\mathcal{CH}$ *is a set of concepts formulated over signature in* $\mathcal{I}$.)

Signature morphisms $\sigma : \Sigma \to \Sigma'$ *are functions between the sets of concepts.*

Given a $\mathcal{CH}$*-signature* Σ, *we define* Σ*-sentences in the following way:*

$$Sen^{\mathcal{CH}}(\Sigma) ::= \{c \sqsubseteq d \mid c, d \in \Sigma\}$$

Given a signature morphism $\sigma : \Sigma \to \Sigma'$, *we have* $Sen^{\mathcal{CH}}(\sigma) : Sen^{\mathcal{CH}}(\Sigma) \to Sen^{\mathcal{CH}}(\Sigma')$, *renames concepts according to* σ, *i.e.* $\sigma(c \sqsubseteq d) = \sigma(c) \sqsubseteq \sigma(d)$.

The semantics of $\mathcal{CH}$ *is defined by means of interpretations* $\mathcal{M} = (\Delta^{\mathcal{M}}, \cdot^{\mathcal{M}})$, *which are objects in the category of models. The interpretation domain* $\Delta^{\mathcal{M}}$ *is a set, and* $\cdot^{\mathcal{M}}$ *is a function mapping each concept name* $c \in \Sigma$ *to a subset* $c^{\mathcal{M}}$ *of* $\Delta^{\mathcal{M}}$. *Given a signature morphism* $\sigma : \Sigma \to \Sigma'$, *the reduct functor* $\mathcal{M} \restriction_\sigma$ *maps a* Σ'*-model* $\mathcal{M}'$ *to the* Σ*-model* $\mathcal{M} = \sigma; \mathcal{M}'$ *and* $\mathcal{M} \restriction_\sigma$ *is defined by* $c^{\mathcal{M}\restriction_\sigma} = \sigma(c)^{\mathcal{M}}$.

A straightforward argument shows that the satisfaction condition holds for $\mathcal{CH}$, i.e. given a signatures Σ, Σ', signature morphism $\sigma : \Sigma \to \Sigma'$, $\varphi \in Sen^{\mathcal{CH}}(\Sigma)$ and $\mathcal{M} \in |Mod^{\mathcal{CH}}(\Sigma')|$ the following holds:

$$\mathcal{M}\restriction_\sigma \models_\Sigma \varphi \quad \textit{iff} \quad \mathcal{M} \models_{\Sigma'} \sigma(\varphi) \ ,$$

Proof: Let $\varphi = c \sqsubseteq d$ and $\mathcal{M} \in \mathsf{Mod}(\Sigma')$.

$$\mathcal{M}\restriction_\sigma \models_\Sigma c \sqsubseteq d$$

iff

$$c^{\mathcal{M}\restriction_\sigma} \subseteq d^{\mathcal{M}\restriction_\sigma}$$

iff

$$\sigma(c)^{\mathcal{M}} \subseteq \sigma(d)^{\mathcal{M}}$$

iff

$$\mathcal{M} \models_{\Sigma'} \sigma(c) \sqsubseteq \sigma(d)$$

iff

$$\mathcal{M} \models_{\Sigma'} \sigma(c \sqsubseteq d)$$

□

Example 2.4.9. *Description logic $\mathcal{EL}$. An $\mathcal{EL}$-signature is a pair (P, R), where P is a set of 'concept names' and R is a set of 'role names'.*

Signature morphisms $\sigma : (P, R) \to (P', R')$ consist of two functions between the sets of concept names and sets of role names respectively, i.e., $\sigma = (f, g)$ with $f : P \to P'$ and $g : R \to R'$. Composition is defined pairwise $(f, g); (f', g') = (f; f', g; g')$.

Given an $\mathcal{EL}$-signature $\Sigma = (P, R)$, we define Σ-concepts $Con^{\mathcal{EL}}(\Sigma)$ using the following syntax rule:

$$Con^{\mathcal{EL}}(\Sigma) ::= \top \mid P \mid Con^{\mathcal{EL}}(\Sigma) \sqcap Con^{\mathcal{EL}}(\Sigma) \mid \exists R.Con^{\mathcal{EL}}(\Sigma) \ .$$

Where P and R are considered to be syntactic categories of the BNF definition. For every signature Σ, $Sen^{\mathcal{EL}}(\Sigma)$ is the set of General Concept Inclusions (GCI) over Σ,

$$Sen^{\mathcal{EL}}(\Sigma) ::= Con^{\mathcal{EL}}(\Sigma) \sqsubseteq Con^{\mathcal{EL}}(\Sigma) \ .$$

Given a signature morphism $\sigma : \Sigma \to \Sigma'$, we have

$$Sen^{\mathcal{EL}}(\sigma) : Sen^{\mathcal{EL}}(\Sigma) \to Sen^{\mathcal{EL}}(\Sigma')$$

this is done by renaming concept and role names according to σ.

The semantics of $\mathcal{EL}$ is defined by means of models $\mathcal{M} = (\Delta^{\mathcal{M}}, \cdot^{\mathcal{M}})$, which are objects in the category of models. The interpretation domain $\Delta^{\mathcal{M}}$ is a non-empty set, and $\cdot^{\mathcal{M}}$ is a function mapping each concept name $A \in P$ to a subset $A^{\mathcal{M}}$ of $\Delta^{\mathcal{M}}$, and each role name $r \in R$ to a binary relation $r^{\mathcal{M}} \subseteq \Delta^{\mathcal{M}} \times \Delta^{\mathcal{M}}$. The function $\cdot^{\mathcal{M}}$ is inductively extended to arbitrary concepts by setting

$$\top^{\mathcal{M}} := \Delta^{\mathcal{M}} \ ,$$

$$(C \sqcap D)^{\mathcal{M}} := C^{\mathcal{M}} \cap D^{\mathcal{M}} \ ,$$

and

$$(\exists r.C)^{\mathcal{M}} := \{d \in \Delta^{\mathcal{M}} \mid \text{ there is an } e \in \Delta^{\mathcal{M}} \text{ such that } (d,e) \in r^{\mathcal{M}} \text{ and } e \in C^{\mathcal{M}}\} .$$

This applies to every description logic.

When convenient we will refer to $\mathcal{EL}$ models as triples of the form:

$$(\Delta^{\mathcal{M}}, (p^{\mathcal{M}})_{p\in P}, (r^{\mathcal{M}})_{r\in R}) .$$

Model reduct is defined similarly as in FOL. Satisfaction in $\mathcal{EL}$ is the standard satisfaction of description logics. An interpretation $\mathcal{M}$ satisfies a GCI $C \sqsubseteq D$ (written $\mathcal{M} \models C \sqsubseteq D$) iff $C^{\mathcal{M}} \subseteq D^{\mathcal{M}}$.

The proof that the satisfaction condition holds for $\mathcal{EL}$ is similar to that for $\mathcal{CH}$.

Example 2.4.10. ***Description logic $\mathcal{EL}^+$.*** *Signatures of $\mathcal{EL}^+$ and $\mathcal{EL}$ are exactly the same.*

The description logic $\mathcal{EL}^+$ is an extension of $\mathcal{EL}$ with role inclusions. Thus $Sen^{\mathcal{EL}}$ is extended to $Sen^{\mathcal{EL}^+}(\Sigma)$ by adding role inclusions of the form $r_1 \circ \cdots \circ r_n \sqsubseteq r$ and $r \sqsubseteq s$, where $r, r_1, \cdots r_n, s$ are roles. We call a set of RIs an RBox.

Signature morphisms induce adequate changes in concept and role names used for sentence formulation just as in the case for $\mathcal{EL}$.

$\mathcal{EL}^+$-models and $\mathcal{EL}$-models are exactly the same.

Satisfaction is the standard satisfaction of description logics. Conditions under which model $\mathcal{M}$ satisfies GCIs and ontologies are exactly the same as for $\mathcal{EL}$. In addition to satisfaction for $\mathcal{EL}$ we have the following condition for RIs. An interpretation $\mathcal{M}$ satisfies an RI $r_1 \circ \cdots \circ r_n \sqsubseteq r$, $n \geq 1$, (written $\mathcal{M} \models r_1 \circ \cdots \circ r_n \sqsubseteq r$) if $r_1^{\mathcal{M}} \circ \cdots \circ r_n^{\mathcal{M}} \subseteq r^{\mathcal{M}}$, where '$\circ$' is interpreted as the composition of binary relations (i.e., we consider '$\circ$' to be defined as $R \circ S = \{(d, d'') \mid dRd', d'Sd'', \text{for some } d'\}$). $\mathcal{M}$ is a model of an RBox $\mathcal{R}$ if it satisfies all RIs in $\mathcal{R}$. Model reduct is the same as for $\mathcal{EL}$.

A straightforward inductive argument shows that the satisfaction condition holds.

Example 2.4.11. ***Description logic $\mathcal{ALC}$.*** *$\mathcal{ALC}$-signatures and signature morphisms are exactly the same as for $\mathcal{EL}$.*

The description logic $\mathcal{ALC}$ is an extension of $\mathcal{EL}$ with negation ($\neg$).

Thus given $\mathcal{ALC}$-signature $\Sigma = (P, R)$, we define Σ-concepts $Con^{\mathcal{ALC}}(\Sigma)$ using the following syntax rule:

$$Con^{\mathcal{ALC}}(\Sigma) ::= \top \mid P \mid \neg Con^{\mathcal{ALC}}(\Sigma) \mid Con^{\mathcal{ALC}}(\Sigma) \sqcap Con^{\mathcal{ALC}}(\Sigma) \mid \exists R.Con^{\mathcal{ALC}}(\Sigma) .$$

It is easy to see that using negation, we can define also $\bot, Con^{\mathcal{ALC}}(\Sigma) \sqcup Con^{\mathcal{ALC}}(\Sigma)$ and $\forall R.Con^{\mathcal{ALC}}(\Sigma)$ in the standard way.

For every signature Σ, $Sen^{\mathcal{ALC}}(\Sigma)$ is the set of GCIs over Σ,

$$Sen^{\mathcal{ALC}}(\Sigma) ::= Con^{\mathcal{ALC}}(\Sigma) \sqsubseteq Con^{\mathcal{ALC}}(\Sigma) .$$

Again, P and R are considered to be syntactic categories of the BNF definition.

Signature morphisms induce adequate changes in concept and role names used for sentence formulation just as in the case for $\mathcal{EL}$.

$\mathcal{ALC}$-models are of the same nature as $\mathcal{EL}$-models, as negation is available in $\mathcal{ALC}$ we have $\neg C^{\mathcal{M}} = \Delta^{\mathcal{M}} \setminus C^{\mathcal{M}}$. Model reduct is defined similarly as in FOL.

Satisfaction is the standard satisfaction of description logics. Conditions under which model $\mathcal{M}$ satisfies GCIs and ontologies are exactly the same as for $\mathcal{EL}$.

The proof that the satisfaction condition holds for $\mathcal{ALC}$ is similar to that for $\mathcal{EL}$.

2.4.1 Inclusive Institutions

Inclusions are very simple and natural, yet important type of maps. As already suggested in [42] inclusions are very important for modularisation and with the use of institutions it is possible to formulate this notion in an independent way. [42] left axiomatising and exploiting inclusions for modularisation amongst the open problems, and the notion was first formalized in [35] where it was used for simplification of the semantics of module systems over an institution. The notion of inclusion system received attention in the literature, for instance [29, 30, 67]. This notion was also discussed in [43], here we present the formulation of inclusive institution presented there.

Definition 2.4.12. *An* ***inclusive institution*** *is an institution with its category of signatures and its Sen functor both inclusive, in other words the category of signatures is equipped with an inclusion system such that $\Sigma \hookrightarrow \Sigma'$ implies $Sen(\Sigma) \subseteq Sen(\Sigma')$. An inclusive institution is* ***distributive*** *iff its category of signatures is distributive.*

Example 2.4.13 ($\mathcal{EL}$ is an inclusive institution)**.** *We have to show the following:*

1. *the category of $\mathcal{EL}$-signatures is inclusive, and*

2. *the functor Sen is inclusive.*

For (1), it is enough to notice that category of $\mathcal{EL}$-signatures has a broad subcategory **I** *such that $(P,R) \overset{(f,g)}{\hookrightarrow} (P',R')$ is in* **I** *iff $P \overset{f}{\hookrightarrow} P'$ and $R \overset{g}{\hookrightarrow} R'$ are inclusions in category of Sets, i.e. $P \subseteq P'$ and $R \subseteq R'$. As already mentioned signatures behave like sets and any $\mathcal{EL}$ signature consists of two disjoint sets P and R. Thanks to that, for arbitrary $\mathcal{EL}$-signatures (P,R) and (P',R') we have that $P \hookrightarrow P \cup P' \hookleftarrow P'$ is a pushout of $P \hookleftarrow P \cap P' \hookrightarrow P'$ and $R \hookrightarrow R \cup R' \hookleftarrow R'$ is a pushout of $R \hookleftarrow R \cap R' \hookrightarrow R'$. By taking these two together we receive that $(P,R) \hookrightarrow (P \cup P', R \cup R') \hookleftarrow (P',R')$ is a pushout of $(P,R) \hookleftarrow (P \cap P', R \cap R') \hookrightarrow (P',R')$.*

For (2) it is enough to recall the syntax rule for $\mathcal{EL}$. It tells us that given an $\mathcal{EL}$-signature Σ, $\mathcal{EL}$-concepts are built in the following way: $Con^{\mathcal{EL}}(\Sigma) ::= \top \mid P \mid Con^{\mathcal{EL}}(\Sigma) \sqcap Con^{\mathcal{EL}}(\Sigma) \mid \exists R.Con^{\mathcal{EL}}(\Sigma)$. As a signature inclusion $(P,R) \hookrightarrow (P',R')$ consists of two inclusions $(P) \hookrightarrow (P')$ and $(R) \hookrightarrow (R')$, it is easy to see that this entails concept inclusion $Con^{\mathcal{EL}}(P,R) \hookrightarrow Con^{\mathcal{EL}}(P',R')$.

Above we have shown that $\mathcal{EL}$ is inclusive and distributive. In a very similar way we may show that all the institutions discussed above have these properties as well.

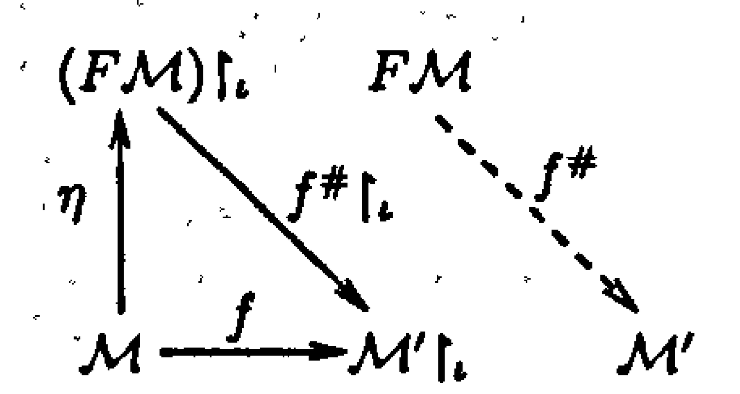
One of the important properties of many logical systems is that of preserving finite colimits by the model functor. Thus given signatures Σ_1 and Σ_2 such that signature Σ' is their colimit we would expect that $\mathsf{Mod}(\Sigma')$ is the limit. In particular we would expect that a Σ'-model would consist of a pair, a Σ_1-model and a Σ_2-model, in other words, we expect $\mathsf{Mod}(\Sigma')$ to be $\mathsf{Mod}(\Sigma_1) \times \mathsf{Mod}(\Sigma_2)$. Analogously for pushouts, if signature Σ' is a pushout of $\Sigma \to \Sigma_1$ and $\Sigma \to \Sigma_2$, then we would like $\mathsf{Mod}(\Sigma')$ to be the pullback of $\mathsf{Mod}(\Sigma_1) \to \mathsf{Mod}(\Sigma)$ and $\mathsf{Mod}(\Sigma_2) \to \mathsf{Mod}(\Sigma)$. This property is called exactness and originates in Tarlecki's work [70] and [76]. Meseguer in [64] introduces the term **exactness**, but it refers to the situation which we call **semi-exactness** here as this terminology was used by Diaconescu in [35]

Definition 2.4.14. *An institution is **exact** iff the functor $Mod : Sig^{op} \to \mathcal{CAT}$ preserves finite colimits and is **semi-exact** iff it preserves pushouts, i.e., it takes pushouts in Sig to pullbacks in $\mathcal{CAT}$.*

As already pointed out by Diaconescu in [32] semi-exactness is a very widespread property, all institutions of conventional or non-conventional logics are at least semi-exact. In fact, in [35] Diaconescu et al. tell us that institutions of many sorted logics are exact and institutions of unsorted (or one-sorted) logics are semi-exact.

Definition 2.4.15. *In any institution $\mathcal{I}$, a signature morphism $\iota : \Sigma \to \Sigma'$ is liberal if and only if the reduct functor $Mod(\iota) : Mod(\Sigma') \to Mod(\Sigma)$ has a left adjoint.*

In other words, for each Σ-model $\mathcal{M}$ there exists a Σ'-model $F\mathcal{M}$ and a Σ-model homomorphism $\eta : \mathcal{M} \to (F\mathcal{M})\restriction_\iota$



such that for each Σ'-model $\mathcal{M}'$ and for each Σ-model homomorphism $f : \mathcal{M} \to \mathcal{M}'\restriction_\iota$, there exists a unique Σ'-model homomorphism $f^{\#} : FM \to M'$ such that $\eta; f^{\#}\restriction_\iota = f$.

An institution $\mathcal{I}$ is **liberal** if and only if every signature morphism in $\mathcal{I}$ is liberal.

A special case of this property is model extension along inclusions. This notion can be found in [32].

Definition 2.4.16. *Institution $\mathcal{I}$ has **model extension along inclusions** iff for $\Sigma \hookrightarrow \Sigma'$ in $Sig^{\mathcal{I}}$ every Σ-model $\mathcal{M}$ has a Σ'-model $\mathcal{M}'$, such that $\mathcal{M}'\restriction^*_\Sigma = \mathcal{M}^*$*

Example 2.4.17 ($\mathcal{EL}$ has model extension along inclusions)**.** *To show that we simply show how given a signature inclusion $\Sigma \hookrightarrow \Sigma'$ and a Σ-model $\mathcal{M}$ we construct a desired Σ'-model $\mathcal{M}'$.*

Let $\mathcal{M} = (\Delta^{\mathcal{M}}, (p^{\mathcal{M}})_{p \in P}, (r^{\mathcal{M}})_{r \in R})$, we define $\mathcal{M}'$ in the following way:

- $\Delta^{\mathcal{M}'} = \Delta^{\mathcal{M}}$,
- *for* $p \in P'$:
  - *if* $p \in P : p^{\mathcal{M}'} := p^{\mathcal{M}}$,
  - *if* $p \notin P : p^{\mathcal{M}'} := \emptyset$
- *for* $r \in R'$:
  - *if* $r \in R : r^{\mathcal{M}'} := r^{\mathcal{M}}$,
  - *if* $r \notin R : r^{\mathcal{M}'} := \emptyset$

Now we only have to show that $\mathcal{M}'{\restriction}^{*}_{\Sigma} = \mathcal{M}^{*}$. *To do that we introduce the following lemma.*

Lemma 2.4.18. *For any* $\Sigma, \Sigma' \in Sig^{\mathcal{EL}}$ *such that* $\Sigma \hookrightarrow \Sigma'$ *and any* Σ*-concept* C *we have that* $C^{\mathcal{M}'{\restriction}_{\Sigma}} = C^{\mathcal{M}}$.

Proof: *The proof is by induction on the structure of* C.

- *In the induction base, we have that* $C = \top$ *or* $C = A$ *with* $A \in \Sigma$. *The former is trivial, for the latter we have:*

$$A^{\mathcal{M}'{\restriction}_{\Sigma}} = A^{\mathcal{M}'} = A^{\mathcal{M}} \ .$$

- *The case for* $C_1 \sqcap C_2$ *is trivial.*
- *For* $C = \exists r.C'$

$$\begin{aligned} \exists r.C'^{\mathcal{M}'{\restriction}_{\Sigma}} &= \{x \in \Delta^{\mathcal{M}'{\restriction}_{\Sigma}} \mid \textit{for some } y,\ r^{\mathcal{M}'{\restriction}_{\Sigma}}(x,y) \wedge y \in C'^{\mathcal{M}'{\restriction}_{\Sigma}}\} \\ &= \{x \in \Delta^{\mathcal{M}} \mid \textit{for some } y,\ r^{\mathcal{M}}(x,y) \wedge y \in C'^{\mathcal{M}}\} \\ &= \exists r.C'^{\mathcal{M}} \end{aligned}$$

□

2.4.2 Morphisms and Comorphisms

One of the very important benefits that we get from using category theory is that translations between institutions (logical systems) can be treated in a systematic way. Thanks to that we can integrate theories over different logics, which can be very useful in practice. For different purposes there are various kinds of translation available. Probably the two best-known and most basic are institution morphisms and institution comorphisms. Morphisms were originally introduced by Goguen and Burstall in [42]. Comorphisms were first introduced by Meseguer in [64] but then were called 'plain maps', later the same structures were renamed by Tarlecki in [77] and called 'representations'. We prefer the name 'comorphism' since it emphasizes the relation of this structure to morphisms.

Intuitively, institution morphisms are truth preserving translations from one logical system to another. An institution morphism shows how a 'richer' institution $\mathcal{I}$ is built over a 'poorer' institution $\mathcal{I}'$. This is done by defining a 'forgetful' operation from $\mathcal{I}$ to $\mathcal{I}'$. A functor Ψ translates the signatures of $\mathcal{I}$ into the signatures of $\mathcal{I}'$. A natural transformation γ translates sentences of $\mathcal{I}'$ over $\Psi(\Sigma)$ into sentences of $\mathcal{I}$ over Σ. A natural transformation δ translates Σ-models of $\mathcal{I}$ into $\Psi(\Sigma)$-models of $\mathcal{I}'$.

Definition 2.4.19 (Institution morphism). *Let* $\mathcal{I} = (Sig, Sen, Mod, \models)$ *and* $\mathcal{I}' = (Sig', Sen', Mod', \models')$ *be two institutions. An institution-morphism* $\mu = (\Psi^\mu, \gamma^\mu, \delta^\mu) : \mathcal{I} \to \mathcal{I}'$ *consists of:*

- *a functor* $\Psi^\mu : Sig \to Sig'$
- *a natural transformation* $\gamma^\mu : Sen' \circ \Psi^\mu \Rightarrow Sen$
- *a natural transformation* $\delta^\mu : Mod \Rightarrow Mod' \circ (\Psi^\mu)^{op}$

such that the following ***satisfaction condition*** *holds:*
for all $\Sigma \in |Sig|$, *for all* $\mathcal{M} \in |Mod(\Sigma)|$, *for all* $\varphi' \in Sen'(\Psi^\mu(\Sigma))$

$$\mathcal{M} \models^{\mathcal{I}}_{\Sigma} \gamma^\mu_\Sigma(\varphi') \quad \textit{iff} \quad \delta^\mu_\Sigma(\mathcal{M}) \models^{\mathcal{I}'}_{\Psi^\mu(\Sigma)} \varphi'$$

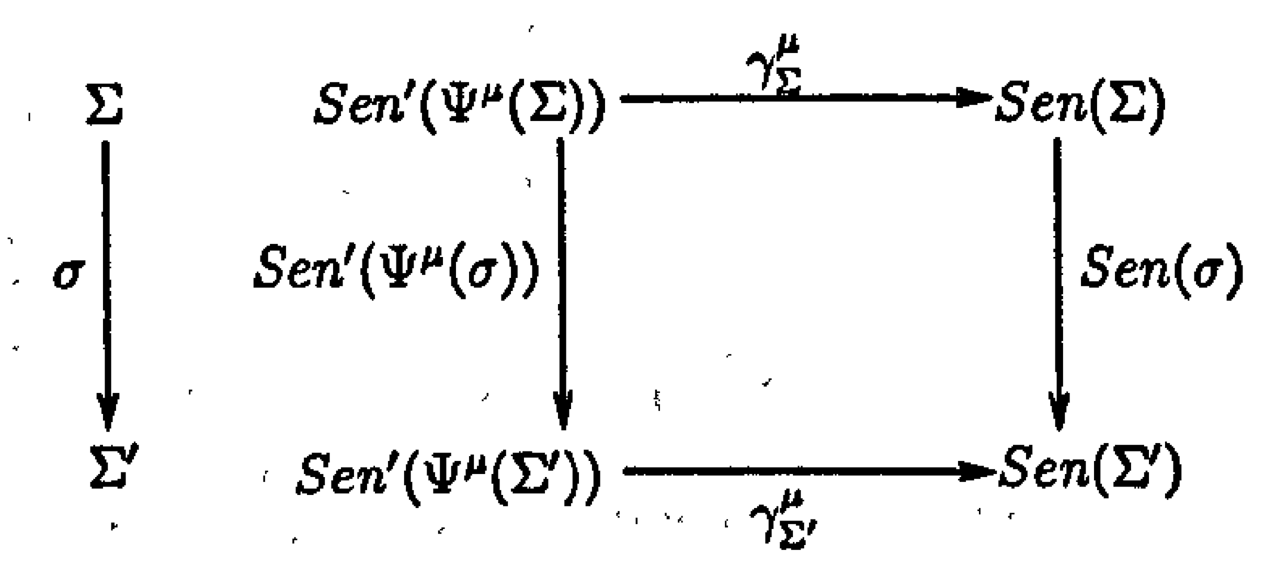


Figure 2.1: The Sentence Natural Transformation

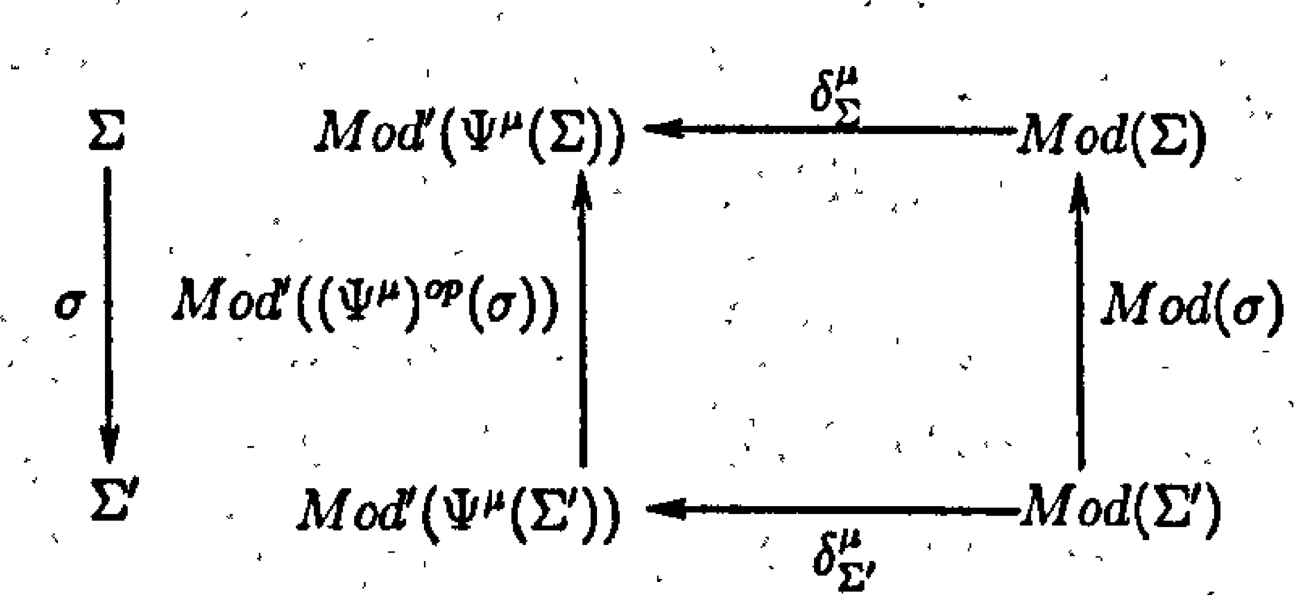


Figure 2.2: The Model Natural Transformation

Note that the functor Ψ and the natural transformation δ go in the same direction, whereas the natural transformation γ goes in the opposite direction.

Below we present possible morphisms between institutions that are of interest to us. Before presenting the first example, morphism fa : $FOL \to \mathcal{ALC}$, we need some additional explanation. While discussing natural transformation γ^{fa} we introduce counter $1-n$ for formulae using $\exists$ and $\forall$ as a way to introduce new variables, so we have the following: $\overline{\exists r.C}^n = ((\exists x_{1-n})\ r) \wedge \overline{C}^{1-n}$, and $\overline{\forall r.C}^n = ((\forall x_{1-n})\ r) \wedge \overline{C}^{1-n}$. The reason for introducing that counter is the fact that in case for morphism fa we are given FOL-signature Π, which is translated into $\mathcal{ALC}$ where we construct concepts that are translated back to FOL. But in FOL we can only express that some variable is an element of the interpretation of the predicate of our interest (these predicates are now concept names translated from $\mathcal{ALC}$). This becomes crucial when the $\mathcal{ALC}$ concept is of the form $\exists r.C$ or $\forall r.C$, for simplicity reasons we focus here only on $\exists r.C$ as the intuition for $\forall r.C$ is similar. When we deal with concept $\exists r.C$ translated into FOL we need two variables, one that is an element of the interpretation of $\exists r.C$ translated into FOL and another one which is related via r with the previous one as well as is an element of the interpretation of translated concept C. One of the problems here is that we may only talk about one variable at any time, another is that the concept translated from $\mathcal{ALC}$ may be of arbitrary depth, for instance of the form $\exists r.(\exists s.C)$ and so on, so for each role we need new variable. To solve that we introduce the counter $1-n$ which allows us to introduce new variables when needed and deals with the nesting problem. As a simple example consider an $\mathcal{ALC}$-concept $\exists r.C$, we translate it into FOL and set $n=0$, in other words we choose variable x_0 to be an element of the interpretation of $\overline{\exists r.C}^0$ in FOL. By the way how $\mathcal{ALC}$ concepts are translated into FOL we have that there is a variable x_1, such that x_0 and x_1 are related via r and x_1 is an element of the interpretation of $\overline{C}$, so we have $\overline{C}^1$. Now, let C be of the form $\exists s.C'$, thus we get $\overline{\exists s.C'}^1$, with $n=1$ received from the previous step. So we have that x_1 is in the interpretation of $\overline{\exists s.C'}$, thus there is a variable, which is related with x_1 via s and is in the interpretation of $\overline{C'}$. We apply our definition to introduce that variable, as at this point $n=1$ we receive that x_0 is that variable. This procedure allows us to introduce new variables as needed. To see how this definition is working consider the following example.

Example 2.4.20. *Let $C = \exists r.((\exists s.\varphi) \sqcap \psi) \sqcap \phi$. We translate this concept into FOL and set $n=0$, so we have:*

$$\overline{\exists r.((\exists s.\varphi) \sqcap \psi) \sqcap \phi}^0$$

this gives us

$$\overline{\exists r.((\exists s.\varphi) \sqcap \psi)}^0 \wedge \overline{\phi}^0.$$

By definition we receive

$$((\exists x_1) r(x_0, x_1) \wedge \overline{((\exists s.\varphi) \sqcap \psi)}^1) \wedge \phi(x_0),$$

which roughly states that x_0 has property ϕ and there is x_1, such that $r(x_0, x_1)$ and x_1 is an element in the interpretation of $\overline{((\exists s.\varphi) \sqcap \psi)}$. But after applying the definition to $\overline{(\exists s.\varphi) \sqcap \psi}$ we receive: $\overline{(\exists s.\varphi)}^1 \wedge \overline{\psi}^1$, and thus

$$((\exists x_0) s(x_1, x_0) \wedge \overline{(\varphi)}^0) \wedge \psi(x_1),$$

which by definition is

$$((\exists x_0)s(x_1, x_0) \wedge \varphi(x_0)) \wedge \psi(x_1).$$

This together with

$$((\exists x_1)r(x_0, x_1) \wedge \overline{((\exists s.\varphi) \sqcap \psi)}^{1}) \wedge \phi(x_0),$$

gives us:

$$((\exists x_1)r(x_0, x_1) \wedge ((\exists x_0)s(x_1, x_0) \wedge \varphi(x_0)) \wedge \psi(x_1)) \wedge \phi(x_0).$$

Example 2.4.21. *Morphism* fa : $FOL \to \mathcal{ALC}$.

Let Π *be a* FOL*-signature, where* Π *is a family of sets of predicate names* $(\Pi_n)_{n\in\omega}$ *where n is an arity. A morphism* fa : $FOL \to \mathcal{ALC}$ *consists of:*

- *the functor* Ψ^{fa} *translates an* FOL*-signature* Π *into an* $\mathcal{ALC}$*-signature in the following way:* $\Psi^{\mathsf{fa}}(\Pi) = (\Pi^{\{x_0\}}, \Pi^{\{x_0,x_1\}})$, *where*

  $$\Pi^{\{x_0\}} = \{\varphi \in Sen^{FOL}(\Pi) | \varphi \text{ contains exactly one free variable } x_0\}$$

  and

  $$\Pi^{\{x_0,x_1\}} = \{\varphi \in Sen^{FOL}(\Pi) | \varphi \text{ contains exactly two free variables } x_0 \text{ and } x_1\}.$$

  In other words, unary predicates of FOL *are translated into atomic concepts of* $\mathcal{ALC}$, *whereas binary predicates of* FOL *are translated into roles of* $\mathcal{ALC}$,

- *natural transformation* $\gamma^{\mathsf{fa}}_{\Pi}$ *translates* $\mathcal{ALC}$*-sentences, constructed over* $\Psi^{\mathsf{fa}}(\Pi)$, *into* FOL*-sentences with countable set of variables in the following way:*

  $$\gamma^{\mathsf{fa}}_{\Pi}(C \sqsubseteq D) = (\forall x_0)\overline{C}^0 \Rightarrow \overline{D}^0$$

  where we fix some enumeration of the variables and x_0 *is the first one. Concept translation is defined inductively:*

  $$\begin{array}{lrcl} C: & & & \\ & \overline{p}^n & = & p(x_n), \\ & \overline{\neg C}^n & = & \sim \overline{C}^n, \\ & \overline{C_1 \sqcap C_2}^n & = & \overline{C_1}^n \wedge \overline{C_2}^n, \\ & \overline{C_1 \sqcup C_2}^n & = & \overline{C_1}^n \vee \overline{C_2}^n, \\ & \overline{\exists r.C}^n & = & ((\exists x_{1-n})\ r) \wedge \overline{C}^{1-n}, \\ & \overline{\forall r.C}^n & = & ((\forall x_{1-n})\ r) \wedge \overline{C}^{1-n}. \end{array}$$

  where $n = 1, 2$. *For the last two clauses, we use* $1 - n$ *to allow us to introduce another variable.*

- *The natural transformation* $\delta^{\mathsf{fa}}_{\Pi}$ *converts a* FOL *model into an* $\mathcal{ALC}$ *model such that for any* FOL Π*-model* $\mathcal{M}$ *we have* $\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M}) = \mathcal{M}$, *i.e. for* $(P, R) = \Psi^{\mathsf{fa}}(\Pi)$ *we have that every* $p \in P$ *is a formula with one free variable, so* $p^{\mathcal{M}} \subseteq \Delta^{\mathcal{M}}$. *Similarly, every* $r \in R$ *is a formula with two free variables, so* $r^{\mathcal{M}} \subseteq \Delta^{\mathcal{M}} \times \Delta^{\mathcal{M}}$.

Natural transformations $\gamma^{\mathsf{fa}}_{\Pi}$ *and* $\delta^{\mathsf{fa}}_{\Pi}$ *satisfy the satisfaction condition: for all* $\Pi \in |Sig^{FOL}|$, *for all* $\mathcal{M} \in |Mod^{FOL}(\Pi)|$, *for all* $\varphi \in Sen^{\mathcal{ALC}}(\Psi^{\mathsf{fa}}(\Pi))$,

$$\mathcal{M} \models^{FOL}_{\Pi} \gamma^{\mathsf{fa}}_{\Pi}(\varphi) \text{ iff } \delta^{\mathsf{fa}}_{\Pi}(\mathcal{M}) \models^{\mathcal{ALC}}_{\Psi^{\mathsf{fa}}(\Pi)} \varphi\,,$$

this can be shown by induction.

Here we show only the base case. But first of all we need an auxiliary lemma.

Lemma 2.4.22. *For every FOL-signature* Π, *every* $\mathcal{ALC}$*-concept* C *over* $\delta^{\mathsf{fa}}(\Pi)$ *and every FOL-model* $\mathcal{M} \in |Mod(\Pi)|$ *we have* $C^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})} = \overline{C}^{n\mathcal{M}}$.

Proof: *The proof is by induction.*

Let:

- $C = p$

$$p^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})} = \{x \in \Delta^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})} | x \in p^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})}\} = \{x \in \Delta^{\mathcal{M}} | x \in p^{\mathcal{M}}\} = (p(x))^{\mathcal{M}}$$

- $C = \neg p$

$$(\neg p)^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})} = \{x \in \Delta^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})} | x \notin p^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})}\} = \{x \in \Delta^{\mathcal{M}} | x \notin p^{\mathcal{M}}\} = (\sim p(x))^{\mathcal{M}}$$

- $C = C_1 \sqcap C_2$ *is trivial,*

- $C = \exists r.C'$

$$\begin{aligned}
(\exists r.C')^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})} &= \{x_0 \in \Delta^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})} | (\exists x_1)\ x_0\ r^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})} x_1 \ \wedge\ x_1 \in (C')^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})}\} \\
&= \{x_0 \in \Delta^{\mathcal{M}} | (\exists x_1)\ x_0\ r^{\mathcal{M}} x_1 \ \wedge\ x_1 \in \overline{C'}^{1\mathcal{M}}\} \\
&= (((\exists x_1)\ r) \wedge \overline{C}^1)^{\mathcal{M}} \\
&= \overline{(\exists r.C')}^{0\mathcal{M}}
\end{aligned}$$

□

To show that the satisfaction condition holds, let $\Pi \in |Sig^{FOL}|$, $\mathcal{M} \in |Mod^{FOL}(\Pi)|$ *and* $\varphi = C \sqsubseteq D$ *be an* $\mathcal{ALC}$*-sentence over* $\Psi^{\mathsf{fa}}(\Pi)$. *Assume that* $\mathcal{M} \models^{FOL}_{\Pi} \gamma^{\mathsf{fa}}_{\Pi}(\varphi)$. *Thus*

$$\mathcal{M} \models^{FOL}_{\Pi} \gamma^{\mathsf{fa}}_{\Pi}(C \sqsubseteq D)$$

iff

$$\mathcal{M} \models^{FOL}_{\Pi} (\forall x_0)\overline{C}^0 \Rightarrow \overline{D}^0$$

iff

$$C^{\mathcal{M}} \subseteq D^{\mathcal{M}}$$

iff

$$C^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})} \subseteq D^{\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M})}$$

iff

$$\delta^{\mathsf{fa}}_{\Pi}(\mathcal{M}) \models^{\mathcal{ALC}}_{\Psi^{\mathsf{fa}}(\Pi)} C \sqsubseteq D$$

As it is easy to see that natural transformations in the remaining examples of (co)morphisms satisfy the satisfaction conditions we omit most of the proofs, in Example 2.4.25 we only present part of the proof where we consider the base case for RIs.

Example 2.4.23. *Morphism* $\mathsf{ae} : \mathcal{ALC} \to \mathcal{EL}$. *For morphism* ae *we let the functor* Ψ^{ae} *be the identity functor. For any* $\mathcal{ALC}$ *signature* $\Sigma = (P, R)$ *we have* $Sen^{\mathcal{EL}}(\Sigma) \subseteq Sen^{\mathcal{ALC}}(\Sigma)$, *thus we define* $\gamma^{\mathsf{ae}}_{\Sigma}$ *to be the inclusion* $Sen^{\mathcal{EL}}(\Sigma) \hookrightarrow Sen^{\mathcal{ALC}}(\Sigma)$. *The natural transformation* $\delta^{\mathsf{ae}}_{\Sigma}$ *is the identity.*

Example 2.4.24. *Morphism* $\mathsf{e^+e} : \mathcal{EL}^+ \to \mathcal{EL}$.
As $\mathcal{EL}$ *and* $\mathcal{EL}^+$ *use the same signatures we let the functor* $\Psi^{\mathsf{e^+e}}$ *be the identity functor. For any signature* $\Sigma = (P, R)$ *we have* $Sen^{\mathcal{EL}}(\Sigma) \subseteq Sen^{\mathcal{EL}^+}(\Sigma)$, *so we may take* $\gamma^{\mathsf{e^+e}}_{\Sigma}$ *to be the inclusion* $Sen^{\mathcal{EL}}(\Sigma) \hookrightarrow Sen^{\mathcal{EL}^+}(\Sigma)$. *We take* $\delta^{\mathsf{e^+e}}_{\Sigma}$ *to be the identity natural transformation, as* Σ*-models in* $\mathcal{EL}$ *and* $\mathcal{EL}^+$ *are the same.*

Example 2.4.25. *Morphism* $\mathsf{fe^+} : FOL \to \mathcal{EL}^+$.

We have noted above that signatures of $\mathcal{EL}$, $\mathcal{EL}^+$ *and* $\mathcal{ALC}$ *are built in the very same way, for that reason functor* $\Psi^{\mathsf{fe^+}}$ *works exactly like* Ψ^{fa} *for the morphism described in Example 2.4.21.*

Natural transformation $\gamma^{\mathsf{fe^+}}$ *translates* $\mathcal{EL}^+$*-sentences, constructed over* $\Psi^{\mathsf{fe^+}}(\Pi)$, *into* FOL*-sentences with a fixed countable set of variables. Note that for every signature* Π *the set of* $\mathcal{EL}^+$ *GCIs built over* $\Psi^{\mathsf{fe^+}}(\Pi)$ *is a subset of* $\mathcal{ALC}$ *GCIs built over* $\Psi^{\mathsf{fe^+}}(\Pi)$. *Thus* $\gamma^{\mathsf{fe^+}}$ *translates* $\mathcal{EL}^+$ *GCIs as in Example 2.4.21, with the restriction to atomic concepts and those using '*$\sqcap$*' and '*$\exists$*'. Since* $\mathcal{EL}^+$ *allows also role inclusion axioms,* $\gamma^{\mathsf{fe^+}}$ *translates them in the following way:*

$$\begin{aligned}\gamma^{\mathsf{fe^+}}(r \sqsubseteq s) &= (\forall x, x')\ r(x, x') \Rightarrow s(x, x'),\\ \gamma^{\mathsf{fe^+}}(r_1 \circ r_2 \sqsubseteq r) &= (\forall x, x', x'')\ r_1(x, x') \wedge r_2(x', x'') \Rightarrow r(x, x''),\end{aligned}$$

Note that models of $\mathcal{ALC}$ *and* $\mathcal{EL}^+$ *are the same, thus the natural transformation* $\delta^{\mathsf{fe^+}}$ *converts a* FOL *model into an* $\mathcal{EL}^+$ *model in the very same way as* δ^{fa} *described in Example 2.4.21.*

Natural transformations $\gamma^{\mathsf{fe^+}}$ *and* $\delta^{\mathsf{fe^+}}$ *satisfy the satisfaction condition: for all* $\Pi \in |Sig^{FOL}|$, *for all* $\mathcal{M} \in |Mod^{FOL}(\Pi)|$, *for all* $\varphi \in Sen^{\mathcal{EL}^+}(\Psi^{\mathsf{fe^+}}(\Pi))$,

$$\mathcal{M} \models^{FOL}_{\Pi} \gamma^{\mathsf{fe^+}}_{\Pi}(\varphi) \textit{ iff } \delta^{\mathsf{fe^+}}_{\Pi}(\mathcal{M}) \models^{\mathcal{EL}^+}_{\Psi^{\mathsf{fe^+}}(\Pi)} \varphi.$$

The satisfaction condition for GCIs is as in Example 2.4.21. Here we present the base case for RIs. But first we have to notice that by definition $r^{\delta^{\mathsf{fe^+}}_{\Pi}(\mathcal{M})} = r^{\mathcal{M}}$.

Let $\Pi \in |Sig^{FOL}|$, $\mathcal{M} \in |Mod^{FOL}(\Pi)|$ *and* $\varphi \in Sen^{\mathcal{EL}^+}(\Psi^{\mathsf{fe^+}}(\Pi))$. *Let us set* $r, s \in \Pi_2$ *and* $\varphi = r \sqsubseteq s$.

Assume that $\mathcal{M} \models^{FOL}_{\Pi} \gamma^{\mathsf{fe^+}}_{\Pi}(\varphi)$. *Thus*

$$
\begin{array}{llll}
 & \mathcal{M} & \models^{FOL}_{\Pi} & \gamma^{\mathsf{fe}^+}(r \sqsubseteq s) \\
\textit{iff} & & & \\
 & \mathcal{M} & \models^{FOL}_{\Pi} & (\forall x, y)\ r(x,y) \Rightarrow s(x,y) \\
\textit{iff} & & & \\
 & & r^{\mathcal{M}} \subseteq s^{\mathcal{M}} & \\
\textit{iff} & & & \\
 & & r^{\delta^{\mathsf{fe}^+}_{\Pi}\mathcal{M}} \subseteq s^{\delta^{\mathsf{fe}^+}_{\Pi}\mathcal{M}} & \\
\textit{iff} & & & \\
 & \delta^{\mathsf{fe}^+}_{\Pi}(\mathcal{M}) & \models^{\mathcal{EL}^+}_{\Psi^{\mathsf{fe}^+}(\Pi)} & r \sqsubseteq s
\end{array}
$$

Role inclusion axioms of the form $r_1 \circ r_2 \sqsubseteq r$ are treated in a similar way.

Definition 2.4.26 (Composition of morphisms). *Let $\mathcal{I}$, $\mathcal{I}_1$, $\mathcal{I}_2$ be institutions such that there are morphisms $\mu_1 = (\Psi^{\mu_1}, \gamma^{\mu_1}, \delta^{\mu_1}) : \mathcal{I} \to \mathcal{I}_1$ and $\mu_2 = (\Psi^{\mu_2}, \gamma^{\mu_2}, \delta^{\mu_2}) : \mathcal{I}_1 \to \mathcal{I}_2$. Then we can define a morphism $\mu = (\Psi^{\mu}, \gamma^{\mu}, \delta^{\mu}) : \mathcal{I} \to \mathcal{I}_2$ to be the composition of morphisms $\mu_1; \mu_2$, i.e. μ is a triple defined by means of composition in the following way: the functor $\Psi^{\mu} = \Psi^{\mu_1}; \Psi^{\mu_2}$, natural transformation $\gamma^{\mu} = \Psi^{\mu_1}\gamma^{\mu_2}; \gamma^{\mu_1}$ i.e., $\gamma^{\mu}_{\Sigma}(e) = \gamma^{\mu_1}_{\Sigma}(\gamma^{\mu_2}_{\Psi^{\mu_1}(\Sigma)}(e))$ and $\delta^{\mu} = \delta^{\mu_1}; \Psi^{\mu_1 op}\delta^{\mu_2}$ i.e., $\delta^{\mu}_{\Sigma}(\mathcal{M}) = \delta^{\mu_2}_{\Psi^{\mu_1}(\Sigma)}(\delta^{\mu_1}_{\Sigma}(\mathcal{M}))$. This is graphically represented in Figure 2.3.*

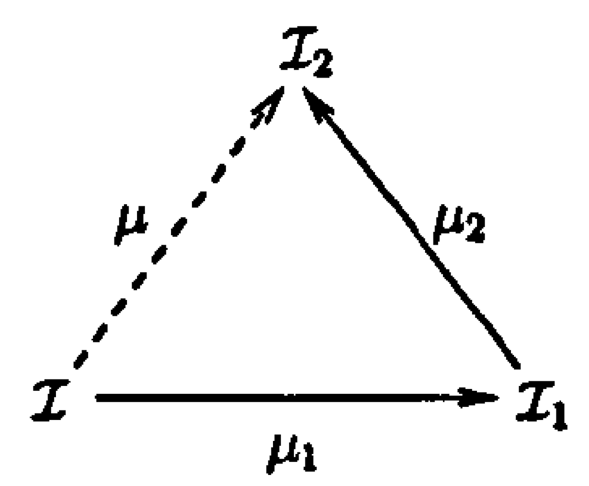


Figure 2.3: A composition of morphisms $\mu = \mu_1; \mu_2$

In more detail, Definition 2.4.26 states that given $\Sigma \in |\mathsf{Sig}^{\mathcal{I}}|$ functor Ψ^{μ_1} translates Σ into $\mathcal{I}_1$-signature $\Psi^{\mu_1}(\Sigma)$, which is then translated by Ψ^{μ_2} into $\mathcal{I}_1$-signature $\Psi^{\mu_2}(\Psi^{\mu_1}(\Sigma))$. Natural transformation γ^{μ_1} is defined in the following way: $\gamma^{\mu_1}_{\Sigma} : \mathsf{Sen}^{\mathcal{I}_1}(\Psi^{\mu_1}(\Sigma)) \to \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ and natural transformation γ^{μ_2} is defined in the following way: $\gamma^{\mu_2}_{\Sigma'} : \mathsf{Sen}^{\mathcal{I}_2}(\Psi^{\mu_2}(\Sigma')) \to \mathsf{Sen}^{\mathcal{I}_1}(\Sigma')$, where $\Sigma' \in |\mathsf{Sig}^{\mathcal{I}_1}|$. But as signatures of $\mathcal{I}_1$ are in fact signatures of $\mathcal{I}$ translated along Ψ^{μ_1} we have that $\gamma^{\mu_2}_{\Psi^{\mu_1}(\Sigma)} : \mathsf{Sen}^{\mathcal{I}_2}(\Psi^{\mu_2}(\Psi^{\mu_1}(\Sigma))) \to \mathsf{Sen}^{\mathcal{I}_1}(\Psi^{\mu_1}(\Sigma))$. Thus $\gamma^{\mu_2}_{\Psi^{\mu_1}(\Sigma)}; \gamma^{\mu_1}_{\Sigma} : \mathsf{Sen}^{\mathcal{I}_2}(\Psi^{\mu_2}(\Psi^{\mu_1}(\Sigma))) \to \mathsf{Sen}^{\mathcal{I}}(\Sigma)$. Natural transformation δ^{μ_1} is defined in the following way: $\delta^{\mu_1}_{\Sigma} : \mathsf{Mod}^{\mathcal{I}}(\Sigma) \to \mathsf{Mod}^{\mathcal{I}_1}(\Psi^{\mu_1}(\Sigma))$ and natural transformation δ^{μ_2} is defined in the following way: $\delta^{\mu_2}_{\Sigma'} : \mathsf{Mod}^{\mathcal{I}_1}(\Sigma') \to \mathsf{Mod}^{\mathcal{I}_2}(\Psi^{\mu_2}(\Sigma'))$, where $\Sigma' \in |\mathsf{Sig}^{\mathcal{I}_1}|$. But as signatures of $\mathcal{I}_1$ are in fact signatures of $\mathcal{I}$ translated along Ψ^{μ_1} we have that $\delta^{\mu_2}_{\Psi^{\mu_1}(\Sigma)} : \mathsf{Mod}^{\mathcal{I}_1}(\Psi^{\mu_1}(\Sigma)) \to \mathsf{Mod}^{\mathcal{I}_2}(\Psi^{\mu_2}(\Psi^{\mu_1}(\Sigma)))$. Thus $\delta^{\mu_1}_{\Sigma}; \Psi^{\mu_1 op}\delta^{\mu_2}_{\Psi^{\mu_1}(\Sigma)} : \mathsf{Mod}^{\mathcal{I}}(\Sigma) \to \mathsf{Mod}^{\mathcal{I}_2}(\Psi^{\mu_2}(\Psi^{\mu_1}(\Sigma)))$.

Example 2.4.27. *Morphism* fe : $\mathcal{FOL} \to \mathcal{EL}$. *Definition 2.4.26 leads us to the observation that the morphism* fe $= (\Psi^{fe}, \gamma^{fe}, \delta^{fe}) : \mathcal{FOL} \to \mathcal{EL}$, *can be conceived as a composition of morphisms* fa $= (\Psi^{fa}, \gamma^{fa}, \delta^{fa}) : \mathcal{FOL} \to \mathcal{ALC}$ *and* ae $= (\Psi^{ae}, \gamma^{ae}, \delta^{ae}) : \mathcal{ALC} \to \mathcal{EL}$, *or as a composition of morphisms* $fe^+ = (\Psi^{fe^+}, \gamma^{fe^+}, \delta^{fe^+}) : \mathcal{FOL} \to \mathcal{EL}^+$ *and* $e^+e = (\Psi^{e^+e}, \gamma^{e^+e}, \delta^{e^+e}) : \mathcal{EL}^+ \to \mathcal{EL}$. *This gives us the commutative diagram shown in Figure 2.4.*

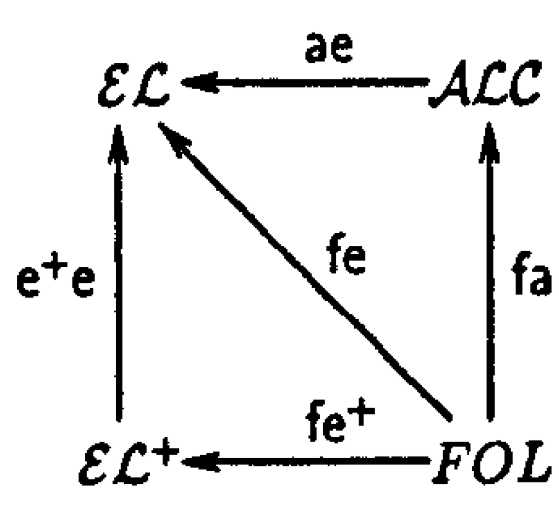


Figure 2.4: A commuting diagram of morphisms

To see that $fe^+; e^+e = fa; ae$ first recall that the signatures of $\mathcal{EL}$, $\mathcal{EL}^+$ and $\mathcal{ALC}$ are the same; this also applies to the models. For that reason $\Psi^{ae}, \Psi^{e^+e}, \delta^{ae}$ and δ^{e^+e} are identities. As noticed above $\Psi^{fe^+} = \Psi^{fa}$ and thus $\Psi^{fe^+}; \Psi^{e^+e} = \Psi^{fa}; \Psi^{ae}$, similarly $\delta^{fe^+} = \delta^{fa}$ and thus $\delta^{fe^+}; \delta^{e^+e} = \delta^{fa}; \delta^{ae}$. Therefore we only have to show that $\gamma^{fe^+}; \gamma^{e^+e} = \gamma^{fa}; \gamma^{ae}$. As presented in Example 2.4.25 γ^{fe^+} translates $\mathcal{EL}^+$-sentences over $\Psi^{fe^+}(\Pi)$ into $\mathcal{FOL}$-sentences, where Π is a $\mathcal{FOL}$-signature. In a sense γ^{fe^+} restricts our attention only to these $\mathcal{FOL}$-sentences that were translated from $\mathcal{EL}^+$. Similarly γ^{e^+e} further restricts our attention only to the sentences that were first constructed in $\mathcal{EL}$. Therefore $\gamma^{fe^+}; \gamma^{e^+e}$ allows us to consider only these $\mathcal{FOL}$-sentences that are translated from $\mathcal{EL}$. In Example 2.4.21 we showed how γ^{fa} translates $\mathcal{ALC}$-sentences over $\Psi^{fa}(\Pi)$ into $\mathcal{FOL}$-sentences, where Π is a $\mathcal{FOL}$-signature. Again, γ^{fa} restricts our attention only to these $\mathcal{FOL}$-sentences that were translated from $\mathcal{ALC}$. Similarly γ^{ae} further restricts our attention only to the sentences that were first constructed in $\mathcal{EL}$. Therefore $\gamma^{fa}; \gamma^{ae}$ allows us to consider only these $\mathcal{FOL}$-sentences that are translated from $\mathcal{EL}$. To sum up, we have that $\Psi^{fe^+}; \Psi^{e^+e} = \Psi^{fa}; \Psi^{ae}$ and $\delta^{fe^+}; \delta^{e^+e} = \delta^{fa}; \delta^{ae}$, these were immediate consequences of previously considered morphisms, and $\gamma^{fe^+}; \gamma^{e^+e} = \gamma^{fa}; \gamma^{ae}$, which is a straight forward consequence of morphisms presented above. Thus we have that $fe^+; e^+e = fa; ae$.

We let $\mathbb{I}_{mor}$ denote the category of institutions with institution morphisms.

A "dual" notion for institution morphism is **institution comorphism.**

Definition 2.4.28 (Institution comorphism).
Let $\mathcal{I} = (Sig, Sen, Mod, \models)$ *and* $\mathcal{I}' = (Sig', Sen', Mod', \models')$ *be two institutions. An institution comorphism* $\mu = (\Phi^{\mu}, \alpha^{\mu}, \beta^{\mu}) : \mathcal{I} \to \mathcal{I}'$ *consists of:*

- *a functor* $\Phi^{\mu} : Sig \to Sig'$

- *a natural transformation* $\alpha^{\mu} : Sen \Rightarrow Sen' \circ \Phi^{\mu}$
- *a natural transformation* $\beta^{\mu} : Mod' \circ (\Phi^{\mu})^{op} \Rightarrow Mod$

such that the following satisfaction condition holds:
for all $\Sigma \in |Sig|$, *for all* $\mathcal{M}' \in |Mod'(\Phi^{\mu}(\Sigma))|$, *for all* $\varphi \in Sen(\Sigma)$

$$\mathcal{M}' \models^{\mathcal{I}'}_{\Phi^{\mu}(\Sigma)} \alpha^{\mu}_{\Sigma}(\varphi) \text{ iff } \beta^{\mu}_{\Sigma}(\mathcal{M}') \models^{\mathcal{I}}_{\Sigma} \varphi .$$

Intuitively, an institution comorphism shows how a 'poorer' institution $\mathcal{I}$ is embedded in a 'richer' institution $\mathcal{I}'$. As for the case of institution morphism the functor Φ translates the signatures of $\mathcal{I}$ into the signatures of $\mathcal{I}'$. But the natural transformation α translates sentences of $\mathcal{I}$ over Σ into sentences of $\mathcal{I}'$ over $\Phi(\Sigma)$. And the natural transformation β translates $\Phi(\Sigma)$-models of $\mathcal{I}'$ into Σ-models of $\mathcal{I}$.

Note that the functor Φ and the natural transformation α go in the same direction, whereas the natural transformation β goes in the opposite direction.

Comorphisms are transformations of special interest to us, since we will use them in our approach to the Σ-entailment and the query answering problems, presented in section 3.2.

The examples below show comorphisms between logical systems that are of interest to us.

Example 2.4.29. *Comorphism* af : $\mathcal{ALC} \rightarrow FOL$.
In this comorphism functor Φ^{af} *translates* $\mathcal{ALC}$*-signature into* FOL*-signature in the following way:*

$$\begin{aligned} \Phi^{af}(P,R) = (\emptyset,\ & \Pi), \text{ where} \\ & \Pi_0 = \emptyset \\ & \Pi_1 = P \\ & \Pi_2 = R \\ & \Pi_n = \emptyset \text{ for } n > 2 \end{aligned}$$

Natural transformation α^{af} *translates* $\mathcal{ALC}$*-sentences into* FOL*-sentences with countable set of variables in the following way:*

$$\alpha^{af}(C \sqsubseteq D) = (\forall x_0)[C]^{x_0} \Rightarrow [D]^{x_0}$$

where we fix some enumeration of the variables and x_0 *is the first one. Concept translation is defined inductively:*

C :

$$\begin{aligned} [p]^x &= p(x), \\ [\neg C]^x &= \sim [C]^x, \\ [C \sqcap D]^x &= [C]^x \wedge [D]^x, \\ [C \sqcup D]^x &= [C]^x \vee [D]^x, \\ [\exists r.C]^x &= (\exists x')\, r(x,x') \wedge [C]^{x'}, \\ [\forall r.C]^x &= (\forall x')\, r(x,x') \Rightarrow [C]^{x'}. \end{aligned}$$

where, for the last two clauses, x' is the variable after x in the enumeration.

For any FOL $\Phi^{af}(\Sigma)$-model $\mathcal{M}$ we define natural transformation β^{af}_{Σ} to do the following transformation: $\beta^{af}_{\Sigma}(\mathcal{M}) = \mathcal{M}$, i.e. for $\Phi^{af}(P,R) = (\emptyset, \Pi)$ we have that for $p \in P$ we have $p \in \Pi_1$, thus $p^{\mathcal{M}} \subseteq \Delta^{\mathcal{M}}$, and for $r \in R$ we have $r \in \Pi_2$, thus $r^{\mathcal{M}} \subseteq \Delta^{\mathcal{M}} \times \Delta^{\mathcal{M}}$.

Natural transformations α^{af} and β^{af} satisfy the following condition:

for all $\Sigma \in |Sig^{\mathcal{ALC}}|$, for all $\mathcal{M} \in |Mod^{FOL}(\Phi^{af}(\Sigma))|$, for all $\varphi \in Sen^{\mathcal{ALC}}(\Sigma)$

$$\mathcal{M} \models^{FOL}_{\Phi^{af}(\Sigma)} \alpha^{af}_{\Sigma}(\varphi) \text{ iff } \beta^{af}_{\Sigma}(\mathcal{M}) \models^{\mathcal{ALC}}_{\Sigma} \varphi.$$

This can be shown by induction. Here we show only the base case. But first of all we need an auxiliary lemma.

Lemma 2.4.30. *For every $\mathcal{ALC}$-signature Σ, every $\mathcal{ALC}$-concept C and every FOL-model $\mathcal{M} \in |Mod(\Phi^{af}(\Sigma)|$ we have $C^{\beta^{af}_{\Sigma}(\mathcal{M})} = [C]^{x\mathcal{M}}$.*

Proof: *The proof is by induction.*

Let:

- $C = p$

$$p^{\beta^{af}_{\Sigma}(\mathcal{M})} = \{x \in \Delta^{\beta^{af}_{\Sigma}(\mathcal{M})} | x \in p^{\beta^{af}_{\Sigma}(\mathcal{M})}\} = \{x \in \Delta^{\mathcal{M}} | x \in p^{\mathcal{M}}\} = (p(x))^{\mathcal{M}}$$

- $C = \neg p$

$$\neg p^{\beta^{af}_{\Sigma}(\mathcal{M})} = \{x \in \Delta^{\beta^{af}_{\Sigma}(\mathcal{M})} | x \notin p^{\beta^{af}_{\Sigma}(\mathcal{M})}\} = \{x \in \Delta^{\mathcal{M}} \mid x \notin p^{\mathcal{M}}\} = (\sim p(x))^{\mathcal{M}},$$

- $C = C_1 \sqcap C_2$ *is trivial,*

- $C = \exists r.C'$

$$\begin{aligned} (\exists r.C')^{\beta^{af}_{\Sigma}(\mathcal{M})} &= \{x \in \Delta^{\beta^{af}_{\Sigma}(\mathcal{M})} | (\exists x')\ x\ r^{\beta^{af}_{\Sigma}(\mathcal{M})} x' \wedge x' \in (C')^{\beta^{af}_{\Sigma}(\mathcal{M})}\} \\ &= \{x \in \Delta^{\mathcal{M}} | (\exists x')\ x\ r^{\mathcal{M}} x' \wedge x' \in ([C'])^{\mathcal{M}}\} \\ &= ((\exists x')\ x\ r\ x' \wedge [C']^{x'})^{\mathcal{M}} \\ &= [\exists r.C']^{x\mathcal{M}} \end{aligned}$$

To show that the satisfaction condition holds, let $\Sigma \in |Sig^{\mathcal{ALC}}|$, $\mathcal{M} \in |Mod^{FOL}(\Phi^{af}(\Sigma))|$ and $\varphi = C \sqsubseteq D$ be an $\mathcal{ALC}$-sentence over Σ. Assume that $\beta^{af}_{\Sigma}(\mathcal{M}) \models^{\mathcal{ALC}}_{\Sigma} \varphi$. Thus

$$\beta^{af}_{\Sigma}(\mathcal{M}) \models^{\mathcal{ALC}}_{\Sigma} C \sqsubseteq D$$

iff

$$C^{\beta^{af}_{\Sigma}(\mathcal{M})} \subseteq D^{\beta^{af}_{\Sigma}(\mathcal{M})}$$

iff

$$[C]^{x_0\mathcal{M}} \subseteq [D]^{x_0\mathcal{M}}$$

iff

$$\mathcal{M} \models^{FOL}_{\Phi^{af}(\Sigma)} (\forall x_0)[C]^{x_0} \Rightarrow [D]^{x_0}$$

iff

$$\mathcal{M} \models^{FOL}_{\Phi^{af}(\Sigma)} \alpha^{af}_{\Sigma}(C \sqsubseteq D)$$

□

Example 2.4.31. *Comorphism* ea : $\mathcal{EL} \to \mathcal{ALC}$.
For the comorphism ea : $\mathcal{EL} \to \mathcal{ALC}$ *we take* Φ^{ea} *to be the identity functor. We know that for any signature* $\Sigma = (P, R)$ *we have* $Sen^{\mathcal{EL}}(\Sigma) \subseteq Sen^{\mathcal{ALC}}(\Sigma)$, *thus* α_Σ *is an inclusion* $Sen^{\mathcal{EL}}(\Sigma) \hookrightarrow Sen^{\mathcal{ALC}}(\Sigma)$. *Natural transformation* β_Σ *is the identity.*

Example 2.4.32. *Comorphism* $\mathsf{e^+f} : \mathcal{EL}^+ \to FOL$.
In the case for comorphism $\mathsf{e^+f} : \mathcal{EL}^+ \to FOL$ *the functor* $\Phi^{\mathsf{e^+f}}$ *is the same as* Φ^{af} *in Example 2.4.29. Natural transformation* $\alpha^{\mathsf{e^+f}}$ *GCIs into FOL-sentences as* α^{af} *in Example 2.4.29.* $\alpha^{\mathsf{e^+f}}$ *translates role inclusion axioms just as* $\gamma^{\mathsf{fe^+}}$ *in Example 2.4.25. Natural transformation* $\beta^{\mathsf{e^+f}}$ *is as in Example 2.4.29.*

Example 2.4.33. *Comorphism* $\mathsf{ee^+} : \mathcal{EL} \to \mathcal{EL}^+$.
For comorphism $\mathsf{ee^+} : \mathcal{EL} \to \mathcal{EL}^+$ *we define* $\Phi^{\mathsf{ee^+}}$ *to be the identity. For any signature* $\Sigma = (P, R)$ *we have* $Sen^{\mathcal{EL}}(\Sigma) \subseteq Sen^{\mathcal{EL}^+}(\Sigma)$, *thus* $\alpha_\Sigma^{\mathsf{ee^+}}$ *is an inclusion* $Sen^{\mathcal{EL}}(\Sigma) \hookrightarrow Sen^{\mathcal{EL}^+}(\Sigma)$. *Natural transformation* $\beta_\Sigma^{\mathsf{ee^+}}$ *is the identity.*

Example 2.4.34. *Comorphism* ef : $\mathcal{EL} \to FOL$.
It should not be difficult to see that this comorphism is just a special case of the comorphism presented in Example 2.4.29, it is also a special case of comorphism presented in Example 2.4.32. In both cases the only difference is that α^{ef} *translates only sentences built with atomic concepts and those using '*$\exists r.$*' or '*$\sqcap$*'.*

Figure 2.5 is a graphical representation of the above comorphisms. Note that we have no comorphisms between $\mathcal{ALC}$ and $\mathcal{EL}^+$, this is because each of them has expressions that are not available in the other one.

In Definition 2.4.26, we presented how morphisms are composed. A similar definition can be formulated for comorphism. In this case it is even simpler as the directions of Φ and α are the same [43].

Definition 2.4.35 (Composition of comorphisms). *Let* $\mathcal{I}, \mathcal{I}_1, \mathcal{I}_2$ *be institutions such that there are comorphisms* $\mu_1 = (\Phi^{\mu_1}, \alpha^{\mu_1}, \beta^{\mu_1}) : \mathcal{I} \to \mathcal{I}_1$ *and* $\mu_2 = (\Phi^{\mu_2}, \alpha^{\mu_2}, \beta^{\mu_2}) : \mathcal{I}_1 \to \mathcal{I}_2$. *Then we can define a comorphism* $\mu = (\Phi^\mu, \alpha^\mu, \beta^\mu) : \mathcal{I} \to \mathcal{I}_2$ *to be the composition of comorphisms* $\mu_1; \mu_2$, *i.e.* μ *is a triple defined by means of composition in the following way: the functor* $\Phi^\mu = \Phi^{\mu_1}; \Phi^{\mu_2}$, *natural transformation* $\alpha^\mu = \alpha^{\mu_1}; \Phi^{\mu_1}\alpha^{\mu_2}$ *i.e.,* $\alpha^\mu_\Sigma(e) = \alpha^{\mu_2}_{\Phi^{\mu_1}(\Sigma)}(\alpha^{\mu_1}_\Sigma(e))$ *and* $\beta^\mu = \Phi^{\mu_1 op}\beta^{\mu_2}; \beta^{\mu_1}$ *i.e.,* $\beta^\mu_\Sigma(\mathcal{M}) = \beta^{\mu_1}_\Sigma(\beta^{\mu_2}_{\Phi^{\mu_1}(\Sigma)}(\mathcal{M}))$

This allows us to consider the comorphism ef as a composition of comorphisms ea and af for the first case and a composition of comorphisms $\mathsf{ee^+}$ and $\mathsf{e^+f}$ for the second case. This gives us a commutative diagram shown in a Figure 2.5.

To see that $\mathsf{ee^+}; \mathsf{e^+f} = \mathsf{ea}; \mathsf{af}$ again we will use the fact that the signatures and the models of $\mathcal{EL}$, $\mathcal{EL}^+$ and $\mathcal{ALC}$ are the same. For that reason $\Phi^{\mathsf{ea}}, \Phi^{\mathsf{ee^+}}, \beta^{\mathsf{ea}}$ and $\beta^{\mathsf{ee^+}}$ are

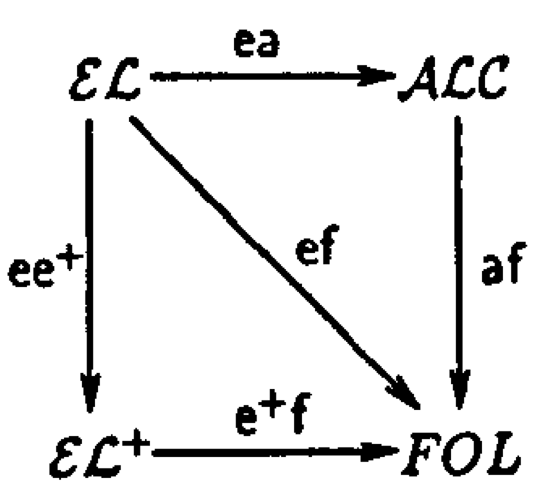


Figure 2.5: A commuting diagram of comorphisms

identities. As noticed above $\Phi^{\mathsf{e^+f}} = \Phi^{\mathsf{af}}$ and thus $\Phi^{\mathsf{ee^+}};\Phi^{\mathsf{e^+f}} = \Phi^{\mathsf{ea}};\Phi^{\mathsf{af}}$, similarly $\beta^{\mathsf{e^+f}} = \beta^{\mathsf{af}}$ and thus $\beta^{\mathsf{ee^+}};\beta^{\mathsf{e^+f}} = \beta^{\mathsf{ea}};\beta^{\mathsf{af}}$. Therefore we only have to show that $\alpha^{\mathsf{ee^+}};\alpha^{\mathsf{e^+f}} = \alpha^{\mathsf{ea}};\alpha^{\mathsf{af}}$. But this is the same as for the composition of morphisms $\mathsf{fe^+};\mathsf{e^+e}$ and $\mathsf{fa};\mathsf{ae}$ above.

A useful property of comorphisms is

Lemma 2.4.36. *For $\mu : \mathcal{I} \to \mathcal{I}'$, $E \subseteq Sen^{\mathcal{I}}(\Sigma)$ and $\varphi \in Sen^{\mathcal{I}}(\Sigma)$, if $E \models^{\mathcal{I}}_{\Sigma} \varphi$, then $\alpha^{\mu}_{\Sigma}(E) \models^{\mathcal{I}'}_{\Phi^{\mu}(\Sigma)} \alpha^{\mu}_{\Sigma}(\varphi)$; moreover, the converse implication holds if β^{μ} is surjective on models.*

Proof: The forwards implication is a straightforward consequence of the satisfaction condition for μ. For the converse implication, let M be a Σ-model such that $M \models^{\mathcal{I}}_{\Sigma} E$. If β^{μ} is surjective, there is a $\Phi^{\mu}(\Sigma)$-model M' such that $\beta^{\mu}(M') = M$, so $\beta^{\mu}(M') \models^{\mathcal{I}}_{\Sigma} E$, which gives $M' \models^{\mathcal{I}'}_{\Phi^{\mu}(\Sigma)} \alpha^{\mu}_{\Sigma}(E)$, and if $\alpha^{\mu}_{\Sigma}(E) \models^{\mathcal{I}'}_{\Phi^{\mu}(\Sigma)} \alpha^{\mu}_{\Sigma}(\varphi)$, then we have $M' \models^{\mathcal{I}'}_{\Phi^{\mu}(\Sigma)} \alpha^{\mu}_{\Sigma}(\varphi)$, so $M = \beta^{\mu}(M') \models^{\mathcal{I}}_{\Sigma} \varphi$ as desired. □

We let $\mathbb{I}_{comor}$ denote the category of institutions with institution comorphisms.

Arrais and Fiadeiro [5] note that adjunctions on signatures can be lifted to adjunctions of theories provided that the left adjoint be associated with a comorphism and the right adjoint with a morphism of institutions. Their results also show the fact that in such a case no new theorems arise when a theory is translated from one formalism to another.

We have already noticed that morphisms and comorphisms basically use the same type of transformations - the difference is in the direction of the natural transformations, as noticed when we introduced Definition 2.4.19 and Definition 2.4.28. A result presented by Arrais and Fiadeiro confirms that there is a very strong relationship between institution morphisms and comorphisms. After noticing that morphisms and comorphisms correspond to the two directions of an adjunction it becomes much easier to grasp the difference between them. We can see that morphisms take the direction of the right adjoint while comorphisms take the direction of the left adjoint. And that in turn is consistent with the view of morphisms as projections of one institution into another and comorphisms as providing representations.

Arrais and Fiadeiro showed that given an adjunction between signature categories of two institutions, an institution morphism gives us an institution comorphism and vice versa. And that it guarantees adjunctions for the functor between the corresponding categories of theories.

Theorem 2.4.37. *Let $\mathcal{I}$ and $\mathcal{I}'$ be institutions.*

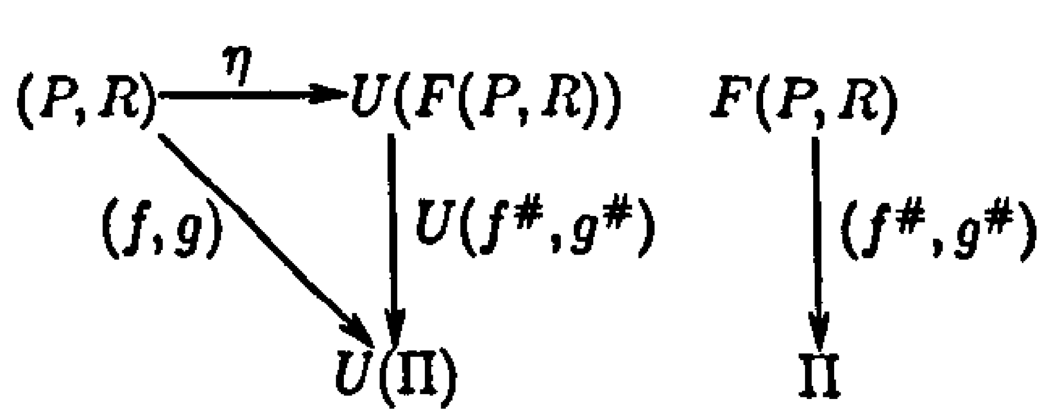
1. *If $(\Phi, \alpha, \beta) : \mathcal{I} \to \mathcal{I}'$ is a comorphism such that the functor Φ has a right adjoint Ψ, then:*
   a) *the triple (Ψ, γ, δ), where γ is the natural transformation defined by $\gamma_{\Sigma'} = \alpha_{\Psi(\Sigma')}; Sen'(\varepsilon_{\Sigma'})$, where ε is the counit of the adjunction, and δ is the natural transformation defined by $\delta_{\Sigma'} = Mod'(\varepsilon_{\Sigma'}); \beta_{\Psi(\Sigma')}$, is an institution morphism from $\mathcal{I}'$ to $\mathcal{I}$.*
   b) *the functor $F : TH_{\mathcal{I}} \to TH_{\mathcal{I}'}$ induced by the comorphism has a right adjoint - the functor $U : TH_{\mathcal{I}'} \to TH_{\mathcal{I}}$ which is induced by the morphism.*
2. *If $(\Psi, \gamma, \delta) : \mathcal{I}' \to \mathcal{I}$ is a morphism such that the functor Ψ has a left adjoint Φ, then:*
   a) *the triple (Φ, α, β), where α is the natural transformation defined by $\alpha_\Sigma = Sen(\eta_\Sigma); \gamma_{\Phi(\Sigma)}$, where η is the unit of the adjunction, and β is the natural transformation defined by $\beta_\Sigma = \delta_{\Phi(\Sigma)}; Mod(\eta_\Sigma)$, is an institution comorphism from $\mathcal{I}$ to $\mathcal{I}'$.*
   b) *the functor $U : TH_{\mathcal{I}'} \to TH_{\mathcal{I}}$ induced by the morphism has a right adjoint - the functor $F : TH_{\mathcal{I}} \to TH_{\mathcal{I}'}$ which is induced by the comorphism.*

Then Goguen and Roşu in [43] showed that this result is a natural consequence of the fact that an adjoint between signature categories lifts contravariantly to functor categories.

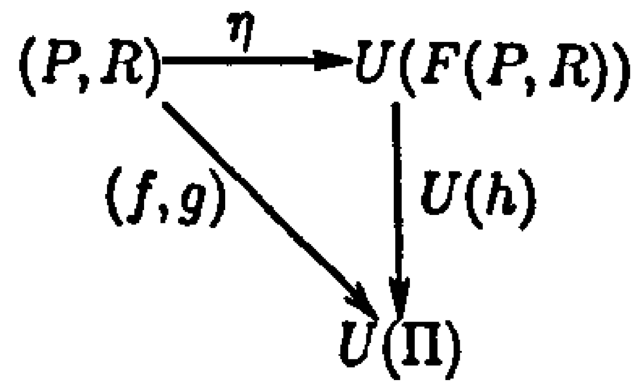
Theorem 2.4.38. *For comorphism $\mathrm{ef} = (\Phi^{\mathrm{ef}}, \alpha^{\mathrm{ef}}, \beta^{\mathrm{ef}})$ of Example 2.4.34 and morphism $\mathrm{fe} = (\Psi^{\mathrm{fe}}, \gamma^{\mathrm{fe}}, \delta^{\mathrm{fe}})$ of Example 2.4.27 we have that $\Phi^{\mathrm{ef}} \dashv \Psi^{\mathrm{fe}}$.*

Proof: Let (P, R) be an $\mathcal{EL}$-signature and Π a FOL-signature, then $\Phi^{\mathrm{ef}}(P, R) = \Pi$, where $\Pi_0 = \emptyset$, $\Pi_1 = P$, $\Pi_2 = R$ and $\Pi_n = \emptyset$ for $n \geq 3$ (which is also written as $(\emptyset, P, R, \emptyset, \ldots)$) and $\Psi^{\mathrm{fe}}(\Pi) = (\Pi^{\{x_0\}}, \Pi^{\{x_0, x_1\}})$.

First note that for any $\mathcal{EL}$-signature Σ we have that $\Sigma = \Psi^{\mathrm{fe}}(\Phi^{\mathrm{ef}}(\Sigma))$. Thus we only have to show that for any arrow $(f, g) : (P, R) \to \Psi^{\mathrm{fe}}(\Pi)$ there is a unique arrow $(f^\#, g^\#) : \Phi^{\mathrm{ef}}(P, R) \to \Pi$, such that



commutes, where $\Phi^{\mathrm{ef}}(P, R) = (\emptyset, P, R, \emptyset, \ldots)$, $(f^\#, g^\#) = (\emptyset \hookrightarrow \Pi_0, f, g, \emptyset \hookrightarrow \Pi_3, \ldots)$ and $\eta = 1_{(P,R)}$. Assume that there is another arrow $h = (h_0, h_1, h_2, h_3, \ldots) : \Phi^{\mathrm{ef}}(P, R) \to \Pi$, such that the triangle



commutes. This implies that $\Psi^{\mathsf{fe}}(h) = \Psi^{\mathsf{fe}}(f^{\#}, g^{\#})$ and thus $h_1 = f$ and $h_2 = g$. Therefore $h = (h_0, f, g, h_3, ...)$, but since $\Phi^{\mathsf{ef}}(P, R) = (\emptyset, P, R, \emptyset, ...)$, we have that $h = (\emptyset \hookrightarrow \Pi_0, f, g, \emptyset \hookrightarrow \Pi_3, ...)$. But this is exactly how $(f^{\#}, g^{\#})$ is built, so we have $h = (f^{\#}, g^{\#})$ and therefore $(f^{\#}, g^{\#})$ is unique. □

Since signatures of $\mathcal{EL}$, $\mathcal{ALC}$ and $\mathcal{EL}^+$ are identical, the following two corollaries have proofs similar to the proof of Theorem 2.4.38.

Corollary 2.4.39. *For morphism* $\mathsf{fa} = (\Psi^{\mathsf{fa}}, \gamma^{\mathsf{fa}}, \delta^{\mathsf{fa}}) : FOL \to \mathcal{ALC}$ *and comorphism* $(\Phi^{\mathsf{af}}, \alpha^{\mathsf{af}}, \beta^{\mathsf{af}}) : \mathcal{ALC} \to FOL$ *we have that* $\Phi^{\mathsf{af}} \dashv \Psi^{\mathsf{fa}}$.

Corollary 2.4.40. *For morphism* $\mathsf{fe} = (\Psi^{\mathsf{fe}}, \gamma^{\mathsf{fe}}, \delta^{\mathsf{fe}}) : FOL \to \mathcal{EL}^+$ *and comorphism* $(\Phi^{\mathsf{fe}}, \alpha^{\mathsf{fe}}, \beta^{\mathsf{fe}}) : \mathcal{EL}^+ \to FOL$ *we have that* $\Phi^{\mathsf{fe}} \dashv \Psi^{\mathsf{fe}}$.

Using theorem 2.4.37 we have the following:

1. theorem 2.4.39 together with the morphism from FOL to $\mathcal{ALC}$, presented in example 2.4.21, are sufficient to give us the comorphism from $\mathcal{ALC}$ to FOL, presented in example 2.4.29,

2. theorem 2.4.39 together with the comorphism from $\mathcal{ALC}$ to FOL, presented in example 2.4.29, are sufficient to give us the morphism from FOL to $\mathcal{ALC}$, presented in example 2.4.21,

3. theorem 2.4.38 together with the morphism from FOL to $\mathcal{EL}$, presented in example 2.4.27, are sufficient to give us the comorphism from $\mathcal{EL}$ to FOL, presented in example 2.4.34,

4. theorem 2.4.38 together with the comorphism from $\mathcal{EL}$ to FOL, presented in example 2.4.34, are sufficient to give us the morphism from FOL to $\mathcal{EL}$, presented in example 2.4.27,

5. theorem 2.4.40 together with the morphism from FOL to $\mathcal{EL}^+$, presented in example 2.4.25, are sufficient to give us the comorphism from $\mathcal{EL}^+$ to FOL, presented in example 2.4.32,

6. theorem 2.4.40 together with the comorphism from $\mathcal{EL}^+$ to FOL, presented in example 2.4.32, are sufficient to give us the morphism from FOL to $\mathcal{EL}^+$, presented in example 2.4.25,

As signatures in $\mathcal{EL}$, $\mathcal{EL}^+$ and $\mathcal{ALC}$ are identical, we can show similar relationship between the morphism and comorphism between $\mathcal{EL}$ and $\mathcal{EL}^+$ described in examples 2.4.24 and 2.4.33 and for morphism and comorphism between $\mathcal{EL}$ and $\mathcal{ALC}$ described in examples 2.4.23 and 2.4.31. These examples show that there is a one to one correspondence between morphisms and comorphisms between the above pairs of institutions.

2.5 Craig Interpolation Property and Conservative comorphisms

We have already studied institution morphisms and comorphisms. In this section first we introduce the notion of Craig interpolation property (CIP) and the notion of conservative comorphism. Then following [3] we present the problem of preservation of CIP along institution (co)morphisms.

The Craig interpolation property is a useful property of a logical system. Roughly stated, given two formulae φ and ψ such that φ entails ψ, we can always find a formula χ, called the interpolant of φ and ψ , which uses only symbols that occur in both, φ and ψ, and φ implies χ and χ entails ψ. In other words, the interpolant carries all the information needed for implying consequences of the original formula, so reasoning is not affected, in the same time it is formulated in the shared signature, so it carries relevant information only, which allows for simplifying reasoning. Craig interpolation property has a number of applications, to mention only a few of them, it is used in model checking, proofs in modular specifications, modular ontologies. Craig interpolation is also one of the basic properties of FOL. Much attention has been paid to CIP in the literature, but for us the most interesting are institution independent formulations of CIP [76] (which was one of the first institution independent formulations of CIP), other are [18,19,35,36,78]. But even these formulations impose additional requirements on the squares of signature morphisms. First of all, only squares with intersection and union of signatures are taken into consideration, moreover, in [18,19,36,78] it is required that these squares are pushouts and [35] requires them to be inclusions. In our work we follow the formulation of CIP presented in [32]. In contrast to previous formulations this one can capture any square of signature morphisms, the only requirement is that the square commutes.

Definition 2.5.1 (Craig interpolation square). *A commutative square of signature morphisms*

$$\begin{array}{ccc} \Sigma & \xrightarrow{\sigma_1} & \Sigma_1 \\ {\scriptstyle \sigma_2}\downarrow & & \downarrow{\scriptstyle \sigma_1'} \\ \Sigma_2 & \xrightarrow[\sigma_2']{} & \Sigma' \end{array}$$

*is a **Craig interpolation square** if and only if for every set E_1 of Σ_1-sentences and set E_2 of Σ_2-sentences such that $Sen(\sigma_1')(E_1) \models_{\Sigma'} Sen(\sigma'2)(E_2)$, there exists a set E of Σ-sentences such that $E_1 \models_{\Sigma_1} Sen(\sigma_1)(E)$ and $Sen(\sigma_2)(E) \models_{\Sigma_2} E_2$. The set E is called an interpolant of E_1 and E_2. If this property holds for all such sets of sentences, we say the square has CIP. Similarly, we say an institution has CIP if all commuting squares have CIP. However, this is a strong requirement that is not met in many institutions. Weaker notions have been obtained by considering only certain classes of commuting squares. For example, classical CIP (which we shall call **weak interpolation**) has $\Sigma = \Sigma_1 \cap \Sigma_2$ and $\Sigma' = \Sigma_1 \cup \Sigma_2$. More generally, [32] requires the top arrow to belong to a class L of morphisms, and the left arrow* *to belong to a class R, giving a notion of (L, R)-CIP. Examples of such classes might be: inclusions, surjective morphisms, etc., and we say an institution has (L, R)-CIP if every square has CIP, provided the top arrow of the square is in L and the left arrow in R.*

In some cases proving that a particular institution $\mathcal{I}$ has CIP may be difficult or laborious. But it may be the case that we can find another institution, which is related to $\mathcal{I}$ by a particular (co)morphism and for which it is easier to prove CIP. Then we can use this (co)morphism to prove CIP for $\mathcal{I}$. For this purpose here we present the notion of conservative comorphism which was introduced by Aiguier and Barbier in [3], where they also show that conservative comorphisms preserve CIP.

Notation 2.5.2. *Let $\mathcal{I}$ be an institution. Let $\sigma : \Sigma \to \Sigma'$ be a signature morphism and let $\mathcal{O} \subseteq Sen(\Sigma')$. Let us write $\mathcal{O}_\sigma = \{\varphi \mid \varphi \in Sen(\Sigma)$ and $\mathcal{O} \models_{\Sigma'} \sigma(\varphi)\}$.*

If we think of σ as a signature inclusion $\Sigma \hookrightarrow \Sigma'$, then Notation 2.5.2 says that for $\mathcal{O} \subseteq \mathrm{Sen}(\Sigma')$ we define $\mathcal{O}_\sigma$ to be a set of consequences of $\mathcal{O}$ restricted along σ to the sentences over Σ. Thus, defining $\mathcal{O}^*$ to be the set of consequences of $\mathcal{O}$, then $\mathcal{O}^*\!\restriction_\sigma$ is $\mathcal{O}^*$ restricted along σ to the sentences over Σ, for short $\mathcal{O}_\sigma$. Note that $\mathcal{O}_{1_\Sigma} = \mathcal{O}^*$.

Definition 2.5.3. *An institution comorphism $\mu : \mathcal{I} \to \mathcal{I}'$ is* **conservative** *iff every signature morphism $\sigma : \Sigma \to \Sigma'$ and every set of sentences $\mathcal{O} \subseteq Sen(\Sigma')$ in $\mathcal{I}$ satisfy:*

$$\alpha^\mu_\Sigma(\mathcal{O}_\sigma) \models^{\mathcal{I}'}_{\Phi^\mu(\Sigma)} (\alpha^\mu_{\Sigma'}(\mathcal{O}))_{\Phi^\mu(\sigma)} .$$

Conservativity means that moving to a richer logic introduces no new consequences (this is represented in Figure 2.6).

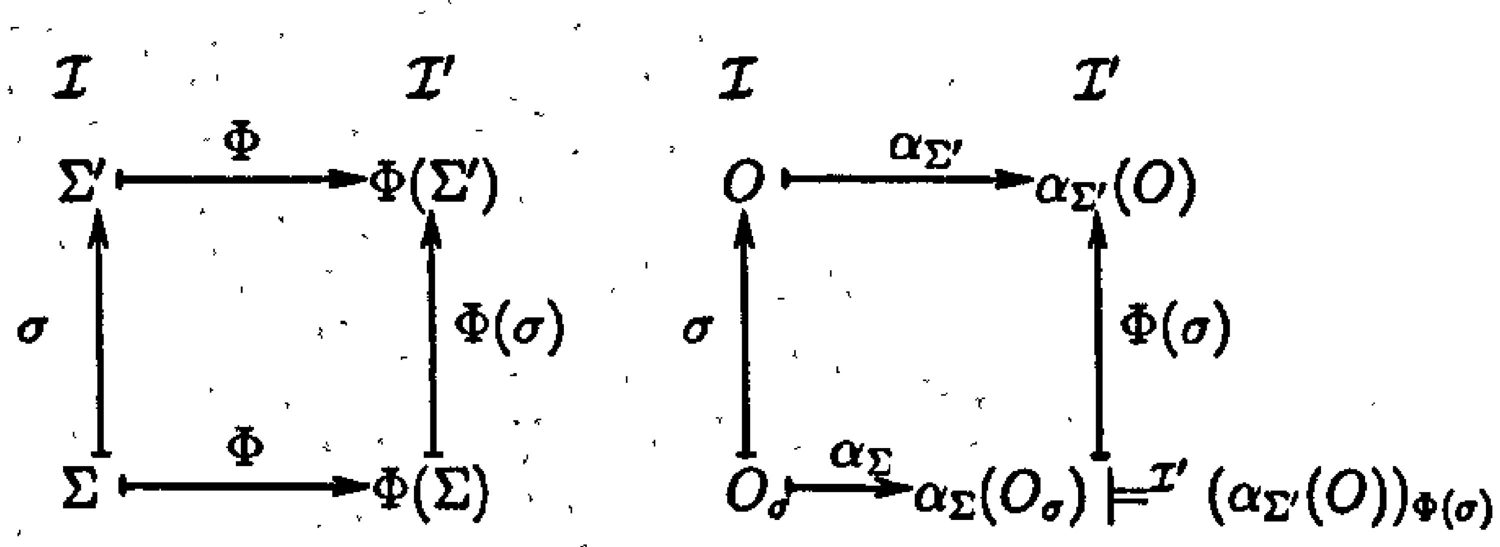


Figure 2.6: Conservative comorphism

As pointed out in [3], a sufficient condition for conservative comorphism is $(\alpha_{\Sigma'}(\mathcal{O}))_{\Phi(\sigma)} \subseteq \alpha_\Sigma(\mathcal{O}_\sigma)$, which is called **restriction adequateness**. Restriction adequateness is exactly the notion defined in [69] to obtain preservation of CIP along institution transformations, but this notion is more restrictive than the notion of conservativeness.

The following observation is a consequence of the fact that $\mathcal{O}^*\!\restriction_1 = \mathcal{O}^*$ and presents a version of Definition 2.5.3 for the case where we take the signature morphism σ to be the identity 1_Σ.

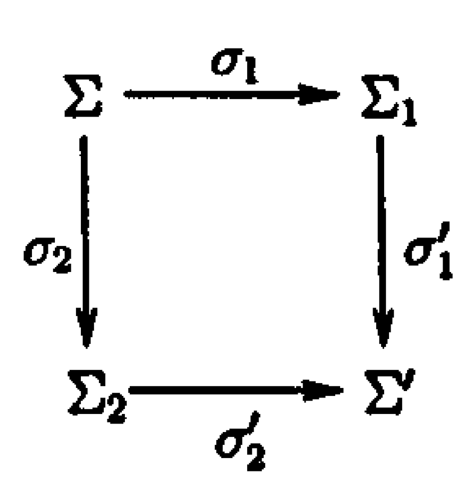
Observation 2.5.4. *For any conservative institution comorphism $\mu = (\Phi, \alpha, \beta) : \mathcal{I} \to \mathcal{I}'$, for any signature $\Sigma \in |Sig|$ if we take σ to be the identity, then we have that every set of sentences $\mathcal{O} \subseteq Sen(\Sigma)$ in $\mathcal{I}$ satisfy:*

$$\alpha_\Sigma(\mathcal{O}^*) \models^{\mathcal{I}'} \alpha_\Sigma(\mathcal{O})^*$$

A condition for preservation of CIP that was presented in [3] uses a conservative institution morphism. Nevertheless the authors suggest that a similar result can be presented using conservative institution comorphism. Since we use comorphisms rather that morphisms we present a version using conservative institution comorphism. But still the proof remains very similar to the one presented in [3].

Theorem 2.5.5. *Let $\mu = (\Phi, \alpha, \beta) : \mathcal{I}' \to \mathcal{I}$ be a conservative institution comorphism such that for every signature Σ in $\mathcal{I}'$, β_Σ is surjective. Then, if $\mathcal{I}$ has CIP then so does $\mathcal{I}'$.*

Proof: Let



be a commutative square of signature morphisms in $\mathsf{Sig}^{\mathcal{I}'}$. Let $E_1 \subseteq \mathsf{Sen}'(\Sigma_1)$ and $E_2 \subseteq \mathsf{Sen}'(\Sigma_2)$, such that $\sigma_1'(E_1) \models^{\mathcal{I}'}_{\Sigma'} \sigma_2'(E_2)$. Let $E' = E_{1_{\sigma_1}}$, by Notation 2.5.2 we have $E_1 \models^{\mathcal{I}'}_{\Sigma_1} \sigma_1(E')$, we show that $\sigma_2(E') \models^{\mathcal{I}'}_{\Sigma_2} E_2$. Let $\mathcal{M}$ be a $\Phi(\Sigma')$-model such that $\mathcal{M} \models^{\mathcal{I}}_{\Phi(\Sigma')} \alpha_{\Sigma'}(\sigma_1'(E_1))$. By the satisfaction condition, we have:

$$\begin{aligned}
\mathcal{M} \models^{\mathcal{I}}_{\Phi(\Sigma')} \alpha_{\Sigma'}(\sigma_1'(E_1)) &\Leftrightarrow \beta_{\Sigma'}(\mathcal{M}) \models^{\mathcal{I}'}_{\Sigma'} \sigma_1'(E_1) \\
&\Rightarrow \beta_{\Sigma'}(\mathcal{M}) \models^{\mathcal{I}'}_{\Sigma'} \sigma_2'(E_2) \\
&\Leftrightarrow \mathcal{M} \models^{\mathcal{I}}_{\Phi(\Sigma')} \alpha_{\Sigma'}(\sigma_2'(E_2)).
\end{aligned}$$

Thus, we have $\alpha_{\Sigma'}(\sigma_1'(E_1)) \models^{\mathcal{I}}_{\Phi(\Sigma')} \alpha_{\Sigma'}(\sigma_2'(E_2))$. Since $\mathcal{I}$ has CIP there exists $E \subseteq \mathsf{Sen}(\Phi(\Sigma))$ such that $\alpha_{\Sigma_1}(E_1) \models^{\mathcal{I}}_{\Phi(\Sigma_1)} \Phi(\sigma_1)(E)$ and $\Phi(\sigma_2)(E) \models^{\mathcal{I}}_{\Phi(\Sigma_2)} \alpha_{\Sigma_2}(E_2)$. This implies that $E \subseteq (\alpha_{\Sigma_1}(E_1))_{\Phi(\sigma_1)}$. By conservativity of μ we get that $\alpha_\Sigma(E') \models^{\mathcal{I}}_{\Sigma} E$, thus $\Phi(\sigma_2)(\alpha_\Sigma(E')) \models^{\mathcal{I}}_{\Phi(\Sigma_2)} \alpha_{\Sigma_2}(E_2)$. Let $\mathcal{M}'$ be a Σ_2-model, such that $\mathcal{M}' \models^{\mathcal{I}'}_{\Sigma_2} \sigma_2(E')$. By surjectivity of β_{Σ_2}, there is a $\Phi(\Sigma_2)$-model $\mathcal{M}$, such that $\beta_{\Sigma_2}(\mathcal{M}) = \mathcal{M}'$. By satisfaction condition we have:

$$\begin{aligned}
\mathcal{M}' \models^{\mathcal{I}'}_{\Sigma_2} \sigma_2(E') &\Leftrightarrow \mathcal{M} \models^{\mathcal{I}}_{\Phi(\Sigma_2)} \Phi(\sigma_2)(\alpha_\Sigma(E')) \\
&\Rightarrow \mathcal{M} \models^{\mathcal{I}}_{\Phi(\Sigma_2)} \alpha_{\Sigma_2}(E_2) \\
&\Leftrightarrow \mathcal{M}' \models^{\mathcal{I}'}_{\Sigma'} E_2.
\end{aligned}$$

□

Below we present examples showing that comorphisms $\mathcal{EL} \to \mathcal{ALC}$ and $\mathcal{ALC} \to \mathcal{FOL}$ are not conservative.

Example 2.5.6. *$\mathcal{EL} \to \mathcal{ALC}$. Let $\mathcal{EL}$-signature be defined in the following way $\Sigma =$ {Human, Plant, Vegetable, Healty, Area, eats, grows-in} and define the ontologies:*

$$\begin{array}{llcl} \mathcal{O}_1: & \mathit{Human} & \sqsubseteq & \exists \mathit{eats}.\top \\ & \mathit{Plant} & \sqsubseteq & \exists \mathit{grows\text{-}in}.\mathit{Area} \\ & \mathit{Vegetable} & \sqsubseteq & \mathit{Healthy} \end{array}$$

$$\begin{array}{llcl} \mathcal{O}_2: & \mathit{Human} & \sqsubseteq & \exists \mathit{eats}.\mathit{Food} \\ & \mathit{Food} \sqcap \mathit{Plant} & \sqsubseteq & \mathit{Vegetable} \\ & \mathit{Human} & \sqsubseteq & \exists \mathit{eats}.\top \\ & \mathit{Plant} & \sqsubseteq & \exists \mathit{grows\text{-}in}.\mathit{Area} \\ & \mathit{Vegetable} & \sqsubseteq & \mathit{Healthy}. \end{array}$$

Then, $\mathcal{O}_2$ is a conservative extension of $\mathcal{O}_1$ w.r.t. $\mathcal{EL}$. However, $\mathcal{O}_2$ is not a conservative extension of $\mathcal{O}_1$ w.r.t. $\mathcal{ALC}$, as witnessed by

$$\mathit{Human} \sqcap \forall \mathit{eats}.\mathit{Plant} \sqsubseteq \exists \mathit{eats}.\mathit{Vegetable}\ .$$

Example 2.5.7. *$\mathcal{ALC} \to FOL$. Let Σ and Σ' be $\mathcal{ALC}$ signatures defined: $\Sigma = (\emptyset, \{r\})$, $\Sigma' = (\{A\}, \{r\})$, such that $\Sigma \xrightarrow{\sigma} \Sigma'$. Let $T \subseteq Sen^{\mathcal{ALC}}(\Sigma')$ be of the form $T = \{\top \sqsubseteq \exists r.A \sqcap \exists r.\neg A\}$, then $T_\sigma = \{\top \sqsubseteq \exists r.\top\}$, which is translated into FOL in the following way $\alpha_\Sigma(T_\sigma) = \{\forall x\ \exists y\ r(x,y)\}$. We translate T into FOL in the following way: $\alpha_{\sigma'}(T) = \{(\forall x)((\exists y')\ r(x,y') \wedge [A]^{y'}) \wedge ((\exists y'')\ r(x,y'') \wedge [\neg A]^{y''})\}$. Then we can calculate $(\alpha_{\sigma'}(T))_{\Phi(\sigma)} = \{\forall x\ \exists y'\ \exists y''\ (y' \neq y'' \wedge r(x,y') \wedge r(x,y''))\}$. Now we can see that $Mod^{FOL}(\alpha_\Sigma(T_\sigma)) \not\subseteq Mod^{FOL}((\alpha_{\Sigma'}(T))_{\Phi(\sigma)})$, due to the fact that the former models tell us that every object has a successor in the relation r, whereas the latter models assert that every point has two different successors, both in the relation r.*

Chapter 3

Frameworks

3.1 Introduction

In Section 1.1 we have already mentioned that in the standard approach, the function of an ontology is to state, explicitly, a conceptualisation. We have also pointed out that in practice one also wants to use existing ontologies, perhaps to browse the induced concept hierarchy, or to access instance data, or perhaps to create a new ontology that extends either an entire ontology or a manageably small fragment of one. Or perhaps one may want to test whether one ontology is in some way consistent with another, or provides the same information regarding some subset of concepts.

We have also mentioned that the increasing number of ontologies available leads to the situation where there are several ontologies describing one domain of interest within one field. Often they complement each other by focusing on different aspects of that domain. This together with the fact that we can observe an increasing interest in multiple use of ontologies may raise some problems. We have identified three main issues we want to solve, these are:

1. ontologies may be formulated in distinct formalisms,
2. different formalisms may be used for ontology and query formulation,
3. different signatures may be used for ontologies or an ontology and a query.

As the core of our solution to these problems we adopt a **functional approach** to ontologies. In this chapter we provide a structure, a **framework** over a **query basis**, which works as a bridge allowing us to bring together arbitrary ontologies and queries, regardless of their signatures and formalisms used for their construction. The notion of framework captures the situation of a 'global' language into which both an 'ontology' language and a 'query' language can be translated, in a more general and abstract way. Within a framework, it is possible to capture a general notion of consequence, whereby an ontology answers a query, when both are translated into the global language. This in turn gives rise to an institution-independent notion of **entailment** of ontologies with respect to some signature. To formulate the notion of frameworks we use comorphisms, which tell us how a 'poorer' institution is embedded in a 'richer' one. We use the fact that the relationships between many ontology languages are well understood, and translating between them, or embedding them into richer languages, is often straightforward. Of course, the details of how one particular language is translated into another are necessarily *ad hoc*.

We also discuss three types of robustness of frameworks and present relations between robustness properties of frameworks and Craig interpolation properties. We also investigate the inheritance of robustness and interpolation properties between frameworks.

3.2 Frameworks

In this part we focus on the fundamental notions of description logics; Σ-entailment, Σ-inseparability w.r.t. a query language and Σ-conservative extension. This section generalizes the results presented by B. Konev, C. Lutz, D. Walther, and F. Wolter in [50]. Our goal is to present these notions in an institutional setting, and thus independently of particular ontology or query languages. To achieve this we introduce the notion of a framework, which allows us to study entailment even if an ontology language differs from a query language, or even if these languages are incomparable, in the sense that there is no (co)morphism between them.

Notation 3.2.1. *In what follows we introduce three types of institutions: $\mathcal{L}$ which is called the* ***ontology institution****, $\mathcal{Q}$ which is called the* ***query institution****, and $\mathcal{G}$ which is called the* ***global institution****.*

For the sake of simplicity we introduce a convention regulating how we name components of particular comorphisms.

Convention 3.2.2. *For institutions and $\mathcal{I}$ and $\mathcal{I}'$ with a comorphism $\mu : \mathcal{I} \to \mathcal{I}'$ we define $\mu = (\Phi^\mu, \alpha^\mu, \beta^\mu)$.*

Definition 3.2.3. *A* ***query basis*** *is an inclusive comorphism $\eta : \mathcal{Q} \to \mathcal{G}$. A* ***framework over a query basis*** *$\eta : \mathcal{Q} \to \mathcal{G}$ is an inclusive comorphism $\mu : \mathcal{L} \to \mathcal{G}$. We call $\mathcal{L}$ the* ***ontology language****, $\mathcal{G}$ the* ***global language****, and $\mathcal{Q}$ the* ***query language*** *of the framework.*

The intuition behind this construct is that given an ontology and a query represented in two institutions, $\mathcal{L}$ and $\mathcal{Q}$ respectively, we chose a global institution $\mathcal{G}$ such that there are comorphisms $\mathcal{L} \xrightarrow{\mu} \mathcal{G}$ and $\mathcal{Q} \xrightarrow{\eta} \mathcal{G}$. Using these comorphisms we can translate the ontology and the query into $\mathcal{G}$, then we can check whether the query is a consequence of the ontology. The use of comorphisms not only allows us to bring the ontology and the query together, but also addresses the problem of potential differences in vocabularies used for ontology and query formulation. This is possible as the comorphisms map these vocabularies into a common vocabulary in $\mathcal{G}$. This mapping may involve renaming in case synonyms (or homonyms) are present in the vocabularies of the ontology and the query or if they are formulated in different natural languages.

We allow more than one framework over a query basis. Figure 3.1 is a graphical representation of frameworks μ_1 and μ_2 over query basis η. This situation would arise, for

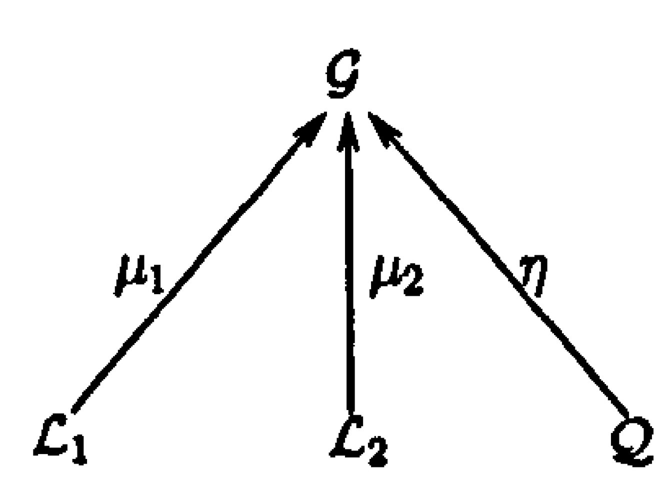


Figure 3.1: Frameworks μ_1 and μ_2 over a query basis η

example, if we wanted to merge two ontologies written in different languages.

It is easy to see that one can introduce a similar definition using morphisms (we do that in the Appendix). In such a case the direction of arrows will change and we will have morphisms $\mu_1 : \mathcal{G} \to \mathcal{L}_1$, $\mu_2 : \mathcal{G} \to \mathcal{L}_2$ and $\eta : \mathcal{G} \to \mathcal{Q}$.

One of the reasons why we use comorphisms rather than morphisms in our construct is the origin of the signature; for comorphism $\mu : \mathcal{I} \to \mathcal{I}'$ signatures originate in $\mathcal{I}$, whereas for morphism $\nu : \mathcal{I}' \to \mathcal{I}$ signatures originate in $\mathcal{I}'$. So in comorphism framework case we start with separate signatures for each ontology and query language, whereas for morphism framework we have one signature, which is translated to ontology and query languages. So use of comorphisms in a very natural way reflects the intuition that ontologies and queries originate from possibly different languages. On the other hand, this fact also plays its role when we are investigating cases with signature inclusions. For instance, we may want to make explicit that we add fresh symbols to the query signature only (or to the ontology signature only). Or perhaps that we are given two ontologies formulated over two different signatures. Again, use of comorphisms allows us to express that in a very natural way. Also the alignment of functor Φ and natural transformation α for comorphisms fits well with the intuition that ontologies and queries may originate from different languages and are brought together, in order to determine if the ontology entails the query. Whereas, for case with morphisms the signature is given in the global language and then translated to an ontology (a query) language, where we formulate our ontology (resp. query) and then translate it back to the global language. For all these reasons we find comorphisms more intuitive and appealing to use in our construct. Nevertheless, using morphisms we are still able to show all the properties of frameworks that are presented below.

We start with defining the notion of an ontology in a framework.

Definition 3.2.4. *Given a framework $\mu : \mathcal{L} \to \mathcal{G}$ and $\Lambda \in |Sig^{\mathcal{L}}|$ we say that $\mathcal{O} \subseteq Sen^{\mathcal{L}}(\Lambda)$ is a Λ-ontology for μ.*

The consequence relation is a basic notion in our studies, it is used to solve query answering problems and is the base for other definitions. Therefore before continuing we define this notion in the framework setting.

Definition 3.2.5. *Let $\mu : \mathcal{L} \to \mathcal{G}$ be a framework over η, and let $\mathcal{O}$ be a Λ-ontology for μ and $\varphi \in Sen^{\mathcal{Q}}(\Sigma)$ be a query, with Σ in $Sig^{\mathcal{Q}}$. We say that φ is a* ***consequence of $\mathcal{O}$ with respect to η*** *(written $\mathcal{O} \models^{\eta}_{\Sigma} \varphi$) iff*

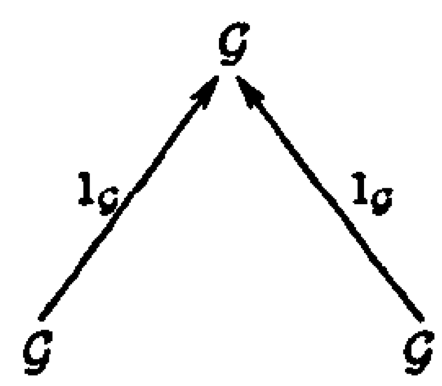
$$\alpha^{\mu}_{\Lambda}(\mathcal{O}) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda) \cup \Phi^{\eta}(\Sigma)} \alpha^{\eta}_{\Sigma}(\varphi).$$

This says that in a framework μ, over a query basis η, a query φ is a consequence of an ontology $\mathcal{O}$ iff $\mathcal{O}$ translated along comorphism μ into $\mathcal{G}$ entails in $\mathcal{G}$, with respect to the union of translations of signatures Λ and Σ, the translation of φ along η. A very similar notion was introduced by Schorlemmer and Kalfoglou in [71] as the notion of **ontology-based consequence** with the difference that they fix the global institution to be *FOL*.

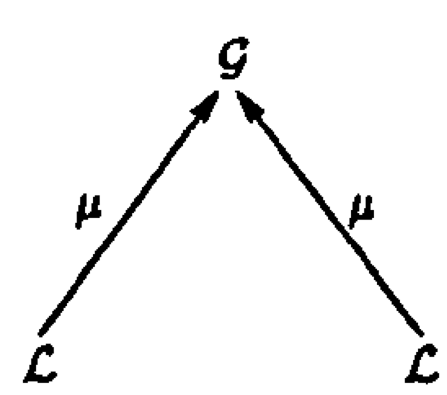
3.2.1 Basic framework structures

Here we present six special cases of frameworks (listed below), where entailment can be simplified. Each special case has its graphical representation. These frameworks differ in relations between institutions used in each construct.

1. Let $\mathcal{L} = \mathcal{G} = \mathcal{Q}$, in this case the comorphisms are identities. Proposition 3.2.11 below shows that the entailment in framework $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$ over the same query basis is the same as entailment in $\mathcal{G}$.



2. Let $\mathcal{L} = \mathcal{Q}$, i.e. an ontology $\mathcal{O}$ and a query φ are expressed in the same language. Proposition 3.2.12 states that for this framework there may be entailments that arise from the greater power of $\mathcal{G}$, and that the ontology $\mathcal{O}$ in this framework entails exactly the same consequences as in $\mathcal{L}$ only if β^{μ} is surjective.



Example 3.2.6. *As an example take $\mathcal{L} = \mathcal{Q} = \mathcal{EL}$ and $\mathcal{G} = FOL$, with comorphism $\mu =$ ef from Example 2.4.34. Let $\Lambda = (P, R)$ with $P = \{\mathsf{Toe}, \mathsf{Foot}, \mathsf{Leg}\}$, $R = \{\mathsf{isPartOf}\}$ be an $\mathcal{EL}$-signature, $\mathcal{O} \subseteq \mathsf{Sen}^{\mathcal{EL}}(\Lambda)$ be an $\mathcal{EL}$-ontology consisting of the following axioms:*

$$\begin{aligned} \mathsf{Toe} &\sqsubseteq \exists \mathsf{isPartOf.Foot} \\ \mathsf{Foot} &\sqsubseteq \exists \mathsf{isPartOf.Leg} \end{aligned}$$

Let $\varphi \in \mathsf{Sen}^{\mathcal{EL}}(\Lambda)$ be the $\mathcal{EL}$-query $\mathsf{Toe} \sqsubseteq \exists \mathsf{isPartOf.Leg}$. We begin by translating both into FOL, i.e. along μ. As a result we receive the FOL-ontology $\alpha^{\mu}_{\Lambda}(\mathcal{O})$ with axioms:

$$\begin{aligned} (\forall x)\mathsf{Toe}(x) &\Rightarrow (\exists y)\ \mathsf{isPartOf}(x,y) \wedge \mathsf{Foot}(y) \\ (\forall x)\mathsf{Foot}(x) &\Rightarrow (\exists y)\ \mathsf{isPartOf}(x,y) \wedge \mathsf{Leg}(y) \end{aligned}$$

whereas $\alpha^{\mu}_{\Lambda}(\varphi) = (\forall x)\mathsf{Toe}(x) \Rightarrow (\exists y)\ \mathsf{isPartOf}(x,y) \wedge \mathsf{Leg}(y)$. Then we have that

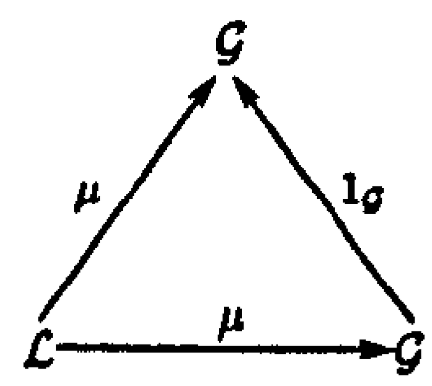
$$\alpha^{\mu}_{\Lambda}(\mathcal{O}) \models (\forall x)\mathsf{Toe}(x) \Rightarrow (\exists y)\ \mathsf{isPartOf}(x,y) \wedge (\exists z)\ \mathsf{isPartOf}(y,z) \wedge \mathsf{Leg}(z)\ ,$$

but since we do not have an axiom stating that isPartOf is transitive therefore

$$\alpha^{\mu}_{\Lambda}(\mathcal{O}) \not\models (\forall x)\mathsf{Toe}(x) \Rightarrow (\exists z)\ \mathsf{isPartOf}(x,z) \wedge \mathsf{Leg}(z)\ .$$

And this is as expected, since axioms of the form $r \circ r = r$ (or in general $r \circ s = t$) are disallowed in $\mathcal{EL}$.

3. For the case where $\mathcal{G} = \mathcal{Q}$ with a framework $\mu : \mathcal{L} \to \mathcal{G}$ over a query base $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$, i.e. $\mathcal{L}$ is a sublanguage of $\mathcal{G}$, we translate an $\mathcal{L}$-ontology into a richer language $\mathcal{G}$, and then, in $\mathcal{G}$, we check whether a query is a consequence of the ontology. Proposition 3.2.14, states that entailment in this framework is the same as entailment in $\mathcal{G}$.



Example 3.2.7. *Here we consider the case with $\mathcal{EL}$ as $\mathcal{L}$ and $\mathcal{ALC}$ as $\mathcal{G}$, with comorphism* ea *from Example 2.4.31. Let*

$$\Lambda = \{\mathit{Parent}, \mathit{Father}, \mathit{Male}, \mathit{Mother}, \mathit{Female}, \mathit{has\_Child}\}$$

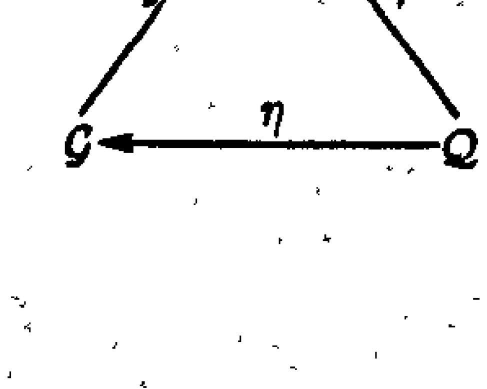
be an $\mathcal{EL}$-signature (note that it is also an $\mathcal{ALC}$-signature), $\mathcal{O} \subseteq \mathit{Sen}^{\mathcal{EL}}(\Lambda)$ be an $\mathcal{EL}$-ontology consisting of the following axioms:

$$\begin{aligned} \mathit{Parent} &\equiv \exists \mathit{has\_Child}.\top, \\ \mathit{Father} &\sqsubseteq \mathit{Male} \sqcap \exists \mathit{has\_Child}.\top, \\ \mathit{Mother} &\sqsubseteq \mathit{Female} \sqcap \exists \mathit{has\_Child}.\top, \end{aligned}$$

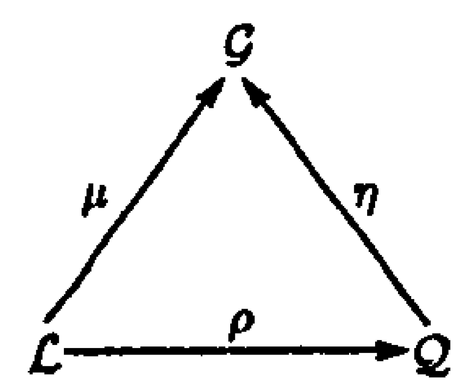
Let $\mathcal{ALC}$-query φ be of the form $\mathit{Father} \sqcup \mathit{Mother} \sqsubseteq \mathit{Parent}$. After transforming $\mathcal{O}$ into $\mathcal{ALC}$ (in this case it is simply an inclusion) we can answer the query. In this particular case we have $\mathcal{O} \models^{\mathcal{ALC}}_{\Sigma} \varphi$.

Note that in cases 3 and 2 there may be entailments that arise from the greater power of $\mathcal{G}$.

4. For the case where $\mathcal{G} = \mathcal{L}$ and a framework $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$ over a query base $\eta : \mathcal{Q} \to \mathcal{G}$, i.e. $\mathcal{Q}$ is a sublanguage of $\mathcal{G}$, we translate a $\mathcal{Q}$-query into a richer language $\mathcal{G}$, and then, in $\mathcal{G}$, we check whether the query is a consequence of the ontology. Proposition 3.2.15 shows that entailment in this framework is the same as entailment in $\mathcal{G}$.



5. Consider a scenario with distinct institutions $\mathcal{L}$, $\mathcal{G}$ and $\mathcal{Q}$, together with a framework $\mu : \mathcal{L} \to \mathcal{G}$ over a query basis $\eta : \mathcal{Q} \to \mathcal{G}$ and a comorphism $\rho : \mathcal{L} \to \mathcal{Q}$. We can consider $\mu : \mathcal{L} \to \mathcal{G}$ as a composition $\mu = \rho;\eta$. So, in fact we are translating an $\mathcal{L}$-ontology into $\mathcal{G}$ via $\mathcal{Q}$. Properties of this framework are presented in Corollary 3.2.27 and Proposition 3.2.28.



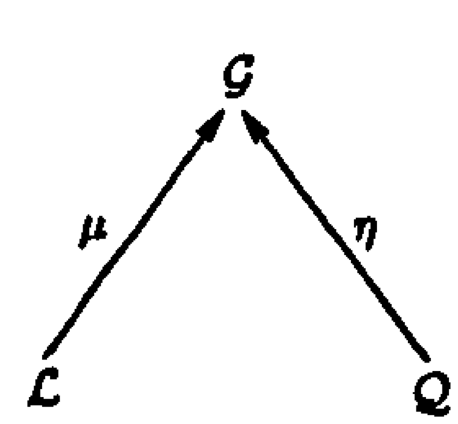
Example 3.2.8. *In this example, again, we take $\mathcal{L}$ and $\mathcal{Q}$ to be $\mathcal{EL}$ and $\mathcal{ALC}$ respectively, and we chose FOL to be $\mathcal{G}$. Take Λ, $\mathcal{O}$ and φ as in Example 3.2.7. We already know that we have comorphism* af : $\mathcal{ALC} \to FOL$, *recall Example 2.4.29, we also know that there is comorphism* ea : $\mathcal{EL} \to \mathcal{ALC}$, *recall Example 2.4.31. By composition of comorphisms we receive comorphism* ea ; af. *As it was mentioned in Example 2.4.34 this comorphism is identical to comorphism* ef : $\mathcal{EL} \to FOL$, *i.e. we can translate the $\mathcal{EL}$-ontology $\mathcal{O}$ into FOL via $\mathcal{ALC}$ without any harm. Thus $\alpha^{\mu}_{\Lambda}(\mathcal{O})$ consists of:*

$$\begin{aligned} (\forall x)\mathit{Parent}(x) &\Leftrightarrow (\exists y)\ \mathit{has\_Child}(x,y), \\ (\forall x)\mathit{Father}(x) &\Rightarrow \mathit{Male}(x) \wedge (\exists y)\ \mathit{has\_Child}(x,y), \\ (\forall x)\mathit{Mother}(x) &\Rightarrow \mathit{Female}(x) \wedge (\exists y)\mathit{has\_Child}(x,y), \end{aligned}$$

and $\alpha^{\mathrm{af}}_{\Sigma}(\varphi) = (\forall x)\mathit{Father}(x) \vee \mathit{Mother}(x) \Rightarrow \mathit{Parent}(x)$. It is not difficult to see that $\mathcal{O} \models^{\rho}_{\Sigma} \varphi$.

Remark 3.2.9. *Note that even though structures in Examples 3.2.7 and 3.2.8 differ, both give exactly the same answers to the queries. This only shows that for a particular pair of $\mathcal{L}$ and $\mathcal{Q}$ there can be more than one framework.*

6. Let $\mu : \mathcal{L} \to \mathcal{G}$ be a framework over a query basis $\eta : \mathcal{Q} \to \mathcal{G}$ with incomparable $\mathcal{L}$ and $\mathcal{Q}$. In this case we have to translate an ontology $\mathcal{O}$ and a query φ into a language in which we can check whether φ is a consequence of $\mathcal{O}$, i.e. we have to find a $\mathcal{G}$ such that there are comorphisms from $\mathcal{L}$ and $\mathcal{Q}$ to $\mathcal{G}$.



For instance, consider a situation with $\mathcal{EL}^{+}$ as $\mathcal{L}$ and $\mathcal{ALC}$ as $\mathcal{Q}$. To check whether a query is a consequence of an ontology we translate both into FOL, i.e. FOL is $\mathcal{G}$.

Example 3.2.10. *Let* $e^+f : \mathcal{EL}^+ \to FOL$ *be a framework over query basis* $af : \mathcal{ALC} \to FOL$. *Let* $\Lambda = (P, R)$ *with*

$$P = \{\mathit{Grandparent}, \mathit{Grandfather}, \mathit{Grandmother}, \mathit{Male}, \mathit{Female}\},$$

and

$$R = \{\mathit{has\_Grandchild}, \mathit{has\_Child}\}$$

be an $\mathcal{EL}^+$*-signature,* $\mathcal{O} \subseteq Sen^{\mathcal{EL}^+}(\Lambda)$ *be an* $\mathcal{EL}^+$ *ontology consisting the following axioms:*

$$\begin{aligned} \mathit{Grandparent} &\equiv \exists \mathit{has\_Grandchild}.\top, \\ \mathit{Grandfather} &\sqsubseteq \mathit{Male} \sqcap \exists \mathit{has\_Child}.\exists \mathit{has\_Child}.\top, \\ \mathit{Grandmother} &\sqsubseteq \mathit{Female} \sqcap \exists \mathit{has\_Child}.\exists \mathit{has\_Child}.\top, \\ \mathit{has\_Grandchild} &\equiv \mathit{has\_Child} \circ \mathit{has\_Child}. \end{aligned}$$

and let $\Sigma = \{\mathit{Grandparent}, \mathit{Grandfather}, \mathit{Grandmother}\}$ *be an* $\mathcal{ALC}$*-signature, and* $\varphi \in Sen^{\mathcal{ALC}}(\Sigma)$ *be an* $\mathcal{ALC}$*-query of the form* $\varphi = \mathit{Grandfather} \sqcup \mathit{Grandmother} \sqsubseteq \mathit{Grandparent}$.

To answer whether $\mathcal{O} \models^{af}_{\Sigma} \varphi$ *first we have translate both* $\mathcal{O}$ *and* φ *into FOL and we receive:*

$\alpha^{e^+f}_{\Lambda}(\mathcal{O})$:

$$\begin{aligned} (\forall x)\mathit{Grandparent}(x) &\Leftrightarrow (\exists y)\mathit{has\_Grandchild}(x,y), \\ (\forall x)\mathit{Grandfather}(x) &\Rightarrow \mathit{Male}(x) \wedge (\exists y,z)\mathit{has\_Child}(x,y) \wedge \mathit{has\_Child}(y,z) \\ (\forall x)\mathit{Grandmother}(x) &\Rightarrow \mathit{Female}(x) \wedge (\exists y,z)\mathit{has\_Child}(x,y) \wedge \mathit{has\_Child}(y,z) \\ (\forall x,y,z)\mathit{has\_Grandchild}(x,y) &\Leftrightarrow \mathit{has\_Child}(x,z) \wedge \mathit{has\_Child}(z,y). \end{aligned}$$

and $\alpha^{af}_{\Sigma}(\varphi) = (\forall x)\mathit{Grandfather}(x) \vee \mathit{Grandmother}(x) \Rightarrow \mathit{Grandparent}(x)$. *Now we can see that* $\alpha^{e^+f}_{\Lambda}(\mathcal{O}) \models^{FOL}_{\Phi^{e^+f}(\Lambda)\cup\Phi^{af}(\Sigma)} \alpha^{af}_{\Sigma}(\varphi)$.

For any institution $\mathcal{G}$, the identity $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$ is a query basis, and also provides a framework over itself, which is just the institution $\mathcal{G}$, and so the notion of consequence in a framework generalizes entailment in a fixed institution.

Proposition 3.2.11. *For framework* $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$ *over query basis* $1_{\mathcal{G}}$ *itself, consequence in the framework is just consequence in* $\mathcal{G}$*; i.e.,* $\mathcal{O} \models^{1_{\mathcal{G}}}_{\Sigma} \varphi$ *iff* $\mathcal{O} \models^{\mathcal{G}}_{\Lambda\cup\Sigma} \varphi$ *for any* Λ*-ontology* $\mathcal{O}$ *and* Σ*-sentence* φ.

Proof: This follows directly from Definition 3.2.5, noting that $\Phi^{1_{\mathcal{G}}}$ and $\alpha^{1_{\mathcal{G}}}$ are identities. □

Now we show that in certain cases there is a very close relation between entailment in an institution and consequence relation w.r.t. a query basis.

First we show that for framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis μ itself there is a very close relation between consequence relation in $\mathcal{L}$ and consequence relation w.r.t. μ.

Proposition 3.2.12. *For any framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis μ itself, for any Λ-ontology $\mathcal{O}$ for μ, and any $\varphi \in Sen^{\mathcal{L}}(\Lambda')$ we have:*

$$\mathcal{O} \models^{\mathcal{L}}_{\Lambda\cup\Lambda'} \varphi \quad \textit{implies} \quad \mathcal{O} \models^{\mu}_{\Lambda'} \varphi \ .$$

Moreover, if β^{μ} is surjective on models, then the converse implication also holds.

Proof: The implication is a straightforward application of the satisfaction condition.

For the converse implication, assume $\mathcal{O} \models^{\mu}_{\Lambda'} \varphi$ and let M be a $(\Lambda\cup\Lambda')$-model in $\mathcal{L}$ such that $M \models^{\mathcal{L}}_{\Lambda\cup\Lambda'} \mathcal{O}$. By surjectivity of β^{μ} there is a $\Phi^{\mu}(\Lambda \cup \Lambda')$-model M' in $\mathcal{G}$ such that $\beta^{\mu}_{\Lambda\cup\Lambda'}(M') = M$. By assumption, $M = \beta^{\mu}_{\Lambda\cup\Lambda'}(M') \models^{\mathcal{L}}_{\Lambda\cup\Lambda'} \mathcal{O}$, so $M' \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda\cup\Lambda')} \alpha^{\mu}_{\Lambda\cup\Lambda'}(\mathcal{O})$, and therefore $M' \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda\cup\Lambda')} \alpha^{\mu}_{\Lambda\cup\Lambda'}(\varphi)$, whence $M = \beta^{\mu}_{\Lambda\cup\Lambda'}(M') \models^{\mathcal{L}}_{\Lambda\cup\Lambda'} \varphi$ as desired. □

Because $\mathcal{EL}$ and FOL-models are essentially the same (see Section 2.4), we have

Corollary 3.2.13. *As comorphism $\mathsf{ef} : \mathcal{EL} \to FOL$ has surjective β^{ef}, we have that for all $\mathcal{EL}$-signatures Λ, Λ' and any Λ-ontology $\mathcal{O}$ for ef, and $\varphi \in Sen^{\mathcal{EL}}(\Lambda')$:*

$$\mathcal{O} \models^{\mathsf{ef}}_{\Lambda'} \varphi \quad \textit{iff} \quad \mathcal{O} \models^{\mathcal{EL}}_{\Lambda\cup\Lambda'} \varphi \ .$$

In a similar way there is a close relation between entailment in the global institution and consequence relation w.r.t. query basis $1_{\mathcal{G}}$.

Proposition 3.2.14. *For any framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$, for any Λ-ontology $\mathcal{O}$ for $1_{\mathcal{G}}$, and any $\varphi \in Sen^{\mathcal{G}}(\Sigma)$ we have:*

$$\mathcal{O} \models^{1_{\mathcal{G}}}_{\Sigma} \varphi \quad \textit{iff} \quad \alpha^{\mu}_{\Lambda}(\mathcal{O}) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)\cup\Sigma} \varphi \ .$$

Proof: This follows directly from Definition 3.2.5. □

Also for framework $1_{\mathcal{G}}$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$ there is a close relation between entailment in $\mathcal{G}$ and consequence relation w.r.t. η.

Proposition 3.2.15. *For any framework $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$, for any Λ-ontology $\mathcal{O}$ for η, and any $\varphi \in Sen^{\mathcal{Q}}(\Sigma)$ we have:*

$$\mathcal{O} \models^{\eta}_{\Sigma} \varphi \quad \textit{iff} \quad \mathcal{O} \models^{\mathcal{G}}_{\Lambda\cup\Phi^{\eta}(\Sigma)} \alpha^{\eta}_{\Sigma}(\varphi) \ .$$

Proof: This follows directly from Definition 3.2.5, noting that $\Phi^{1_{\mathcal{G}}}$ and $\alpha^{1_{\mathcal{G}}}$ are identities. □

3.2.2 Σ-entailment in frameworks

The notion of Σ-entailment is basic to many studies of description logics in the literature, for instance see [50,51,54,60,61]. Closely related notions are **inseparability** and **conservative extension**. Here we present these notions in the framework setting.

Very often we are interested in comparing ontologies written in different languages (frameworks) over a given query basis η. That is, we have two frameworks $\mu_1 : \mathcal{L}_1 \to \mathcal{G}$ and $\mu_2 : \mathcal{L}_2 \to \mathcal{G}$; we shall refer to such a situation as a **binary framework**, with notation $\mathfrak{F} = (\mu_1, \mu_2)$.

Definition 3.2.16 (Σ-entailment and inseparability). *For a binary framework* $\mathfrak{F} = (\mu_1, \mu_2)$ *query basis* η, Λ_1*-ontology* $\mathcal{O}_1$ *for* μ_1 *and* Λ_2*-ontology* $\mathcal{O}_2$ *for* μ_2, *we say that*

- $\mathcal{O}_1$ ***Σ-entails*** $\mathcal{O}_2$ *with respect to* η, *and write* $\mathcal{O}_1 \sqsubseteq^\eta_\Sigma \mathcal{O}_2$, *iff for all* $\varphi \in Sen^{\mathcal{Q}}(\Sigma)$ *we have:*

$$\mathcal{O}_2 \models^\eta_\Sigma \varphi \quad \textit{implies} \quad \mathcal{O}_1 \models^\eta_\Sigma \varphi \, .$$

- $\mathcal{O}_1$ *and* $\mathcal{O}_2$ *are **Σ-inseparable** with respect to* η, *written* $\mathcal{O}_1 \approx^\eta_\Sigma \mathcal{O}_2$, *iff:*

$$\mathcal{O}_1 \sqsubseteq^\eta_\Sigma \mathcal{O}_2 \quad \textit{and} \quad \mathcal{O}_2 \sqsubseteq^\eta_\Sigma \mathcal{O}_1 \, .$$

- $\mathcal{O}_2$ *is a **Σ-conservative extension** of* $\mathcal{O}_1$ *with respect to* η *iff* $\alpha^{\mu_2}_{\Lambda_2}(\mathcal{O}_2) \supseteq \alpha^{\mu_1}_{\Lambda_1}(\mathcal{O}_1)$ *and* $\mathcal{O}_1$ *and* $\mathcal{O}_2$ *are Σ-inseparable with respect to* η.

- $\mathcal{O}_2$ *is a **conservative extension** of* $\mathcal{O}_1$ *with respect to* η *iff* $\mathcal{O}_2$ *is a Σ-conservative extension of* $\mathcal{O}_1$ *with respect to* η *for all* $\Sigma \in |Sig^{\mathcal{Q}}|$ *such that* $\Phi^\eta(\Sigma) \subseteq \Phi^{\mu_1}(\Lambda_1)$.

The terminology 'Σ-inseparability' is taken from [50], and says that two ontologies are **equivalent** in the sense that they entail exactly the same consequences with respect to the signature Σ. Indeed, for any query basis η and signature Σ in $\mathcal{Q}$, the relation $\approx^\eta_\Sigma$ with respect to η is an equivalence relation.

In the situation of the previous definition, we say that φ separates $\mathcal{O}_1$ and $\mathcal{O}_2$ iff $\mathcal{O}_1 \models^\eta_\Sigma \varphi$ and $\mathcal{O}_2 \not\models^\eta_\Sigma \varphi$ or, vice versa.

Notation 3.2.17. *In what follows we often leave inclusions implicit, i.e. if* $\iota : \Sigma \hookrightarrow \Sigma'$ *and* $\varphi \in Sen(\Sigma)$ *we write* $\alpha^\eta_{\Sigma'}(\varphi)$ *for* $Sen(\iota)(\alpha^\eta_\Sigma(\varphi))$.

The following lemma states that if we have a consequence relative to some signature, then extending the $\mathcal{Q}$-signature with fresh symbols has no impact on the consequence relation in the framework for the queries formulated in the original signature. In addition, it tells us that if the square in Figure 3.2 has CIP in the global institution, then the converse holds,

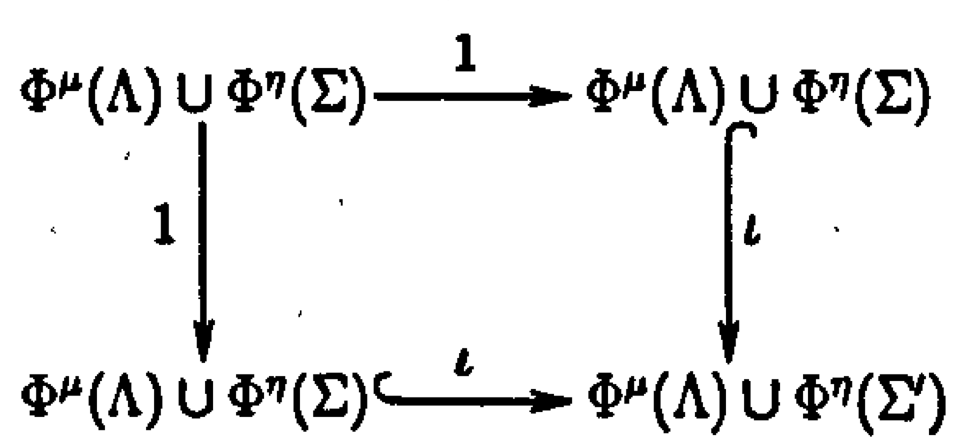


Figure 3.2.

i.e. every $\varphi \in \mathrm{Sen}^{\mathcal{Q}}(\Sigma)$ which is a consequence of O, relative to Σ' is also a consequence of O, relative to Σ.

Lemma 3.2.18. *For any framework* $\mu : \mathcal{L} \to \mathcal{G}$ *over query basis* $\eta : \mathcal{Q} \to \mathcal{G}$ *and signatures* Σ, Σ' *in* $Sig^{\mathcal{Q}}$, *such that* $\Sigma \subseteq \Sigma'$, *any* Λ*-ontology* $\mathcal{O}$ *for* μ, *and any query* $\varphi \in Sen^{\mathcal{Q}}(\Sigma)$, *the following property holds:*

$$\mathcal{O} \models^\eta_\Sigma \varphi \quad \textit{implies} \quad \mathcal{O} \models^\eta_{\Sigma'} \varphi \, .$$

Moreover, if Figure 3.2 is a CIP-square, then the converse implication also holds.

Proof: For "⇒" assume $\mathcal{O} \models^{\eta}_{\Sigma} \varphi$. Let ι be the inclusion:

$$\Phi^{\mu}(\Lambda) \cup \Phi^{\eta}(\Sigma) \subseteq \Phi^{\mu}(\Lambda) \cup \Phi^{\eta}(\Sigma')$$

and let M be a $(\Phi^{\mu}(\Lambda) \cup \Phi^{\eta}(\Sigma'))$-model such that $M \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma')} \alpha^{\mu}_{\Lambda}(\mathcal{O})$. By the satisfaction condition we have

$$M{\restriction_{\iota}} \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma)} \alpha^{\mu}_{\Lambda}(\mathcal{O}), \quad \text{and thus} \quad M{\restriction_{\iota}} \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma)} \alpha^{\eta}_{\Sigma}(\varphi)\ ,$$

so $M \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma')} \alpha^{\eta}_{\Sigma'}(\varphi)$ as desired.

For the converse implication, if the diagram above is a CIP-square, then $\mathcal{O} \models^{\eta}_{\Sigma'} \varphi$ implies that there is an interpolant $I \subseteq \mathsf{Sen}^{\mathcal{G}}(\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma))$. Thus for every $(\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma))$-model M, if $M \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma)} \alpha^{\mu}_{\Lambda}(\mathcal{O})$, then $M \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma)} I$, and therefore $M \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma)} \alpha^{\eta}_{\Sigma}(\varphi)$, showing $\mathcal{O} \models^{\eta}_{\Sigma} \varphi$ as desired. □

The property described in Lemma 3.2.18 also extends to entailment:

Proposition 3.2.19. *For any binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis η, and Λ_1-ontology $\mathcal{O}_1$ for μ_1, Λ_2-ontology $\mathcal{O}_2$ for μ_2, and signatures $\Sigma \subseteq \Sigma'$ in $Sig^{\mathcal{Q}}$, if Figure 3.2 has CIP, then:*

$$\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma'} \mathcal{O}_2 \quad \textit{implies} \quad \mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2\ .$$

Proof: Let Λ_1, Λ_2, Σ, Σ', $\mathcal{O}_1$ and $\mathcal{O}_2$ be as in the statement of the proposition.

Assume that $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma'} \mathcal{O}_2$ and $\mathcal{O}_2 \models^{\eta}_{\Sigma} \varphi$, for an arbitrary $\varphi \in \mathsf{Sen}^{\mathcal{Q}}(\Sigma)$; it follows by Lemma 3.2.18 that $\mathcal{O}_2 \models^{\eta}_{\Sigma'} \varphi$. From the assumption that $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma'} \mathcal{O}_2$ it follows that $\mathcal{O}_1 \models^{\eta}_{\Sigma'} \varphi$. As $\varphi \in \mathsf{Sen}^{\mathcal{Q}}(\Sigma)$ and $\Sigma \subseteq \Sigma'$ we get $\mathcal{O}_1 \models^{\eta}_{\Sigma} \varphi$, and thus $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2$. □

Note that the opposite direction does not hold because it would imply extending the signature over which queries may be expressed.

3.2.3 Frameworks with attached comorphisms

Investigating frameworks gives us an insight into properties of entailment also in more complex situations. For instance we can consider frameworks with attached comorphisms. So we may have a comorphism attached on the query language side, intuitively it is a situation when we already have a framework with $\mathcal{L}$ and $\mathcal{Q}$ and we have a query formulated in a language $\mathcal{Q}'$, such that there is a comorphism $\xi : \mathcal{Q}' \to \mathcal{Q}$, i.e. $\mathcal{Q}'$ is a weaker language than $\mathcal{Q}$. We may also have a situation when we already have a framework with $\mathcal{L}$ and $\mathcal{Q}$ and we have an ontology formulated in a language $\mathcal{L}'$, such that there is a comorphism $\zeta : \mathcal{L}' \to \mathcal{L}$, i.e. $\mathcal{L}'$ is a weaker language than $\mathcal{L}$. We also present the consequences of these two situations.

First we consider a framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$ with an additional comorphism $\xi : \mathcal{Q}' \to \mathcal{Q}$, Figure 3.3 represents that situation. Using comorphisms composition $\xi;\eta = \eta'$ we receive a framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $\eta' : \mathcal{Q}' \to \mathcal{G}$. This is illustrated in Figure 3.4.

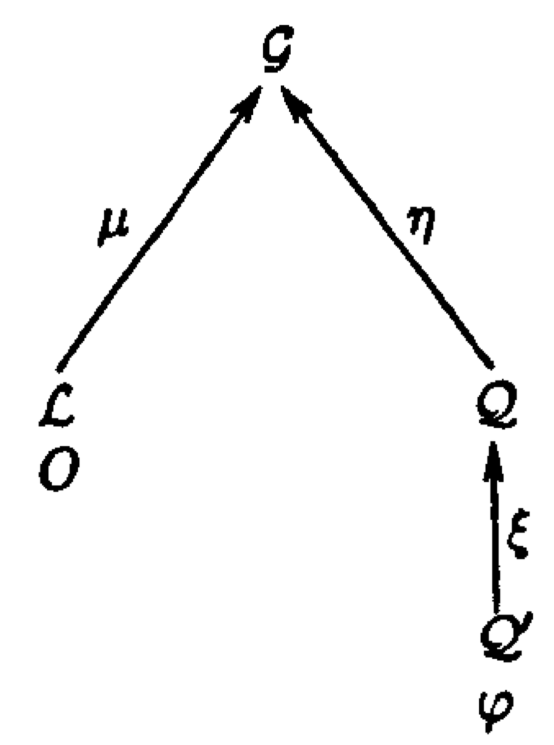


Figure 3.3.

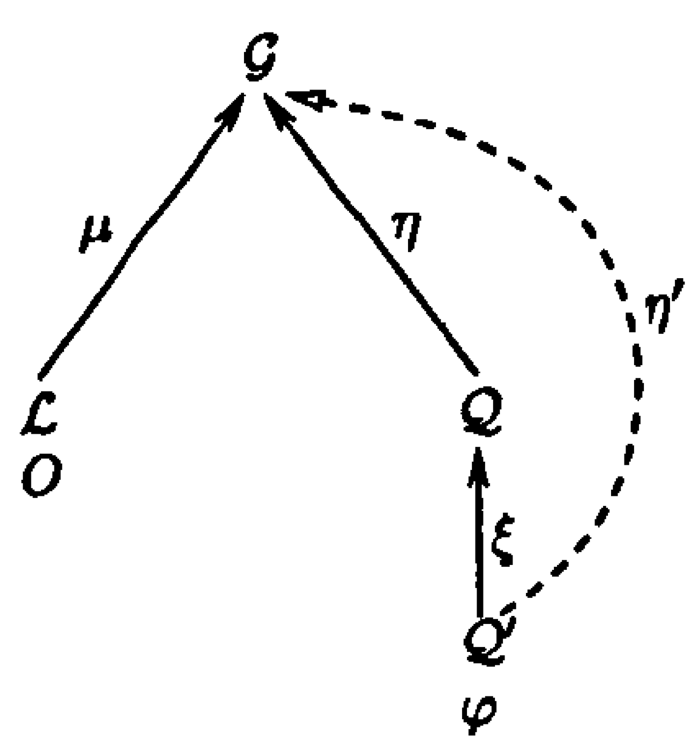


Figure 3.4.

Lemma 3.2.20. *Given a framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$ with an attached comorphism $\xi : \mathcal{Q}' \to \mathcal{Q}$, we get a framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $\eta' : \mathcal{Q}' \to \mathcal{G}$, with $\eta' = \xi;\eta$, such that for any Λ-ontology $\mathcal{O}$ for μ, and a query $\varphi \in Sen^{\mathcal{Q}'}(\Sigma')$, where $\Sigma' \in |Sig^{\mathcal{Q}'}|$, the following holds:*

$$\mathcal{O} \models^{\eta'}_{\Sigma'} \varphi \quad \textit{iff} \quad \mathcal{O} \models^{\eta}_{\Phi^{\xi}(\Sigma')} \alpha^{\xi}_{\Sigma'}(\varphi) \,.$$

Proof. The proof is as follows:

$$\begin{array}{rrl} & \mathcal{O} & \models^{\eta'}_{\Sigma'} \varphi \\ \text{iff} & \alpha^{\mu}_{\Lambda}(\mathcal{O}) & \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)\cup\Phi^{\eta'}(\Sigma')} \alpha^{\eta'}_{\Sigma'}(\varphi) \\ \text{iff} & \alpha^{\mu}_{\Lambda}(\mathcal{O}) & \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Phi^{\xi}(\Sigma'))} \alpha^{\eta}_{\Phi^{\xi}(\Sigma')}(\alpha^{\xi}_{\Sigma'}(\varphi)) \\ \text{iff} & \mathcal{O} & \models^{\eta}_{\Phi^{\xi}(\Sigma')} \alpha^{\xi}_{\Sigma'}(\varphi) \,. \end{array}$$

□

Note that the third line of the proof follows from the fact that $\eta' = \xi;\eta$.

In other words Lemma 3.2.20 states that given a framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$ and a query formulated in $\mathcal{Q}'$, such that there is a comorphism $\mathcal{Q}' \to \mathcal{Q}$ then we can safely lift the query to $\mathcal{Q}$ and then translate it into $\mathcal{G}$. That gives us exactly the same results as creating a framework μ over query basis $\eta' : \mathcal{Q}' \to \mathcal{G}$ (using composition of comorphisms) for answering the query. In other words, to answer a query which is formulated in a sublanguage of $\mathcal{Q}$ we do not have to introduce new query basis over which we use the framework.

Example 3.2.21. *As an example consider a scenario where we have a framework* $\mathsf{e^+f}$: $\mathcal{EL}^+ \rightarrow FOL$ *over query basis* $\mathsf{af} : \mathcal{ALC} \rightarrow FOL$*, and we use it to answer a query formulated in* $\mathcal{ALC}$*. Now, Lemma 3.2.20 tells us that if we have another query which is formulated in a sublanguage of* $\mathcal{ALC}$*, for instance in* $\mathcal{EL}$*, we can "reuse" the framework and the query base. To do this we can "attach" comorphism* $\mathsf{ea} : \mathcal{EL} \rightarrow \mathcal{ALC}$*, then we can lift the query to* $\mathcal{ALC}$ *and answer it in the framework and we will receive the same answer as in the scenario with query basis* $\mathsf{ef} : \mathcal{EL} \rightarrow FOL$*. This is illustrated by Figure 3.5.*

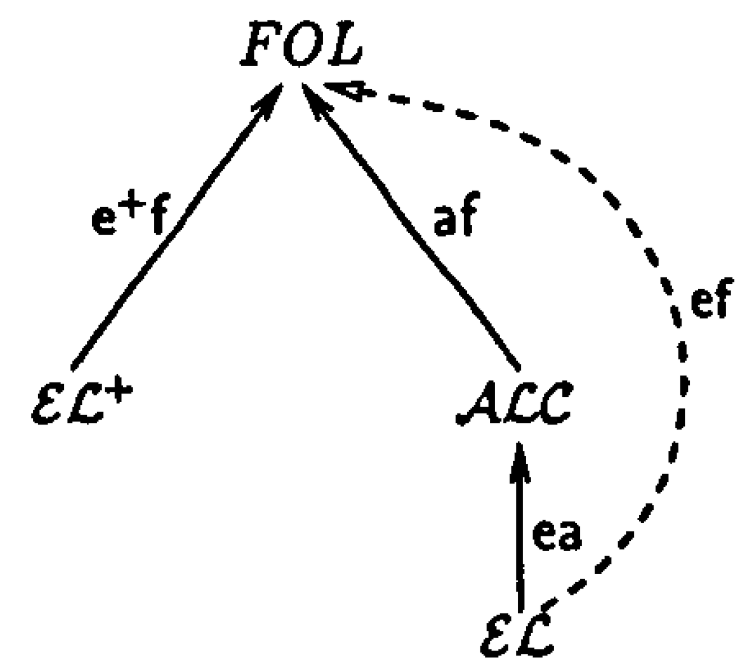


Figure 3.5.

The next proposition extends the result of Lemma 3.2.20 to Σ-entailment.

Proposition 3.2.22. *For any binary framework* $\mathfrak{F} = (\mu_1, \mu_2)$ *over query basis* $\eta : \mathcal{Q} \rightarrow \mathcal{G}$ *with an attached comorphism* $\xi : \mathcal{Q}' \rightarrow \mathcal{Q}$*, and the same binary framework over* $\eta' : \mathcal{Q}' \rightarrow \mathcal{G}$*, where* $\eta' = \xi;\eta$*, for signatures* $\Lambda_1 \in |Sig^{\mathcal{L}_1}|$*,* $\Lambda_2 \in |Sig^{\mathcal{L}_2}|$ *and* $\Sigma \in |Sig^{\mathcal{Q}'}|$*, and ontologies* $\mathcal{O}_1 \subseteq Sen^{\mathcal{L}_1}(\Lambda_1)$ *and* $\mathcal{O}_2 \subseteq Sen^{\mathcal{L}_2}(\Lambda_2)$ *the following implication holds:*

$$\mathcal{O}_1 \sqsubseteq^{\eta}_{\Phi^{\xi}(\Sigma)} \mathcal{O}_2 \quad \textit{implies} \quad \mathcal{O}_1 \sqsubseteq^{\eta'}_{\Sigma} \mathcal{O}_2 \ .$$

Proof. Assume $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Phi^{\xi}(\Sigma)} \mathcal{O}_2$ and $\mathcal{O}_2 \models^{\eta'}_{\Sigma} \varphi$. By Lemma 3.2.20 we have that $\mathcal{O}_2 \models^{\eta}_{\Phi^{\xi}(\Sigma)} \alpha^{\xi}_{\Sigma}(\varphi)$. By the assumption $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Phi^{\xi}(\Sigma)} \mathcal{O}_2$, we have $\mathcal{O}_1 \models^{\eta}_{\Phi^{\xi}(\Sigma)} \alpha^{\xi}_{\Sigma}(\varphi)$. Again by the Lema 3.2.20 we receive that $\mathcal{O}_1 \models^{\eta'}_{\Sigma} \varphi$. □

This result further extends to inseparability of ontologies.

Corollary 3.2.23. *For any binary framework* $\mathfrak{F} = (\mu_1, \mu_2)$ *over query basis* $\eta : \mathcal{Q} \rightarrow \mathcal{G}$ *with an attached comorphism* $\xi : \mathcal{Q}' \rightarrow \mathcal{Q}$*, and the same binary framework over* $\eta' : \mathcal{Q}' \rightarrow \mathcal{G}$*, where* $\eta' = \xi;\eta$*, for signatures* $\Lambda_1 \in |Sig^{\mathcal{L}_1}|$*,* $\Lambda_2 \in |Sig^{\mathcal{L}_2}|$ *and* $\Sigma \in |Sig^{\mathcal{Q}'}|$*, and ontologies* $\mathcal{O}_1 \subseteq Sen^{\mathcal{L}_1}(\Lambda_1)$ *and* $\mathcal{O}_2 \subseteq Sen^{\mathcal{L}_2}(\Lambda_2)$ *we have that* $\mathcal{O}_1 \approx^{\eta}_{\Phi^{\xi}(\Sigma)} \mathcal{O}_2$ *implies* $\mathcal{O}_1 \approx^{\eta'}_{\Sigma} \mathcal{O}_2$.

As promised above we now consider a framework with an attached comorphism on the ontology language side.

Given a framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$, such that there is an attached comorphism $\zeta : \mathcal{L}' \to \mathcal{L}$, Figure 3.6 represents that situation, by composition of comorphisms $\zeta;\mu = \mu'$, we can construct a framework $\mu' : \mathcal{L}' \to \mathcal{G}$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$. This is illustrated in Figure 3.7. Note that given a $\mathcal{O} \subseteq \mathrm{Sen}^{\mathcal{L}'}(\Lambda)$ we have that $\alpha^{\zeta}_{\Lambda}(\mathcal{O})$ has the same set of consequences in μ over η as $\mathcal{O}$ in μ' over η. This is the statement of Lemma 3.2.24.

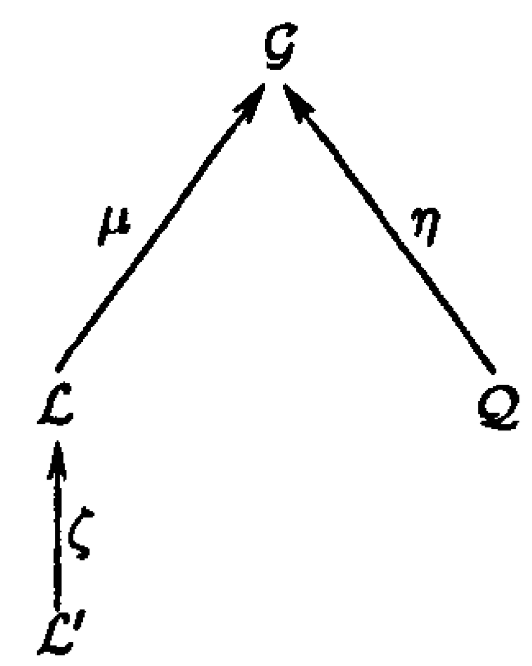


Figure 3.6.

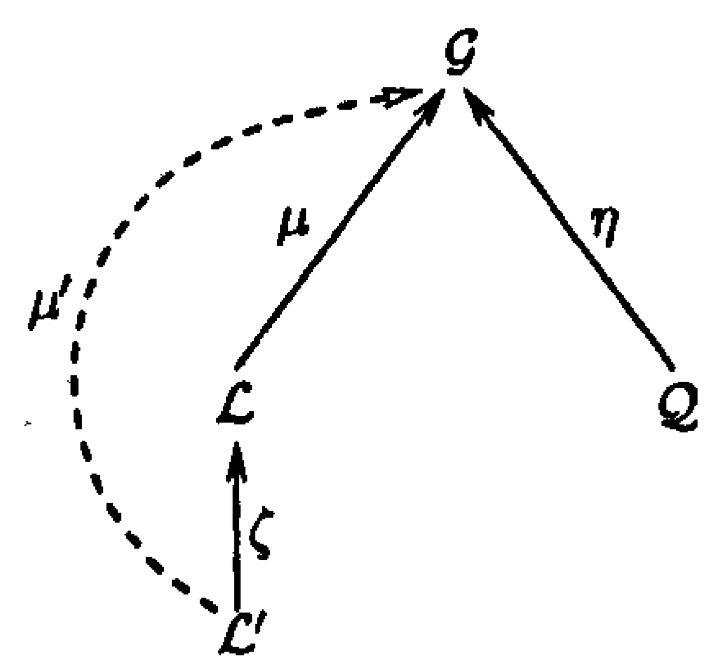


Figure 3.7.

Lemma 3.2.24. *For a framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$ with an attached comorphism $\zeta : \mathcal{L}' \to \mathcal{L}$, we can create framework $\mu' : \mathcal{L}' \to \mathcal{G}$ over query basis η with $\zeta;\mu = \mu'$, such that:*

$$\mathcal{O} \models^{\eta}_{\Sigma} \varphi \quad \textit{iff} \quad \alpha^{\zeta}_{\Lambda}(\mathcal{O}) \models^{\eta}_{\Sigma} \varphi$$

for any $\mathcal{L}'$-signature Λ, ontology $\mathcal{O} \subseteq Sen^{\mathcal{L}'}(\Lambda)$, and a query $\varphi \in Sen^{\mathcal{Q}}(\Sigma)$.

Proof. The proof is as follows:

$$\begin{array}{lrl} & \mathcal{O} & \models^{\eta}_{\Sigma} \varphi \\ \text{iff} & \alpha^{\mu'}_{\Lambda}(\mathcal{O}) & \models^{\mathcal{G}}_{\Phi^{\mu'}(\Lambda)\cup\Phi^{\eta}(\Sigma)} \alpha^{\eta}_{\Sigma}(\varphi) \\ \text{iff} & \alpha^{\mu}_{\Phi^{\zeta}(\Lambda)}(\alpha^{\zeta}_{\Lambda}(\mathcal{O})) & \models^{\mathcal{G}}_{\Phi^{\mu}(\Phi^{\zeta}(\Lambda))\cup\Phi^{\eta}(\Sigma)} \alpha^{\eta}_{\Sigma}(\varphi) \\ \text{iff} & \alpha^{\zeta}_{\Lambda}(\mathcal{O}) & \models^{\eta}_{\Sigma} \varphi \end{array}$$

□

This is to say, that given a framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$ with an attached comorphism $\zeta : \mathcal{L}' \to \mathcal{L}$ and an ontology $\mathcal{O} \subseteq \mathsf{Sen}^{\mathcal{L}'}(\Lambda)$, we can safely lift $\mathcal{O}$ to $\mathcal{L}$ and after translating it into $\mathcal{G}$ answer the query. That gives us exactly the same results as creating a framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$ using composition of comorphisms and then answering the query in $\mathcal{G}$.

Example 3.2.25. *As an example consider a scenario where we are given a framework* $\mathsf{af} : \mathcal{ALC} \to FOL$ *over query basis* $\mathsf{e^+f} : \mathcal{EL}^+ \to FOL$ *and an ontology* $\mathcal{O}$ *for* af*. Assume that from* $\mathcal{O}$ *we have extracted a module* $\mathcal{O}'$ *which is formulated in* $\mathcal{EL}$*. Lemma 3.2.24 tells us that in order to use it for query answering we do not have to introduce new framework but we can attach comorphism* ea *to the framework and then we can lift* $\mathcal{O}'$ *to* $\mathcal{ALC}$*. In other words we can keep using the original framework when we are using a module, which is formulated in a weaker language. This is illustrated by Figure 3.8.*

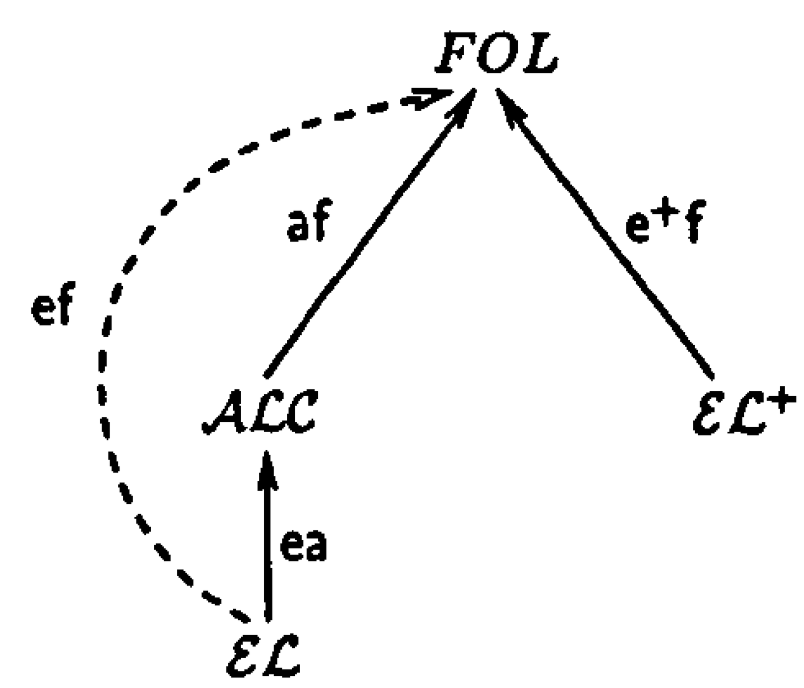


Figure 3.8.

The following proposition extends the result of Lemma 3.2.24 to Σ-entailment of ontologies.

Proposition 3.2.26. *For a binary framework* $\mathfrak{F} = (\mu_1, \mu_2)$ *over query basis* $\eta : \mathcal{Q} \to \mathcal{G}$ *with attached comorphisms* $\zeta_i : \mathcal{L}'_i \to \mathcal{L}_i$ *and a binary framework* $\mathfrak{F} = (\mu'_1, \mu'_2)$ *over query basis* η *with* $\zeta_i ; \mu_i = \mu'_i$ *(for* $i = 1, 2$*), we have that:*

$$\alpha^{\zeta_1}_{\Lambda_1}(\mathcal{O}_1) \sqsubseteq^{\eta}_{\Sigma} \alpha^{\zeta_2}_{\Lambda_2}(\mathcal{O}_2) \quad \textit{iff} \quad \mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2$$

for any signatures $\Sigma \in |Sig^{\mathcal{Q}}|$, $\Lambda_i \in |Sig^{\mathcal{L}'_i}|$, *and ontologies* $\mathcal{O}_i \subseteq Sen^{\mathcal{L}'_i}(\Lambda_i)$.

Proof. For the direction "$\Rightarrow$" assume that $\alpha^{\zeta_1}_{\Lambda_1}(\mathcal{O}_1) \sqsubseteq^{\eta}_{\Sigma} \alpha^{\zeta_2}_{\Lambda_2}(\mathcal{O}_2)$ and $\mathcal{O}_2 \models^{\eta}_{\Sigma} \varphi$, for $\varphi \in \mathsf{Sen}^{\mathcal{Q}}(\Sigma)$. By Lemma 3.2.24 we have $\alpha^{\zeta_2}_{\Lambda_2}(\mathcal{O}_2) \models^{\eta}_{\Sigma} \varphi$, and by the assumption $\alpha^{\zeta_1}_{\Lambda_1}(\mathcal{O}_1) \sqsubseteq^{\eta}_{\Sigma} \alpha^{\zeta_2}_{\Lambda_2}(\mathcal{O}_2)$, we get $\alpha^{\zeta_1}_{\Lambda_1}(\mathcal{O}_1) \models^{\eta}_{\Sigma} \varphi$. Again by Lemma 3.2.24 we have $\mathcal{O}_1 \models^{\eta}_{\Sigma} \varphi$. Thus $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2$ as desired.

For "$\Leftarrow$" assume that $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2$ and $\alpha^{\zeta_2}_{\Lambda_2}(\mathcal{O}_2) \models^{\eta}_{\Sigma} \varphi$, for $\varphi \in \mathsf{Sen}^{\mathcal{Q}}(\Sigma)$. By Lemma 3.2.24 we have $\mathcal{O}_2 \models^{\eta}_{\Sigma} \varphi$ and by the assumption $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2$, we get $\mathcal{O}_1 \models^{\eta}_{\Sigma} \varphi$. Again by Lemma 3.2.24 we have $\alpha^{\zeta_1}_{\Lambda_1}(\mathcal{O}_1) \models^{\eta}_{\Sigma} \varphi$. Thus $\alpha^{\zeta_1}_{\Lambda_1}(\mathcal{O}_1) \sqsubseteq^{\eta}_{\Sigma} \alpha^{\zeta_2}_{\Lambda_2}(\mathcal{O}_2)$ as desired. $\square$

The next corollary is a consequence of Proposition 3.2.26. It shows that for framework $\mu : \mathcal{L} \to \mathcal{G}$ with comorphism $\rho : \mathcal{L} \to \mathcal{Q}$, over query basis $\eta : \mathcal{Q} \to \mathcal{G}$, and framework $\eta : \mathcal{Q} \to \mathcal{G}$, with an attached comorphism $\rho : \mathcal{L} \to \mathcal{Q}$ over query basis η itself, Σ-entailment of ontologies $\mathcal{O}_1, \mathcal{O}_2 \subseteq \mathsf{Sen}^{\mathcal{L}}(\Lambda)$ coincides.

Corollary 3.2.27. *Let $\mathfrak{F} = (\mu_1, \mu_2)$ a binary framework over a query basis $\eta : \mathcal{Q} \to \mathcal{G}$, such that there are comorphisms $\rho_i : \mathcal{L}_i \to \mathcal{Q}$ and $\mu_i = \rho_i; \eta$, for $i = 1, 2$. Let $\Sigma \in |Sig^{\mathcal{Q}}|$ and ontologies $\mathcal{O}_i \subseteq Sen^{\mathcal{L}'_i}(\Lambda_i)$ then:*

$$\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2 \quad \textit{iff} \quad \alpha^{\rho_1}_{\Lambda_1}(\mathcal{O}_1) \sqsubseteq^{\eta}_{\Sigma} \alpha^{\rho_2}_{\Lambda_2}(\mathcal{O}_2) \; .$$

If an ontology language can be translated directly into the query language, then entailment can be reduced to showing that each sentence in one ontology is a consequence of the other:

Proposition 3.2.28. *Given a framework $\mathcal{L} \xrightarrow{\mu} \mathcal{G}$ over a query basis $\mathcal{Q} \xrightarrow{\eta} \mathcal{G}$ with a comorphism $\mathcal{L} \xrightarrow{\rho} \mathcal{Q}$, such that $\mu = \rho; \eta$, and Λ-ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, we have:*

$$\mathcal{O}_1 \sqsubseteq^{\eta}_{\Phi^{\rho}(\Lambda)} \mathcal{O}_2 \quad \textit{iff} \quad \mathcal{O}_1 \models^{\eta}_{\Phi^{\rho}(\Lambda)} \varphi \, ,$$

for all $\varphi \in \alpha^{\rho}_{\Lambda}(\mathcal{O}_2)$.

Proof: In this proof we use the fact that $\mu = \rho; \eta$, thus $\Phi^{\mu}(\Lambda) = \Phi^{\eta}(\Phi^{\rho}(\Lambda))$ therefore $\Phi^{\mu}(\Lambda) = \Phi^{\mu}(\Lambda) \cup \Phi^{\eta}(\Phi^{\rho}(\Lambda))$. Thus we write $\alpha^{\mu}_{\Lambda}(\mathcal{O}_2) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)} \alpha^{\eta}_{\Phi^{\rho}(\Lambda)}(\psi)$ instead of $\alpha^{\mu}_{\Lambda}(\mathcal{O}_2) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda) \cup \Phi^{\eta}(\Phi^{\rho}(\Lambda))} \alpha^{\eta}_{\Phi^{\rho}(\Lambda)}(\psi)$.
For "$\Rightarrow$" assume $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Phi^{\rho}(\Lambda)} \mathcal{O}_2$, i.e. $\mathcal{O}_2 \models^{\eta}_{\Phi^{\rho}(\Lambda)} \psi$ implies $\mathcal{O}_1 \models^{\eta}_{\Phi^{\rho}(\Lambda)} \psi$ for any $\psi \in \mathsf{Sen}^{\mathcal{Q}}(\Phi^{\rho}(\Lambda))$. Note that for every $\varphi \in \alpha^{\rho}_{\Lambda}(\mathcal{O}_2)$ we have $\varphi = \alpha^{\rho}_{\Lambda}(\varphi')$ with $\varphi' \in \mathcal{O}_2$, thus by the composition of comorphisms we have that:

$$\alpha^{\mu}_{\Lambda}(\varphi') = \alpha^{\eta}_{\Phi^{\rho}(\Lambda)}(\alpha^{\rho}_{\Lambda}(\varphi')) = \alpha^{\eta}_{\Phi^{\rho}(\Lambda)}(\varphi).$$

Thus for every $\varphi \in \alpha^{\rho}_{\Lambda}(\mathcal{O}_2)$ we have $\alpha^{\mu}_{\Lambda}(\mathcal{O}_2) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)} \alpha^{\eta}_{\Phi^{\rho}(\Lambda)}(\varphi)$. By the assumption we have $\alpha^{\mu}_{\Lambda}(\mathcal{O}_1) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)} \alpha^{\eta}_{\Phi^{\rho}(\Lambda)}(\varphi)$, and thus $\mathcal{O}_1 \models^{\eta}_{\Phi^{\rho}(\Lambda)} \varphi$.
For "$\Leftarrow$" assume $\mathcal{O}_1 \models^{\eta}_{\Phi^{\rho}(\Lambda)} \varphi$, i.e., $\alpha^{\mu}_{\Lambda}(\mathcal{O}_1) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)} \alpha^{\eta}_{\Phi^{\rho}(\Lambda)}(\varphi)$ for all $\varphi \in \alpha^{\rho}_{\Lambda}(\mathcal{O}_2)$. Since $\mu = \rho; \eta$, this means $\alpha^{\mu}_{\Lambda}(\mathcal{O}_1) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)} \alpha^{\mu}_{\Lambda}(\mathcal{O}_2)$. Therefore $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Phi^{\mu}(\Lambda)} \mathcal{O}_2$, as required. □

Moving to a richer global language preserves consequences:

Lemma 3.2.29. *For framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$, if we have a comorphism $\lambda : \mathcal{G} \to \mathcal{G}'$, there is a framework $\mu' = \mu; \lambda$ over query basis $\eta' = \eta; \lambda$, and we have:*

$$\mathcal{O} \models^{\eta}_{\Sigma} \varphi \quad \textit{implies} \quad \mathcal{O} \models^{\eta'}_{\Sigma} \varphi$$

for any Λ-ontology for μ and any query $\varphi \in Sen^{\mathcal{Q}}(\Sigma)$, with $\Sigma \in Sig^{\mathcal{Q}}$. Moreover, if β^{λ} is surjective, then the converse implication also holds.

Proof: Suppose $\mathcal{O} \models^{\eta}_{\Sigma} \varphi$, i.e., $\alpha^{\mu}_{\Lambda}(\mathcal{O}) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma)} \alpha^{\eta}_{\Sigma}(\varphi)$. By Lemma 2.4.36 it follows that (or is equivalent to, if β^{λ} is surjective)

$$\alpha^{\lambda}_{\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma)}(\alpha^{\mu}_{\Lambda}(\mathcal{O})) \models^{\mathcal{G}'}_{\Phi^{\lambda}(\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma))} \alpha^{\lambda}_{\Phi^{\mu}(\Lambda)\cup\Phi^{\eta}(\Sigma)}(\alpha^{\eta}_{\Sigma}(\varphi)) \ .$$

Since Φ^{λ} distributes over unions, and α^{λ} commutes with inclusions, this is equivalent to

$$\alpha^{\lambda}_{\Phi^{\mu}(\Lambda)}(\alpha^{\mu}_{\Lambda}(\mathcal{O})) \models^{\mathcal{G}'}_{\Phi^{\lambda}(\Phi^{\mu}(\Lambda))\cup\Phi^{\lambda}(\Phi^{\eta}(\Sigma))} \alpha^{\lambda}_{\Phi^{\eta}(\Sigma)}(\alpha^{\eta}_{\Sigma}(\varphi)) \ .$$

But since $\mu' = \mu;\lambda$ and $\eta' = \eta;\lambda$, this is $\alpha^{\mu'}_{\Lambda}(\mathcal{O}) \models^{\mathcal{G}}_{\Phi^{\mu'}(\Lambda)\cup\Phi^{\eta'}(\Sigma)} \alpha^{\eta'}_{\Sigma}(\varphi)$, i.e., $\mathcal{O} \models^{\eta'}_{\Sigma} \varphi$ as desired. □

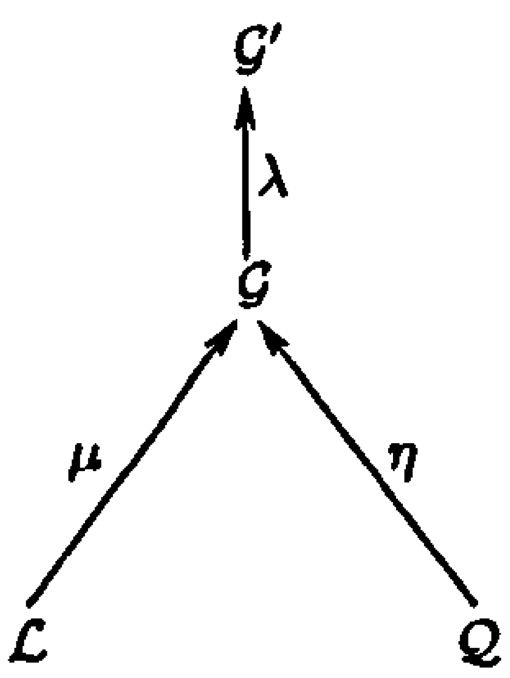


Figure 3.9.

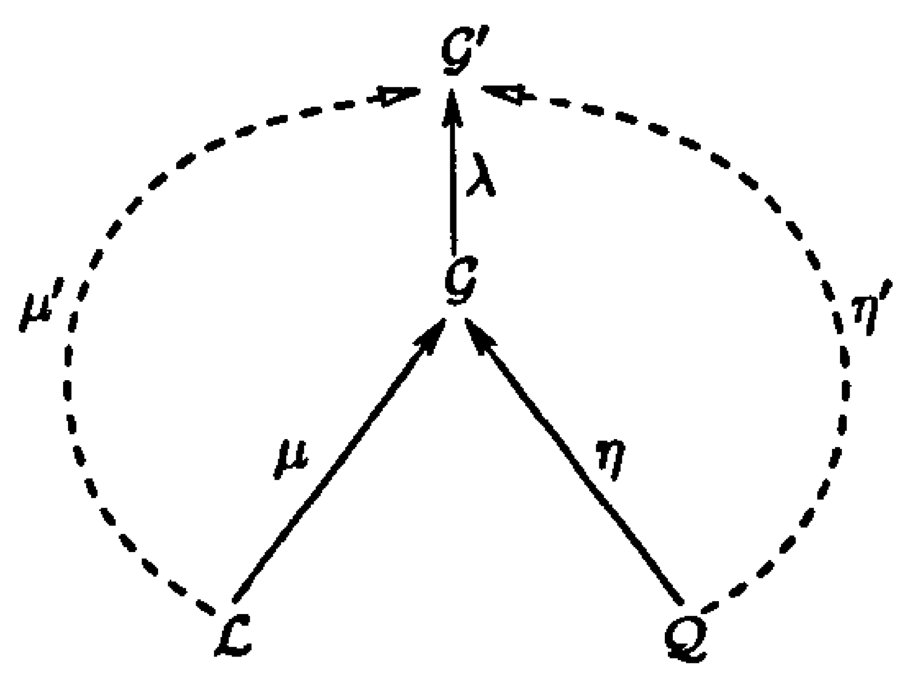


Figure 3.10.

Example 3.2.30. *As an example consider a scenario with framework* ea : $\mathcal{EL} \rightarrow \mathcal{ALC}$ *over query basis* $1_{\mathcal{ALC}} : \mathcal{ALC} \rightarrow \mathcal{ALC}$ *and ontology* $\mathcal{O}$ *for* ea. *In such a case Lemma 3.2.29 tells us that moving to FOL, using attached comorphism* af, *we will preserve all the consequences of* $\mathcal{O}$ *already present in* $\mathcal{ALC}$. *This is illustrated by Figure 3.11.*

Now we compare two frameworks presented in Figure 3.12 and in Figure 3.13. The former is simply the case when the ontology language is a sublanguage of the query language, but the query is expressed in $\mathcal{G}$ over a translated $\mathcal{L}$-signature. Whereas the latter is the case when the ontology language is the same as the query language, but we translate both

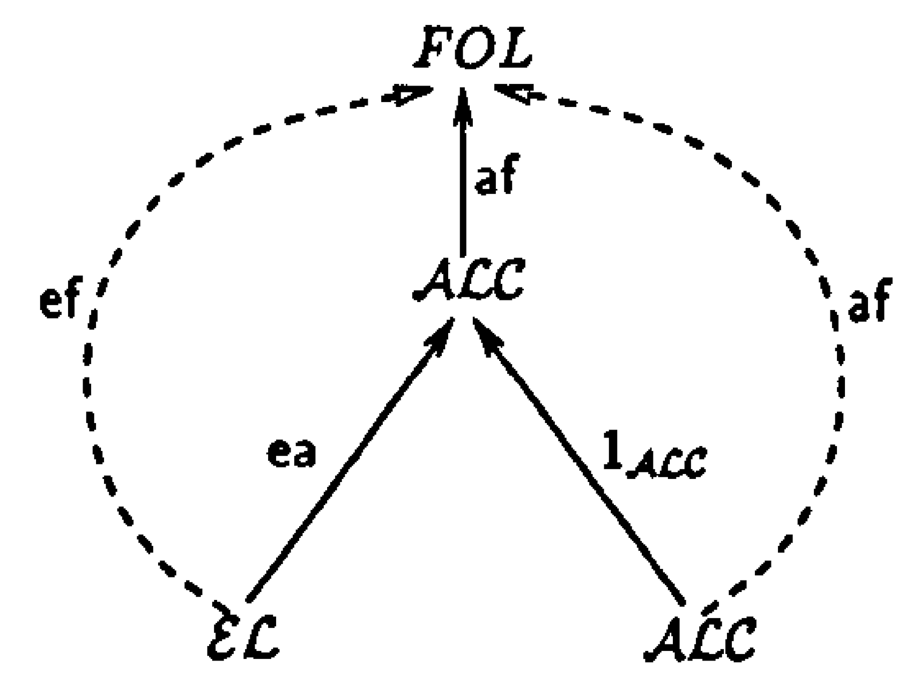


Figure 3.11.

into $\mathcal{G}$, which in this case is the global language. Additionally, in both cases comorphism $\mu : \mathcal{L} \to \mathcal{G}$ is conservative. We show a correlation of conservativity of comorphism $\mu : \mathcal{L} \to \mathcal{G}$ with coincidence of $\mathcal{O}_1 \approx^1_{\Phi^\mu(\Lambda)} \mathcal{O}_2$ and $\mathcal{O}_1 \approx^\eta_\Lambda \mathcal{O}_2$. To do that first we need an auxiliary lemma. First note that it uses notation introduced in Notation 2.5.2. The statement of this auxiliary lemma is that given two signatures $\Lambda, \Lambda' \in |\mathsf{Sig}^{\mathcal{L}}|$, such that there is a signature morphism $\sigma : \Lambda \to \Lambda'$ and an ontology $\mathcal{O} \subseteq \mathsf{Sen}^{\mathcal{L}}(\Lambda')$ we are guaranteed that $\mathcal{O}$ itself and the set of consequences of $\mathcal{O}$, restricted to these sentences that were originally expressed in Λ and then translated into Λ' using $\mathsf{Sen}^{\mathcal{L}}(\sigma)$ (i.e. $\mathcal{O}_\sigma$), give us exactly the same set of consequences over sentences expressed in Λ.

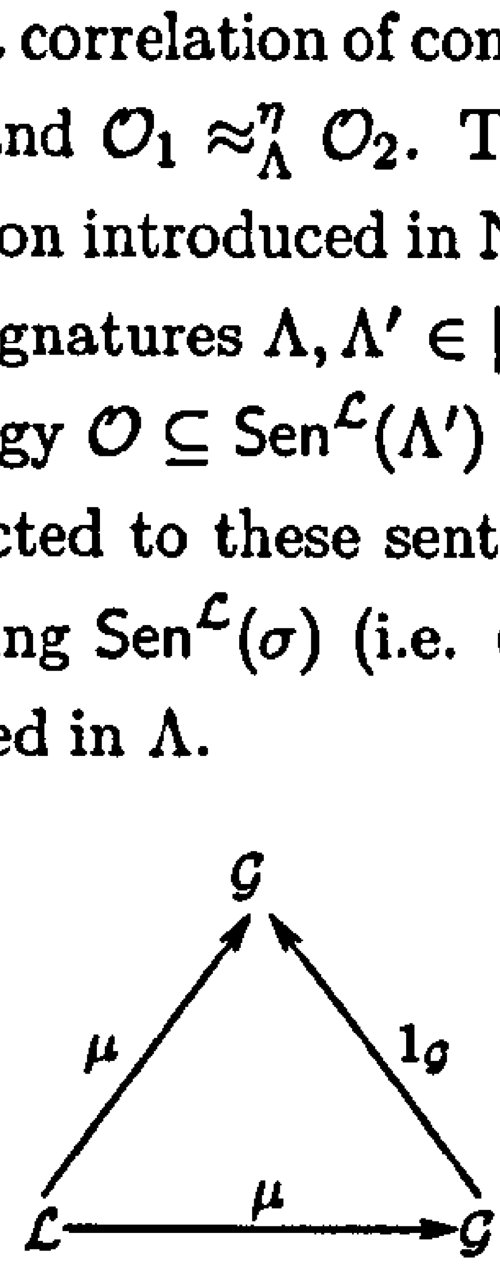


Figure 3.12: Framework μ over query basis 1

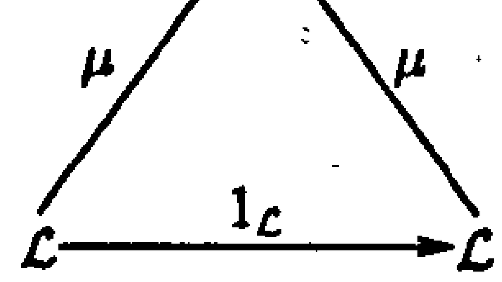


Figure 3.13: Framework μ over query basis μ

Lemma 3.2.31. *For any framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $\mu : \mathcal{L} \to \mathcal{G}$ and all signatures $\Lambda, \Lambda' \in |Sig^{\mathcal{L}}|$, such that $\Lambda \xrightarrow{\sigma} \Lambda'$ and an ontology $\mathcal{O} \subseteq Sen^{\mathcal{L}}(\Lambda')$ the following holds: $\mathcal{O} \approx^\mu_\Lambda Sen^{\mathcal{L}}(\sigma)(\mathcal{O}_\sigma)$, i.e. for every $\varphi \in Sen^{\mathcal{L}}(\Lambda)$ we have that $\mathcal{O} \models^\mu_\Lambda \varphi$ iff $Sen^{\mathcal{L}}(\sigma)(\mathcal{O}_\sigma) \models^\mu_\Lambda \varphi$.*

Proof. Let $\Lambda, \Lambda' \in |\mathsf{Sig}^{\mathcal{L}}|$, such that $\Lambda \xrightarrow{\sigma} \Lambda'$, let $\mathcal{O} \subseteq \mathsf{Sen}^{\mathcal{L}}(\Lambda')$ and $\varphi \in \mathsf{Sen}^{\mathcal{L}}(\Lambda)$.

"$\Rightarrow$". Assume $\alpha^{\mu}_{\Lambda'}(\mathcal{O}) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda\cup\Lambda')} \alpha^{\mu}_{\Lambda}(\varphi)$. Let $\mathcal{M} \in |\mathsf{Mod}^{\mathcal{G}}(\Phi^{\mu}(\Lambda\cup\Lambda'))|$, such that $\mathcal{M} \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda')} \alpha^{\mu}_{\Lambda'}(\mathsf{Sen}^{\mathcal{L}}(\sigma)(\mathcal{O}_{\sigma}))$. Thus $\beta^{\mu}_{\Lambda\cup\Lambda'}(\mathcal{M}) \models^{\mathcal{L}}_{\Lambda'} \mathsf{Sen}^{\mathcal{L}}(\sigma)(\mathcal{O}_{\sigma})$, by definition, $\varphi \in \mathcal{O}_{\sigma}$ and therefore $\mathsf{Sen}^{\mathcal{L}}(\sigma)(\varphi) \in \mathsf{Sen}^{\mathcal{L}}(\sigma)(\mathcal{O}_{\sigma})$. Thus $\beta^{\mu}_{\Lambda\cup\Lambda'}(\mathcal{M}) \models^{\mathcal{L}}_{\Lambda'} \mathsf{Sen}^{\mathcal{L}}(\sigma)(\varphi)$, this implies $\beta^{\mu}_{\Lambda\cup\Lambda'}(\mathcal{M}) \models^{\mathcal{L}}_{\Lambda} \varphi$. Therefore $\mathcal{M} \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)} \alpha^{\mu}_{\Lambda}(\varphi)$.

"$\Leftarrow$". Assume $\alpha^{\mu}_{\Lambda'}(\mathsf{Sen}^{\mathcal{L}}(\sigma)(\mathcal{O}_{\sigma})) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda\cup\Lambda')} \alpha^{\mu}_{\Lambda}(\varphi)$. Let $\mathcal{M} \in |\mathsf{Mod}^{\mathcal{G}}(\Phi^{\mu}(\Lambda\cup\Lambda'))|$, such that $\mathcal{M} \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda')} \alpha^{\mu}_{\Lambda'}(\mathcal{O})$. Thus $\beta^{\mu}_{\Lambda\cup\Lambda'}(\mathcal{M}) \models^{\mathcal{L}}_{\Lambda'} \mathcal{O}$, which implies $\beta^{\mu}_{\Lambda'}(\mathcal{M}) \models^{\mathcal{L}}_{\Lambda'} \mathsf{Sen}^{\mathcal{L}}(\sigma)(\mathcal{O}_{\sigma})$. Thus $\mathcal{M} \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda')} \alpha^{\mu}_{\Lambda'}(\mathsf{Sen}^{\mathcal{L}}(\sigma)(\mathcal{O}_{\sigma}))$. By the assumption we get $\mathcal{M} \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)} \alpha^{\mu}_{\Lambda}(\varphi)$, as required. □

Next proposition presents close correlation between the inseparability problem in two types of frameworks presented above and conservativity of comorphism $\mu : \mathcal{L} \to \mathcal{G}$.

Proposition 3.2.32. *For framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$ and framework μ over itself as query basis, comorphism μ is conservative iff $\Phi^{\mu}(\Lambda)$-inseparability w.r.t. $1_{\mathcal{G}}$ coincides with Λ-inseparability w.r.t. μ, for any signature $\Lambda \in |Sig^{\mathcal{L}}|$.*

Note that $\mathcal{O}_1 \approx^{1_{\mathcal{G}}}_{\Phi^{\mu}(\Lambda)} \mathcal{O}_2$ means that $\mathcal{O}_1$ and $\mathcal{O}_2$ are indistinguishable relative to the sentences from the set $Sen^{\mathcal{G}}(\Phi^{\mu}(\Lambda))$, whereas $\mathcal{O}_1 \approx^{\mu}_{\Lambda} \mathcal{O}_2$ means that $\mathcal{O}_1$ and $\mathcal{O}_2$ are indistinguishable relative to the sentences from the set $Sen^{\mathcal{L}}(\Lambda)$.

Proof. "$\Rightarrow$". Let $\mu : \mathcal{L} \to \mathcal{G}$ be a framework over query basis $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$, and let it also be a framework over itself as a query basis, moreover, μ is conservative. Let $\Lambda, \Lambda' \in |\mathsf{Sig}^{\mathcal{L}}|$, and $\mathcal{O}_1, \mathcal{O}_2 \subseteq \mathsf{Sen}^{\mathcal{L}}(\Lambda')$. By Corollary 3.2.23 we only show $\mathcal{O}_1 \approx^{\mu}_{\Lambda} \mathcal{O}_2$ implies $\mathcal{O}_1 \approx^{1_{\mathcal{G}}}_{\Phi^{\mu}(\Lambda)} \mathcal{O}_2$.

Assume that $\mathcal{O}_1 \approx^{\mu}_{\Lambda} \mathcal{O}_2$, i.e. for any $\varphi \in \mathsf{Sen}^{\mathcal{L}}(\Lambda)$ the following holds:

$$\alpha^{\mu}_{\Lambda'}(\mathcal{O}_1) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda\cup\Lambda')} \alpha^{\mu}_{\Lambda}(\varphi) \quad \text{iff} \quad \alpha^{\mu}_{\Lambda'}(\mathcal{O}_2) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda\cup\Lambda')} \alpha^{\mu}_{\Lambda}(\varphi) .$$

We only show that $\mathcal{O}_2 \sqsubseteq^{1_{\mathcal{G}}}_{\Phi^{\mu}(\Lambda)} \mathcal{O}_1$, as the opposite entailment is shown by replacing with each other $\mathcal{O}_1$ and $\mathcal{O}_2$. Suppose $\alpha^{\mu}_{\Lambda'}(\mathcal{O}_1) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda\cup\Lambda')} \psi$, for some $\psi \in \mathsf{Sen}^{\mathcal{G}}(\Phi^{\mu}(\Lambda))$. Let $\mathcal{M} \in |\mathsf{Mod}^{\mathcal{G}}(\Phi^{\mu}(\Lambda\cup\Lambda'))|$, such that $\mathcal{M} \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda')} \alpha^{\mu}_{\Lambda'}(\mathcal{O}_2)$. This implies $\beta^{\mu}_{\Lambda'}(\mathcal{M}) \models^{\mathcal{L}}_{\Lambda'} \mathcal{O}_2$, since signature morphism σ is the identity we get $\beta^{\mu}_{\Lambda'}(\mathcal{M}) \models^{\mathcal{L}}_{\Lambda'} (\mathcal{O}_2)^{*}$. By the assumption $\mathcal{O}_1 \approx^{\mu}_{\Lambda'} \mathcal{O}_2$ we get $\beta^{\mu}_{\Lambda'}(\mathcal{M}) \models^{\mathcal{L}}_{\Lambda'} (\mathcal{O}_1)^{*}$. This implies $\mathcal{M} \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda')} \alpha^{\mu}_{\Lambda'}((\mathcal{O}_1)^{*})$. By the conservativity of μ we have that

$$\mathsf{Mod}^{\mathcal{G}}(\alpha^{\mu}_{\Lambda'}((\mathcal{O}_1)^{*}) \subseteq \mathsf{Mod}^{\mathcal{G}}((\alpha^{\mu}_{\Lambda'}(\mathcal{O}_1))^{*}) ,$$

and since $\psi \in (\alpha^{\mu}_{\Lambda'}(\mathcal{O}_1))^{*}$ we get that $\mathcal{M} \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)} \psi$. Thus $\mathcal{O}_2 \sqsubseteq^{1_{\mathcal{G}}}_{\Phi^{\mu}(\Lambda)} \mathcal{O}_1$, as required.

"$\Leftarrow$". Assume that for any Λ we have $\mathcal{O}_1 \approx^{\mu}_{\Lambda} \mathcal{O}_2$ iff $\mathcal{O}_1 \approx^{1_{\mathcal{G}}}_{\Lambda} \mathcal{O}_2$.

Suppose that μ is not conservative. Then there is a signature morphism $\Lambda \xrightarrow{\sigma} \Lambda'$ in $\mathcal{L}$, such that for some ontology $\mathcal{O} \subseteq \mathsf{Sen}^{\mathcal{L}}(\Lambda')$ we have

$$\mathsf{Mod}^{\mathcal{G}}(\alpha^{\mu}_{\Lambda}(\mathcal{O}_{\sigma})) \not\subseteq \mathsf{Mod}^{\mathcal{G}}((\alpha^{\mu}_{\Lambda'}(\mathcal{O}))_{\Phi^{\mu}(\sigma)}) ,$$

i.e. there is a model $\mathcal{M} \in \mathsf{Mod}^{\mathcal{G}}(\Phi^{\mu}(\Lambda))$, such that $\mathcal{M} \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)} \alpha^{\mu}_{\Lambda}(\mathcal{O}_{\sigma})$ but $\mathcal{M} \not\models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)} (\alpha^{\mu}_{\Lambda'}(\mathcal{O}))_{\Phi^{\mu}(\sigma)}$. Thus for some $\psi \in \mathsf{Sen}^{\mathcal{G}}(\Phi^{\mu}(\Lambda))$ we have

$$\alpha^{\mu}_{\Lambda'}(\mathcal{O}) \models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda')} \mathsf{Sen}\Phi^{\mu}(\sigma)(\psi) \quad \text{but} \quad \alpha^{\mu}_{\Lambda}(\mathcal{O}_{\sigma}) \not\models^{\mathcal{G}}_{\Phi^{\mu}(\Lambda)} \psi .$$

Now, let $\mathcal{O}_1 = \mathcal{O}$ and $\mathcal{O}_2 = \mathrm{Sen}^{\mathcal{L}}(\sigma)(\mathcal{O}_\sigma)$. Then by Lemma 3.2.31 we get $\mathcal{O}_1 \approx^{1_\mathcal{G}}_{\Phi^\mu(\Lambda)} \mathcal{O}_2$ but $\mathcal{O}_1 \not\approx^\mu_\Lambda \mathcal{O}_2$, which is in contradiction to our assumption. □

3.3 Robustness properties and Craig interpolation property

In this section we consider how ontologies can safely be combined. If we have frameworks μ_1 and μ_2 over the same query basis η, then a Λ_1-ontology $\mathcal{O}_1$ for μ_1 can be combined with a Λ_2-ontology $\mathcal{O}_2$ for μ_2 by taking the union of $\alpha^{\mu_1}(\mathcal{O}_1)$ and $\alpha^{\mu_2}(\mathcal{O}_2)$ in $\mathcal{G}$, even if the ontology languages of μ_1 and μ_2 are different. This justifies using a notation for union for ontologies, so we can write, for example, $\mathcal{O}_1 \cup \mathcal{O}_2 \sqsubseteq^\eta_\Sigma \mathcal{O}_1$.

3.3.1 Robustness in frameworks

In the previous section, we studied some properties of $\approx^\eta_\Sigma$, but we are mainly interested in determining what robustness properties it has, i.e., how safely ontologies can be combined. We introduce three types of robustness properties in the framework setting.

Definition 3.3.1. *For any binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis η we say that $\mathfrak{F}$ is* ***robust under:***

- ***vocabulary extension*** *if for all signatures Λ_1 in $Sig^{\mathcal{L}_1}$, Λ_2 in $Sig^{\mathcal{L}_2}$, Σ, Σ' in $Sig^{\mathcal{Q}}$, such that $\Phi^\eta(\Sigma') \cap (\Phi^{\mu_1}(\Lambda_1) \cup \Phi^{\mu_2}(\Lambda_2)) \subseteq \Phi^\eta(\Sigma)$, all ontologies $\mathcal{O}_1 \subseteq Sen^{\mathcal{L}_1}(\Lambda_1)$ and $\mathcal{O}_2 \subseteq Sen^{\mathcal{L}_2}(\Lambda_2)$, the following holds:*

$$\mathcal{O}_1 \sqsubseteq^\eta_\Sigma \mathcal{O}_2 \quad \textit{implies} \quad \mathcal{O}_1 \sqsubseteq^\eta_{\Sigma'} \mathcal{O}_2,$$

- ***joins*** *if for all signatures Λ_1 in $Sig^{\mathcal{L}_1}$, Λ_2 in $Sig^{\mathcal{L}_2}$ and Σ in $Sig^{\mathcal{Q}}$, such that $\Phi^{\mu_1}(\Lambda_1) \cap \Phi^{\mu_2}(\Lambda_2) \subseteq \Phi^\eta(\Sigma)$ and all ontologies $\mathcal{O}_1 \subseteq Sen^{\mathcal{L}_1}(\Lambda_1)$ and $\mathcal{O}_2 \subseteq Sen^{\mathcal{L}_2}(\Lambda_2)$, the following holds for $i = 1, 2$:*

$$\mathcal{O}_1 \approx^\eta_\Sigma \mathcal{O}_2 \quad \textit{implies} \quad \mathcal{O}_i \approx^\eta_\Sigma \mathcal{O}_1 \cup \mathcal{O}_2,$$

- ***replacement in framework*** *$\mu : \mathcal{L} \to \mathcal{G}$ if for all signatures Λ_1 in $Sig^{\mathcal{L}_1}$, Λ_2 in $Sig^{\mathcal{L}_2}$, Λ in $Sig^{\mathcal{L}}$ and Σ in $Sig^{\mathcal{Q}}$, such that $\Phi^\mu(\Lambda) \cap (\Phi^{\mu_1}(\Lambda_1) \cup \Phi^{\mu_2}(\Lambda_2)) \subseteq \Phi^\eta(\Sigma)$, for all ontologies $\mathcal{O}_1 \subseteq Sen^{\mathcal{L}_1}(\Lambda_1), \mathcal{O}_2 \subseteq Sen^{\mathcal{L}_2}(\Lambda_2), \mathcal{O} \subseteq Sen^{\mathcal{L}}(\Lambda)$, the following holds:*

$$\mathcal{O}_1 \sqsubseteq^\eta_\Sigma \mathcal{O}_2 \quad \textit{implies} \quad \mathcal{O}_1 \cup \mathcal{O} \sqsubseteq^\eta_\Sigma \mathcal{O}_2 \cup \mathcal{O}\,.$$

We briefly discuss the intuitions behind those three types of robustness.

Robustness under vocabulary extension assures us that we can extend signature Σ with fresh symbols, which do not occur in $\mathcal{O}_1$ nor $\mathcal{O}_2$ and this extension has no impact on Σ-inseparability of $\mathcal{O}_1$ and $\mathcal{O}_2$.

Robustness under joins. We will use a typical example to present the importance of this type of robustness. First we introduce a proposition which is a consequence of robustness under joins:

Proposition 3.3.2. *For binary frameworks (μ, μ_1), (μ, μ_2), over η, both robust under joins, given Λ-ontology $\mathcal{O}$ for μ, Λ_1-ontology $\mathcal{O}_1$ for μ_1, and Λ_2-ontology $\mathcal{O}_2$ for μ_2, with*

$$\Phi^{\mu}(\Lambda) \cap \Phi^{\mu_1}(\Lambda_1) \subseteq \Phi^{\eta}(\Sigma) ,$$

and

$$\Phi^{\mu}(\Lambda) \cap \Phi^{\mu_2}(\Lambda_2) \subseteq \Phi^{\eta}(\Sigma) ,$$

where $\Sigma \in Sig^{\mathcal{Q}}$: if $\mathcal{O} \cup \mathcal{O}_i$ is a conservative extension of $\mathcal{O}$ (for $i = 1, 2$), then also $\mathcal{O} \cup \mathcal{O}_1 \cup \mathcal{O}_2$ is a conservative extension of $\mathcal{O}$.

Proof: Let $\mathcal{O}$, $\mathcal{O}_1$ and $\mathcal{O}_2$ be as in the statement of the proposition. Then

$$\mathcal{O} \cup \mathcal{O}_1 \approx^{\eta}_{\Sigma} \mathcal{O} \approx^{\eta}_{\Sigma} \mathcal{O} \cup \mathcal{O}_2 ,$$

and therefore by robustness under joins, $\mathcal{O} \approx^{\eta}_{\Sigma} \mathcal{O} \cup \mathcal{O}_1 \approx^{\eta}_{\Sigma} \mathcal{O} \cup \mathcal{O}_1 \cup \mathcal{O}_2$. □

For example, suppose that two groups of ontology designers independently refine an ontology $\mathcal{O} \subseteq \mathsf{Sen}(\Lambda)$ by creating their own set of axioms, say $\mathcal{O}_1 \subseteq \mathsf{Sen}(\Lambda_1)$ and $\mathcal{O}_2 \subseteq \mathsf{Sen}(\Lambda_2)$ respectively. Both teams ensure that their ontologies are conservative extensions of $\mathcal{O}$. Assume that these groups decide to merge their ontologies at some point, they would like $\mathcal{O} \cup \mathcal{O}_1 \cup \mathcal{O}_2$ to be a conservative extension of $\mathcal{O}$, but the fact that $\mathcal{O}_1$ and $\mathcal{O}_2$ are conservative extensions of $\mathcal{O}$ is not sufficient to guarantee that. Robustness under joins and the above proposition give a sufficient condition.

Robustness under replacement. This allows modules of existing ontologies to be reused in new applications. For instance, assume that a group of ontology designers is developing an ontology which is supposed to use terminology over Σ. They know that there is another ontology $\mathcal{O}'$, which already defines symbols of Σ, so instead of creating this part from scratch they would prefer to reuse $\mathcal{O}'$. However, instead of importing whole $\mathcal{O}'$, it would be more efficient to import a Σ-module $\mathcal{O}_{\Sigma}$ of $\mathcal{O}'$. If the framework $\mathfrak{F}$ used for answering queries is robust under replacement then from $\mathcal{O}_{\Sigma} \approx^{\eta}_{\Sigma} \mathcal{O}'$, it follows that $\mathcal{O} \cup \mathcal{O}_{\Sigma} \approx^{\eta}_{\Sigma} \mathcal{O} \cup \mathcal{O}'$. Therefore importing $\mathcal{O}_{\Sigma}$ instead $\mathcal{O}'$ still gives the same consequences.

3.3.2 Robustness properties and interpolation

Now we study correlations of different types of robustness and interpolation. The following proposition shows that weak interpolation in $\mathcal{G}$ implies robustness under vocabulary extension.

Proposition 3.3.3. *Let $\mathfrak{F} = (\mu_1, \mu_2)$ be a binary framework over query basis $\eta : \mathcal{Q} \to \mathcal{G}$, with η conservative and such that for every signature Σ in $Sig^{\mathcal{Q}}$, β^{η}_{Σ} is surjective. Moreover, let there be a comorphism $\rho : \mathcal{L}_2 \to \mathcal{Q}$ such that $\mu_2 = \rho; \eta$. If $\mathcal{G}$ has weak interpolation, then $\mathfrak{F}$ is robust under vocabulary extension.*

Proof: Let $\mathcal{O}_1$ be a Λ_1-ontology for μ_1 and $\mathcal{O}_2$ be a Λ_2-ontology for μ_2 and let Σ, Σ' be $\mathsf{Sig}^{\mathcal{Q}}$ signatures such that $\Phi^{\eta}(\Sigma') \cap (\Phi^{\mu_1}(\Lambda_1) \cup \Phi^{\mu_2}(\Lambda_2)) \subseteq \Phi^{\eta}(\Sigma)$. Assume $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2$. Let

$\varphi \in \mathrm{Sen}^{\mathcal{Q}}(\Sigma')$ and suppose $\mathcal{O}_2 \models^{\eta}_{\Sigma'} \varphi$, i.e.,

$$\alpha^{\mu_2}_{\Lambda_2}(\mathcal{O}_2) \models^{\mathcal{G}}_{\Phi^{\mu_2}(\Lambda_2)\cup\Phi^{\eta}(\Sigma')} \alpha^{\eta}_{\Sigma'}(\varphi) .$$

We need to show that $\mathcal{O}_1 \models^{\eta}_{\Sigma'} \varphi$.

First note that $\alpha^{\rho}_{\Lambda_2}(\mathcal{O}_2) \models^{\mathcal{Q}}_{\Phi^{\rho}(\Lambda_2)\cup\Sigma'} \varphi$ by Proposition 3.2.12. If $\mathcal{G}$ has weak interpolation, then from the results of [3] we get that $\mathcal{Q}$ also has. Thus we can find $I \subseteq \mathrm{Sen}^{\mathcal{Q}}(\Phi^{\rho_2}(\Lambda_2) \cap \Sigma') \subseteq \mathrm{Sen}^{\mathcal{Q}}(\Sigma)$ such that

$$\alpha^{\rho_2}_{\Lambda_2}(\mathcal{O}_2) \models^{\mathcal{Q}}_{\Phi^{\rho_2}(\Lambda_2)} I \quad \text{and} \quad I \models^{\mathcal{Q}}_{\Sigma'} \varphi .$$

This gives us

$$\alpha^{\eta}_{\Phi^{\rho_2}(\Lambda_2)}(\alpha^{\rho_2}_{\Lambda_2}(\mathcal{O}_2)) \models^{\mathcal{G}}_{\Phi^{\eta}(\Phi^{\rho_2}(\Lambda_2))} \alpha^{\eta}_{\Phi^{\rho_2}(\Lambda_2)}(I)$$

i.e., $\alpha^{\mu_2}_{\Lambda_2}(\mathcal{O}_2) \models^{\mathcal{G}}_{\Phi^{\mu_2}(\Lambda_2)} \alpha^{\eta}_{\Phi^{\rho_2}(\Lambda_2)}(I)$, and also $\alpha^{\eta}_{\Sigma'}(I) \models^{\mathcal{G}}_{\Phi^{\eta}(\Sigma')} \alpha^{\eta}_{\Sigma'}(\varphi)$. The former together with $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2$ and the fact that $I \subseteq \mathrm{Sen}^{\mathcal{Q}}(\Sigma)$ gives us

$$\alpha^{\mu_1}_{\Lambda_1}(\mathcal{O}_1) \models^{\mathcal{G}}_{\Phi^{\mu_1}(\Lambda_1)\cup\Phi^{\mu_2}(\Lambda_2)} \alpha^{\eta}_{\Phi^{\rho_2}(\Lambda_2)}(I) .$$

This gives us $\alpha^{\mu_1}_{\Lambda_1}(\mathcal{O}_1) \models^{\mathcal{G}}_{\Phi^{\mu_1}(\Lambda_1)\cup\Phi^{\eta}(\Sigma')} \alpha^{\eta}_{\Sigma'}(\varphi)$ as desired. □

The following can be understood as a partial converse of Proposition 3.3.3.

Proposition 3.3.4. *If framework $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$ over query basis $1_{\mathcal{G}}$ is robust under vocabulary extension, then $\mathcal{G}$ has weak interpolation.*

Proof: Let Λ, Λ' be signatures in $\mathrm{Sig}^{\mathcal{G}}$ and set $\Lambda_0 = \Lambda \cap \Lambda'$. Let $\mathcal{O} \subseteq \mathrm{Sen}^{\mathcal{G}}(\Lambda)$ and $\varphi \in \mathrm{Sen}^{\mathcal{G}}(\Lambda')$, and assume $\mathcal{O} \models^{\mathcal{G}}_{\Lambda\cup\Lambda'} \varphi$.

Let $\mathcal{O}' = \{\psi \in \mathrm{Sen}^{\mathcal{G}}(\Lambda_0) \mid \mathcal{O} \models^{\mathcal{G}}_{\Lambda} \psi\}$, then $\mathcal{O} \approx^{1_{\mathcal{G}}}_{\Lambda_0} \mathcal{O}'$ and $\mathcal{O} \models^{\mathcal{G}}_{\Lambda_0} \mathcal{O}'$. By robustness under vocabulary extension we get $\mathcal{O} \approx^{1_{\mathcal{G}}}_{\Lambda'} \mathcal{O}'$ and $\mathcal{O} \approx^{1_{\mathcal{G}}}_{\Lambda} \mathcal{O}'$. From $\mathcal{O} \models^{\mathcal{G}}_{\Lambda\cup\Lambda'} \varphi$ we get $\mathcal{O}' \models^{\mathcal{G}}_{\Lambda\cup\Lambda'} \varphi$. Thus $\mathcal{O}'$ is the required interpolant. □

Further results require institutions that have Boolean operators:

Definition 3.3.5. *Let $\varphi, \psi \in \mathrm{Sen}^{\mathcal{I}}(\Sigma)$ in some institution $\mathcal{I}$. A* **conjunction of** φ **and** ψ *is a sentence $\varphi\wedge\psi \in \mathrm{Sen}^{\mathcal{I}}(\Sigma)$ such that for every Σ-model M,*

$$M \models_{\Sigma} \varphi\wedge\psi \quad \textit{iff} \quad M \models_{\Sigma} \varphi \ \textit{and} \ M \models_{\Sigma} \psi .$$

The **negation of** φ *is a sentence $\neg\varphi \in \mathrm{Sen}^{\mathcal{I}}(\Sigma)$ such that*

$$M \models_{\Sigma} \neg\varphi \quad \textit{iff} \quad M \not\models_{\Sigma} \varphi .$$

We say $\mathcal{I}$ **has negation** *if every sentence has a negation, and is* **closed under Boolean operators** *if it has negation, and every pair of sentences has a conjunction. If $\mathcal{I}$ is closed under Boolean operators, and $\mathcal{O}$ is a finite subset of $\mathrm{Sen}^{\mathcal{I}}(\Sigma)$, we write $\bigwedge(\mathcal{O})$ for the conjunction of all the sentences in $\mathcal{O}$.*

Note that, as usual, the other logical connectives can be defined in terms of negation and conjunction; in particular, implication $\varphi \Rightarrow \psi$ is $\neg(\varphi \wedge \neg\psi)$, and it is straightforward to see that a deduction theorem holds for implications:

$$\mathcal{O} \models_{\Sigma} \varphi \Rightarrow \psi \quad \text{iff} \quad \mathcal{O} \cup \{\varphi\} \models_{\Sigma} \psi \,.$$

Note also that for any comporphism $\mu : \mathcal{I} \to \mathcal{I}'$, if $\varphi \wedge \psi$ is a conjunction in $\mathcal{I}$, then $\alpha^{\mu}_{\Sigma}(\varphi \wedge \psi)$ is a conjunction in $\mathcal{I}'$ of $\alpha^{\mu}_{\Sigma}(\varphi)$ and $\alpha^{\mu}_{\Sigma}(\psi)$.

The following proposition shows a correlation between robustness under joins of $\mathfrak{F}$ and interpolation.

Proposition 3.3.6. *For any binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis η, s.t. there is a comorphism $\mathcal{L}_2 \overset{\rho_3}{\to} \mathcal{Q}$ and $\mathcal{G}$ is closed under Boolean operators, if $\mathcal{G}$ has weak interpolation, then $\mathfrak{F}$ is robust under joins.*

Proof: *Let $\Lambda_1 \in |Sig^{\mathcal{L}_1}|$, $\Lambda_2 \in |Sig^{\mathcal{L}_2}|$, $\Sigma \in |Sig^{\mathcal{Q}}|$ and O_1 be a Λ_1-ontology for μ_1, O_2 be a Λ_2-ontology for μ_2 and $\varphi \in Sen^{\mathcal{Q}}(\Sigma)$.*

Suppose that $\mathcal{G}$ has weak interpolation, $O_1 \approx^{\eta}_{\Sigma} O_2$ and $\Phi^{\mu_1}(\Lambda_1) \cap \Phi^{\mu_2}(\Lambda_2) \hookrightarrow \Phi^{\eta}(\Sigma)$. Assume that

$$\alpha^{\mu_1}_{\Lambda_1}(O_1) \cup \alpha^{\mu_2}_{\Lambda_2}(O_2) \models^{\mathcal{G}}_{\Phi^{\mu_1}(\Lambda_1)\cup\Phi^{\mu_2}(\Lambda_2)\cup\Phi^{\eta}(\Sigma)} \alpha^{\eta}_{\Sigma}(\varphi).$$

Then

$$\alpha^{\mu_1}_{\Lambda_1}(O_1) \models^{\mathcal{G}}_{\Phi^{\mu_1}(\Lambda_1)\cup\Phi^{\mu_2}(\Lambda_2)\cup\Phi^{\eta}(\Sigma)} \alpha^{\mu_2}_{\Lambda_2}(\bigwedge O_2) \Rightarrow \alpha^{\eta}_{\Sigma}(\varphi). \tag{3.1}$$

Take an interpolant I for (3.1), it is straightforward to see that $I \subseteq Sen^{\mathcal{Q}}(\Sigma)$. Since institution $\mathcal{G}$ is closed under Boolean operators and $O_1 \approx^{\eta}_{\Sigma} O_2$, we obtain

$$\alpha^{\mu_2}_{\Lambda_2}(O_2) \models^{\mathcal{G}}_{\Phi^{\mu_2}(\Lambda_2)\cup\Phi^{\eta}(\Sigma)} I.$$

This entails

$$\alpha^{\mu_2}_{\Lambda_2}(O_2) \models^{\mathcal{G}}_{\Phi^{\mu_2}(\Lambda_2)\cup\Phi^{\eta}(\Sigma)} \alpha^{\mu_2}_{\Lambda_2}(\bigwedge O_2) \Rightarrow \alpha^{\eta}_{\Sigma}(\varphi),$$

i.e. $\alpha^{\mu_2}_{\Lambda_2}(O_2) \models^{\mathcal{G}}_{\Phi^{\mu_2}(\Lambda_2)\cup\Phi^{\eta}(\Sigma)} \alpha^{\eta}_{\Sigma}(\varphi)$. □

The following proposition shows how framework $\mathfrak{F}'$, received from framework $\mathfrak{F}$ by attaching comorphisms to its ontology languages, inherits robustness properties after $\mathfrak{F}$.

Proposition 3.3.7. *For any binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$ with a comorphisms $\zeta_1 : \mathcal{L}'_1 \to \mathcal{L}_1$ $\zeta_2 : \mathcal{L}'_2 \to \mathcal{L}_2$ attached and binary framework $\mathfrak{F}' = (\mu'_1, \mu'_2)$ over η, where $\mu'_i = \zeta_i; \mu_i$ for $i = 1,2$ we have that robustness of any type of $\mathfrak{F}$ implies the same type of robustness of $\mathfrak{F}'$.*

Proof: The proof is given by case by case consideration.

Let μ, μ' and η be as in the theorem.

1. Robustness under vocabulary extension.

   Let $\Lambda_1 \in |\mathsf{Sig}^{\mathcal{L}'_1}|$, $\Lambda_2 \in |\mathsf{Sig}^{\mathcal{L}'_2}|$ and $\Sigma, \Sigma' \in |\mathsf{Sig}^{\mathcal{Q}}|$, such that $\Sigma \hookrightarrow \Sigma'$ and $\Phi^{\eta}(\Sigma') \cap (\Phi^{\mu_1}(\Phi^{\zeta_1}(\Lambda_1)) \cup \Phi^{\mu_2}(\Phi^{\zeta_2}(\Lambda_2))) \subseteq \Phi^{\eta}(\Sigma)$, where $\mathcal{O}_1 \subseteq \mathsf{Sen}^{\mathcal{L}'_1}(\Lambda_1)$ and $\mathcal{O}_2 \subseteq \mathsf{Sen}^{\mathcal{L}'_2}(\Lambda_2)$.

   Assume that $\mathfrak{F}$ is robust under vocabulary extension, and $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2$.

   By Proposition 3.2.26 we get $\alpha^{\zeta_1}_{\Lambda}(\mathcal{O}_1) \sqsubseteq^{\eta}_{\Sigma} \alpha^{\zeta_2}_{\Lambda}(\mathcal{O}_2)$. Since $\mathfrak{F}$ is robust under vocabulary extension we get $\alpha^{\zeta_1}_{\Lambda}(\mathcal{O}_1) \sqsubseteq^{\eta}_{\Sigma'} \alpha^{\zeta_1}_{\Lambda}(\mathcal{O}_2)$. Again by Proposition 3.2.26 we receive $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma'} \mathcal{O}_2$, as desired. Thus $\mathfrak{F}'$ is robust under vocabulary extension.

2. Robustness under joins.

   Let $\Lambda_1 \in |\mathsf{Sig}^{\mathcal{L}'_1}|$, $\Lambda_2 \in |\mathsf{Sig}^{\mathcal{L}'_2}|$ and $\Sigma \in |\mathsf{Sig}^{\mathcal{Q}}|$, such that $\Phi^{\mu_1}(\Phi^{\zeta_1}(\Lambda_1)) \cap \Phi^{\mu_2}(\Phi^{\zeta_2}(\Lambda_2)) \subseteq \Phi^{\eta}(\Sigma)$, and $\mathcal{O}_1 \subseteq \mathsf{Sen}^{\mathcal{L}'_1}(\Lambda_1)$ and $\mathcal{O}_2 \subseteq \mathsf{Sen}^{\mathcal{L}'_2}(\Lambda_2)$.

   Assume that $\mathfrak{F}$ is robust under joins, and $\mathcal{O}_1 \approx^{\eta}_{\Sigma} \mathcal{O}_2$.

   By Proposition 3.2.26 we get $\alpha^{\zeta_1}_{\Lambda_1}(\mathcal{O}_1) \approx^{\eta}_{\Sigma} \alpha^{\zeta_2}_{\Lambda_2}(\mathcal{O}_2)$. Since $\mathfrak{F}$ is robust under joins we get $\alpha^{\zeta_i}_{\Lambda_i}(\mathcal{O}_i) \approx^{\eta}_{\Sigma} \alpha^{\zeta_1}_{\Lambda_1}(\mathcal{O}_1) \cup \alpha^{\zeta_2}_{\Lambda_2}(\mathcal{O}_2)$ for $i = 1, 2$. Again by Proposition 3.2.26 we receive $\mathcal{O}_i \approx^{\eta}_{\Sigma} \mathcal{O}_1 \cup \mathcal{O}_2$, as desired. Thus $\mathfrak{F}'$ is robust under joins.

3. Robustness under replacement. Let $\Lambda_1 \in \mathsf{Sig}^{\mathcal{L}'_1}$, $\Lambda_2 \in \mathsf{Sig}^{\mathcal{L}'_2}$, $\Lambda \in \mathsf{Sig}^{\mathcal{L}'}$ and Σ in $\mathsf{Sig}^{\mathcal{Q}}$, such that

   $$\Phi^{\mu}(\Phi^{\zeta}(\Lambda)) \cap (\Phi^{\mu_1}(\Phi^{\zeta_1}(\Lambda_1)) \cup \Phi^{\mu_2}(\Phi^{\zeta_2}(\Lambda_2))) \subseteq \Phi^{\eta}(\Sigma) ,$$

   for all ontologies $\mathcal{O}_1 \subseteq \mathsf{Sen}^{\mathcal{L}'_1}(\Lambda_1), \mathcal{O}_2 \subseteq \mathsf{Sen}^{\mathcal{L}'_2}(\Lambda_2), \mathcal{O} \subseteq \mathsf{Sen}^{\mathcal{L}'}(\Lambda)$.

   Assume that $\mathfrak{F}$ is robust under replacement, and $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2$.

   By Proposition 3.2.26 we get $\alpha^{\zeta_1}_{\Lambda_1}(\mathcal{O}_1) \sqsubseteq^{\eta}_{\Sigma} \alpha^{\zeta_2}_{\Lambda_2}(\mathcal{O}_2)$. Since $\mathfrak{F}$ is robust under replacement we get $\alpha^{\zeta_1}_{\Lambda_1}(\mathcal{O}_1) \cup \alpha^{\zeta}_{\Lambda}(\mathcal{O}) \sqsubseteq^{\eta}_{\Sigma} \alpha^{\zeta_2}_{\Lambda_2}(\mathcal{O}_2) \cup \alpha^{\eta}_{\Lambda}(\mathcal{O})$. Again by Proposition 3.2.26 we receive $\mathcal{O}_1 \cup \mathcal{O} \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2 \cup \mathcal{O}$, as desired, so $\mathfrak{F}'$ is robust under replacement. □

Now we show how interpolation in global institution closed under Boolean operators implies robustness properties.

Proposition 3.3.8. *Let $\mathcal{G}$ be an institution closed under Boolean operators and with interpolation, then any binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$ is robust under vocabulary extensions, joins, and under replacement in any framework $\mu : \mathcal{L} \to \mathcal{G}$ over $1_{\mathcal{G}}$ for finite ontologies.*

Proof: Robustness under vocabulary extension is a consequence of Proposition 3.3.3 and the fact that $\mathcal{G}$ has interpolation.

Robustness under joins is a consequence of Proposition 3.3.6 and the fact that $\mathcal{G}$ has interpolation and is closed under Boolean operators.

To prove robustness under replacement in μ, let $\mathcal{O}_i$ be a Λ_i-ontology for μ_i (for $i = 1, 2$) let $\mathcal{O}$ be a Λ-ontology for μ, and let Σ in $\mathsf{Sig}^{\mathcal{G}}$ be such that

$$\Phi^{\mu}(\Lambda) \cap (\Phi^{\mu_1}(\Lambda_1) \cup \Phi^{\mu_2}(\Lambda_2)) \subseteq \Sigma .$$

Assume $\mathcal{O}_1 \sqsubseteq_{\Sigma}^{1_{\mathcal{G}}} \mathcal{O}_2$ and let $\varphi \in \mathrm{Sen}^{\mathcal{G}}(\Sigma)$ be such that $\mathcal{O} \cup \mathcal{O}_2 \models_{\Sigma}^{1_{\mathcal{G}}} \varphi$; we need to show $\mathcal{O} \cup \mathcal{O}_1 \models_{\Sigma}^{1_{\mathcal{G}}} \varphi$.

By robustness under vocabulary extension, we have $\mathcal{O}_1 \sqsubseteq_{\Phi^{\mu}(\Lambda)}^{1_{\mathcal{G}}} \mathcal{O}_2$ and therefore $\mathcal{O}_1 \sqsubseteq_{\Phi^{\mu}(\Lambda)\cup\Sigma}^{1_{\mathcal{G}}} \mathcal{O}_2$. From $\mathcal{O} \cup \mathcal{O}_2 \models_{\Sigma}^{1_{\mathcal{G}}} \varphi$ it follows that

$$\mathcal{O}_2 \models_{\Phi^{\mu}(\Lambda)\cup\Sigma}^{1_{\mathcal{G}}} \bigwedge(\alpha_{\Lambda}^{\mu}(\mathcal{O})) \Rightarrow \varphi$$

and therefore $\mathcal{O}_1 \models_{\Phi^{\mu}(\Lambda)\cup\Sigma}^{1_{\mathcal{G}}} \bigwedge(\alpha_{\Lambda}^{\mu}(\mathcal{O})) \Rightarrow \varphi$, giving $\mathcal{O} \cup \mathcal{O}_1 \models_{\Sigma}^{1_{\mathcal{G}}} \varphi$ as desired. □

The above proposition implies the corollary stating that any framework over query basis $1_{FOL} : FOL \to FOL$ is robust under vocabulary extensions, joins, and under replacement for finite ontologies.

Corollary 3.3.9. *Any binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis $1_{FOL} : FOL \to FOL$ is robust under vocabulary extensions, joins, and under replacement in any $\mu : \mathcal{L} \to FOL$ over 1_{FOL} for finite ontologies.*

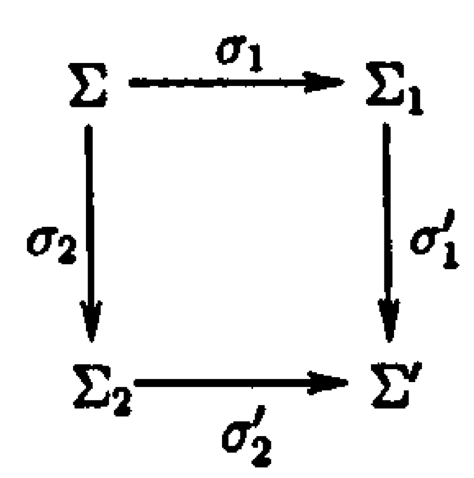
The following corollary is a direct consequence of Corollary 3.3.9 and Lemma 3.2.29.

Corollary 3.3.10. *Any binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis $\eta : FOL \to \mathcal{G}$, with comorphisms $\rho_1 : \mathcal{L}_1 \to FOL$ and $\rho_2 : \mathcal{L}_2 \to FOL$, is robust under vocabulary extensions, joins, and under replacement in any $\mu : \mathcal{L} \to \mathcal{G}$, such that there is comorphism $\rho : \mathcal{L} \to FOL$, for finite ontologies.*

The following proposition shows a correlation between interpolation and robustness under joins for any framework $\mu : \mathcal{L} \to \mathcal{G}$ over itself as query basis, where $\mathcal{L}$ is an institution closed under Boolean operators.

Proposition 3.3.11. *Let $\mathcal{L}$ be an institution closed under Boolean operators, let $\mu : \mathcal{L} \to \mathcal{G}$ be a framework over itself as query basis, such that μ is robust under joins for possibly infinite ontologies. Then $\mathcal{L}$ has interpolation.*

Proof: Let $\Sigma, \Sigma', \Sigma_1, \Sigma_2 \in |\mathrm{Sig}^{\mathcal{L}}|$, such that the square



commutes. Let $\mathcal{O} \subseteq \mathrm{Sen}^{\mathcal{L}}(\Sigma_1)$ and $\varphi \in \mathrm{Sen}^{\mathcal{L}}(\Sigma_2)$.

Assume $\alpha_{\Sigma'}^{\eta}(\sigma_1'(\mathcal{O})) \models_{\Phi^{\eta}(\Sigma')}^{\mathcal{G}} \alpha_{\Sigma'}^{\eta}(\sigma_2'(\varphi))$. Define

$$\mathcal{O}' = \{\chi \in \mathrm{Sen}^{\mathcal{L}}(\Sigma) \mid \alpha_{\Sigma_1}^{\eta}(\mathcal{O}) \models_{\Phi^{\eta}(\Sigma_1)}^{\mathcal{G}} \alpha_{\Sigma_1}^{\eta}(\sigma_1(\chi))\} \, .$$

We show that $\mathcal{O}'$ is an interpolant for $\alpha_{\Sigma'}^{\eta}(\sigma_1'(\mathcal{O})) \models_{\Phi^{\eta}(\Sigma')}^{\mathcal{G}} \alpha_{\Sigma'}^{\eta}(\sigma_2'(\varphi))$. Suppose not. Then $\alpha_{\Sigma_2}^{\eta}(\sigma_2(\mathcal{O}')) \cup \alpha_{\Sigma_2}^{\eta}(\neg\varphi)$ is satisfiable. Take a $\mathcal{G}$-model $\mathcal{M}$ satisfying $\alpha_{\Sigma_2}^{\eta}(\sigma_2(\mathcal{O}')) \cup \alpha_{\Sigma_2}^{\eta}(\neg\varphi)$. Let $\mathcal{O}_1 \subseteq \mathrm{Sen}^{\mathcal{L}}(\Sigma)$ be a set of all sentences $\psi \in \mathrm{Sen}^{\mathcal{L}}(\Sigma)$, such that $\mathcal{M} \models_{\Phi^{\eta}(\Sigma)}^{\mathcal{G}}$

$\alpha^{\eta}_{\Sigma}(\psi)$. Then both $\sigma_1(\mathcal{O}_1) \cup \mathcal{O}$ and $\sigma_2(\mathcal{O}_1) \cup \neg\varphi$ are Σ-conservative extensions of $\mathcal{O}_1$. By robustness under joins, $\sigma'_2(\sigma_2(\mathcal{O}_1)) \cup \sigma'_2(\mathcal{O}) \cup \sigma'_1(\neg\varphi)$ is a Σ-conservative extension of $\mathcal{O}_1$, thus it is consistent. Therefore $\alpha^{\eta}_{\Sigma_1}(\sigma'_1(\mathcal{O})) \not\models^{\mathcal{G}}_{\Phi^{\eta}(\Sigma')} \alpha^{\eta}_{\Sigma_2}(\sigma'_2(\varphi))$, but this is in contradiction to our main assumption. This implies that $\mathcal{O}'$ is an interpolant for $\alpha^{\eta}_{\Sigma'}(\sigma'_1(\mathcal{O})) \models^{\mathcal{G}}_{\Phi^{\eta}(\Sigma')} \alpha^{\eta}_{\Sigma'}(\sigma'_2(\varphi))$ as required. □

The following corollary is a direct consequences of Proposition 3.3.11.

Corollary 3.3.12. *Let $\mathcal{L}$ be an institution of any fragment of first-order logic closed under Boolean operators, let $\mu : \mathcal{L} \to \mathcal{G}$ be a framework over itself as query basis, such that μ is robust under joins for possibly infinite ontologies. Then $\mathcal{L}$ has interpolation.*

A particular case of Corollary 3.3.12 is the case where *FOL* is the global institution.

Chapter 4

Institutions with Individuals and Frameworks with ABoxes

4.1 Introduction

In the previous chapter we introduced frameworks as constructions allowing us to use an ontology to answer a query, even if they are formulated in different formalisms. In this chapter we restrict our attention to an abstract notion of description logics and investigate how we can work with them within the framework setting.

When working with description logics we can distinguish two main tasks: determining concept hierarchies and answering queries about individuals. For determining concept hierarchies we need an ontology only, but to answer queries about individuals we need an ontology and an ABox. In some sense the ontology fixes the terminology, whereas the ABox tells us what is known about individuals. The ontology helps us to infer new information about individuals with the use of concept hierarchies. One of the main issues is the interaction between the ontology and the ABox, as combining them may lead to potential difficulties. In particular, one is interested whether adding the ABox introduces new concept inclusions.

This chapter has two main parts. The first part of the chapter defines description logics in categorical setting and shows how concept hierarchies are introduced into the framework. We start with introducing a general definition of description logic which shows how any description logic is built over a variant of $\mathcal{CH}$ (we use the notion of slice category to formulate that definition). This definition allows us to treat description logics in a general and systematic way and is used to show how queries about concept hierarchies can be formulated in a framework. Second part of the chapter shows how ABoxes and queries about individuals can be introduced into the framework setting. As the first step we show how any description logic extends to description logic with individuals, this is necessary for introducing ABoxes. But to be able to translate the ABox together with the ontology into the global language in the framework we also show how a comorphism between description logics extends to a comorphism between description logics with individuals. This fact is then used to show that any framework built from description logics extends to a framework allowing for the use of individuals. We describe the relations between both structures and investigate the relations between consequence relations in both types of structures as well as relations between Σ-entailment between ontologies and Σ-entailment between ontologies in the presence of ABoxes.

We also discuss how introducing individuals may affect the consequence relation in an institution of description logic. We define notions of query conservativity and query expansion which tell us how an institution of description logic behaves when we introduce ABoxes. We also define the notions of consequence of an ABox, which eventually allows us to define the notion of concept interpolation, which 'splits' the consequence relation for assertions into two types of reasoning.

4.2 Description Logics in a Categorical Setting

To be able to talk about description logics in a general way we need to find and categorically present some properties shared by all of them. In our work we use $\mathcal{CH}$ (or rather a variant of it as we will justify below) as a benchmark to discriminate description logics from other logical systems. In general, we require that description logics are built over $\mathcal{CH}$, this is based on the fact that one of the main properties of description logics is the ability to build hierarchies of concepts and, as mentioned in Example 2.4.8, $\mathcal{CH}$ is a basic description logic and the specification of concept hierarchies is its main property. More specifically, we will define an institution $\mathcal{I}$ to be a description logic if there is a morphism from $\mathcal{I}$ to $\mathcal{CH}$. Intuitively, that morphism provides us with means to translate signatures of $\mathcal{I}$ to sets of $\mathcal{I}$-concepts in $\mathcal{CH}$, where we can build sentences which are subsumptions. These sentences can be translated (via the morphism) back to $\mathcal{I}$, which in turn provides us with models that are used for interpreting the translated $\mathcal{CH}$-sentences. We also know how to translate these models to $\mathcal{CH}$ in order to receive interpretative structures in $\mathcal{CH}$. This definition allows us to extract all the logics that are description logics and to treat them in a systematic way. In fact we are also able to systematically relate such description logics in a categorical setting. To this end we use the notion of a slice category introduced in Definition 2.3.10. We relate the notion of slice category to the definition of description logic, where we have a category of institutions *Inst*, an institution $\mathcal{CH}$ is the 'target' of the slice category and morphisms to $\mathcal{CH}$ are the objects of the slice category. In this way we say that description logics are objects in the slice category $Inst/\mathcal{CH}$. For instance consider $\mathcal{EL}$ with a morphism ec to $\mathcal{CH}$.

Example 4.2.1. *We show how components of morphism $\mathcal{EL} \overset{\text{ec}}{\to} \mathcal{CH}$ are constructed.*

- Ψ^{ec} *maps an $\mathcal{EL}$-signature (P,R) to the set $Con(P,R)$ of $\mathcal{EL}$-concepts.*
- *Natural transformation $\gamma^{\text{ec}}_{(P,R)}$. By definition $Sen^{\mathcal{CH}}(\Pi)$ is a set of sentences of the form $p \sqsubseteq q$, with $p,q \in \Pi$, so $Sen^{\mathcal{CH}}(\Psi^{\text{ec}}(P,R))$ is the set of subsumptions $C \sqsubseteq D$, where $C, D \in Con^{\mathcal{EL}}(P,R)$. Therefore, we may take $\gamma^{\text{ec}}_{(P,R)}$ to be the identity.*
- *Natural transformation $\delta^{\text{ec}}_{(P,R)}$. For any $\mathcal{M} \in |Mod^{\mathcal{EL}}(P,R)|$, we set $\delta^{\text{ec}}_{(P,R)}(\mathcal{M}) = \mathcal{M}$. This is due to the fact that (P,R)-models in $\mathcal{EL}$ are also interpretative structures for $\mathcal{EL}$-concepts built over (P,R), i.e. $Con^{\mathcal{EL}}(P,R)$.*

Now we show that the satisfaction condition holds for every signature $(P,R) \in |Sig^{\mathcal{EL}}|$, model $\mathcal{M} \in |Mod^{\mathcal{EL}}(P,R)|$ and concept inclusion $C \sqsubseteq D \in Sen^{\mathcal{CH}}(\Psi^{\text{ec}}(P,R))$:

$$\mathcal{M} \models^{\mathcal{EL}}_{(P,R)} \gamma^{\text{ec}}_{(P,R)}(C \sqsubseteq D) \text{ iff } \delta^{\text{ec}}_{(P,R)}(\mathcal{M}) \models^{\mathcal{CH}}_{\Psi^{\text{ec}}(P,R)} C \sqsubseteq D.$$

Proof:

$$\mathcal{M} \models^{\mathcal{EL}}_{(P,R)} \gamma^{\text{ec}}_{(P,R)}(C \sqsubseteq D)$$

iff

$$\mathcal{M} \models^{\mathcal{EL}}_{(P,R)} C \sqsubseteq D$$

iff

$$C^{\mathcal{M}} \subseteq D^{\mathcal{M}}$$

iff

$$C^{\delta^{\text{ec}}_{(P,R)}(\mathcal{M})} \subseteq D^{\delta^{\text{ec}}_{(P,R)}(\mathcal{M})}$$

iff

$$\delta^{\text{ec}}_{(P,R)}(\mathcal{M}) \models^{\mathcal{CH}}_{\Psi^{\text{ec}}(P,R)} C \sqsubseteq D$$

□

The composition of the morphism described in Example 4.2.1 with the morphism from Example 2.4.24 gives a morphism $\mathsf{e^+e}$; $\mathsf{ec} : \mathcal{EL}^+ \to \mathcal{CH}$, which tells us how $\mathcal{EL}^+$ is built over $\mathcal{CH}$. As it was already mentioned $\mathcal{EL}^+$ extends $\mathcal{EL}$ with role inclusion axioms but as we are unable to formulate hierarchies of roles in $\mathcal{CH}$ we can, without any harm, define morphism $\mathsf{e^+c} : \mathcal{EL}^+ \to \mathcal{CH}$ to be identical to ec.

Similarly, the composition of the morphism described in Example 4.2.1 with the morphism from Example 2.4.23 gives a morphism ae ; ec : $\mathcal{ALC} \to \mathcal{CH}$, which tells us how $\mathcal{ALC}$ is built over $\mathcal{CH}$: functor $\Psi^{\mathsf{ae;ec}}$ maps an $\mathcal{ALC}$-signature (P,R) to $\mathcal{CH}$-signature in two steps, first to $\mathcal{EL}$ signature (P,R) (as $\mathcal{EL}$ and $\mathcal{ALC}$ signatures are identical Ψ^{ae} is the identity), then this $\mathcal{EL}$ signature is mapped to $\mathcal{CH}$ as presented in Example 4.2.1. So effectively $\mathcal{ALC}$-signatures get mapped to the set $\mathsf{Con}(P,R)$ of $\mathcal{EL}$-concepts. The natural transformation $\gamma^{\mathsf{ae;ec}}_{(P,R)}$ is also a composition of the identity from Example 4.2.1 and the inclusion from Example 2.4.23. As models of $\mathcal{ALC}$ and $\mathcal{EL}$ are essentially the same we have $\delta^{\mathsf{ec}}_{(P,R)} = \delta^{\mathsf{ae;ec}}_{(P,R)}$.

The fact that $\Psi^{\mathsf{ae;ec}}$ allows us to build hierarchies of concepts in $\mathcal{CH}$ with use of $\mathcal{EL}$-concepts only clearly is not satisfactory. For that reason we formulate an alternative morphism directly from $\mathcal{ALC}$ to $\mathcal{CH}$, we call it ac. Morphism ac is built in the following way: functor Ψ^{ac} maps an $\mathcal{ALC}$-signature (P,R) to the set $\mathsf{Con}^{\mathcal{ALC}}(P,R)$ of $\mathcal{ALC}$-concepts. Natural transformation $\gamma^{\mathsf{ac}}_{(P,R)}$ is defined to be the identity, as $\mathsf{Sen}^{\mathcal{CH}}(\Psi^{\mathsf{ac}}(P,R))$ is the set of subsumptions $C \sqsubseteq D$, where $C, D \in \mathsf{Con}^{\mathcal{ALC}}(P,R)$. For any $\mathcal{M} \in |\mathsf{Mod}^{\mathcal{ALC}}(P,R)|$, we have that $\delta^{\mathsf{ac}}_{(P,R)}(\mathcal{M}) = \mathcal{M}$. Proof that the satisfaction condition holds is similar to that for morphism ec.

The composition of the morphisms described in Example 4.2.1 with the morphism from Example 2.4.27 gives a morphism fe ; ec : $FOL \to \mathcal{CH}$. Similarly as in the case for ae ; ec above, we have that FOL-signature is first translated into $\mathcal{EL}$ and then to $\mathcal{CH}$. So first the functor Ψ^{fe} translates an FOL-signature Π into an $\mathcal{EL}$-signature in the following way: $\Psi^{\mathsf{fe}}(\Pi) = (\Pi^{\{x_0\}}, \Pi^{\{x_0,x_1\}})$, where

$$\Pi^{\{x_0\}} = \{\varphi \in \mathsf{Sen}^{FOL}(\Pi) | \varphi \text{ contains exactly one free variable } x_0\}$$

and

$$\Pi^{\{x_0,x_1\}} = \{\varphi \in \mathsf{Sen}^{FOL}(\Pi) | \varphi \text{ contains exactly two free variables } x_0 \text{ and } x_1\}.$$

Then the functor Ψ^{ec} translates $\Psi^{\mathsf{fe}}(\Pi)$ into $\mathsf{Con}^{\mathcal{EL}}(\Psi^{\mathsf{fe}}(\Pi))$. Even though that means that we have only $\mathcal{EL}$ concepts available this is not problematic as $\Pi^{\{x_0\}}$ contains all the formulae with one free variable. $\Psi^{\mathsf{fe}}(\Pi)$ is translated into $\mathsf{Con}^{\mathcal{EL}}(\Psi^{\mathsf{fe}}(\Pi))$. In this case FOL-models over Π are first converted into $\mathcal{EL}$-models and then into $\mathcal{CH}$-models.

These examples illustrate our argument that any institution that is the source of a morphism to $\mathcal{CH}$ can be thought of as a 'description logic'. Specifically, given a morphism $\mu : \mathcal{I} \to \mathcal{CH}$, every signature Σ in $\mathcal{I}$ gives a set $\Psi^{\mu}(\Sigma)$ of concepts, while every $\mathcal{CH}$-sentence $c \sqsubseteq c'$ with $c, c' \in \Psi^{\mu}(\Sigma)$ can be translated to a Σ-sentence in $\mathcal{I}$, and every Σ-model M gives rise to an interpretative structure $\delta^{\mu}_{\Sigma}(M)$ that interprets concepts as subsets of a domain.

The satisfaction condition tells us that

$$M \models^{\mathcal{I}}_{\Sigma} \gamma^{\mu}_{\Sigma}(c \sqsubseteq c') \text{ iff } c^{\delta^{\mu}_{\Sigma}(M)} \subseteq c'^{\delta^{\mu}_{\Sigma}(M)},$$

where the sentence $\gamma^{\mu}_{\Sigma}(c \sqsubseteq c')$ states the concept c is subsumed by the concept c'.

Institution $\mathcal{CH}$ as introduced in Definition 2.4.8 has only unary concepts available for sentence construction. However, in practice the ability to express hierarchies of n-ary concepts in the target of the slice category is desired. The main reason for that is the fact that we will use the target of the slice category for extending description logics with individuals. In our approach to formulate statements about individuals we will use the signatures in the target of the slice category. The idea is to introduce sentences of the form $c(i)$, where c is a concept and i is an individual. As in $\mathcal{CH}$-signature we only have unary concepts available we will be able to formulate statements about individuals with use of unary predicates only, for instance we will be unable to state that two individuals are related with each other via binary concept. This also means that can only refer to unary properties of individuals, which can be inconvenient in practice.

To avoid this problem we introduce an institution $\overline{\mathcal{CH}}$ which extends $\mathcal{CH}$ with n-ary concepts and we use $\overline{\mathcal{CH}}$ for defining description logics.

Notation 4.2.2. *To simplify the notation, in what follows for any model $\mathcal{M}$ we will denote the domain of $\mathcal{M}$ by $|\mathcal{M}|$.*

Definition 4.2.3 (Institution of Conceptual Hierarchies $\overline{\mathcal{CH}}$). *A $\overline{\mathcal{CH}}$-signature is an ω-indexed family of sets of n-ary predicates $(\Pi_n)_{n\in\omega}$. We call such families ω-sets. Signature morphisms $\sigma : \Pi \to \Pi'$ consist of a family of arity respecting functions between sets of predicates, i.e. $\sigma_n : \Pi_n \to \Pi'_n$, for $n \in \omega$. Given a $\overline{\mathcal{CH}}$-signature Π, we define sentences over Π in the following way:*

$$Sen^{\overline{\mathcal{CH}}}(\Pi) ::= \sum_{n\in\omega}(\Pi_n \times \Pi_n),$$

in other words, $Sen^{\overline{\mathcal{CH}}}(\Pi)$ is a disjoint union of sets of sentences of the form $p \sqsubseteq q$, where $p, q \in \Pi_n$, for some $n \in \omega$.

Given a signature morphism $\sigma : \Pi \to \Pi'$, we have $Sen^{\overline{\mathcal{CH}}}(\sigma) : Sen^{\overline{\mathcal{CH}}}(\Pi) \to Sen^{\overline{\mathcal{CH}}}(\Pi')$, this is done by renaming predicates according to σ.

The semantics of $\overline{\mathcal{CH}}$ is defined by means of interpretations $\mathcal{M} = (|\mathcal{M}|, \cdot^{\mathcal{M}})$, which are objects in the category of models, where for each $\Pi \in Sig^{\overline{\mathcal{CH}}}$ we have category $Mod^{\overline{\mathcal{CH}}}(\Pi)$. The interpretation domain $|\mathcal{M}|$ is a non-empty set, and $\cdot^{\mathcal{M}}$ is a function mapping each n-ary predicate $p \in \Pi_n$ to a subset $p^{\mathcal{M}}$ of $|\mathcal{M}|^n$, i.e. n-tuples of $|\mathcal{M}|$. Given a signature morphism $\sigma : \Pi \to \Pi'$ the reduct $\mathcal{M} \upharpoonright_{\sigma}$ is defined by $|\mathcal{M} \upharpoonright_{\sigma}| = |\mathcal{M}|$ and by $p^{\mathcal{M}\upharpoonright_{\sigma}} = \sigma(p)^{\mathcal{M}}$.

An interpretation $\mathcal{M}$ satisfies $p \sqsubseteq q$ (written $\mathcal{M} \models p \sqsubseteq q$) iff $p^{\mathcal{M}} \subseteq q^{\mathcal{M}}$.

A straightforward argument shows that the satisfaction condition holds for $\overline{\mathcal{CH}}$, i.e. given a signature Π, signature morphism $\sigma : \Pi \to \Pi'$, $\varphi \in Sen^{\overline{\mathcal{CH}}}(\Pi)$ and $\mathcal{M} \in |Mod^{\overline{\mathcal{CH}}}(\Pi')|$ the following holds:

$$\mathcal{M} \upharpoonright_{\sigma} \models^{\overline{\mathcal{CH}}}_{\Pi} \varphi \text{ iff } \mathcal{M} \models^{\overline{\mathcal{CH}}}_{\Pi'} \sigma(\varphi).$$

4.2.1 Description logics as objects of the slice category $Inst/\overline{\mathcal{CH}}$

Now we provide a definition of description logics which allows us to treat them in a general and systematical way. This definition shows how description logics are built over $\overline{\mathcal{CH}}$. More specifically, we define an institution $\mathcal{I}$ to be a description logic if there is a morphism from $\mathcal{I}$ to $\overline{\mathcal{CH}}$. Intuitively, that morphism provides us with the means to translate signatures of $\mathcal{I}$ to sets of $\mathcal{I}$-concepts in $\overline{\mathcal{CH}}$, where we can build sentences which are subsumptions. These sentences can be translated (via the morphism) back to $\mathcal{I}$, which in turn provides us with models that are used for interpreting the translated $\overline{\mathcal{CH}}$-sentences. We also know how to translate these models to $\overline{\mathcal{CH}}$ in order to receive interpretative structures in $\overline{\mathcal{CH}}$. The fact that description logics are defined with the use of morphism to $\overline{\mathcal{CH}}$ will be used to introduce individuals into DL-signatures, as well as to formulate sentences with individuals.

Definition 4.2.4. *A description logic is an institution $\mathcal{I}$ together with a morphism $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$.*

Categorically, description logics are the objects of the slice category $Inst/\overline{\mathcal{CH}}$, where *Inst* is the category of institutions and their morphisms. If $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ and $\nu : \mathcal{I}' \to \overline{\mathcal{CH}}$ are description logics, an arrow $\mu \to \nu$ is a morphism $\xi : \mathcal{I} \to \mathcal{I}'$ such that $\mu = \xi \, ; \nu$.

Example 4.2.5. *$\mathcal{EL}$ is a description logic.*
We show how components of morphism $\mathcal{EL} \overset{e\bar{c}}{\to} \overline{\mathcal{CH}}$ are constructed.

- *Functor $\Psi^{e\bar{c}}$. For any $\mathcal{EL}$-signature (P,R) we define $(\Psi^{e\bar{c}}(P,R))_1 = Con^{\mathcal{EL}}(P,R)$ and $(\Psi^{e\bar{c}}(P,R))_n = \emptyset$ for all other n. We sometimes use the notation $\Psi^{e\bar{c}}(P,R) = (\emptyset, Con^{\mathcal{EL}}(P,R), \emptyset, ...)$.*
- *Natural transformation $\gamma^{e\bar{c}}_{(P,R)}$. By definition $Sen^{\overline{\mathcal{CH}}}(\Pi)$ is a disjoint union of sets of sentences of the form $p \sqsubseteq q$, with $p,q \in \Pi_n$, but $\Psi^{e\bar{c}}(P,R)$ only has unary concepts, so $Sen^{\overline{\mathcal{CH}}}(\Psi^{e\bar{c}}(P,R))$ is the set of subsumptions $C \sqsubseteq D$, where $C, D \in Con^{\mathcal{EL}}(P,R)$. Therefore, we may take $\gamma^{e\bar{c}}_{(P,R)}$ to be the identity.*
- *Natural transformation $\delta^{e\bar{c}}_{(P,R)}$. For any $\mathcal{M} \in |Mod^{\mathcal{EL}}(P,R)|$, we set $\delta^{e\bar{c}}_{(P,R)}(\mathcal{M}) = \mathcal{M}$. This make sense because (P,R)-models in $\mathcal{EL}$ are also interpretative structures for $\mathcal{EL}$-concepts built over (P,R), i.e. $Con^{\mathcal{EL}}(P,R)$.*

Now we show that the satisfaction condition holds for all $(P,R) \in |Sig^{\mathcal{EL}}|$, $\mathcal{M} \in |Mod^{\mathcal{EL}}(P,R)|$ and $C \sqsubseteq D \in Sen^{\overline{\mathcal{CH}}}(\Psi^{e\bar{c}}(P,R))$:

$$\mathcal{M} \models^{\mathcal{EL}}_{(P,R)} \gamma^{e\bar{c}}_{(P,R)}(C \sqsubseteq D) \quad \textit{iff} \quad \delta^{e\bar{c}}_{(P,R)}(\mathcal{M}) \models^{\overline{\mathcal{CH}}}_{\Psi^{e\bar{c}}(P,R)} C \sqsubseteq D.$$

Proof:

$$\mathcal{M} \models^{\mathcal{EL}}_{(P,R)} \gamma^{e\bar{c}}_{(P,R)}(C \sqsubseteq D)$$

iff

$$\mathcal{M} \models^{\mathcal{EL}}_{(P,R)} C \sqsubseteq D$$

iff

$$C^{\mathcal{M}} \subseteq D^{\mathcal{M}}$$

iff

$$C^{\delta^{\mathsf{e}\bar{\mathsf{c}}}_{(P,R)}(\mathcal{M})} \subseteq D^{\delta^{\mathsf{e}\bar{\mathsf{c}}}_{(P,R)}(\mathcal{M})}$$

iff

$$\delta^{\mathsf{e}\bar{\mathsf{c}}}_{(P,R)}(\mathcal{M}) \models^{\overline{\mathcal{CH}}}_{\Psi^{\mathsf{e}\bar{\mathsf{c}}}(P,R)} C \sqsubseteq D$$

□

Example 4.2.6. *$\mathcal{ALC}$ is a description logic. This follows from the fact that we can compose morphisms and construct a composition* ae ; e$\bar{\text{c}}$: $\mathcal{ALC} \to \overline{\mathcal{CH}}$. *But in this case (similarly to* ae ; ec*) we have that $\mathcal{ALC}$ signature (P, R) gets mapped into $Con^{\mathcal{EL}}$, which is not satisfactory as we would be able to formulate concept inclusions using $\mathcal{EL}$-concepts only. This means that we cannot use negation, which distinguishes $\mathcal{ALC}$ from $\mathcal{EL}$. For that reason we define an alternative morphism to $\overline{\mathcal{CH}}$, a morphism* a$\bar{\text{c}}$ *which tells us how $\mathcal{ALC}$ is built over $\overline{\mathcal{CH}}$. Functor $\Psi^{\mathsf{a}\bar{\mathsf{c}}}$ is defined in the following way: $\Psi^{\mathsf{e}\bar{\mathsf{c}}}(P,R) = (\emptyset, Con^{\mathcal{ALC}}(P,R), \emptyset, ...)$. Natural transformation $\gamma^{\mathsf{a}\bar{\mathsf{c}}}_{(P,R)}$ is the identity as $\Psi^{\mathsf{a}\bar{\mathsf{c}}}(P,R)$ only has unary concepts, and so $Sen^{\overline{\mathcal{CH}}}(\Psi^{\mathsf{a}\bar{\mathsf{c}}}(P,R))$ is the set of subsumptions $C \sqsubseteq D$, where $C, D \in Con^{\mathcal{ALC}}(P,R)$. Natural transformation $\delta^{\mathsf{a}\bar{\mathsf{c}}}_{(P,R)}$ is the identity, so for any $\mathcal{M} \in |Mod^{\mathcal{ALC}}(P,R)|$, we have that $\delta^{\mathsf{a}\bar{\mathsf{c}}}_{(P,R)}(\mathcal{M}) = \mathcal{M}$. The proof that the satisfaction condition holds is very similar to the one for Example 4.2.5.*

Example 4.2.7. *Now we show that FOL is a description logic, i.e. we define a morphism* f$\bar{\text{c}}$: *$FOL \to \overline{\mathcal{CH}}$. We only sketch how the components of morphism* f$\bar{\text{c}}$: *$FOL \to \overline{\mathcal{CH}}$ are constructed.*

Functor $\Psi^{\mathsf{f}\bar{\mathsf{c}}}$. For any FOL-signature $(\Pi_n)_{n\in\omega}$ we define

$$\Psi^{\mathsf{f}\bar{\mathsf{c}}}(\Pi) = (Form^{FOL}(\Pi, n))_{n\in\omega}.$$

where $Form^{FOL}(\Pi, n)$ is the set of FOL formulae over Π with n free variables $(x_0, \ldots, x_n)$.

Sentences in $\overline{\mathcal{CH}}$ over $\Psi^{\mathsf{f}\bar{\mathsf{c}}}(\Pi)$ are defined in the following way

$$Sen^{\overline{\mathcal{CH}}}(\Psi^{\mathsf{f}\bar{\mathsf{c}}}(\Pi)) = \sum_{n\in\omega}((\Psi^{\mathsf{f}\bar{\mathsf{c}}}(\Pi))_n \times (\Psi^{\mathsf{f}\bar{\mathsf{c}}}(\Pi))_n) \ .$$

Natural transformation $\gamma^{\mathsf{f}\bar{\mathsf{c}}}_{\Pi}$ translates $\overline{\mathcal{CH}}$-sentences, constructed over $\Psi^{\mathsf{f}\bar{\mathsf{c}}}(\Pi)$, into FOL-sentences with countable set of variables in the following way:

$$\gamma^{\mathsf{f}\bar{\mathsf{c}}}_{\Pi}(C \sqsubseteq D) = (\forall x_0, \ldots, x_n)\overline{C} \Rightarrow \overline{D}$$

where $C, D \in Form^{FOL}(\Pi, n)$.

The natural transformation $\delta^{\mathsf{f}\bar{\mathsf{c}}}_{\Pi}$ is the identity, so for any FOL Π-model $\mathcal{M}$ we have $\delta^{\mathsf{f}\bar{\mathsf{c}}}_{\Pi}(\mathcal{M}) = \mathcal{M}$. It is straightforward to see that the satisfaction condition holds for all $\Pi \in |Sig^{FOL}|$, $\mathcal{M} \in |Mod^{FOL}(\Pi)|$ and $C \sqsubseteq D \in Sen^{\overline{\mathcal{CH}}}(\Psi^{\mathsf{f}\bar{\mathsf{c}}}(\Pi))$:

$$\mathcal{M} \models^{FOL}_{(\Pi)} \gamma^{\mathsf{f}\bar{\mathsf{c}}}_{(\Pi)}(C \sqsubseteq D) \text{ iff } \delta^{\mathsf{f}\bar{\mathsf{c}}}_{(\Pi)}(\mathcal{M}) \models^{\overline{\mathcal{CH}}}_{\Psi^{\mathsf{f}\bar{\mathsf{c}}}(\Pi)} C \sqsubseteq D.$$

Using composition of morphisms $\mathsf{e}^{+}\mathsf{e} : \mathcal{EL}^{+} \to \mathcal{EL}$ and $\mathsf{e}\bar{\mathsf{c}} : \mathcal{EL} \to \overline{\mathcal{CH}}$ we can show that $\mathcal{EL}^{+}$ is a description logic, but we find it useful for further studies to distinguish different types of $\mathcal{EL}^{+}$ and show that each of them is a description logic.

We distinguish different types of $\mathcal{EL}^{+}$, insofar as they allow different role inclusion axioms. So we distinguish the following:

- $\mathcal{EL}_0^{+}$ allows axioms of the form: $r \sqsubseteq s$,
- $\mathcal{EL}_1^{+}$ allows for $\mathcal{EL}_0^{+}$ axioms together with axioms of the form: $r \circ r \sqsubseteq r$,
- $\mathcal{EL}_2^{+}$ allows axioms of the form: $r_1 \circ \cdots \circ r_n \sqsubseteq r$, with $n \geq 1$,
- $\mathcal{EL}_3^{+}$ allows axioms of the form: $r_1 \circ \cdots \circ r_n \sqsubseteq s_1 \circ \cdots \circ s_m$ with $m, n \geq 1$.

Example 4.2.8. *$\mathcal{EL}_i^{+}$ is a description logic, where $i = \{0,1,2,3\}$.*
We show how components of morphism $\mathsf{e}_i^{+}\bar{\mathsf{c}} : \mathcal{EL}_i^{+} \to \overline{\mathcal{CH}}$ are constructed.

- *Functor $\Psi^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}$. For any $\mathcal{EL}_i^{+}$-signature (P,R) we define*

$$\Psi^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}(P,R) = (\emptyset, Con^{\mathcal{EL}_i^{+}}(P,R), R, \emptyset, \ldots),$$

  i.e. $\Psi^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}$ translates P part of the signature into $\mathcal{EL}_i^{+}$-concepts over (P,R) and is the identity for R part of the signature.

- *Natural transformation $\gamma^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}_{(P,R)}$. By definition $Sen^{\overline{\mathcal{CH}}}(\Pi)$ is a disjoint union of sets of sentences of the form $p \sqsubseteq q$, with $p, q \in \Pi_n$, but since*

$$\Psi^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}(P,R) = (\emptyset, Con^{\mathcal{EL}_i^{+}}(P,R), R, \emptyset, \ldots)\ ,$$

  we have that

$$p, q \in Con^{\mathcal{EL}_i^{+}}(P,R) \ \textit{or}\ p, q \in R\ .$$

  Since

$$Sen^{\mathcal{EL}_i^{+}}(P,R) ::= Con^{\mathcal{EL}_i^{+}}(P,R) \sqsubseteq Con^{\mathcal{EL}_i^{+}}(P,R) \uplus R \sqsubseteq R\ ,$$

  we have that

$$\gamma^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}_{(P,R)}(Sen^{\overline{\mathcal{CH}}}(\Psi^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}(P,R))) \subseteq Sen^{\mathcal{EL}_i^{+}}(P,R) \ \textit{for}\ i > 0$$

  and

$$\gamma^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}_{(P,R)}(Sen^{\overline{\mathcal{CH}}}(\Psi^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}(P,R))) = Sen^{\mathcal{EL}_i^{+}}(P,R) \ \textit{for}\ i = 0\ .$$

- *Natural transformation $\delta^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}_{(P,R)}$. For any $\mathcal{M} \in |Mod^{\mathcal{EL}_i^{+}}(P,R)|$, we define $\delta^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}_{(P,R)}(\mathcal{M}) = \mathcal{M}$.*

Now we show that the satisfaction condition holds for every signature $(P,R) \in |Sig^{\mathcal{EL}_i^{+}}|$, model $\mathcal{M} \in |Mod^{\mathcal{EL}_i^{+}}(P,R)|$ and sentence $\varphi \in Sen^{\overline{\mathcal{CH}}}(\Psi^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}(P,R))$.

$$\mathcal{M} \models^{\mathcal{EL}_i^{+}}_{(P,R)} \gamma^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}_{(P,R)}(\varphi) \quad \textit{iff} \quad \delta^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}_{(P,R)}(\mathcal{M}) \models^{\overline{\mathcal{CH}}}_{\Psi^{\mathsf{e}_i^{+}\bar{\mathsf{c}}}(P,R)} \varphi\ .$$

Proof: In the proof we distinguish two cases:

1. φ is of the form $\mathsf{Con}^{\mathcal{EL}_i^+}(P,R) \sqsubseteq \mathsf{Con}^{\mathcal{EL}_i^+}(P,R)$,
2. φ is of the form $R \sqsubseteq R$.

Example 4.2.5 already shows the case (1), so we only show for the case (2).

$$\mathcal{M} \models^{\mathcal{EL}}_{(P,R)} \gamma^{\mathsf{e}_i^+\mathsf{c}}_{(P,R)}(r \sqsubseteq s)$$

iff (by definition)

$$\mathcal{M} \models^{\mathcal{EL}}_{(P,R)} r \sqsubseteq s$$

iff

$$r^{\mathcal{M}} \subseteq s^{\mathcal{M}}$$

iff (by definition)

$$r^{\delta^{\mathsf{e}_i^+\mathsf{c}}_{(P,R)}(\mathcal{M})} \subseteq s^{\delta^{\mathsf{e}_i^+\mathsf{c}}_{(P,R)}(\mathcal{M})}$$

iff

$$\delta^{\mathsf{e}_i^+\mathsf{c}}_{(P,R)}(\mathcal{M}) \models^{\overline{\mathcal{CH}}}_{\Psi^{\mathsf{e}_i^+\mathsf{c}}(P,R)} r \sqsubseteq s$$

□

Example 4.2.8 shows how different variants of $\mathcal{EL}^+$ are description logic. The reason why it works for all of them is the fact that $\Psi^{\mathsf{e}_i^+\mathsf{c}}$ is the identity on R part of the signature. In this way only role inclusions of the form $r \sqsubseteq s$, with $r, s \in R$, are allowed. Now we show a morphism $\mathsf{e}_3^+\mathsf{c}$, which allows for arbitrary role inclusions by translating R part of the signature into $R^* = \{r_1 \circ \cdots \circ r_n | r_1 \dots r_n \in R\}$.

Example 4.2.9. *$\mathcal{EL}_3^+$ is a description logic.*
We show how components of morphism $\mathsf{e}_3^+\mathsf{c} : \mathcal{EL}_3^+ \to \overline{\mathcal{CH}}$ are constructed.

- *Functor $\Psi^{\mathsf{e}_3^+\mathsf{c}}$. For any $\mathcal{EL}_3^+$-signature (P,R) we define*

  $$\Psi^{\mathsf{e}_3^+\mathsf{c}}(P,R) = (\emptyset, Con^{\mathcal{EL}_3^+}(P,R), R^*, \emptyset, \dots)$$

  where $R^* = \{r_1 \circ \cdots \circ r_n | r_1 \dots r_n \in R\}$. In other words $\Psi^{\mathsf{e}_3^+\mathsf{c}}$ translates P part of the signature into $\mathcal{EL}_3^+$-concepts over (P,R) and R part of the signature into a set of expressions of the form $r_1 \circ \cdots \circ r_n$.

- *Natural transformation $\gamma^{\mathsf{e}_3^+\mathsf{c}}_{(P,R)}$. By definition $Sen^{\overline{\mathcal{CH}}}(\Pi)$ is a disjoint union of sets of sentences of the form $p \sqsubseteq q$, with $p, q \in \Pi_n$, since*

  $$\Psi^{\mathsf{e}_3^+\mathsf{c}}(P,R) = (\emptyset, Con^{\mathcal{EL}_3^+}(P,R), R^*, \emptyset, \dots) \ ,$$

  we have that $p, q \in Con^{\mathcal{EL}_3^+}(P,R)$ or $p, q \in R^*$. From this it follows that

  $$Sen^{\overline{\mathcal{CH}}}(\Psi^{\mathsf{e}_3^+\mathsf{c}}(P,R)) = Con^{\mathcal{EL}_3^+}(P,R) \times Con^{\mathcal{EL}_3^+}(P,R) \uplus R^* \times R^* \ .$$

  Thus it is not difficult to see that $\gamma^{\mathsf{e}_3^+\mathsf{c}}_{(P,R)}$ is the identity and we have

  $$\gamma^{\mathsf{e}_3^+\mathsf{c}}_{(P,R)}(Sen^{\overline{\mathcal{CH}}}(\Psi^{\mathsf{e}_3^+\mathsf{c}}(P,R))) = Sen^{\mathcal{EL}_3^+}(P,R) \ .$$

- *Natural transformation* $\delta^{e_3^+\bar{c}}_{(P,R)}$. *For any* $\mathcal{M} \in |Mod^{\mathcal{EL}_3^+}(P,R)|$, *we define* $\delta^{e_3^+\bar{c}}_{(P,R)}(\mathcal{M}) = \mathcal{M}$, *with*

$$(r_1 \circ \cdots \circ r_n)^{\delta^{e_3^+\bar{c}}_{(P,R)}(\mathcal{M})} = r_1^{\mathcal{M}} \circ \cdots \circ r_n^{\mathcal{M}} \subseteq |\mathcal{M}|^2$$

Now we show that the satisfaction condition holds for every signature $(P,R) \in |Sig^{\mathcal{EL}_3^+}|$, *model* $\mathcal{M} \in |Mod^{\mathcal{EL}_3^+}(P,R)|$ *and sentence* $\varphi \in Sen^{\overline{\mathcal{CH}}}(\Psi^{e_3^+\bar{c}}(P,R))$:

$$\mathcal{M} \models^{\mathcal{EL}_3^+}_{(P,R)} \gamma^{e_3^+\bar{c}}_{(P,R)}(\varphi) \quad \textit{iff} \quad \delta^{e_3^+\bar{c}}_{(P,R)}(\mathcal{M}) \models^{\overline{\mathcal{CH}}}_{\Psi^{e_3^+\bar{c}}(P,R)} \varphi\,.$$

Proof: In the proof we distinguish two cases:

1. φ is of the form $\mathsf{Con}^{\mathcal{EL}_3^+}(P,R) \sqsubseteq \mathsf{Con}^{\mathcal{EL}_3^+}(P,R)$,

2. φ is of the form $R^* \sqsubseteq R^*$.

As Example 4.2.5 already shows for the case (1), we only show for the case (2). Let $\varphi = r_1 \circ \cdots \circ r_n \sqsubseteq s_1 \circ \cdots \circ s_m$

$$\mathcal{M} \models^{\mathcal{EL}_3^+}_{(P,R)} \gamma^{e_3^+\bar{c}}_{(P,R)}(r_1 \circ \cdots \circ r_n \sqsubseteq s_1 \circ \cdots \circ s_m)$$

iff

$$\mathcal{M} \models^{\mathcal{EL}_3^+}_{(P,R)} r_1 \circ \cdots \circ r_n \sqsubseteq s_1 \circ \cdots \circ s_m$$

iff

$$r_1^{\mathcal{M}} \circ \cdots \circ r_n^{\mathcal{M}} \sqsubseteq s_1^{\mathcal{M}} \circ \cdots \circ s_m^{\mathcal{M}}$$

iff

$$r_1^{\delta^{e_3^+\bar{c}}_{(P,R)}(\mathcal{M})} \circ \cdots \circ r_n^{\delta^{e_3^+\bar{c}}_{(P,R)}(\mathcal{M})} \sqsubseteq s_1^{\delta^{e_3^+\bar{c}}_{(P,R)}(\mathcal{M})} \circ \cdots \circ s_m^{\delta^{e_3^+\bar{c}}_{(P,R)}(\mathcal{M})}$$

iff

$$\delta^{e_3^+\bar{c}}_{(P,R)}(\mathcal{M}) \models^{\overline{\mathcal{CH}}}_{\Psi^{e_3^+\bar{c}}(P,R)} r_1 \circ \cdots \circ r_n \sqsubseteq s_1 \circ \cdots \circ s_m$$

□

4.2.2 Description logics with individuals

Now we show how any description logic can be extended to an institution with individuals.

Definition 4.2.10. *Given a description logic* $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$, *define the institution* $\mathbb{I}\mu$ *as follows.*

- $Sig^{\mathbb{I}\mu} = Sig^{\mathcal{I}} \times \mathsf{Set}$. *That is, signatures are pairs* (Σ, I), *where* Σ *is an* $\mathcal{I}$-*signature and* I *is a set; similarly, signature morphisms are pairs* $(\sigma, g) : (\Sigma, I) \to (\Sigma', I')$ *with* $\sigma : \Sigma \to \Sigma'$ *in* $Sig^{\mathcal{I}}$ *and* $g : I \to I'$ *a function in* Set.

- $Sen^{\mathbb{I}\mu}(\Sigma, I) = Sen^{\mathcal{I}}(\Sigma) + \sum_{n\in\omega}(\Psi^{\mu}(\Sigma_n) \times I^n)$. *That is, sentences are either* Σ-*sentences in* $\mathcal{I}$, *or pairs* (c, i) *with* $c \in \Psi^{\mu}(\Sigma)_n$ *an* n-*ary concept and* $i \in I^n$ *a tuple of individuals; we use the notation* $c(i)$ *to suggest the intended meaning that the tuple of individuals* i *is an instance of the concept* c.

  For $(\sigma, g) : (\Sigma, I) \to (\Sigma', I')$, *we let* $Sen^{\mathbb{I}\mu}(\sigma, g) = Sen^{\mathcal{I}}(\sigma) + \Psi^{\mu}(\sigma) \times g$, *which maps a* Σ-*sentence* φ *to* $Sen^{\mathcal{I}}(\sigma)(\varphi)$, *and a sentence* $c(i)$, *with* $c \in \Psi^{\mu}(\Sigma)$, *to* $\Psi^{\mu}(\sigma)(c)(g(i))$.

- *$Mod^{\mathbb{I}\mu}(\Sigma, I)$ has as objects pairs (M, f), with M a Σ-model in $\mathcal{I}$, and $f : I \to |\delta^{\mu}_{\Sigma}(M)|$ a function that 'interprets' individuals as elements of the domain. Arrows $h : (M, f) \to (M', f')$ are arrows $h : M \to M'$ in $Mod^{\mathcal{I}}(\Sigma)$ such that $f' = f \,;\, \delta^{\mu}_{\Sigma}(h)$.*

  For signature morphism $(\sigma, g) : (\Sigma, I) \to (\Sigma', I')$, $Mod^{\mathbb{I}\mu}(\sigma, g)$ maps a (Σ', I')-model (M', f') to $(Mod^{\mathcal{I}}(\sigma)(M'),\ g\,;\, f')$.

- *For $\varphi \in Sen^{\mathcal{I}}(\Sigma)$,*

$$(M, f) \models^{\mathbb{I}\mu}_{(\Sigma,I)} \varphi \quad \text{iff} \quad M \models^{\mathcal{I}}_{\Sigma} \varphi \,,$$

  and for sentences $c(i)$ with $c \in \Psi^{\mu}(\Sigma_n)$ and $i \in I^n$,

$$(M, f) \models^{\mathbb{I}\mu}_{(\Sigma,I)} c(i) \quad \text{iff} \quad \overline{f}(i) \in c^{\delta^{\mu}_{\Sigma}(M)} \,,$$

  where by $\overline{f}$ we mean the extension of f to n-tuples.

To show that $\mathbb{I}\mu$ is an institution we still have to show that the satisfaction condition holds. Here we only show the satisfaction condition for the case with sentences of the form $c(i)$. The case for sentences without individuals is straightforward.

Given a signature (Σ, I) and a signature morphism $\sigma : (\Sigma, I) \to (\Sigma', I')$, with $\sigma = (\sigma_{\Sigma}, \sigma_I)$ where $\sigma_{\Sigma} : \Sigma \to \Sigma'$ and $\sigma_I : I \to I'$, a sentence $c(i) \in \mathsf{Sen}^{\mathbb{I}\mu}(\Sigma, I)$ and a model (M', f') in $\mathsf{Mod}^{\mathbb{I}\mu}(\Sigma', I')$, the following holds:

$$(M', f')\restriction_{\sigma} \models^{\mathbb{I}\mu}_{(\Sigma,I)} c(i) \;\text{ iff }\; (M', f') \models^{\mathbb{I}\mu}_{(\Sigma',I')} \sigma(c(i)).$$

Proof:

$$(M', f')\restriction_{\sigma} \models^{\mathbb{I}\mu}_{(\Sigma,I)} c(i)$$

iff

$$(M'\restriction_{\sigma_\Sigma}, f'\restriction_{\sigma_I}) \models^{\mathbb{I}\mu}_{(\Sigma,I)} c(i)$$

iff

$$\overline{f'\restriction_{\sigma_I}}(i) \in c^{\delta^{\mu}_{\Sigma}(M'\restriction_{\sigma_\Sigma})}$$

iff

$$\overline{f'}(\sigma_I(i)) \in \sigma_{\Sigma}(c)^{\delta^{\mu}_{\Sigma}(M')}$$

iff

$$(M', f') \models^{\mathbb{I}\mu}_{(\Sigma',I')} \sigma(c(i))$$

□

Now we spell out the details of how to extend $\mathcal{EL}$ to $\mathcal{EL}$ with individuals in the signature. We show that using the morphism $\mathsf{e}\bar{\mathsf{c}}$ from Example 4.2.5.

Example 4.2.11. *The institution $\mathbb{I}\mathsf{e}\bar{\mathsf{c}}$ has as signatures triples of sets (P, R, I) with triples of functions as signature morphisms. A (P, R, I)-sentence is either a GCI of the form $C \sqsubseteq D$, where C and D are (P, R)-concepts, or is of the form $C(i)$, where C is a (P, R)-concept and $i \in I$. (P, R, I)-models are pairs (M, f), where M is the usual (P, R)-model in $\mathcal{EL}$, and $f : I \to |M|$. $(M, f) \models_{(P,R,I)} C \sqsubseteq D$ iff $M \models^{\mathcal{EL}}_{(P,R)} C \sqsubseteq D$, and $(M, f) \models_{(P,R,I)} C(i)$ iff $f(i) \in C^M$.*

Since limits and especially colimits of signatures are essential for modularity, it is useful to note that the following holds:

Proposition 4.2.12. *$Sig^{\mathbb{I}\mu}$ has all small (co)limits that $Sig^{\mathcal{I}}$ has.*
Proof: *This follows from the fact that* Set *is (co)complete, and (co)limits can be taken pointwise in $Sig^{\mathbb{I}\mu} = Sig^{\mathcal{I}} \times$* Set.

Similarly, we have

Proposition 4.2.13. *$\mathbb{I}\mu$ is (semi-)exact iff $\mathcal{I}$ is.*

Proof: We only show the case for semi-exactness, as the case for exactness considers in addition only finite (co)limits and therefore is simpler.

For the direction "$\Rightarrow$" first assume that $\mathbb{I}\mu$ is semi-exact. Now assume that the signature morphisms in $\mathcal{I}$ shown in Figure 4.1 form a pushout. This implies that after extending

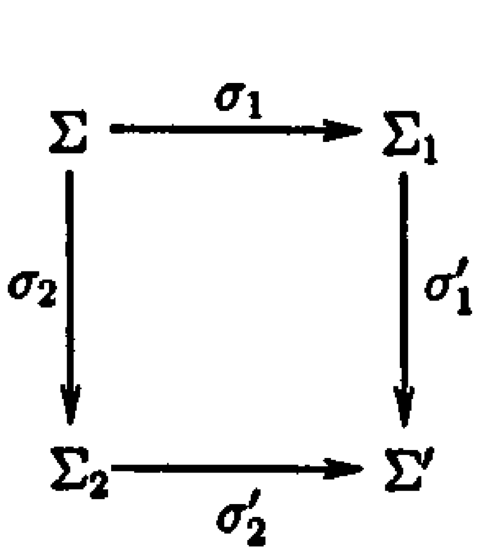


Figure 4.1: Signature morphism square in $\mathcal{I}$.

these signatures with the empty set of individuals, as shown in Figure 4.2, the signature morphisms form a pushout square in $\mathsf{Sig}^{\mathbb{I}\mu}$. But by the assumption we have that the model

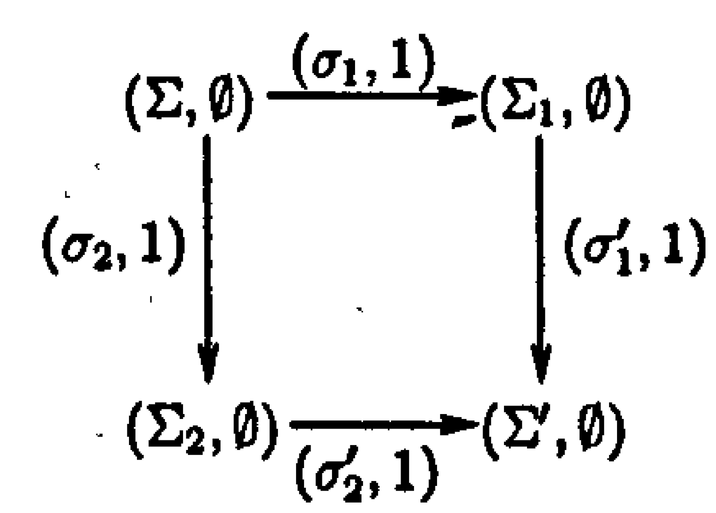


Figure 4.2: Signature morphism square in $\mathbb{I}\mu$ with empty sets of individuals.

morphisms, as shown in Figure 4.3, form a pullback square in $\mathbb{I}\mu$. Note that a $(\Sigma, \emptyset)$-model in $\mathbb{I}\mu$ is of the form (M, ι), where ι is the inclusion $\emptyset \hookrightarrow |\beta^{\mu}_{\Sigma}(M)|$, and is equivalent to a Σ-model M in $\mathcal{I}$. Therefore we have that the model morphisms, as shown in Figure 4.4, in $\mathcal{I}$ form a pullback, as desired.

For the direction "$\Leftarrow$" assume $\mathcal{I}$ is semi-exact. Assume we are given a pushout square of signature morphisms in $\mathbb{I}\mu$ (see Figure 4.5). We want to show that the model morphisms in $\mathbb{I}\mu$ form a pullback square (see Figure 4.6). To do that we take a commuting square of model morphisms in $\mathbb{I}\mu$ with category $\mathcal{C}$ (see Figure 4.7) and show that Figure 4.6 has the

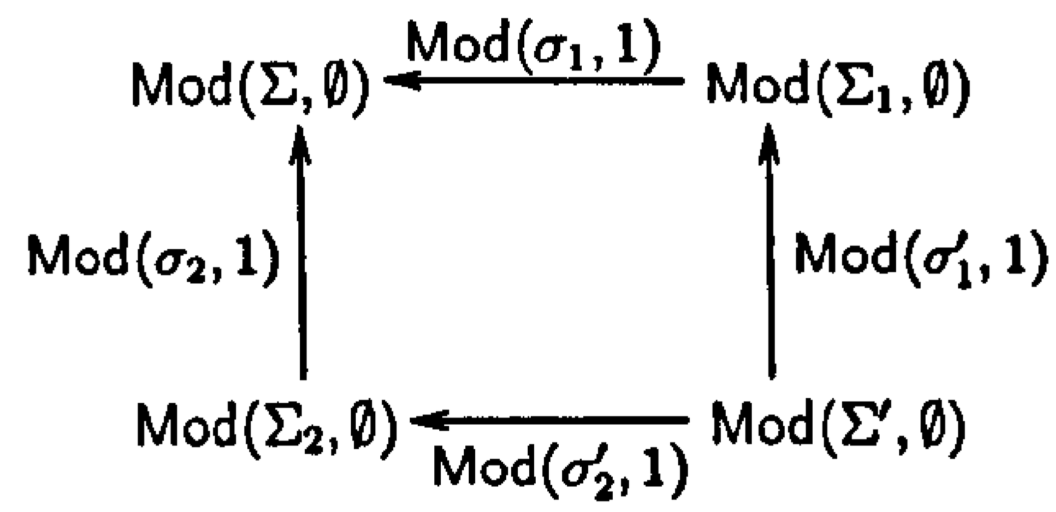


Figure 4.3: Model morphism square in $\mathbb{I}\mu$ with empty sets of individuals.

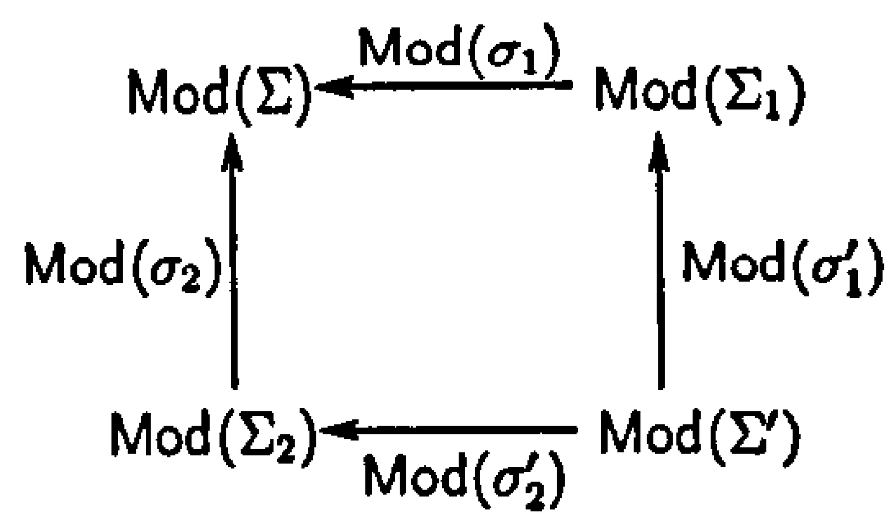


Figure 4.4: Model morphism square in $\mathcal{I}$.

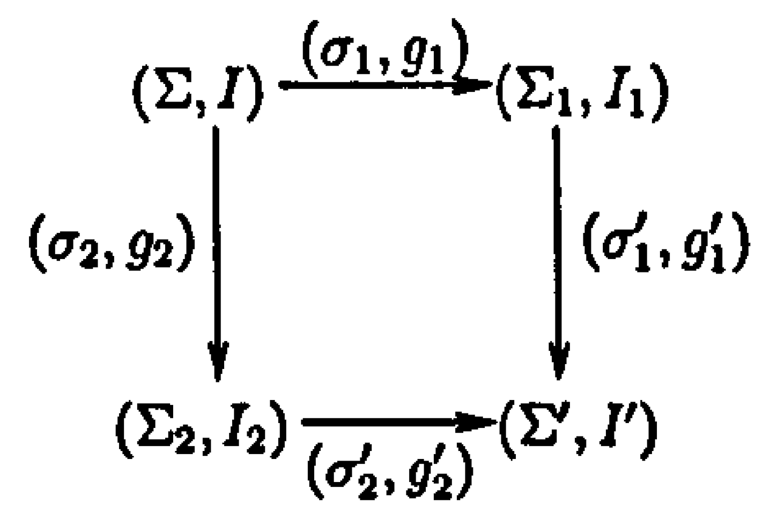


Figure 4.5: Signature morphism square in $\mathbb{I}\mu$.

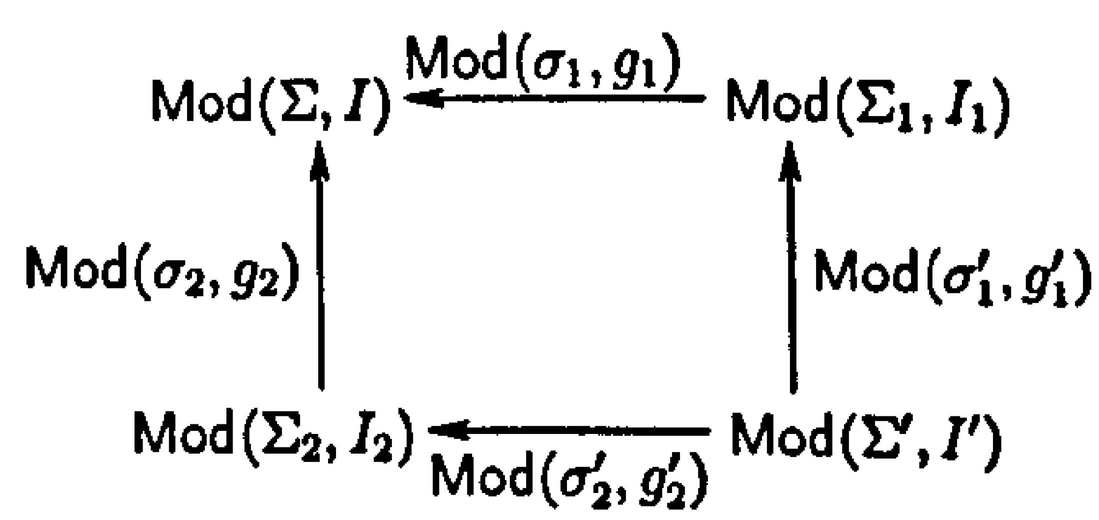


Figure 4.6: Model morphism square in $\mathbb{I}\mu$.

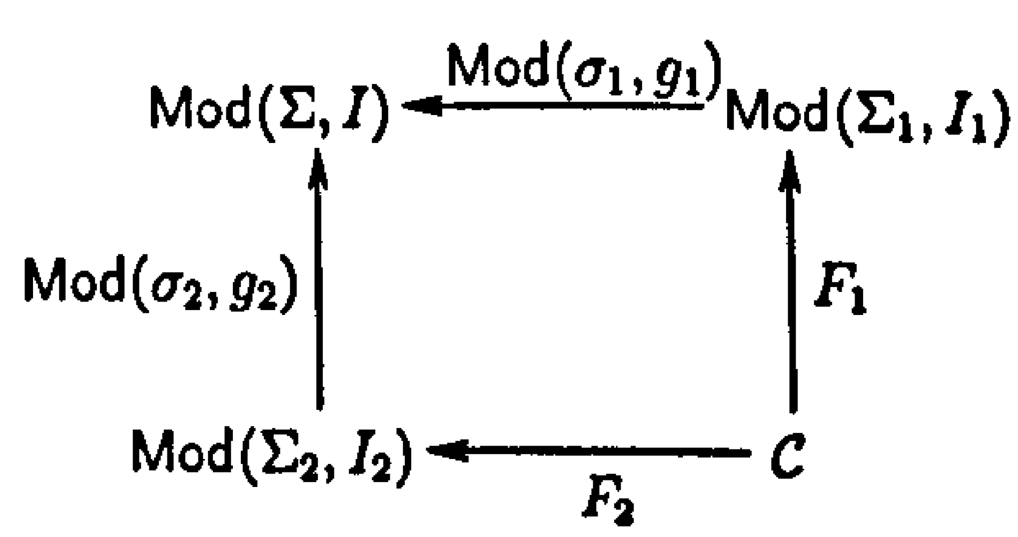


Figure 4.7: Model morphism square in $\mathbb{I}\mu$ with category $\mathcal{C}$.

universal property of pullback by constructing a unique morphism $F : \mathcal{C} \to \mathsf{Mod}(\Sigma', I')$ such that $F; \mathsf{Mod}(\sigma_i', g_i') = F_i$ (with $i = 1,2$).

Any (Σ, I)-model $\mathcal{M}$ is of the form $\mathcal{M} = (M, f)$, where M is a Σ-model. This gives us a functor $U_\Sigma : \mathsf{Mod}(\Sigma, I) \to \mathsf{Mod}(\Sigma)$ sending (M, F) to M. Now if Figure 4.6 commutes, this implies that the outer square of Figure 4.8 commutes, inducing a functor $U : \mathcal{C} \to \mathsf{Mod}(\Sigma')$. For c in $\mathcal{C}$ we write (M_i, f_i) for $F_i c$ (with $i = 1,2$). Then we

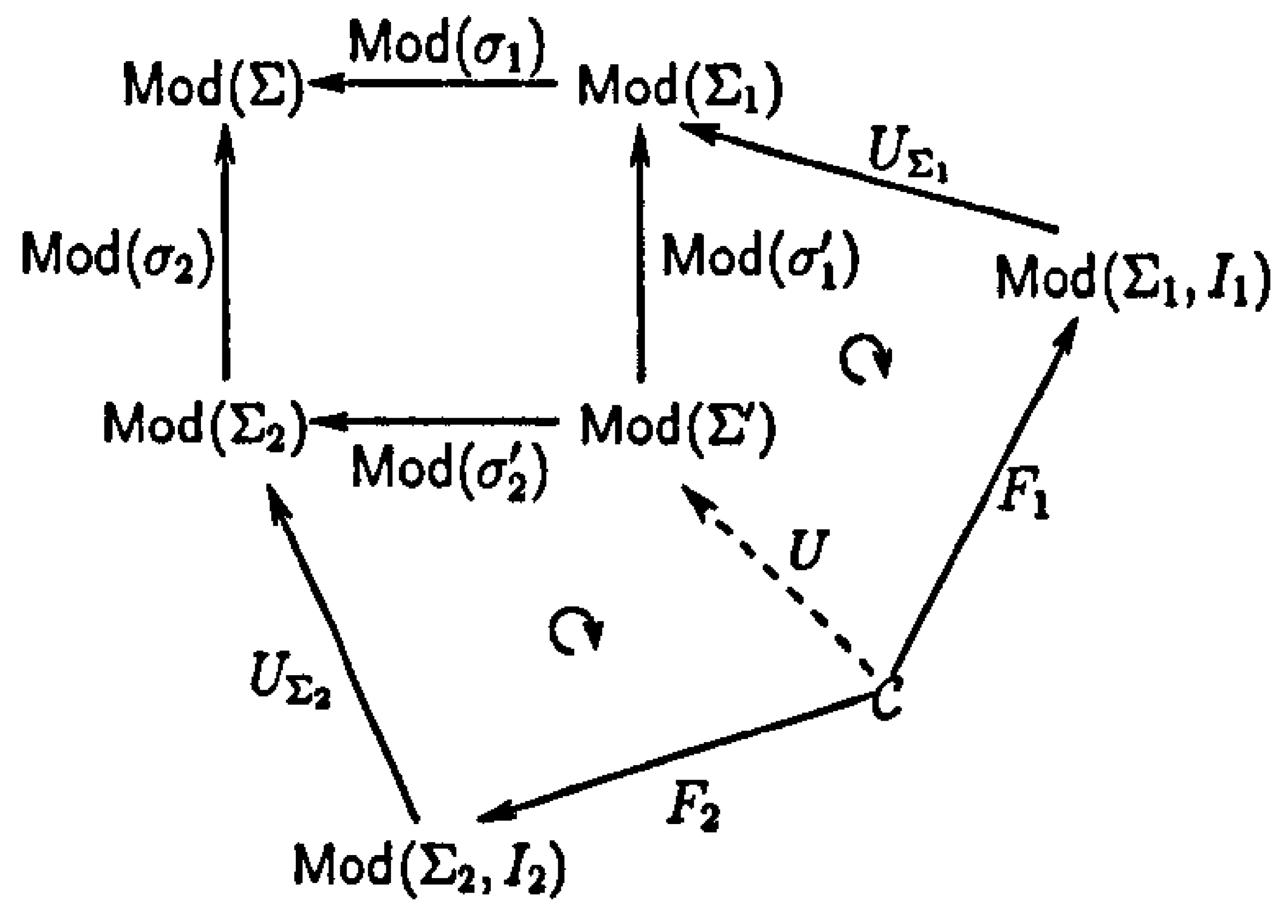


Figure 4.8.

have $\mathsf{Mod}(\sigma_1')(Uc) = U_{\Sigma_1}(F_1 c) = U_{\Sigma_1}(M_1, f_1) = M_1$ and $\mathsf{Mod}(\sigma_2')(Uc) = U_{\Sigma_2}(F_2 c) = U_{\Sigma_2}(M_2, f_2) = M_2$. Now we have to determine the mapping $I' \to |\delta^{\mu}_{\Sigma'}(Uc)|$. But we have models (M_1, f_1) and (M_2, f_2) and so mappings $f_1 : I_1 \to |\delta^{\mu}_{\Sigma_1}(M_1)|$ and $f_2 : I_2 \to |\delta^{\mu}_{\Sigma_2}(M_2)|$, such that $\mathsf{Mod}(\sigma_1, g_1)(M_1, f_1) = \mathsf{Mod}(\sigma_2, g_2)(M_2, f_2)$, because Figure 4.7 commutes. We have $\mathsf{Mod}(\sigma_1, g_1)(M_1, f_1) = (\mathsf{Mod}(\sigma_1)(M_1), g_1; f_1 : I \to |\delta^{\mu}_{\Sigma}(\mathsf{Mod}(\sigma_1)(M_1))|)$ and similarly for $\mathsf{Mod}(\sigma_2, g_2)(M_2, f_2)$; therefore $|\delta^{\mu}_{\Sigma}(\mathsf{Mod}(\sigma_1)(M_1))| = |\delta^{\mu}_{\Sigma}(\mathsf{Mod}(\sigma_2)(M_2))|$ and $g_1; f_1 = g_2; f_2$. We also have that $|\delta^{\mu}_{\Sigma}(\mathsf{Mod}(\sigma_1)(M_1))| = |\mathsf{Mod}(\Psi^{\mu}(\sigma_1'))(\delta^{\mu}_{\Sigma_1}(M_1))| = |\delta^{\mu}_{\Sigma_1}(M_1)|$ and similarly for $|\delta^{\mu}_{\Sigma}(\mathsf{Mod}(\sigma_2)(M_2))|$, so $|\delta^{\mu}_{\Sigma}(\mathsf{Mod}(\sigma_1)(M_1))| = |\delta^{\mu}_{\Sigma}(\mathsf{Mod}(\sigma_2)(M_2))|$. We also have that $|\delta^{\mu}_{\Sigma_1}(M_1)| = |\delta^{\mu}_{\Sigma_1}(\mathsf{Mod}(\sigma_1')(Uc))| = |\mathsf{Mod}(\Psi^{\mu}(\sigma_1'))(\delta^{\mu}_{\Sigma'}(Uc))| = |\delta^{\mu}_{\Sigma'}(Uc)|$ and similarly for $|\delta^{\mu}_{\Sigma_2}(M_2)|$, this gives us that $|\delta^{\mu}_{\Sigma'}(Uc)| = |\delta^{\mu}_{\Sigma_1}(M_1)| = |\delta^{\mu}_{\Sigma_2}(M_2)|$. This gives us that we have $f_1 : I_1 \to |\delta^{\mu}_{\Sigma'}(Uc)|$ and $f_2 : I_2 \to |\delta^{\mu}_{\Sigma'}(Uc)|$, so we use the fact that if Figure 4.5 is a pushout, so too is Figure 4.9. Now, we know that $g_1; f_1 = g_2; f_2$,

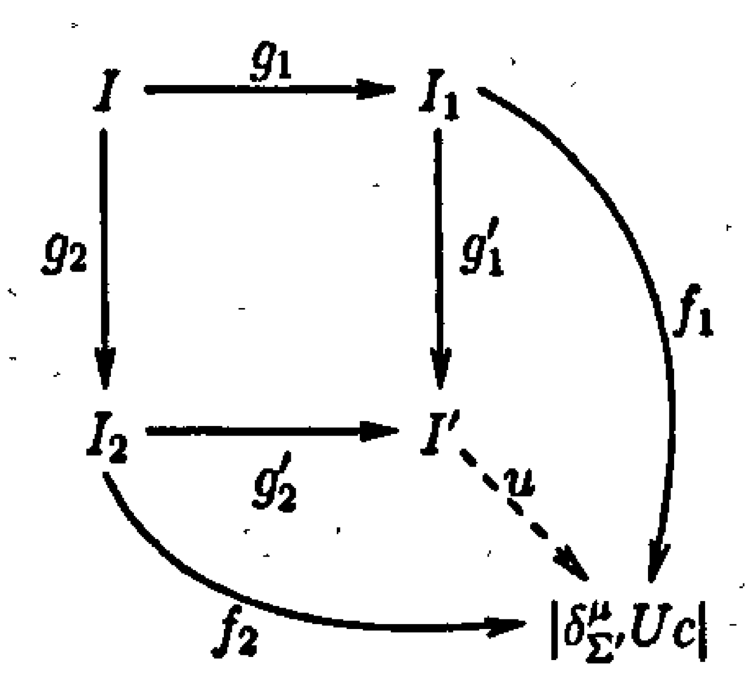


Figure 4.9: Morphism square for part of $\mathbb{I}\mu$-signature with individuals.

so we get $u : I' \to |\delta^{\mu}_{\Sigma'}(Uc)|$, such that $f_i = g_i'; u$ (with $i = 1,2$). We therefore set

$Fc = (Uc, u : I' \to |\delta^{\mu}_{\Sigma'}(Uc)|)$. This defines F on objects of $\mathcal{C}$, we need to show that F is the unique functor such that $F;\mathsf{Mod}(\sigma'_i, g'_i) = F_i$ (with $i = 1, 2$). Assume it is not, then there is another functor F' such that $F';\mathsf{Mod}(\sigma'_i, g'_i) = F_i$ for $i = 1, 2$. We have that $F';\mathsf{Mod}(\sigma'_i, g'_i); U_{\Sigma_i} = F_i; U_{\Sigma_i}$ and $F'; U_{\Sigma'}; \mathsf{Mod}(\sigma'_i) = F_i; U_{\Sigma_i}$, therefore $F'; U_{\Sigma'} = U$. So we have that $(F'c) = (Uc, f : I' \to |\delta^{\mu}_{\Sigma'}(Uc)|)$. But we also have that $g'_i; f = f_i$ therefore $f = u$ by pushout property of I. Thus we have that $(F'c) = (Uc, u : I' \to |\delta^{\mu}_{\Sigma'}(Uc)|)$. But this is exactly how Fc is defined, so $F' = F$. □

The construction of $\mathbb{I}\mu$ is functorial in μ. In other words, $\mathbb{I}$ is a functor $\mathit{Inst}/\overline{\mathcal{CH}} \to \mathit{Inst}$, which means that any morphism of description logics extends to a morphism of description logics with individuals.

Definition 4.2.14. *Let $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ and $\mu' : \mathcal{I}' \to \overline{\mathcal{CH}}$, and let $\xi : \mu \to \mu'$ (i.e., $\mu = \xi ; \mu'$). Morphism $\mathbb{I}\xi : \mathbb{I}\mu \to \mathbb{I}\mu'$ is defined as follows:*

- *Define $\Psi^{\mathbb{I}\xi} = \Psi^{\xi} \times 1_{\mathsf{Set}} \; : \; \mathit{Sig}^{\mathbb{I}\mu} \to \mathit{Sig}^{\mathbb{I}\mu'}$. This maps a $\mathbb{I}\mu$ signature (Σ, I) to the $\mathbb{I}\mu'$ signature $(\Psi^{\xi}(\Sigma), I)$, and similarly for signature morphisms.*
- *For $\varphi \in \mathit{Sen}^{\mathcal{I}'}(\Sigma)$, let $\gamma^{\mathbb{I}\xi}_{(\Sigma,I)}(\varphi) = \gamma^{\xi}_{\Sigma}(\varphi)$, and for $c \in \Psi^{\mu'}(\Psi^{\xi}(\Sigma)) = \Psi^{\mu}(\Sigma)$ and $i \in I$, let $\gamma^{\mathbb{I}\xi}_{(\Sigma,I)}(c(i)) = c(i)$.*
- *$\delta^{\mathbb{I}\xi}_{(\Sigma,I)} : \mathit{Mod}^{\mathbb{I}\mu}(\Sigma, I) \to \mathit{Mod}^{\mathbb{I}\mu'}(\Psi^{\xi}(\Sigma), I)$ is defined by $\delta^{\mathbb{I}\xi}_{(\Sigma,I)}(M, f) = (\delta^{\xi}_{\Sigma}(M), f)$. The actions on model-morphisms and signature morphisms are defined similarly.*

Naturality of $\gamma^{\mathbb{I}\xi}$ and $\delta^{\mathbb{I}\xi}$, as well as the satisfaction condition for $\mathbb{I}\xi$ and preservation of composition and identities follow straightforwardly from:

- $\gamma^{\mathbb{I}\xi}_{(\Sigma,I)} = \gamma^{\xi}_{\Sigma} + 1_{\Psi^{\mu}(\Sigma)} \times 1_I$, and
- $\delta^{\mathbb{I}\xi}_{(\Sigma,I)} = \delta^{\xi} \times 1_{\mathsf{Set}(I, |\delta^{\mu}_{\Sigma}|)}$, where $1_{\mathsf{Set}(I, |\delta^{\mu}_{\Sigma}|)}$ is the identity on the function mapping individuals to the domain in $\overline{\mathcal{CH}}$.

We omit the details here.

In fact, $\mathbb{I}$ extends to a functor $\mathit{Inst}/\overline{\mathcal{CH}} \to \mathit{Inst}/\mathbb{I}(1_{\overline{\mathcal{CH}}})$ from description logics to description logics with individuals. Because $1_{\overline{\mathcal{CH}}} : \overline{\mathcal{CH}} \to \overline{\mathcal{CH}}$ is final in $\mathit{Inst}/\overline{\mathcal{CH}}$, and any $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ in Inst can be seen as $\mu : \mu \to 1_{\overline{\mathcal{CH}}}$ in $\mathit{Inst}/\overline{\mathcal{CH}}$, we get $\mathbb{I}\mu : \mathbb{I}\mu \to \mathbb{I}(1_{\overline{\mathcal{CH}}})$ as a morphism in Inst, and therefore an object in $\mathit{Inst}/\mathbb{I}(1_{\overline{\mathcal{CH}}})$. Similarly, any arrow $\xi : \mu \to \nu$ in $\mathit{Inst}/\overline{\mathcal{CH}}$ gives $\mathbb{I}\xi : \mathbb{I}\mu \to \mathbb{I}\nu$. This overloads the $\mathbb{I}$ functor, so we adopt the following:

Notation 4.2.15. *For a description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$, we write $\mathcal{I}{+}\iota$ for the institution $\mathbb{I}\mu$, and $\mu{+}\iota : \mathcal{I}{+}\iota \to \overline{\mathcal{CH}}{+}\iota$ for the description logic with individuals (i.e., for the application of the functor $\mathit{Inst}/\overline{\mathcal{CH}} \to \mathit{Inst}/\overline{\mathcal{CH}}{+}\iota$).*

For example, we refer to the institution of $\mathcal{EL}$ with individuals presented in Example 4.2.11 as $\mathcal{EL}{+}\iota$.

The final object in $\mathit{Inst}/\overline{\mathcal{CH}}$ is $\overline{\mathcal{CH}}{+}\iota$, and $\overline{\mathcal{CH}}{+}\iota$ gives us a 'minimal' description logic with individuals.

Example 4.2.16. *The institution $\overline{\mathcal{CH}}$+ι has as signatures pairs (C,I), where C is an ω-set and I is a set, and signature morphisms are pairs (f,g), where f is a family of functions $(f_n : C_n \to C'_n)_{n\in\omega}$ and $g : I \to I'$. A (C,I)-sentence is either $c \sqsubseteq d$, with $c,d \in C_n$, or is of the form $c(i)$, with $c \in C_n$ and $i \in I^n$. (C,I)-models are pairs (M,f), where M is a C-model in $\overline{\mathcal{CH}}$ and $f : I \to |M|$. $(M,f) \models_{(C,I)} c \sqsubseteq d$ iff $c^M \subseteq d^M$, and $(M,f) \models_{(C,I)} c(i)$ iff $\overline{f}(i) \in c^M$, where $c \in C_n$ and $i \in I^n$.*

It is natural to think of $\overline{\mathcal{CH}}$+ι as a description logic by forgetting about the individuals that are added to signatures, and we shall return to this idea in Definition 4.2.18 below. There is, however, another way of relating $\overline{\mathcal{CH}}$ and $\overline{\mathcal{CH}}$+ι, by viewing individuals as concepts. That is, we can construct a comorphism $\overline{cc} : \overline{\mathcal{CH}}+\iota \to \overline{\mathcal{CH}}$ as follows. $\Phi^{\overline{cc}}$ maps a $\overline{\mathcal{CH}}$+ι-signature (C,I) to the $\overline{\mathcal{CH}}$-signature $C+I$, and similarly for signature morphisms. Given a C-sentence e (which is also a (C,I)-sentence in $\overline{\mathcal{CH}}$+ι), $\alpha^{\overline{cc}}_C(e) = e$, and given a (C,I)-sentence $c(i)$, where $c \in C_n$ and $i \in I^n$, $\alpha^{\overline{cc}}_C(c(i)) = i \sqsubseteq c$. To translate $\overline{\mathcal{CH}}$-models to $\overline{\mathcal{CH}}$+ι-models, we use the following notation: for a set S, we write 2^S for the set of subsets of S and given $X \in 2^S$, we write $X{\downarrow}$ for the set of all subsets of X. Given a C-model M in $\overline{\mathcal{CH}}$, the (C,I)-model $\beta^{\overline{cc}}_C(M)$ has domain $2^{|M|}$, and given a concept $c \in C_n$, we set $c^{\beta^{\overline{cc}}_C(M)} = (c^M){\downarrow}$ and for $i \in I^n$ we set $i^{\beta^{\overline{cc}}_C(M)} = c^M$. It is straightforward to check that the satisfaction condition holds for $\overline{cc}$. In the later part we give comorphisms more attention, see Corollary 4.2.22 which shows how $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ gives rise to comorphism $\vec{\mu} : \mathcal{I} \to \mathcal{I}+\iota$, Definition 4.2.25 where we introduce comorphisms between description logics or Definition 4.2.26 where we introduce comorphisms between description logics with individuals. Now we return to morphisms.

The following observation follows from the discussion above.

Observation 4.2.17. *Given a description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$, the description logic with individuals $\mu+\iota : \mathcal{I}+\iota \to \overline{\mathcal{CH}}$ is also a description logic: we simply forget about the individuals. This gives a morphism $\mathcal{I}+\iota \to \mathcal{I}$, defined below, which composes with μ to give a morphism $\mathcal{I}+\iota \to \overline{\mathcal{CH}}$.*

Definition 4.2.18. *Given description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$, define the morphism $\mu^- : \mathcal{I}+\iota \to \mathcal{I}$ as follows.*

- *For $\mathcal{I}+\iota$-signature (Σ,I), set $\Psi^{\mu^-}(\Sigma,I) = \Sigma$, and similarly for signature morphisms.*
- *Any Σ-sentence φ is also a (Σ,I)-sentence, and we set $\gamma^{\mu^-}_{(\Sigma,I)}(\varphi) = \varphi$.*
- *Given a (Σ,I)-model (M,f), M is a Σ-model, and we set $\delta^{\mu^-}_{(\Sigma,I)}(M,f) = M$, and similarly for model-homomorphisms.*

While the existence of morphism μ^- was expected, as it shows how $\mathcal{I}+\iota$ is built over $\mathcal{I}$, there is also morphism μ^+, going in the opposite direction. At first this may seem surprising, but this morphism shows how close the relation between $\mathcal{I}$ and $\mathcal{I}+\iota$ is. Intuitively, a signature $\Sigma \in \mathsf{Sig}^{\mathcal{I}}$ gets mapped to $\mathcal{I}+\iota$-signature $(\Sigma,\emptyset)$. Then, from the fact that $\mathsf{Sen}^{\mathcal{I}+\iota}$ is a disjoint union $\mathsf{Sen}^{\mathcal{I}}(\Sigma) + \sum_{n\in\omega}(\Psi^{\mu}(\Sigma_n)\times I^n)$, together with the fact that I is empty we get that $\mathsf{Sen}^{\mathcal{I}}(\Sigma) = \mathsf{Sen}^{\mathcal{I}+\iota}(\Sigma, \emptyset)$. We also show how to construct a $(\Sigma, \emptyset)$-model in $\mathcal{I}+\iota$ given a Σ-model in $\mathcal{I}$. This is formally defined below.

Definition 4.2.19. *Given $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ define the morphism $\mu^+ : \mathcal{I} \to \mathcal{I}+\iota$ as follows.*

- *For $\mathcal{I}$-signature Σ, $\Psi^{\mu^+}(\Sigma) = (\Sigma, \emptyset)$.*
- *For any $\mathcal{I}$-signature Σ, $Sen^{\mathcal{I}+\iota}(\Sigma, \emptyset) = Sen^{\mathcal{I}}\Sigma + \Psi^{\mu}(\Sigma)\times\emptyset = Sen^{\mathcal{I}}(\Sigma)$, and we set $\gamma_{\Sigma}^{\mu^+}(\varphi) = \varphi$. So $\gamma_{\Sigma}^{\mu^+} : Sen^{\mathcal{I}+\iota}(\delta^{\mu^+}(\Sigma)) \to Sen^{\mathcal{I}}(\Sigma)$.*
- *Given a Σ-model M, we construct a $(\Sigma, \emptyset)$-model $\delta_{\Sigma}^{\mu^+}(M) = (M, \emptyset \hookrightarrow |M|)$, where $\emptyset \hookrightarrow |M|$ is the unique inclusion.*

Particular cases of these morphisms are $1^-_{\overline{\mathcal{CH}}} : \overline{\mathcal{CH}}+\iota \to \overline{\mathcal{CH}}$ and $1^+_{\overline{\mathcal{CH}}} : \overline{\mathcal{CH}} \to \overline{\mathcal{CH}}+\iota$.

The close relation between description logics and description logics with individuals is captured in

Proposition 4.2.20. *For any description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$, we have*

$$\begin{aligned} \mu^+ ; \mu^- &= 1_{\mathcal{I}} \\ \mu^+ ; \mu+\iota ; 1^-_{\overline{\mathcal{CH}}} &= \mu \\ \mu ; 1^+_{\overline{\mathcal{CH}}} &= \mu^+ ; \mu+\iota \\ \mu^- ; \mu &= \mu+\iota ; 1^-_{\overline{\mathcal{CH}}} \end{aligned}$$

Proof: This is a straightforward calculation from the definitions. □

Figure 4.10 is a graphical representation of the proposition above.

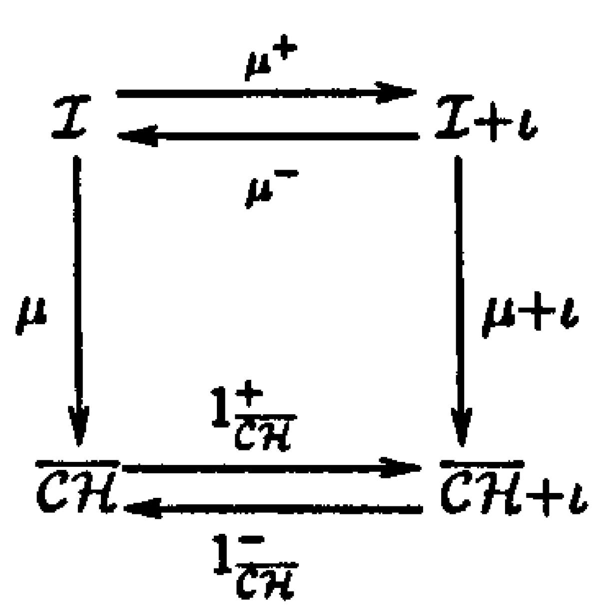


Figure 4.10: Morphisms between DLs and DLs with individuals

So far we showed that description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ generates a description logic with individuals $\mu+\iota : \mathcal{I}+\iota \to \overline{\mathcal{CH}}+\iota$. Then we defined morphisms $\mu^+ : \mathcal{I} \to \mathcal{I}+\iota$ and $\mu^- : \mathcal{I}+\iota \to \mathcal{I}$. As μ^+ and μ^- have opposite directions, functors Ψ^{μ^+} and Ψ^{μ^-} have opposite directions as well. Now we show that the functor $\Psi^{\mu^-} : (\Sigma, I) \mapsto \Sigma$ has left adjoint $\Psi^{\mu^+} : \Sigma \mapsto (\Sigma, \emptyset)$. This is expressed in the theorem below.

Theorem 4.2.21. *For any description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ we have that $\Psi^{\mu^+} \dashv \Psi^{\mu^-}$.*

Proof: The unit of the adjunction η_Σ is the identity, and the proof itself is a straightforward calculation from the definitions. □

The significance of the above theorem becomes clear in the light of results of Arrais and Fiadeiro in [5], showed that given an adjunction between signature categories of two institutions, an institution morphism gives rise to an institution comorphism (This was already mentioned in Section 2.4 when we were discussing adjunctions on signatures.). This result together with Theorem 4.2.21 gives us a comorphism $\vec{\mu} : \mathcal{I} \to \mathcal{I}{+}\iota$. This is made explicit in the corollary below.

Corollary 4.2.22. *For any description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$, there is a comorphism $\vec{\mu} = (\Psi^{\mu^+}, \alpha^{\vec{\mu}}, \beta^{\vec{\mu}}) : \mathcal{I} \to \mathcal{I}{+}\iota$, where $\alpha^{\vec{\mu}}$ is the natural transformation defined by $\alpha^{\vec{\mu}}_\Sigma = Sen(\eta_\Sigma)$; $\gamma^{\mu^-}_{\Psi^{\mu^+}(\Sigma)}$, recall that η_Σ is the identity so $\alpha^{\vec{\mu}}_\Sigma = \gamma^{\mu^-}_{\Psi^{\mu^+}(\Sigma)}$ and $\beta^{\vec{\mu}}$ is the natural transformation defined by $\beta^{\vec{\mu}}_\Sigma = \delta^{\mu^-}_{\Psi^{\mu^+}(\Sigma)}$; $Mod(\eta_\Sigma)$, and as $\eta_\Sigma = 1_\Sigma$ we have $\beta^{\vec{\mu}}_\Sigma = \delta^{\mu^-}_{\Psi^{\mu^+}(\Sigma)}$*

Now we show that $\vec{\mu}$ satisfies a condition that is even stronger than restriction adequateness [3] mentioned in Section 2.5. We will use this fact to show that $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ is conservative.

Proposition 4.2.23. *For any $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ and signatures $\Sigma, \Sigma' \in Sig^{\mathcal{I}}$ with a signature morphism $\sigma : \Sigma \to \Sigma'$ and $\mathcal{O} \subseteq Sen^{\mathcal{I}}(\Sigma')$ we have that*

$$\alpha^{\vec{\mu}}_\Sigma(\mathcal{O}_\sigma) = (\alpha^{\vec{\mu}}_{\Sigma'}(\mathcal{O}))_{\Phi^{\vec{\mu}}(\sigma)} .$$

Proof: In the proof we will use the fact that $\alpha^{\vec{\mu}}_\Sigma(\varphi) = \varphi$ and $\beta^{\vec{\mu}}_\Sigma(\mathcal{M}, f) = \mathcal{M}$.

For "⊆", assume $\mathcal{O} \models^{\mathcal{I}}_{\Sigma'} \sigma(\varphi)$ and $(\mathcal{M}, \_) \models^{\mathcal{I}+\iota}_{\Phi^{\vec{\mu}}(\Sigma')} \alpha^{\vec{\mu}}_{\Sigma'}(\mathcal{O})$, by satisfaction condition $\beta^{\vec{\mu}}_{\Sigma'}(\mathcal{M}, \_) \models^{\mathcal{I}}_{\Sigma'} \mathcal{O}$, thus $\beta^{\vec{\mu}}_{\Sigma'}(\mathcal{M}, \_) \models^{\mathcal{I}}_{\Sigma'} \sigma(\varphi)$, by satisfaction condition $(\mathcal{M}, \_) \models^{\mathcal{I}+\iota}_{\Phi^{\vec{\mu}}(\Sigma')} \alpha^{\vec{\mu}}_{\Sigma'}(\sigma(\varphi))$.

For "⊇", assume $\alpha^{\vec{\mu}}_{\Sigma'}(\mathcal{O}) \models^{\mathcal{I}+\iota}_{\Phi^{\vec{\mu}}(\Sigma')} \Phi^{\vec{\mu}}(\sigma)(\alpha^{\vec{\mu}}_\Sigma(\varphi))$ and $\mathcal{M} \models^{\mathcal{I}}_{\Sigma'} \mathcal{O}$, by the fact that $\beta^{\vec{\mu}}_{\Sigma'}(\mathcal{M}, \_) = \mathcal{M}$ we get $\beta^{\vec{\mu}}_{\Sigma'}(\mathcal{M}, \_) \models^{\mathcal{I}}_{\Sigma'} \mathcal{O}$, by satisfaction condition

$$(\mathcal{M}, \_) \models^{\mathcal{I}+\iota}_{\Phi^{\vec{\mu}}(\Sigma')} \alpha^{\vec{\mu}}_{\Sigma'}(\mathcal{O}) .$$

This implies $(\mathcal{M}, \_) \models^{\mathcal{I}+\iota}_{\Phi^{\vec{\mu}}(\Sigma')} \Phi^{\vec{\mu}}(\sigma)(\alpha^{\vec{\mu}}_\Sigma(\varphi))$, by naturality of $\alpha^{\vec{\mu}}$ we have

$$(\mathcal{M}, \_) \models^{\mathcal{I}+\iota}_{\Phi^{\vec{\mu}}(\Sigma')} \alpha^{\vec{\mu}}_{\Sigma'}(\sigma(\varphi))$$

and by satisfaction condition $\beta^{\vec{\mu}}_{\Sigma'}(\mathcal{M}, \_) \models^{\mathcal{I}}_{\Sigma'} \sigma(\varphi)$, thus $\mathcal{M} \models^{\mathcal{I}}_{\Sigma'} \sigma(\varphi)$. □

As for any $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ we have that $\Phi^{\vec{\mu}}(\Sigma) = (\Sigma, \emptyset)$, i.e. we have no individuals in $\mathcal{I}{+}\iota$-signature, it is not difficult to see that the natural transformation $\beta^{\vec{\mu}}$ is surjective. As mentioned before restriction adequateness is a sufficient condition for conservativity of comorphism, and Proposition 4.2.23 shows that $\vec{\mu}$ has an even stronger property. These two facts are expressed in the following lemma:

Lemma 4.2.24. *For any $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ the comorphism $\vec{\mu}$ is conservative and $\beta^{\vec{\mu}}$ is surjective.*

Although many institution morphisms have corresponding comorphisms that arise from adjoint functors on signatures and vice versa (cf. [5] and the remarks at the end of Section 2.4), it is not always the case. For instance, morphisms to $\overline{\mathcal{CH}}$ usually do not have a corresponding comorphism. On the other hand, in our work we are particularly interested in defining comorphisms between institutions as they are used for constructing frameworks and query bases (in Section 3.1 of Chapter 3 we presented arguments for using comorphisms for constructing frameworks and query bases). But using morphisms and comorphisms at the same time is inconvenient, as we cannot compose them. To avoid that problem but still be able to use comorphisms for constructing frameworks and query bases we introduce the notion of *DL*-comorphism (comorphisms for description logics). To formulate this notion first we use a special case of the notion of an institution modification between institution morphisms, which was presented by Diaconescu in [31]. To be more precise, we use a natural transformation between functors used in morphisms and commorphism of our interest. So our *DL*-comorphisms from $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ to $\nu : \mathcal{I}' \to \overline{\mathcal{CH}}$ are therefore comorphisms from $\mathcal{I}$ to $\mathcal{I}'$ as in Definition 2.4.28 in Chapter 2, together with families of functions $\tau^\eta_\Sigma : \Psi^\mu(\Sigma) \to \Psi^\nu(\Phi^\eta(\Sigma))$, for each signature Σ in $\mathsf{Sig}^\mathcal{I}$.

Definition 4.2.25. *A* ***DL-comorphism*** *from $\mathcal{I} \xrightarrow{\mu} \overline{\mathcal{CH}}$ to $\mathcal{I}' \xrightarrow{\nu} \overline{\mathcal{CH}}$ consists of a comorphism $\eta : \mathcal{I} \to \mathcal{I}'$ together with a natural transformation $\tau^\eta : \Psi^\mu \to \Phi^\eta ; \Psi^\nu$ such that for all Σ in $Sig^\mathcal{I}$,*

$$\delta^\nu_{\Phi^\eta(\Sigma)} ; Mod^{\overline{\mathcal{CH}}}(\tau^\eta_\Sigma) \; = \; \beta^\eta_\Sigma ; \delta^\mu_\Sigma \; . \tag{4.1}$$

For example, consider a comorphism ef (it was presented in Example 2.4.34) from $\mathsf{e}\overline{\mathsf{c}}$ ($\mathcal{EL}$) to $\mathsf{f}\overline{\mathsf{c}}$ (*FOL*). In this case, $\Psi^{\mathsf{e}\overline{\mathsf{c}}}$ maps an $\mathcal{EL}$ signature to the set of all $\mathcal{EL}$ concepts built from that signature. The functor Φ^{ef} would map an $\mathcal{EL}$ signature to the *FOL* signature that had the concept names as unary relations and the roles as binary relations. $\Psi^{\mathsf{f}\overline{\mathsf{c}}}$ would then map this signature to the set of all formulae with one free variable. Clearly, these sets are not equal (and this is one reason *not* to work in $Inst/\overline{\mathcal{CH}}$), but the first set can naturally be included in the second — this is the standard translation of $\mathcal{EL}$ into *FOL*— and this gives the natural transformation τ^{ef}. Note that each $\tau^{\mathsf{ef}}_\Sigma : \Psi^{\mathsf{e}\overline{\mathsf{c}}}(\Sigma) \to \Psi^{\mathsf{f}\overline{\mathsf{c}}}(\Phi^{\mathsf{ef}}(\Sigma))$ is actually a signature morphism in $\mathsf{Sig}^{\overline{\mathcal{CH}}}$, which means that we have the model reduct functor $\mathsf{Mod}^{\overline{\mathcal{CH}}}(\tau^{\mathsf{ef}}_\Sigma) : \mathsf{Mod}^{\overline{\mathcal{CH}}}(\Psi^{\mathsf{f}\overline{\mathsf{c}}}(\Phi^{\mathsf{ef}}(\Sigma))) \to \mathsf{Mod}^{\overline{\mathcal{CH}}}(\Psi^{\mathsf{e}\overline{\mathsf{c}}}(\Sigma))$. This takes a $\Psi^{\mathsf{f}\overline{\mathsf{c}}}(\Phi^{\mathsf{ef}}(\Sigma))$-model M and gives a $\Psi^{\mathsf{e}\overline{\mathsf{c}}}(\Sigma)$ model $M\restriction_{\tau^{\mathsf{ef}}_\Sigma}$ that has the same domain as M, and interprets concept $\psi \in \Phi^{\mathsf{ef}}(\Sigma)$ in the way that M interprets the concept $\tau^{\mathsf{ef}}_\Sigma(\psi)$. Equation (4.1) says that applying this reduct functor to the $\mathsf{f}\overline{\mathsf{c}}$-translation of a *FOL*-model gives the same result as applying the comorphism's translation, then the $\mathsf{e}\overline{\mathsf{c}}$-translation. In practice this is not a serious restriction, as models for most description logics are simply interpretative structures.

But Definition 4.2.25 does not capture individuals which are important in this part of our work. For that reason we introduce its variant, which takes individuals into account.

Definition 4.2.26. *A* ***DL+ι-comorphism*** *from $\mathcal{I} \xrightarrow{\mu} \overline{\mathcal{CH}}{+}\iota$ to $\mathcal{I}' \xrightarrow{\nu} \overline{\mathcal{CH}}{+}\iota$ consists of a comorphism $\eta : \mathcal{I} \to \mathcal{I}'$ together with a natural transformation $\tau^\eta : \Psi^\mu \to \Phi^\eta ; \Psi^\nu$ such that* *for any* Σ *in* $Sig^{\mathcal{I}}$,

$$\delta^{\nu}_{\Phi^{\eta}(\Sigma)} ; Mod^{\overline{\mathcal{CH}}+\iota}(\tau^{\eta}_{\Sigma}) = \beta^{\eta}_{\Sigma} ; \delta^{\mu}_{\Sigma} . \tag{4.2}$$

In fact both, Definition 4.2.25 and Definition 4.2.26 are just special cases of the definition of comorphism between objects of a slice category.

Definition 4.2.27. *For slice category* $Inst/C$ *a* C*-comorphism from* μ *to* ν *consists of a comorphism* $\eta : \mathcal{I} \to \mathcal{I}'$ *together with natural transformation* $\tau^{\eta} : \Psi^{\mu} \to \Phi^{\eta} ; \Psi^{\nu}$ *such that for all* Σ *in* $Sig^{\mathcal{I}}$,

$$\delta^{\nu}_{\Phi^{\eta}(\Sigma)} ; Mod^{\overline{\mathcal{CH}}}(\tau^{\eta}_{\Sigma}) = \beta^{\eta}_{\Sigma} ; \delta^{\mu}_{\Sigma} . \tag{4.3}$$

The following proposition shows that any DL-comorphism can be extended to DL+ι-comorphism, which is expected as we can extend any description logic to a description logic with individuals.

Proposition 4.2.28. *Any* DL*-comorphism* η *from* $\mathcal{I} \xrightarrow{\mu} \overline{\mathcal{CH}}$ *to* $\mathcal{I}' \xrightarrow{\nu} \overline{\mathcal{CH}}$ *extends to a* DL+ι*-comorphism* $\eta^{\iota} : \mu{+}\iota \to \nu{+}\iota$.

Proof: $\Phi^{\eta^{\iota}} = \Phi^{\eta} \times 1_{\mathsf{Set}}$. That is, $\Phi^{\eta^{\iota}}(\Sigma, I) = (\Phi^{\eta}(\Sigma), I)$.

Recall that $\mathsf{Sen}^{\mathcal{I}+\iota}(\Sigma, I) = \mathsf{Sen}^{\mathcal{I}}(\Sigma) + \Psi^{\mu}(\Sigma) \times I$. For $\varphi \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$, let $\alpha^{\eta^{\iota}}_{(\Sigma,I)}(\varphi) = \alpha^{\eta}_{\Sigma}(\varphi)$; for $\psi \in \Psi^{\mu}(\Sigma)$ and $i \in I$, set $\alpha^{\eta^{\iota}}_{(\Sigma,I)}(\psi(i)) = \tau^{\eta}_{\Sigma}(\psi)(i)$.

For any $(\Phi^{\eta}(\Sigma), I)$-model (M, f), let $\beta^{\eta^{\iota}}_{(\Sigma,I)}(M, f) = (\beta^{\eta}_{\Sigma}(M), f)$. Note that this is well-defined, because the codomain of f is $|\delta^{\nu}_{\Phi^{\eta}(\Sigma)}(M)|$, which is the same as $|\mathsf{Mod}^{\overline{\mathcal{CH}}}(\tau^{\eta}_{\Sigma})(\delta^{\nu}_{\Phi^{\eta}(\Sigma)}(M))|$, and by (4.1) in Definition 4.2.25, this is the same as $|\delta^{\mu}_{\Sigma}(\beta^{\eta}_{\Sigma}(M))|$.

We show the satisfaction condition for the case where the $\mathcal{I}$+ι sentence is a predicate on an individual:

$$\beta^{\eta^{\iota}}_{(\Sigma,I)}(M, f) \models^{\mathcal{I}+\iota} \psi(i)$$

is, by definition,

$$(\beta^{\eta}_{\Sigma}(M), f) \models^{\mathcal{I}+\iota} \psi(i)$$

which is, again, by definition,

$$f(i) \in \psi^{\delta^{\mu}_{\Sigma}(\beta^{\eta}_{\Sigma}(M))}$$

by (4.1) in Definition 4.2.25, this is the same as

$$f(i) \in \tau^{\eta}_{\Sigma}(\psi)^{\delta^{\nu}_{\Sigma}(M)}$$

which is, by definition,

$$(M, f) \models^{\mathcal{I}'+\iota} \tau^{\eta}_{\Sigma}(\psi)(i)$$

and this is

$$(M, f) \models^{\mathcal{I}'+\iota} \alpha^{\eta^{\iota}}_{(\Sigma,I)}(\psi(i)) .$$

Now we need only to define natural transformation $\tau^{\eta^{\iota}} : \Psi^{\mu+\iota} \to \Phi^{\eta^{\iota}} ; \Psi^{\nu+\iota}$. First note that $\Psi^{\mu+\iota} = \Psi^{\mu} \times 1_{\mathsf{Set}}$, $\Psi^{\nu+\iota} = \Psi^{\nu} \times 1_{\mathsf{Set}}$ and $\Phi^{\eta^{\iota}} = \Phi^{\eta} \times 1_{\mathsf{Set}}$. So we can define $\tau^{\eta^{\iota}}_{(\Sigma,I)} = \tau^{\eta}_{\Sigma} \times 1_{\mathsf{Set}}$ and we have

$$\tau^{\eta^{\iota}}_{(\Sigma,I)} : \Psi^{\mu+\iota}(\Sigma, I) \to \Psi^{\nu+\iota}(\Phi^{\eta^{\iota}}(\Sigma, I)),$$

i.e.

$$\tau^{\eta^\iota}_{(\Sigma,I)} : (\Psi^\mu(\Sigma), I) \to (\Psi^\nu(\Phi^\eta(\Sigma)), I).$$

So showing

$$\delta^{\nu+\iota}_{\Phi^{\eta^\iota}(\Sigma,I)} ; \mathsf{Mod}^{\overline{\mathcal{CH}}+\iota}(\tau^{\eta^\iota}_{(\Sigma,I)}) = \beta^{\eta^\iota}_{(\Sigma,I)} ; \delta^{\mu+\iota}_{(\Sigma,I)}$$

is straightforward as we have

$$\delta^{\nu}_{\Phi^\eta(\Sigma)} ; \mathsf{Mod}^{\overline{\mathcal{CH}}}(\tau^{\eta}_{(\Sigma)}) = \beta^{\eta}_{\Sigma} ; \delta^{\mu}_{\Sigma} .$$

□

Now we show that for any description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$, comorphism $\vec{\mu}$ can be treated as a DL-comorphism from μ to $\mu+\iota ; 1^-_{\overline{\mathcal{CH}}}$. This theorem uses the fact that composition of comorphisms is a comorphism.

Theorem 4.2.29. *For any description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ comorphism $\vec{\mu}$ is a DL-comorphism from μ to $\mu+\iota ; 1^-_{\overline{\mathcal{CH}}}$.*

Proof: As Corollary 4.2.22 guarantees us the existence of comorphism $\vec{\mu} : \mathcal{I} \to \mathcal{I}+\iota$, so we need only define $\tau^{\vec{\mu}} : \Psi^\mu \to \Phi^{\vec{\mu}}; \Psi^{\mu+\iota;1^-_{\overline{\mathcal{CH}}}}$ such that

$$\delta^{\mu+\iota;1^-_{\overline{\mathcal{CH}}}}_{\Phi^{\vec{\mu}}(\Sigma)} ; \mathsf{Mod}^{\overline{\mathcal{CH}}}(\tau^{\vec{\mu}}_{\Sigma}) = \beta^{\vec{\mu}}_{\Sigma} ; \delta^{\mu}_{\Sigma} .$$

Note that from Corollary 4.2.22 it follows that $\Phi^{\vec{\mu}} = \Psi^{\mu^+}$. From Proposition 4.2.20 it immediately follows that for any $\Sigma \in \mathsf{Sig}^{\mathcal{I}}$ we have:

$$\Psi^{1^-_{\overline{\mathcal{CH}}}}(\Psi^{\mu+\iota}(\Psi^{\mu^+}(\Sigma))) = \Psi^\mu(\Sigma).$$

So we can define

$$\tau^{\vec{\mu}}_{\Sigma} = 1_{\Psi^\mu(\Sigma)} : \Psi^\mu(\Sigma) \to \Psi^{\mu^+} ; \Psi^{\mu+\iota;1^-_{\overline{\mathcal{CH}}}}(\Sigma),$$

i.e.

$$\tau^{\vec{\mu}}_{\Sigma} : \Psi^\mu(\Sigma) \to \Psi^{1^-_{\overline{\mathcal{CH}}}}(\Psi^{\mu+\iota}(\Psi^{\mu^+}(\Sigma))).$$

We now only need to show that

$$\delta^{\mu+\iota;1^-_{\overline{\mathcal{CH}}}}_{\Phi^{\vec{\mu}}(\Sigma)} ; \mathsf{Mod}^{\overline{\mathcal{CH}}}(\tau^{\vec{\mu}}_{\Sigma}) = \beta^{\vec{\mu}}_{\Sigma} ; \delta^{\mu}_{\Sigma} .$$

We have that:

$$\delta^{\mu+\iota;1^-_{\overline{\mathcal{CH}}}}_{\Phi^{\vec{\mu}}(\Sigma)} ; \mathsf{Mod}^{\overline{\mathcal{CH}}}(\tau^{\vec{\mu}}_{\Sigma}) = \delta^{\mu+\iota;1^-_{\overline{\mathcal{CH}}}}_{\Phi^{\vec{\mu}}(\Sigma)} = \delta^{\mu+\iota;1^-_{\overline{\mathcal{CH}}}}_{(\Sigma,\emptyset)} = \delta^{\mu+\iota}_{(\Sigma,\emptyset)} ; \delta^{1^-_{\overline{\mathcal{CH}}}}_{(\Psi^\mu(\Sigma),\emptyset)}.$$

Note that for any $\mathcal{I}+\iota$-model $(\mathcal{M}, f)$ over (Σ, I) we have

$$\delta^{1^-_{\overline{\mathcal{CH}}}}_{(\Psi^\mu(\Sigma),\emptyset)}(\delta^{\mu+\iota}_{(\Sigma,\emptyset)}(\mathcal{M}, f)) = \delta^{1^-_{\overline{\mathcal{CH}}}}_{(\Psi^\mu(\Sigma),\emptyset)}(\delta^{\mu}_{\Sigma}(\mathcal{M}), f) = \delta^{\mu}_{\Sigma}(\mathcal{M})$$

as well as

$$\delta^{\mu}_{\Sigma}(\beta^{\vec{\mu}}_{\Sigma}(\mathcal{M}, f)) = \delta^{\mu}_{\Sigma}(\mathcal{M}),$$

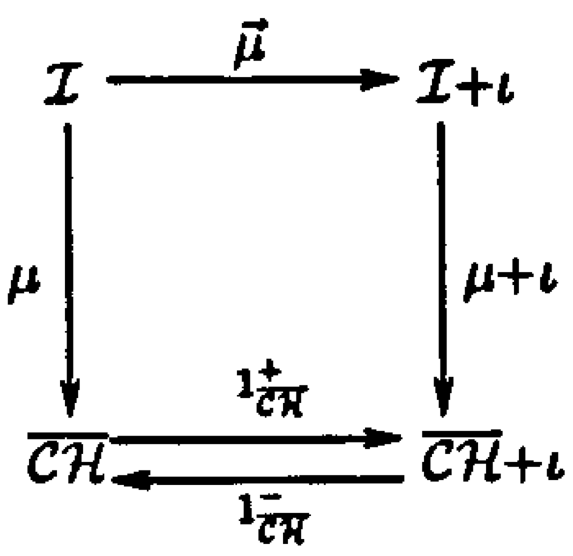


Figure 4.11.

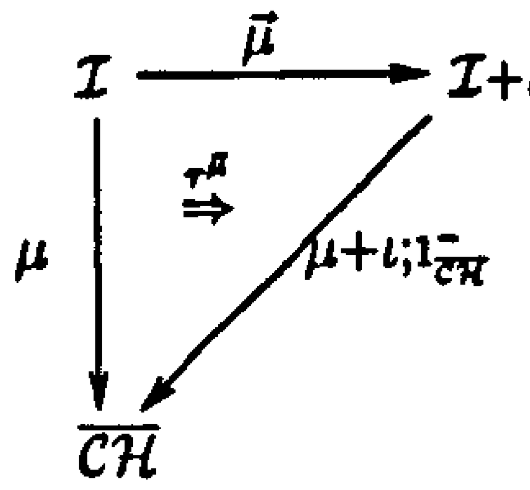


Figure 4.12.

thus

$$\delta^{\mu+\iota}_{(\Sigma,\emptyset)} ; \delta^{1^{-}_{\overline{\mathcal{CH}}}}_{(\Psi^{\mu}(\Sigma),\emptyset)} = \beta^{\vec{\mu}}_{\Sigma} ; \delta^{\mu}_{\Sigma} ,$$

and so

$$\delta^{\mu+\iota;1^{-}_{\overline{\mathcal{CH}}}}_{\Phi^{\vec{\mu}}(\Sigma)} ; \mathsf{Mod}^{\overline{\mathcal{CH}}}(\tau^{\vec{\mu}}_{\Sigma}) = \beta^{\vec{\mu}}_{\Sigma} ; \delta^{\mu}_{\Sigma} ,$$

as desired. □

Similarly we can show that $\vec{\mu}$ is also a DL+ι-comorphism.

Theorem 4.2.30. *For any description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ DL-comorphism $\vec{\mu}$ is also a DL+ι-comorphism from $\mu ; 1^{+}_{\overline{\mathcal{CH}}} : \mathcal{I} \to \overline{\mathcal{CH}}+\iota$ to $\mu+\iota : \mathcal{I}+\iota \to \overline{\mathcal{CH}}+\iota$.*

Proof: We only need to define $\tau^{\vec{\mu}} : \Psi^{\mu;1^{+}_{\overline{\mathcal{CH}}}} \to \Phi^{\vec{\mu}} ; \Psi^{\mu+\iota}$ such that for any Σ in $\mathsf{Sig}^{\mathcal{I}}$,

$$\delta^{\mu+\iota}_{\Phi^{\vec{\mu}}\Sigma} ; \mathsf{Mod}^{\overline{\mathcal{CH}}+\iota}(\tau^{\vec{\mu}}_{\Sigma}) = \beta^{\vec{\mu}}_{\Sigma} ; \delta^{\mu;1^{+}_{\overline{\mathcal{CH}}}}_{\Sigma} .$$

Note that $\Psi^{\mu;1^{+}_{\overline{\mathcal{CH}}}} = \Psi^{\mu} ; \Psi^{1^{+}_{\overline{\mathcal{CH}}}}$ and $\Phi^{\vec{\mu}} ; \Psi^{\mu+\iota} = \Psi^{\mu^{+}} ; \Psi^{\mu+\iota}$, where the latter follows from Corollary 4.2.22. So for any $\Sigma \in \mathsf{Sig}^{\mathcal{I}}$ we have:

$$\Psi^{1^{+}_{\overline{\mathcal{CH}}}}(\Psi^{\mu}(\Sigma)) = (\Psi^{\mu}(\Sigma), \emptyset)$$

and

$$\Psi^{\mu+\iota}(\Phi^{\vec{\mu}}(\Sigma)) = \Psi^{\mu+\iota}(\Psi^{\mu^{+}}(\Sigma)) = \Psi^{\mu+\iota}(\Sigma, \emptyset) = (\Psi^{\mu}(\Sigma), \emptyset) .$$

So we can define

$$\tau^{\vec{\mu}}_{\Sigma} = 1_{\Psi^{\mu;1^{+}_{\overline{\mathcal{CH}}}}(\Sigma)} : \Psi^{\mu;1^{+}_{\overline{\mathcal{CH}}}}(\Sigma) \to \Psi^{\mu+\iota}(\Phi^{\vec{\mu}}(\Sigma)) ,$$

i.e.

$$\tau^{\vec{\mu}}_{\Sigma} : \Psi^{1^{+}_{\overline{\mathcal{CH}}}}(\Psi^{\mu}(\Sigma)) \to \Psi^{\mu+\iota}(\Phi^{\vec{\mu}}(\Sigma)) .$$

We now only need to show that

$$\delta^{\mu+\iota}_{\Phi^{\vec{\mu}}(\Sigma)} \,;\, \mathsf{Mod}^{\overline{\mathcal{CH}}+\iota}(\tau^{\vec{\mu}}_{\Sigma}) \;=\; \beta^{\vec{\mu}}_{(\Sigma)} \,;\, \delta^{\mu;1^{+}_{\overline{\mathcal{CH}}}}_{(\Sigma)} \,.$$

Note that we have that:

$$\delta^{\mu+\iota}_{\Phi^{\vec{\mu}}(\Sigma)} \,;\, \mathsf{Mod}^{\overline{\mathcal{CH}}+\iota}(\tau^{\vec{\mu}}_{\Sigma}) = \delta^{\mu+\iota}_{\Phi^{\vec{\mu}}(\Sigma)} = \delta^{\mu+\iota}_{\Psi^{\mu^{+}}(\Sigma)} = \delta^{\mu+\iota}_{(\Sigma,\emptyset)}$$

and

$$\beta^{\vec{\mu}}_{\Sigma} \,;\, \delta^{\mu;1^{+}_{\overline{\mathcal{CH}}}}_{\Sigma} = \beta^{\vec{\mu}}_{\Sigma} \,;\, \delta^{\mu}_{\Sigma} \,;\, \delta^{1^{+}_{\overline{\mathcal{CH}}}}_{\Psi^{\mu}(\Sigma)} \,.$$

Note that for any $\mathcal{I}+\iota$-model $(\mathcal{M}, f)$ over (Σ, I) we have

$$\delta^{\mu+\iota}_{(\Sigma,\emptyset)}(\mathcal{M}, f) = (\delta^{\mu}_{\Sigma}(\mathcal{M}), f)$$

as well as

$$\delta^{1^{+}_{\overline{\mathcal{CH}}}}_{\Psi^{\mu}(\Sigma)}(\delta^{\mu}_{\Sigma}(\beta^{\vec{\mu}}_{\Sigma}(\mathcal{M}, f))) = \delta^{1^{+}_{\overline{\mathcal{CH}}}}_{\Psi^{\mu}(\Sigma)}(\delta^{\mu}_{\Sigma}(\mathcal{M})) = (\delta^{\mu}_{\Sigma}(\mathcal{M}), f)$$

thus

$$\delta^{\mu+\iota}_{\Phi^{\vec{\mu}}(\Sigma)} = \beta^{\vec{\mu}}_{\Sigma} \,;\, \delta^{\mu}_{\Sigma} \,;\, \delta^{1^{+}_{\overline{\mathcal{CH}}}}_{\Psi^{\mu}(\Sigma)} \,,$$

and so

$$\delta^{\mu+\iota}_{\Phi^{\vec{\mu}}(\Sigma)} \,;\, \mathsf{Mod}^{\overline{\mathcal{CH}}+\iota}(\tau^{\vec{\mu}}_{\Sigma}) \;=\; \beta^{\vec{\mu}}_{(\Sigma)} \,;\, \delta^{\mu;1^{+}_{\overline{\mathcal{CH}}}}_{(\Sigma)} \,,$$

as desired. □

So far we showed that any description logic can be extended to a description logic with individuals, this was presented in Definition 4.2.10. This definition tells us not only how to construct description logics with individuals, but also shows that for description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ with $\Sigma \in \mathsf{Sig}^{\mathcal{I}}$ sentences of $\mu+\iota : \mathcal{I}+\iota \to \overline{\mathcal{CH}}+\iota$ are $\mathsf{Sen}^{\mathcal{I}}(\Sigma) + \sum_{n\in\omega}(\Psi^{\mu}(\Sigma_n)\times I^n)$, i.e. a disjoint union of original Σ-sentences in $\mathcal{I}$ and a set of pairs (c, i) with $c \in \Psi^{\mu}(\Sigma)_n$ an n-ary concept and $i \in I^n$ a tuple of individuals. The set $\sum_{n\in\omega}(\Psi^{\mu}(\Sigma_n)\times I^n)$ provides the information about individuals, this makes it exactly what we defined as an ABox in Section 2.2. This justifies the following notation:

Notation 4.2.31. *Because $Sen^{\mu+\iota}$ is a disjoint union, any set $S \subseteq Sen^{\mu+\iota}(\Sigma, I)$ can be presented as a triple $(\mathcal{O}, \mathcal{R}, \mathcal{A})$, where $\mathcal{O}$ consists of all the concept inclusions over Σ in S, $\mathcal{R}$ consists of all the role inclusion axioms over Σ in S, and $\mathcal{A}$ consists of all the sentences of the form $c(i)$ with $c \in \Psi^{\mu}(\Sigma_n)$ and $i \in I^n$, where $n \in \omega$; we call $\mathcal{O}$ the* **ontology**, *$\mathcal{R}$ the* **RBox** *and $\mathcal{A}$ the* **Abox** *of the set S.*

4.2.3 Query conservativity and query expansion.

We now investigate the interaction between ontologies and ABoxes. Firstly, it should be clear that after adding an ABox to an ontology we will still be able to derive all consequences of the ontology. This is stated by the following result.

Proposition 4.2.32. *For any $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$, any signature $\Sigma \in Sig^{\mathcal{I}}$, any ontology $\mathcal{O} \subseteq Sen^{\mathcal{I}}(\Sigma)$, $\varphi \in Sen^{\mathcal{I}}(\Sigma)$ and any ABox $\mathcal{A}$, the following holds:*

$$\mathcal{O} \models^{\mathcal{I}}_{\Sigma} \varphi \text{ implies } (\mathcal{O}, \mathcal{A}) \models^{\mathcal{I}+\iota}_{\Phi^{\mu}(\Sigma)} \varphi.$$

Proof: Assume $\mathcal{O} \models^{\mathcal{I}}_{\Sigma} \varphi$.

$$\begin{aligned} \text{If } (\mathcal{M}, f) &\models^{\mathcal{I}+\iota}_{\Phi^{\mu}(\Sigma)} (\mathcal{O}, \mathcal{A}), \\ \text{then } \mathcal{M} &\models^{\mathcal{I}}_{\Sigma} \mathcal{O}, \\ \text{so } \mathcal{M} &\models^{\mathcal{I}}_{\Sigma} \varphi, \\ \text{so } (\mathcal{M}, f) &\models^{\mathcal{I}+\iota}_{\Phi^{\mu}(\Sigma)} \varphi, \end{aligned}$$

□

The converse does not always hold. To investigate this, we introduce the definition of **query conservativity**.

Definition 4.2.33. *A description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ is **query conservative** iff for all (Σ, I) in $Sig^{\mathcal{I}+\iota}$ and all $(\mathcal{O}, \mathcal{A}) \subseteq Sen^{\mathcal{I}+\iota}(\Sigma, I)$ and $\varphi \in Sen^{\mathcal{I}}(\Sigma)$*

$$(\mathcal{O}, \mathcal{A}) \models^{\mathcal{I}+\iota}_{(\Sigma,I)} \varphi \quad \textit{iff} \quad \mathcal{O} \models^{\mathcal{I}}_{\Sigma} \varphi \,.$$

The following example shows that $\mathcal{ALC}$ is not query conservative.

Example 4.2.34. *For $\mathcal{O} = \{A \sqsubseteq \forall r.\neg A, \neg A \sqsubseteq \forall r.A\}$ and $\mathcal{A} = \{r(a,a)\}$ we receive that $(\mathcal{O}, \mathcal{A}) \models \top \sqsubseteq \bot$ but $\mathcal{O} \not\models \top \sqsubseteq \bot$.*

Now we define the notion of query expansion. Roughly speaking, if description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ has query expansion, then for any ABox $\mathcal{A} \subseteq \mathsf{Sen}^{\mathcal{I}+\iota}(\Sigma, I)$ we have that for every Σ-model M in $\mathcal{I}$ there is a (Σ, I)-model $M^{\mathcal{A}}$ in $\mathcal{I}+\iota$, which satisfies $\mathcal{A}$ and its reduct to $\mathcal{I}$ satisfies exactly the same Σ-sentences as M does.

Definition 4.2.35. *A description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ has **query expansion** iff for all (Σ, I) in $Sig^{\mathcal{I}+\iota}$ and all $\mathcal{A} \subseteq \sum_{n\in\omega}(\Psi^{\mu}(\Sigma_n) \times I^n)$, for every Σ-model M, there is a (Σ, I)-model $M^{\mathcal{A}}$ such that $M^{\mathcal{A}} \models^{\mathcal{I}+\iota}_{(\Sigma,I)} \mathcal{A}$ and $\delta^{\mu^-}_{(\Sigma,I)}(M^{\mathcal{A}}) \equiv_{\Sigma} M$, where $M' \equiv_{\Sigma} M$ means M' and M satisfy exactly the same Σ-sentences.*

Now we show that query expansion implies query conservativity.

Proposition 4.2.36. *If μ has query expansion, then μ is query conservative.*
Proof: *Suppose $(\mathcal{O}, \mathcal{A}) \models^{\mathcal{I}+\iota}_{(\Sigma,I)} \varphi$, and $M \models^{\mathcal{I}}_{\Sigma} \mathcal{O}$, we need to show $M \models^{\mathcal{I}}_{\Sigma} \varphi$. If μ has query expansion, then there is a model $M^{\mathcal{A}}$ such that $M^{\mathcal{A}} \models^{\mathcal{I}+\iota}_{(\Sigma,I)} \mathcal{A}$ and $\delta^{\mu^-}_{(\Sigma,I)}(M^{\mathcal{A}}) \models^{\mathcal{I}}_{\Sigma} \mathcal{O}$, therefore $\delta^{\mu^-}_{(\Sigma,I)}(M^{\mathcal{A}}) \models^{\mathcal{I}}_{\Sigma} \varphi$, this implies that $M \models^{\mathcal{I}}_{\Sigma} \varphi$ as desired.* □

Now we show query expansion for $\mathcal{EL}$ and $\mathcal{EL}^{+}_{3}$, by Proposition 4.2.36 it also proves query conservativity.

First we show that $\mathsf{e\bar{c}} : \mathcal{EL} \to \overline{\mathcal{CH}}$ has query expansion property.

Theorem 4.2.37. *Description logic* $\mathsf{e}\overline{\mathsf{c}} : \mathcal{EL} \to \overline{\mathcal{CH}}$ *has query expansion. In other words, every* $\mathcal{EL}$ Σ*-model* $\mathcal{M}$ *can be extended to an* $\mathcal{EL}{+}\iota$ $\Phi^{\mathsf{e}\overline{\mathsf{c}}}(\Sigma)$*-model* $(\mathcal{M}', f)$*, s.t. for any* $\varphi \in Sen^{\mathcal{EL}}(\Sigma)$ *the following holds :*

$$(\mathcal{M}, f) \models^{\mathcal{EL}+\iota}_{\Phi^{\mathsf{e}\overline{\mathsf{c}}}(\Sigma)} \varphi \text{ iff } \mathcal{M} \models^{\mathcal{EL}}_{\Sigma} \varphi.$$

To prove the above theorem we need an auxiliary lemma, but first we show how $\mathcal{M}'$ is constructed, given $\mathcal{EL}$-model $\mathcal{M} = (|\mathcal{M}|, \cdot)$, we define:

- $|\mathcal{M}'| = |\mathcal{M}| \uplus I$,
- for any atomic concept p, we define $p^{(\mathcal{M}',f)} = p^{\mathcal{M}} \uplus I$,
- for any role r we define:

$$r^{(\mathcal{M}',f)}(x,y) \quad \text{iff} \quad r^{\mathcal{M}}(x,y), \text{ for any } x,y \in |\mathcal{M}| \quad \text{or} \quad \top \text{ for any } x,y \in I$$

Lemma 4.2.38. *For any* $\mathcal{EL}$*-concept* C *we have* $C^{(\mathcal{M}',f)} = C^{\mathcal{M}} \uplus I$.

Proof: The proof is by induction:
Let:

- $C = p$, it is immediate that $p^{(\mathcal{M}',f)} = p^{\mathcal{M}} \uplus I$,
- $C = C_1 \sqcap C_2$ this case is trivial,
- for $C = \exists r.C'$ we have that:

$$\begin{array}{rcl} \exists r.C'^{(\mathcal{M}',f)} & = & \{x \in |(\mathcal{M}',f)| \mid \text{ for some } y,\ r^{(\mathcal{M}',f)}(x,y) \wedge y \in C'^{(\mathcal{M}',f)}\} \\ & = & (\{x \in |\mathcal{M}| \mid (\exists y) r^{\mathcal{M}}(x,y) \wedge y \in C'^{\mathcal{M}}\} = \exists r.C'^{\mathcal{M}}) \\ & \uplus & (\{x \in I \mid \text{ for some } y,\ r^{(\mathcal{M}',f)}(x,y) \wedge y \in C'^{(\mathcal{M}',f)}\} = I) \\ \text{i.e., } \exists r.C'^{(\mathcal{M},f)} & = & \exists r.C'^{\mathcal{M}} \uplus I \end{array}$$

□

Proof: Now we can prove Theorem 4.2.37. Let $\varphi = C \sqsubseteq D$, where $C, D \in \mathsf{Con}^{\mathcal{EL}}(\Sigma)$.

$$(\mathcal{M}', f) \ \models^{\mathcal{EL}+\iota}_{\Phi^{\mathsf{e}\overline{\mathsf{c}}}(\Sigma)} \ C \sqsubseteq D$$

iff

$$C^{(\mathcal{M}',f)} \subseteq D^{(\mathcal{M}',f)}$$

iff

$$C^{\mathcal{M}} \uplus I \subseteq D^{\mathcal{M}} \uplus I$$

iff

$$C^{\mathcal{M}} \subseteq D^{\mathcal{M}}$$

iff

$$\mathcal{M} \ \models^{\mathcal{EL}}_{\Sigma} \ C \sqsubseteq D$$

□

Similarly we show that $\mathsf{e}^{+}\overline{\mathsf{c}} : \mathcal{EL}^{+} \to \overline{\mathcal{CH}}$ has query expansion property.

Theorem 4.2.39. *Description logic* $e^{+}\bar{c} : \mathcal{EL}^{+} \to \overline{\mathcal{CH}}$ *has query expansion property. In other words every* $\mathcal{EL}^{+}$ Σ*-model* $\mathcal{M}$ *can be extended to a* $\Phi^{\vec{e^{+}\bar{c}}}(\Sigma)$*-model* $(\mathcal{M}', f)$ *in* $\mathcal{EL}^{+}+\iota$*, s.t. for any* $\varphi \in Sen^{\mathcal{EL}^{+}}(\Sigma)$ *the following holds:*

$$(\mathcal{M}', f) \models^{\mathcal{EL}^{+}+\iota}_{(\Sigma,I)} \varphi \text{ iff } \mathcal{M} \models^{\mathcal{EL}^{+}}_{\Sigma} \varphi .$$

Proof: Note that by Lemma 4.2.38 for any $\mathcal{EL}^{+}$-concept C we have $C^{(\mathcal{M},f)} = C^{\mathcal{M}} \uplus I$.

We distinguish two cases:

1. $\varphi = C \sqsubseteq D$ where $C, D \in \mathsf{Con}^{\mathcal{EL}^{+}}(\Sigma)$, but this was already proven in Theorem 4.2.37.
2. $\varphi = r_1 \circ \cdots \circ r_n \sqsubseteq r$ with $n \geq 1$, then we have that

$$r_1^{(\mathcal{M},f)}(x_0, x_1), \ldots, r_n^{(\mathcal{M},f)}(x_{n-1}, x_n) \text{ and } r^{(\mathcal{M},f)}(x_0, x_n) :$$

$$\begin{array}{ll} & (\mathcal{M}, f) \models^{\mathcal{EL}^{+}+\iota}_{(\Sigma,I)} r_1 \circ \cdots \circ r_n \sqsubseteq r \\ \text{iff} & \\ & \text{for } x_0, x_1, \ldots, x_n \in |\mathcal{M}| \\ & r_1^{(\mathcal{M},f)} \circ \cdots \circ r_n^{(\mathcal{M},f)} \subseteq r^{(\mathcal{M},f)} \\ & (\text{for } x_0, x_1, \ldots, x_n \in I \text{ it is trivial}) \\ \text{iff} & \\ & r_1^{\mathcal{M}} \circ \cdots \circ r_n^{\mathcal{M}} \subseteq r^{\mathcal{M}} \\ \text{iff} & \\ & \mathcal{M} \models^{\mathcal{EL}^{+}}_{\Sigma} r_1 \circ \cdots \circ r_n \sqsubseteq r \end{array}$$

Cases (1) and (2) show that $(\mathcal{M}, f) \models^{\mathcal{EL}^{+}+\iota}_{(\Sigma,I)} \varphi$ iff $\mathcal{M} \models^{\mathcal{EL}^{+}}_{\Sigma} \varphi$ as required. □

4.2.4 Concept interpolation

In this section we introduce the notion of **concept interpolation.** We could say that given an assertion $\varphi(i)$ about an individual, which is a consequence of a knowledge base $(\mathcal{O}, \mathcal{A})$, concept interpolation 'splits' the consequence into two types of reasoning: determine whether there is ψ such that $\psi(i)$ is a consequence of $\mathcal{A}$ and $\psi \sqsubseteq \varphi$ is a consequence of $\mathcal{O}$.

Before we can formulate the notion of concept interpolation we have to define what does it mean for $\psi(i)$ to be a consequence of $\mathcal{A}$.

Definition 4.2.40. *Given* $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$*, ABox* $\mathcal{A}$*,* $\psi \in \Psi^{\mu}(\Sigma_n)_{n\in\omega}$ *and* $i \in I^n$*, we say that* $\psi(i)$ *is a consequence of* $\mathcal{A}$ *w.r.t.* μ *and write* $\mathcal{A} \models^{\mu}_{(\Sigma,I)} \psi(i)$ *if for every* Σ*-model* M *in* $\mathcal{I}$ *and every* $f : I \to |\delta^{\mu}_{\Sigma}(M)|$

$$(\delta^{\mu}_{\Sigma}(M), f) \models^{\overline{\mathcal{CH}}+\iota}_{(\Psi^{\mu}(\Sigma),I)} \mathcal{A} \quad \textit{implies} \quad (\delta^{\mu}_{\Sigma}(M), f) \models^{\overline{\mathcal{CH}}+\iota}_{(\Psi^{\mu}(\Sigma),I)} \psi(i) .$$

Now we present the notion of concept interpolation. Recall that, intuitively, concept interpolation tells us that given $(\mathcal{O}, \mathcal{A}) \subseteq \mathsf{Sen}^{\mathcal{I}+\iota}(\Sigma, I)$ and $\varphi(i) \in \mathsf{Sen}^{\mathcal{I}+\iota}(\Sigma', I)$ such that $\varphi(i)$ is a consequence of $(\mathcal{O}, \mathcal{A})$ in $\mathcal{I}+\iota$, then we can find a finite interpolant $\psi \in \Psi^{\mu}(\Sigma)$, such that $\psi(i)$ is a consequence of $\mathcal{A}$ w.r.t. μ and $\psi \sqsubseteq \varphi$ is a consequence of $\mathcal{O}$ in $\mathcal{I}$. So the idea behind this notion is very similar to that of Craig Interpolation.

Definition 4.2.41. *Let $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ be a description logic. We say that μ has* **concept interpolation** *if for any signatures (Σ, I) and (Σ', I) in $Sig^{\mathcal{I}+\iota}$, with $\Sigma \hookrightarrow \Sigma'$, and any $(\mathcal{O}, \mathcal{A}) \subseteq Sen^{\mathcal{I}+\iota}(\Sigma, I)$, $\varphi \in (\Psi^\mu(\Sigma'))_n$, and $i \in I^n$, if $(\mathcal{O}, \mathcal{A}) \models^{\mathcal{I}+\iota}_{(\Sigma', I)} \varphi(i)$, then there exists $\psi \in (\Psi^\mu(\Sigma))_n$ such that $\mathcal{A} \models^\mu_{(\Sigma, I)} \psi(i)$ and $\mathcal{O} \models^{\mathcal{I}}_{\Sigma'} \gamma^\mu_{\Sigma'}(\psi \sqsubseteq \varphi)$.*

As already shown in [62] $\mathcal{EL}+\iota$ has concept interpolation. In a very similar way it is possible to show that $\mathcal{EL}^+_0+\iota$ has concept interpolation. On the other hand, many description logics of higher expressivity do not have this property. For instance $\mathcal{ALC}$, as it is possible in $\mathcal{ALC}$ that after adding a consistent ABox to a consistent TBox we could receive an inconsistent KBox, this was already presented in [50].

Example 4.2.42. *For comorphism $\mathsf{e}_2^+\overline{\mathsf{c}}+\iota$ as defined in Example 4.2.8 we are unable to find concept interpolant, as we could have the following: $\mathcal{O} = \{r_1 \circ r_2 \sqsubseteq r\}$ and $\mathcal{A} = \{r_1(a, a'), r_2(a', b)\}$, then*

$$(\mathcal{O}, \mathcal{A}) \models^{\mathcal{EL}^+_2} r(a, b)$$

but we cannot find an interpolant defined in Definition 4.2.41. This is due to the fact that using morphism $\mathsf{e}_2^+\overline{\mathsf{c}}$ we are unable to formulate expressions with complex roles like $r_1 \circ r_2$ as we have only role names available.

Now we show that comorphism $\mathsf{e}_3^+\overline{\mathsf{c}}+\iota$ from Example 4.2.9 has concept interpolation. But first we will need some auxiliary notions.

Notation 4.2.43. *Let $\Sigma \in Sig^{\mathcal{EL}}$ and $(\mathcal{O}, \mathcal{R}) \subseteq Sen^{\mathcal{EL}^+_3}(\Sigma)$. Let us write*

$$\mathcal{O}^{\mathcal{R}} = \mathcal{O} \cup \{\exists r_1.\exists r_2.\dots\exists r_n.D \sqsubseteq \exists r.D \mid \mathcal{R} \models^{\mathcal{EL}^+_3}_{\Sigma} r_1 \circ \dots \circ r_n \sqsubseteq r$$
$$\text{and } r_1, \dots, r_n, r, D \in Con^{\mathcal{EL}^+_3}(\Sigma)\} .$$

Now we show that answering queries about individuals in $\mathcal{EL}^+_3+\iota$ reduces to answering queries about individuals in $\mathcal{EL}+\iota$.

Lemma 4.2.44. *Let $\Sigma, \Sigma' \in Sig^{\mathcal{EL}^+_3}$ be such that $\Sigma \hookrightarrow \Sigma'$, let $\mathcal{O}, \mathcal{R}, \mathcal{A} \subseteq Sen^{\mathcal{EL}^+_3+\iota}(\Sigma, I)$ and $C \in Con^{\mathcal{EL}^+_3}(\Sigma')$. Then:*

$$(\mathcal{O}^{\mathcal{R}}, \mathcal{A}) \models^{\mathcal{EL}+\iota}_{(\Sigma', I)} C(i) \quad \textit{iff} \quad (\mathcal{O}, \mathcal{R}, \mathcal{A}) \models^{\mathcal{EL}^+_3+\iota}_{(\Sigma', I)} C(i)$$

Proof: The direction "$\Rightarrow$" is straightforward.

For "$\Leftarrow$", first we show how model $\mathcal{M}$ of $(\mathcal{O}^{\mathcal{R}}, \mathcal{A})$ extends to $\mathcal{M}'$ which admits RBoxes, we also show how these extensions interpret roles.

For any signature (P, R, I) given $(\mathcal{O}^{\mathcal{R}}, \mathcal{A})$-model $\mathcal{M}$ we define (P, R, I) model $\mathcal{M}'$ in the following way:

- the domain $|\mathcal{M}| = |\mathcal{M}'|$,
- $C^{\mathcal{M}} = C^{\mathcal{M}'}$ for any $C \in \Sigma$
- $i^{\mathcal{M}} = i^{\mathcal{M}'}$ for any $i \in I$

- let $(r^{\mathcal{M}})_{r\in R}$ be the least family $\subseteq |\mathcal{M}'| \times |\mathcal{M}'|$ such that:
  - $r^{\mathcal{M}} \subseteq r^{\mathcal{M}'}$
  - if $(i_j, i_{j+1}) \in r_j^{\mathcal{M}'}$, for $1 \leq j \leq n$, and $r_1 \circ \cdots \circ r_n \sqsubseteq r \in \mathcal{R}$, then $(1, i_{n+1}) \in r^{\mathcal{M}'}$.

It is easy to see that $\mathcal{M}'$ is a model for $(\mathcal{O},\mathcal{R},\mathcal{A})$ and $C^{\mathcal{M}} = C^{\mathcal{M}'}$.

To prove "$\Leftarrow$", assume that we have $(\mathcal{O}^{\mathcal{R}}, \mathcal{A}) \not\models^{\mathcal{EL}+\iota}_{(\Sigma',I)} C(i)$, let $\mathcal{M}$ be a model such that $\mathcal{M} \models^{\mathcal{EL}+\iota}_{(\Sigma',I)} (\mathcal{O}^{\mathcal{R}}, \mathcal{A})$, but $\mathcal{M} \not\models^{\mathcal{EL}+\iota}_{(\Sigma',I)} C(i)$, as presented above we can extend $\mathcal{M}$ to a model $\mathcal{M}'$ such that $\mathcal{M}' \models^{\mathcal{EL}_3^++\iota}_{(\Sigma',I)} (\mathcal{O},\mathcal{R},\mathcal{A})$ but as $C^{\mathcal{M}} = C^{\mathcal{M}'}$ and $i^{\mathcal{M}} = i^{\mathcal{M}'}$, we have $\mathcal{M}' \not\models^{\mathcal{EL}_3^++\iota}_{(\Sigma',I)} C(i)$. □

Now we show the following proposition:

Proposition 4.2.45. *$\mathcal{EL}_3^++\iota$ has concept interpolation.*

Proof: Let $(\Sigma, I) \in \mathsf{Sig}^{\mathcal{EL}_3^++\iota}$, $(\mathcal{O},\mathcal{R},\mathcal{A}) \subseteq \mathsf{Sen}^{\mathcal{EL}_3^++\iota}(\Sigma, I)$ and $\varphi(i) \in \mathsf{Sen}^{\mathcal{EL}_3^++\iota}(\Sigma, I)$. Assume

$$(\mathcal{O},\mathcal{R},\mathcal{A}) \models^{\mathcal{EL}_3^++\iota}_{(\Sigma,I)} \varphi(i) .$$

We distinguish two cases:

(a) $\varphi(i) = C(i)$,

(b) $\varphi(i) = r_1 \circ \cdots \circ r_m(i_1, i_2)$.

For (a), first note that in Lemma 4.2.44 we already presented that given $\mathcal{O}$, $\mathcal{R}$ and $\mathcal{A}$, the pair $(\mathcal{O}^{\mathcal{R}}, \mathcal{A})$ in $\mathcal{EL}+\iota$ gives exactly the same answers to the queries of the form $C(i)$ as the triple $(\mathcal{O},\mathcal{R},\mathcal{A})$ in $\mathcal{EL}_3^++\iota$. From this it follows that for queries of the form $C(i)$ the problem of finding the interpolant in $\mathcal{EL}_3^++\iota$ reduces to $\mathcal{EL}+\iota$.

For (b), first we need some auxiliary lemmas. First we introduce a lemma stating that if in $\mathcal{EL}_3^++\iota$ a triple $(\mathcal{O},\mathcal{R},\mathcal{A})$ entails a role assertion then $\mathcal{O}$ does not play any role in that entailment and can be removed without any harm.

Lemma 4.2.46. *Let $(\Sigma, I) \in Sig^{\mathcal{EL}_3^++\iota}$ and $(\mathcal{O},\mathcal{R},\mathcal{A}) \subseteq Sen^{\mathcal{EL}_3^++\iota}(\Sigma, I)$. Let $r_1 \circ \cdots \circ r_n(i_1, i_2)$ be a role assertion, with $r_1, \cdots, r_n \in \Sigma$ and $i_1, i_2 \in I$, then:*

$$(\mathcal{O},\mathcal{R},\mathcal{A}) \models^{\mathcal{EL}_3^++\iota}_{(\Sigma,I)} r_1 \circ \cdots \circ r_n(i_1, i_2) \quad \textit{implies} \quad (\mathcal{R},\mathcal{A}) \models^{\mathcal{EL}_3^++\iota}_{(\Sigma,I)} r_1 \circ \cdots \circ r_n(i_1, i_2) .$$

Proof: We distinguish two cases:

(a) $r_1 \circ \cdots \circ r_n(i_1, i_2) \in \mathcal{A}$, then it trivially follows from $(\mathcal{R},\mathcal{A})$,

(b) $r_1 \circ \cdots \circ r_n(i_1, i_2) \notin \mathcal{A}$, but then there is a sequence of role assertions $r_{1,1} \circ \cdots \circ r_{1,m}(i_1, i'), r_{2,1} \circ \cdots \circ r_{2,l}(i', i''), \ldots, r_{k,1} \circ \cdots \circ r_{k,j}(i, i_2)$ in $\mathcal{A}$ and a role inclusion $r_{1,1} \circ \cdots \circ r_{1,m} \circ \cdots \circ r_{k,1} \circ \cdots \circ r_{k,j} \sqsubseteq r_1 \circ \cdots \circ r_n$ in $\mathcal{R}$. These two facts together give us $(\mathcal{R},\mathcal{A}) \models^{\mathcal{EL}_3^++\iota}_{(\Sigma,I)} r_1 \circ \cdots \circ r_n(i_1, i_2)$. □

This leads us to the lemma which, roughly speaking, states that if $(\mathcal{R}, \mathcal{A})$ entails some role assertion $r_1 \circ \cdots \circ r_n(i_1, i_2)$ then there is a sequence of role assertions $r_{1,1} \circ \cdots \circ r_{1,m}(i_1, i'), r_{2,1} \circ \cdots \circ r_{2,l}(i', i''), \ldots, r_{k,1} \circ \cdots \circ r_{k,j}(i, i_2)$ in $\mathcal{A}$, such that for $\psi : r_{1,1} \circ \cdots \circ r_{1,m} \circ \cdots \circ r_{k,1} \circ \cdots \circ r_{k,j}$, we have $\mathcal{A} \models \psi(i_1, i_2)$ and $\mathcal{R} \models \psi \sqsubseteq r_1 \circ \cdots \circ r_n$.

Lemma 4.2.47. *For any role assertion $r_1 \circ \cdots \circ r_n(i_1, i_2) \in Sen^{\mathcal{EL}_3^+ + \iota}(\Sigma, I)$ and $(\mathcal{R}, \mathcal{A}) \subseteq Sen^{\mathcal{EL}_3^+ + \iota}(\Sigma, I)$, such that $(\mathcal{R}, \mathcal{A}) \models^{\mathcal{EL}_3^+ + \iota}_{(\Sigma, I)} r_1 \circ \cdots \circ r_n(i_1, i_2)$ there exists $\psi : r_{1,1} \circ \cdots \circ r_{1,m} \circ \cdots \circ r_{k,1} \circ \cdots \circ r_{k,j}$ such that $\mathcal{A} \models^{e_3^+ \bar{c}}_{(\Sigma, I)} \psi(i_1, i_2)$ and $\mathcal{R} \models^{\mathcal{EL}_3^+ + \iota}_{(\Sigma, I)} \psi \sqsubseteq r_1 \circ \cdots \circ r_n$.*

Proof: The proof is straightforward by Lemma 4.2.46 and the way how model $\mathcal{M}'$ in the proof of Lemma 4.2.44 is constructed. □

But Lemma 4.2.47 shows how to find an interpolant for (b) in Proposition 4.2.45, which proves the Proposition. □

4.3 Constructing a Framework with Individuals

In Section 3.2 we introduced the notion of framework, in order to study in an institutional setting the work of [50] which explored the relationship between robustness properties and interpolation properties. The main idea was to allow ontologies to be written in one language, while their consequences could be tested, or queried, in another language. This requires both ontologies and queries to be translated into a 'global language'.

Now we would like to apply this notion of frameworks to description logics with individuals: given an ontology in one language, we might like to query what it tells us about individuals. In other words, instead of querying it through a query basis $\eta : \mathcal{Q} \to \mathcal{G}$, we would like to query it from the institution $\mathcal{Q}{+}\iota$. However, there may be cases where a query basis has no corresponding morphism; to allow for such cases, in Definition 4.2.25 we defined DL-comorphisms between description logics. Our query bases for description logics $\mathcal{Q} \xrightarrow{\mu} \overline{\mathcal{CH}}$ and $\mathcal{G} \xrightarrow{\nu} \overline{\mathcal{CH}}$ are therefore comorphisms as in Definition 3.2.3, together with natural transformations $\tau^{\eta}_{\Sigma} : \Psi^{\nu}(\Sigma) \to \Psi^{\mu}(\Phi^{\eta}(\Sigma))$, for each signature Σ in $\mathsf{Sig}^{\mathcal{Q}}$.

So far we have established that given a description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$, institution $\mathcal{I}$ extends to an institution $\mathcal{I}{+}\iota$ (cf. Definition 4.2.10). We have also shown that any DL-comorphism extends to $DL{+}\iota$-comorphism (cf. Proposition 4.2.28). Using these two facts we can show that any framework μ over a query basis η can be extended to a framework with individuals μ^{ι} over a query basis with individuals η^{ι}. Thanks to that we can use an ontology $\mathcal{O}$ together with an ABox $\mathcal{A}$ and post queries about individuals. This is presented in the Corollary below.

Remark 4.3.1. *In the remainder of the chapter all the comorphisms are in fact DL-comorphisms.*

Corollary 4.3.2. *Given a framework $\mu : \mathcal{L} \to \mathcal{G}$ over a query basis $\eta : \mathcal{Q} \to \mathcal{G}$, where $\mathcal{L}$, $\mathcal{Q}$ and $\mathcal{G}$ are description logics, we can extend μ and η to a framework with individuals $\mu^{\iota} : \mathcal{L}{+}\iota \to \mathcal{G}{+}\iota$ over the query basis with individuals $\eta^{\iota} : \mathcal{Q}{+}\iota \to \mathcal{G}{+}\iota$.*

This is graphically presented in Figure 4.13.

As we have that given a description logic $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$ there are morphisms $\mu^+ : \mathcal{I} \to \mathcal{I}{+}\iota$ and $\mu^- : \mathcal{I}{+}\iota \to \mathcal{I}$ and a comorphism $\vec{\mu} : \mathcal{I} \to \mathcal{I}{+}\iota$, we are able to establish what are the relations between frameworks $\mu : \mathcal{L} \to \mathcal{G}$ and $\mu^\iota : \mathcal{L}{+}\iota \to \mathcal{G}{+}\iota$, and query bases $\eta : \mathcal{Q} \to \mathcal{G}$ and $\eta^\iota : \mathcal{Q}{+}\iota \to \mathcal{G}{+}\iota$. Similarly we can determine relations between their component institutions. This is presented in Figure 4.13, it also presents the list of morphisms and comorphisms used. For the sake of simplicity, for each pair of institutions $\mathcal{I}$ and $\mathcal{I}{+}\iota$, Figure 4.13 mentions only comorphism $\vec{v} : \mathcal{I} \to \mathcal{I}{+}\iota$, but as Definition 4.2.18 and Definition 4.2.19 tell us that there are also morphisms $v^- : \mathcal{I}{+}\iota \to \mathcal{I}$ and $v^+ : \mathcal{I} \to \mathcal{I}{+}\iota$ respectively. Taking that into account Figure 4.13 to a great extent summarizes relations between frameworks and frameworks with individuals (similarly for query bases) and between their components.

1. Morphisms:
   - $\mu : \mathcal{L} \to \overline{\mathcal{CH}}$
   - $\xi : \mathcal{Q} \to \overline{\mathcal{CH}}$
   - $\nu : \mathcal{G} \to \overline{\mathcal{CH}}$
   - $\mu{+}\iota : \mathcal{L}{+}\iota \to \overline{\mathcal{CH}}{+}\iota$
   - $\xi{+}\iota : \mathcal{Q}{+}\iota \to \overline{\mathcal{CH}}{+}\iota$
   - $\nu{+}\iota : \mathcal{G}{+}\iota \to \overline{\mathcal{CH}}{+}\iota$
2. Comorphisms:
   - $\kappa : \mathcal{L} \to \mathcal{G}$
   - $\lambda : \mathcal{Q} \to \mathcal{G}$
   - $\kappa^\iota : \mathcal{L}{+}\iota \to \mathcal{G}{+}\iota$
   - $\lambda^\iota : \mathcal{Q}{+}\iota \to \mathcal{G}{+}\iota$
   - $\vec{\nu} : \mathcal{G} \to \mathcal{G}{+}\iota$
   - $\vec{\mu} : \mathcal{L} \to \mathcal{L}{+}\iota$
   - $\vec{\xi} : \mathcal{Q} \to \mathcal{Q}{+}\iota$
   - $1_{\vec{\mathcal{CH}}} : \overline{\mathcal{CH}} \to \overline{\mathcal{CH}}{+}\iota$

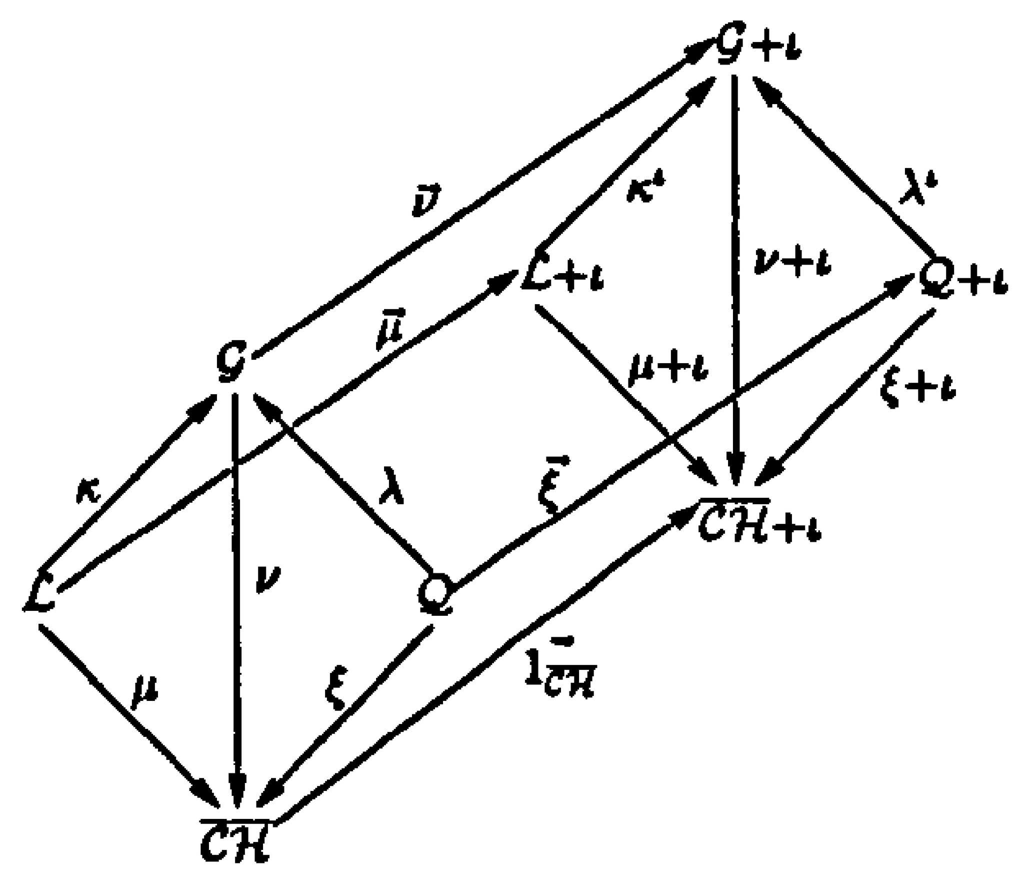


Figure 4.13: Construction of a framework with individuals.

Proposition 4.3.3. *For framework $\kappa : \mathcal{L} \to \mathcal{G}$ over query basis $\lambda : \mathcal{Q} \to \mathcal{G}$ and framework $\kappa{+}\iota : \mathcal{L}{+}\iota \to \mathcal{G}{+}\iota$ over query basis $\lambda{+}\iota : \mathcal{Q}{+}\iota \to \mathcal{G}{+}\iota$. The following holds:*

$$\begin{aligned} \kappa \,;\, \vec{\nu} &= \vec{\mu} \,;\, \kappa^\iota \\ \lambda \,;\, \vec{\nu} &= \vec{\xi} \,;\, \lambda^\iota \end{aligned}$$

Proof: Here we just show the case for $\kappa \,;\, \vec{\nu} = \vec{\mu} \,;\, \kappa^\iota$, as the other one is very similar. The proof has three parts showing composition of three components of comorphisms.

(a) First we show that $\Phi^\kappa \,;\, \Phi^{\vec{\nu}} = \Phi^{\vec{\mu}} \,;\, \Phi^{\kappa^\iota}$. Let $\Sigma \in \mathrm{Sig}^{\mathcal{L}}$ then we have that:

$$\Phi^{\vec{\nu}}(\Phi^\kappa(\Sigma)) = (\Phi^\kappa(\Sigma), \emptyset)$$

and

$$\Phi^{\kappa^{\iota}}(\Phi^{\vec{\mu}}(\Sigma)) = \Phi^{\kappa^{\iota}}(\Sigma, \emptyset) = (\Phi^{\kappa}(\Sigma), \emptyset) .$$

(b) Now we show $\alpha^{\kappa} ; \alpha^{\vec{\nu}} = \alpha^{\vec{\mu}} ; \alpha^{\kappa^{\iota}}$. First note that for any $\Sigma \in \mathsf{Sig}^{\mathcal{L}}$ and $\varphi \in \mathsf{Sen}^{\mathcal{L}}(\Sigma)$ we have that $\alpha^{\vec{\mu}}_{\Sigma}(\varphi) = \varphi$, and for any $\Lambda \in \mathsf{Sig}^{\mathcal{G}}$ and $\psi \in \mathsf{Sen}^{\mathcal{G}}(\Lambda)$ we have that $\alpha^{\vec{\nu}}_{\Lambda}(\psi) = \psi$.

Let $\varphi \in \mathsf{Sen}^{\mathcal{L}}(\Sigma)$, then

$$\alpha^{\vec{\nu}}_{\Phi^{\kappa}(\Sigma)}(\alpha^{\kappa}_{\Sigma}(\varphi)) = \alpha^{\kappa}_{\Sigma}(\varphi)$$

and

$$\alpha^{\kappa^{\iota}}_{\Phi^{\vec{\mu}}(\Sigma)}(\alpha^{\vec{\mu}}_{\Sigma}(\varphi)) = \alpha^{\kappa^{\iota}}_{\Phi^{\vec{\mu}}(\Sigma)}(\varphi) = \alpha^{\kappa}_{\Sigma}(\varphi) .$$

The last step is by the fact that $\varphi \in \mathsf{Sen}^{\mathcal{L}}(\Sigma)$, so $\alpha^{\kappa^{\iota}}_{\Phi^{\vec{\mu}}(\Sigma)}$ translates it just as α^{κ}_{Σ} does.

(c) Finally, we show that $\beta^{\vec{\nu}} ; \beta^{\kappa} = \beta^{\kappa^{\iota}} ; \beta^{\vec{\mu}}$. First note that for any model $(\mathcal{M}, f)$ in $\mathcal{L}{+}\iota$ over (Σ, I) we have $\beta^{\vec{\mu}}_{\Sigma}(\mathcal{M}, f) = \mathcal{M}$, with $\mathcal{M}$ an $\mathcal{L}$-model over Σ and similarly for models in $\mathcal{G}{+}\iota$ and $\mathcal{G}$.

Let $(\mathcal{M}, f)$ be a $\mathcal{G}{+}\iota$-model over $(\Phi^{\kappa}(\Sigma), I)$ then

$$\beta^{\kappa}_{\Sigma}(\beta^{\vec{\nu}}_{\Phi^{\kappa}(\Sigma)}(\mathcal{M}, f)) = \beta^{\kappa}_{\Sigma}(\mathcal{M})$$

and

$$\beta^{\vec{\mu}}_{\Sigma}(\beta^{\kappa^{\iota}}_{\Phi^{\vec{\mu}}(\Sigma)}(\mathcal{M}, f)) = \beta^{\vec{\mu}}_{\Sigma}(\beta^{\kappa}_{\Sigma}(\mathcal{M}), f) = \beta^{\kappa}_{\Sigma}(\mathcal{M}) .$$

□

For the bottom rectangles in Figure 4.13 we have mixed morphisms and comorphisms, we can make these rectangles commute in two ways: by replacing $\vec{\mu}$, $\vec{\xi}$ and $1_{\overrightarrow{\mathcal{CH}}}$ with μ^-, ξ^- and $1^-_{\overline{\mathcal{CH}}}$ respectively, alternatively we can make these rectangles commute by replacing these comorphisms with μ^+, ξ^+ and $1^+_{\overline{\mathcal{CH}}}$ respectively. Then by Proposition 4.2.20 we immediately obtain:

$$\begin{aligned} \mu^- ; \mu &= \mu{+}\iota ; 1^-_{\overline{\mathcal{CH}}} \\ \xi^- ; \xi &= \xi{+}\iota ; 1^-_{\overline{\mathcal{CH}}} \end{aligned}$$

and

$$\begin{aligned} \mu ; 1^+_{\overline{\mathcal{CH}}} &= \mu^+ ; \mu{+}\iota \\ \xi ; 1^+_{\overline{\mathcal{CH}}} &= \xi^+ ; \xi{+}\iota \end{aligned}$$

Notation 4.3.4. *In what follows, unless stated otherwise, we will use the names for morphisms and comorphisms that are introduced in Figure 4.13.*

Lemma 4.3.5. *For framework $\kappa : \mathcal{L} \to \mathcal{G}$ over query basis $\lambda : \mathcal{Q} \to \mathcal{G}$, we also have framework $\kappa ; \vec{\nu} : \mathcal{L} \to \mathcal{G}{+}\iota$ over query basis $\lambda^{\iota} : \mathcal{Q}{+}\iota \to \mathcal{G}{+}\iota$, and the following holds for all signatures $\Lambda \in \mathsf{Sig}^{\mathcal{L}}$, $\Sigma \in \mathsf{Sig}^{\mathcal{Q}}$ and $(\Sigma, I) \in \mathsf{Sig}^{\mathcal{Q}+\iota}$, ontology $\mathcal{O} \subseteq \mathsf{Sen}^{\mathcal{L}}(\Lambda)$ and query $\varphi \in \mathsf{Sen}^{\mathcal{Q}}(\Sigma)$:*

$$\mathcal{O} \models^{\lambda}_{\Sigma} \varphi \quad \textit{iff} \quad \mathcal{O} \models^{\lambda^{\iota}}_{(\Sigma, I)} \varphi .$$

Proof: The direction "⇒" is immediate.
For the direction "⇐", assume $\mathcal{O} \models^{\lambda^\iota}_{(\Sigma,I)} \varphi$. Let $\mathcal{M} \in \mathsf{Mod}^{\mathcal{G}}(\Phi^\kappa(\Sigma) \cup \Phi^\lambda(\Lambda))$ such that $\mathcal{M} \models^{\mathcal{G}}_{(\Phi^\kappa(\Sigma)\cup\Phi^\lambda(\Lambda))} \alpha^\kappa_\Lambda(\mathcal{O})$. We want to construct $\mathcal{G}+\iota$-model $(\mathcal{M}, f)$ with $f : I \to |\delta^\nu_{(\Phi^\kappa(\Sigma)\cup\Phi^\lambda(\Lambda))}(\mathcal{M})|$. By Definition 4.2.3 we know that $|\delta^\nu_{(\Phi^\kappa(\Sigma)\cup\Phi^\lambda(\Lambda))}(\mathcal{M})| \neq \emptyset$, therefore there is $x \in |\delta^\nu_{(\Phi^\kappa(\Sigma)\cup\Phi^\lambda(\Lambda))}(\mathcal{M})|$ and we set $f(i) = x$ for all $i \in I$. Thus we have a model $(\mathcal{M}, f)$, such that $(\mathcal{M}, f) \models^{\mathcal{G}+\iota}_{(\Phi^\kappa(\Sigma)\cup\Phi^\lambda(\Lambda),I)} \alpha^\kappa_\Lambda(\mathcal{O})$ and thus $(\mathcal{M}, f) \models^{\mathcal{G}+\iota}_{(\Phi^\kappa(\Sigma)\cup\Phi^\lambda(\Lambda),I)} \alpha^\lambda_\Sigma(\varphi)$. From this it follows that $\mathcal{M} \models^{\mathcal{G}}_{(\Phi^\kappa(\Sigma)\cup\Phi^\lambda(\Lambda))} \alpha^\lambda_\Sigma(\varphi)$. □

Theorem 4.3.6. *For framework $\kappa : \mathcal{L} \to \mathcal{G}$ over query basis $\lambda : \mathcal{Q} \to \mathcal{G}$ and framework $\kappa ; \vec{\nu} : \mathcal{L} \to \mathcal{G}+\iota$ over query basis $\lambda^\iota : \mathcal{Q}+\iota \to \mathcal{G}+\iota$, signatures $\Lambda \in Sig^{\mathcal{L}}$, $\Sigma \in Sig^{\mathcal{Q}}$ and $(\Sigma, I) \in Sig^{\mathcal{Q}+\iota}$, any ontologies $\mathcal{O}_1, \mathcal{O}_2 \subseteq Sen^{\mathcal{L}}(\Lambda)$, the following holds:*

$$\mathcal{O}_1 \sqsubseteq^{\lambda^\iota}_{(\Sigma,I)} \mathcal{O}_2 \quad \textit{implies} \quad \mathcal{O}_1 \sqsubseteq^\lambda_\Sigma \mathcal{O}_2.$$

Proof uses the fact that for any institutions $\kappa : \mathcal{I} \to \mathcal{I}+\iota$ with $\Sigma \in \mathsf{Sig}^{\mathcal{I}}$ we have that $\mathsf{Sen}^{\mathcal{I}}(\Sigma) \subseteq \mathsf{Sen}^{\mathcal{I}+\iota}(\Phi^\kappa(\Sigma))$.

Proof: Assume $\mathcal{O}_1 \sqsubseteq^{\lambda^\iota}_{\Sigma,I} \mathcal{O}_2$, that is $\mathcal{O}_2 \models^{\lambda^\iota}_{(\Sigma,I)} \varphi$ implies $\mathcal{O}_1 \models^{\lambda^\iota}_{(\Sigma,I)} \varphi$ for $\varphi \in \mathsf{Sen}^{\mathcal{Q}+\iota}(\Sigma, I)$. This gives us that $\alpha^{\kappa;\vec{\nu}}_\Lambda(\mathcal{O}_2) \models^{\lambda^\iota}_{\Phi^{\kappa;\vec{\nu}}(\Lambda)\cup\Phi^{\lambda^\iota}(\Sigma,I)} \alpha^{\lambda^\iota}_{(\Sigma,I)}(\varphi)$ implies $\alpha^{\kappa;\vec{\nu}}_\Lambda(\mathcal{O}_2) \models^{\lambda^\iota}_{\Phi^{\kappa;\vec{\nu}}(\Lambda)\cup\Phi^{\lambda^\iota}(\Sigma,I)} \alpha^{\lambda^\iota}_{(\Sigma,I)}(\varphi)$ for $\varphi \in \mathsf{Sen}^{\mathcal{Q}}(\Sigma)$.

Assume $\alpha^\kappa_\Lambda(\mathcal{O}_2) \models^{\mathcal{G}}_{\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)} \alpha^\lambda_\Sigma(\varphi)$ then

$$\alpha^{\vec{\nu}}_{\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)}(\alpha^\kappa_\Lambda(\mathcal{O}_2)) \models^{\mathcal{G}+\iota}_{((\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)),I)} \alpha^{\vec{\nu}}_{\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)}(\alpha^\lambda_\Sigma(\varphi))$$

thus

$$\alpha^{\kappa;\vec{\nu}}_\Lambda(\mathcal{O}_2) \models^{\mathcal{G}+\iota}_{(\Phi^\kappa(\Lambda)\cup(\Phi^\lambda(\Sigma),I))} \alpha^{\lambda^\iota}_{(\Sigma,I)}(\alpha^\xi_\Sigma(\varphi)) .$$

This gives us

$$\alpha^{\kappa;\vec{\nu}}_\Lambda(\mathcal{O}_2) \models^{\mathcal{G}+\iota}_{(\Phi^\kappa(\Lambda)\cup(\Phi^\lambda(\Sigma),I))} \alpha^{\lambda^\iota}_{(\Sigma,I)}(\varphi) .$$

Therefore

$$\alpha^{\kappa;\vec{\nu}}_\Lambda(\mathcal{O}_1) \models^{\mathcal{G}+\iota}_{(\Phi^\kappa(\Lambda)\cup(\Phi^\lambda(\Sigma),I))} \alpha^{\lambda^\iota}_{(\Sigma,I)}(\varphi) ,$$

and thus

$$\alpha^{\vec{\nu}}_{\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)}(\alpha^\kappa_\Lambda(\mathcal{O}_1)) \models^{\mathcal{G}+\iota}_{(\Phi^\kappa(\Lambda)\cup(\Phi^\lambda(\Sigma),I))} \alpha^{\lambda^\iota}_{(\Sigma,I)}(\varphi) ,$$

this gives us

$$\alpha^\kappa_\Lambda(\mathcal{O}_1) \models^{\mathcal{G}+\iota}_{(\Phi^\kappa(\Lambda)\cup(\Phi^\lambda(\Sigma),I))} \alpha^\lambda_\Sigma(\varphi) .$$

From this it follows that

$$\alpha^\kappa_\Lambda(\mathcal{O}_1) \models^{\mathcal{G}+\iota}_{((\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)),I)} \alpha^\lambda_\Sigma(\varphi) ,$$

so by Lemma 4.3.5

$$\alpha^\kappa_\Lambda(\mathcal{O}_1) \models^{\mathcal{G}}_{(\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma))} \alpha^\lambda_\Sigma(\varphi) .$$

Thus

$$\mathcal{O}_1 \models^\lambda_\Sigma \varphi .$$

as required. □

4.3.1 Σ-entailment for knowledge bases

As already suggested in [50] when we are using ontologies together with ABoxes, we find it more useful to formulate Σ-entailment based on instance checking rather than based on subsumption, which is usually too weak in this case. In addition one has to determine how to include the ABox into the framework. In general, an ABox can be either a part of the ontology or a part of the query language. Here we consider only the case when the ontology and the ABox are closely related and the ABox does not change significantly more often than the ontology.

Above we presented how ABoxes and ontologies may interact together (recall the notion of query conservativity in Definition 4.2.33). Now we investigate the relations between Σ-inseparability and Σ-inseparability w.r.t $\mathcal{Q}+\iota$ instance checking.

Theorem 4.3.7. *For framework $\kappa : \mathcal{L} \to \mathcal{G}$ over query basis $\lambda : \mathcal{Q} \to \mathcal{G}$, with $\nu : \mathcal{G} \to \overline{\mathcal{CH}}$ query conservative, and framework $\kappa^\iota : \mathcal{L}+\iota \to \mathcal{G}+\iota$ over query basis $\lambda^\iota : \mathcal{Q}+\iota \to \mathcal{G}+\iota$, signatures $\Lambda \in Sig^{\mathcal{L}}$, $\Sigma \in Sig^{\mathcal{Q}}$, $(\Lambda, I) \in Sig^{\mathcal{L}+\iota}$, $(\Sigma, I) \in Sig^{\mathcal{Q}+\iota}$, any ontologies $\mathcal{O}_1, \mathcal{O}_2 \subseteq Sen^{\mathcal{L}}(\Lambda)$ and ABox $\mathcal{A}$, such that $\mathcal{A} \subseteq \Psi^{\mu}(\Lambda) \times I$. We have that:*

$$(\mathcal{O}_1, \mathcal{A}) \sqsubseteq^{\lambda^\iota}_{(\Sigma,I)} (\mathcal{O}_2, \mathcal{A}) \quad \textit{implies} \quad \mathcal{O}_1 \sqsubseteq^{\lambda}_{\Sigma} \mathcal{O}_2.$$

Proof: Assume $(\mathcal{O}_1, \mathcal{A}) \sqsubseteq^{\lambda^\iota}_{(\Sigma,I)} (\mathcal{O}_2, \mathcal{A})$, we want to show that $\mathcal{O}_1 \sqsubseteq^{\lambda}_{\Sigma} \mathcal{O}_2$.

Assume $\mathcal{O}_2 \models^{\lambda}_{\Sigma} \varphi$, i.e. $\alpha^{\kappa}_{\Lambda}(\mathcal{O}_2) \models^{\mathcal{G}}_{\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)} \alpha^{\lambda}_{\Sigma}(\varphi)$. Therefore

$$\alpha^{\kappa}_{\Lambda}(\mathcal{O}_2, \mathcal{A}) \models^{\mathcal{G}+\iota}_{((\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)),I)} \alpha^{\lambda}_{\Sigma}(\varphi)\ .$$

Thus

$$\alpha^{\kappa}_{\Lambda}(\mathcal{O}_1, \mathcal{A}) \models^{\mathcal{G}+\iota}_{((\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)),I)} \alpha^{\lambda}_{\Sigma}(\varphi)\ .$$

If $\nu : \mathcal{G} \to \overline{\mathcal{CH}}$ is query conservative then

$$\alpha^{\kappa}_{\Lambda}(\mathcal{O}_1) \models^{\mathcal{G}}_{\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)} \alpha^{\lambda}_{\Sigma}(\varphi)\ .$$

Thus $\mathcal{O}_1 \models^{\lambda}_{\Sigma} \varphi$, as required. □

The converse does not always hold. This was already shown in [50], where an example for $\mathcal{ALC}$ was presented:

Example 4.3.8. *Let $1_{\mathcal{ALC}}$ be a framework over itself as a query basis. Let $\mathcal{O}_1 = \emptyset$, $\mathcal{O}_2 = \{A \sqsubseteq \forall r.\neg A, \neg A \sqsubseteq \forall r.A\}$ be Λ-ontologies for $1_{\mathcal{ALC}}$, and let $\Sigma = \{r\}$, where $\Lambda, \Sigma \in Sig^{\mathcal{ALC}}$. Then we have that $\mathcal{O}_1 \approx^{1_{\mathcal{ALC}}}_{\Sigma} \mathcal{O}_2$. But if we extend framework $1_{\mathcal{ALC}}$ to $1_{\mathcal{ALC}+\iota}$ and add an ABox of the form $\mathcal{A} = \{r(a,a)\}$ to $\mathcal{O}_1$ and $\mathcal{O}_2$ we receive that $(\mathcal{O}_1, \mathcal{A})$ is consistent but $(\mathcal{O}_2, \mathcal{A})$ is not. This in turn gives us that:*

$$(\mathcal{O}_2, \mathcal{A}) \models^{1_{\mathcal{ALC}+\iota}}_{(\Sigma,I)} \bot(a)$$

but

$$(\mathcal{O}_1, \mathcal{A}) \not\models^{1_{\mathcal{ALC}+\iota}}_{(\Sigma,I)} \bot(a)\ .$$

Therefore $(\mathcal{O}_1, \mathcal{A}) \not\approx^{1_{\mathcal{ALC}+\iota}}_{(\Sigma,I)} (\mathcal{O}_2, \mathcal{A})$, as assertion $\bot(a)$ separates $(\mathcal{O}_1, \mathcal{A})$ and $(\mathcal{O}_2, \mathcal{A})$.

The following can be understood as a partial converse of Theorem 4.3.7.

Theorem 4.3.9. *For framework $\kappa : \mathcal{L} \to \mathcal{G}$ over query basis $\lambda : \mathcal{Q} \to \mathcal{G}$ and framework $\kappa^\iota : \mathcal{L}{+}\iota \to \mathcal{G}{+}\iota$ over query basis $\lambda^\iota : \mathcal{Q}{+}\iota \to \mathcal{G}{+}\iota$, such that $\mathcal{G}{+}\iota$ has concept interpolation and $\nu : \mathcal{G} \to \overline{\mathcal{CH}}$ has query conservativity for any signatures $\Lambda \in Sig^{\mathcal{L}}$, $\Sigma \in Sig^{\mathcal{Q}}$, $(\Lambda, I) \in Sig^{\mathcal{L}+\iota}$, $(\Sigma, I) \in Sig^{\mathcal{Q}+\iota}$, any ontologies $\mathcal{O}_1, \mathcal{O}_2 \subseteq Sen^{\mathcal{L}}(\Lambda)$ and ABox $\mathcal{A} \subseteq Sen^{\mathcal{L}+\iota}(\Lambda, I)$. We have that:*

$$\mathcal{O}_1 \sqsubseteq^{\lambda}_{\Sigma} \mathcal{O}_2 \quad \textit{implies} \quad (\mathcal{O}_1, \mathcal{A}) \sqsubseteq^{\lambda^\iota}_{(\Sigma,I)} (\mathcal{O}_2, \mathcal{A}) \ .$$

Proof: Suppose $\mathcal{O}_1 \sqsubseteq^{\lambda}_{\Sigma} \mathcal{O}_2$ and $(\mathcal{O}_1, \mathcal{A}) \not\sqsubseteq^{\lambda^\iota}_{(\Sigma,I)} (\mathcal{O}_2, \mathcal{A})$, so there is $\varphi \in \mathrm{Sen}^{\mathcal{Q}+\iota}(\Sigma, I)$, such that $(\mathcal{O}_2, \mathcal{A}) \models^{\lambda^\iota}_{(\Sigma,I)} \varphi$ but $(\mathcal{O}_1, \mathcal{A}) \not\models^{\lambda^\iota}_{(\Sigma,I)} \varphi$.

- If $\varphi \in \mathrm{Sen}^{\mathcal{Q}}(\Sigma)$, then by query conservativity we have $\mathcal{O}_2 \models^{\lambda}_{\Sigma} \varphi$, whence $\mathcal{O}_1 \models^{\lambda}_{\Sigma} \varphi$ and so $(\mathcal{O}_1, \mathcal{A}) \models^{\lambda^\iota}_{(\Sigma,I)} \varphi$ giving a contradiction.

- If φ is of the form $\chi(a)$ with $\chi \in \Psi^\nu(\Phi^\lambda(\Sigma))$ and $a \in I$ then by concept interpolation there is $\psi \in \Psi^\nu(\Phi^\lambda(\Sigma))$, such that $\mathcal{A} \models^{\xi}_{(\Lambda\cup\Sigma,I)} \psi(a)$ and

$$\alpha^{\kappa}_{\Lambda}(\mathcal{O}_2) \models^{\mathcal{G}}_{\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)} \gamma^{\nu}_{\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)}(\psi \sqsubseteq \chi)$$

  and so

$$\alpha^{\kappa}_{\Lambda}(\mathcal{O}_1) \models^{\mathcal{G}}_{\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)} \gamma^{\nu}_{\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)}(\psi \sqsubseteq \chi) \ .$$

  If

$$(\mathcal{M}, f) \models^{\mathcal{G}+\iota}_{((\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma)),I)} \alpha^{\kappa}_{\Lambda}(\mathcal{O}_1, \mathcal{A}) \ ,$$

  then

$$\mathcal{M} \models^{\mathcal{G}}_{(\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma))} \Psi^{\nu}_{(\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma))}(\psi \sqsubseteq \chi) \ ,$$

  so

$$\psi^{\delta^\nu_{(\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma))}(\mathcal{M})} \subseteq \chi^{\delta^\nu_{(\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma))}(\mathcal{M})} \ ;$$

  moreover, $f(a) \in \psi^{\delta^\nu_{(\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma))}(\mathcal{M})}$ and therefore $f(a) \in \chi^{\delta^\nu_{(\Phi^\kappa(\Lambda)\cup\Phi^\lambda(\Sigma))}(\mathcal{M})}$, contradicting $(\mathcal{O}_1, \mathcal{A}) \not\models^{\lambda^\iota}_{(\Sigma,I)} \varphi$.

In both cases, we obtain a contradiction, so we conclude $(\mathcal{O}_1, \mathcal{A}) \models^{\lambda^\iota}_{(\Sigma,I)} \varphi$ as desired. □

The following theorem was originally introduced in [62]. Here it follows from Theorem 4.3.9 and from the facts that description logic $\mathrm{e}\bar{\mathrm{c}} : \mathcal{EL} \to \overline{\mathcal{CH}}$ has query expansion property and that $\mathcal{EL}{+}\iota$ has concept interpolation.

Theorem 4.3.10. *For framework $1_{\mathcal{EL}} : \mathcal{EL} \to \mathcal{EL}$ over itself as a query basis and framework $1_{\mathcal{EL}+\iota} : \mathcal{EL}{+}\iota \to \mathcal{EL}{+}\iota$ over itself as a query basis, any $(\Sigma, I) \in Sig^{\mathcal{EL}+\iota}$ with $\Sigma \in Sig^{\mathcal{EL}}$, any ontologies $\mathcal{O}_1, \mathcal{O}_2 \subseteq Sen^{\mathcal{EL}}(\Sigma)$ and ABox $\mathcal{A} \subseteq Sen^{\mathcal{EL}+\iota}(\Sigma, I)$, the following holds:*

$$(\mathcal{O}_1, \mathcal{A}) \sqsubseteq^{1_{\mathcal{EL}+\iota}}_{(\Sigma,I)} (\mathcal{O}_2, \mathcal{A}) \quad \textit{iff} \quad \mathcal{O}_1 \sqsubseteq^{1_{\mathcal{EL}}}_{\Sigma} \mathcal{O}_2 \ .$$

Since we know that for an institution of description logic $\mathcal{G}$ there is a description logic $\mathcal{G}+\iota$, such that there is a conservative comorphism $\mathcal{G} \to \mathcal{G}+\iota$, the following theorem is a direct consequence of Theorem 4.3.10 and Lemma 3.2.29.

Theorem 4.3.11. *Let $\mathcal{G}$ be an institution of a description logic and $\mathcal{G}+\iota$ its corresponding institution with individuals in the signature. Then for frameworks $\kappa : \mathcal{EL} \to \mathcal{G}$ over itself as a query basis, with surjective β^{κ} and $\kappa^{\iota} : \mathcal{EL}+\iota \to \mathcal{G}+\iota$ over itself as a query basis, with conservative comorphism κ^{ι}, any $\Lambda, \Sigma \in Sig^{\mathcal{EL}}$ and $(\Sigma, I) \in Sig^{\mathcal{EL}+\iota}$, any ontologies $\mathcal{O}_1, \mathcal{O}_2 \subseteq Sen^{\mathcal{EL}}(\Lambda)$ and ABox $\mathcal{A} \subseteq Sen^{\mathcal{EL}+\iota}(\Lambda)$, the following holds:*

$$\mathcal{O}_1 \sqsubseteq^{\kappa}_{\Sigma} \mathcal{O}_2 \quad \textit{iff} \quad (\mathcal{O}_1, \mathcal{A}) \sqsubseteq^{\kappa^{\iota}}_{(\Sigma,I)} (\mathcal{O}_2, \mathcal{A}) \; .$$

Proof: The direction "⇐" immediately follows from Theorem 4.3.7

For "⇒" first assume that $\mathcal{O}_1 \sqsubseteq^{\kappa}_{\Sigma} \mathcal{O}_2$, by surjectivity of β^{κ} we get that $\mathcal{O}_1 \sqsubseteq^{1_{\mathcal{EL}}}_{\Sigma} \mathcal{O}_2$. By Theorem 4.3.10 we have that

$$(\mathcal{O}_1, \mathcal{A}) \sqsubseteq^{1_{\mathcal{EL}+\iota}}_{(\Sigma,I)} (\mathcal{O}_2, \mathcal{A}) \; .$$

Assume $\alpha^{\kappa^{\iota}}_{(\Sigma,I)}(\mathcal{O}_2, \mathcal{A}) \models^{\mathcal{G}+\iota} \alpha^{\kappa^{\iota}}_{(\Sigma,I)}(\varphi)$ and let $(\mathcal{M}, f) \in \mathsf{Mod}^{\mathcal{G}+\iota}(\Phi^{\kappa^{\iota}}(\Sigma, I))$ such that

$$(\mathcal{M}, f) \models^{\mathcal{G}+\iota}_{(\Phi^{\kappa^{\iota}}(\Sigma,I))} \alpha^{\kappa^{\iota}}_{(\Sigma,I)}(\mathcal{O}_1, \mathcal{A}) \; .$$

Then

$$\beta^{\kappa^{\iota}}_{(\Sigma,I)}(\mathcal{M}, f) \models^{\mathcal{EL}+\iota}_{(\Sigma,I)} (\mathcal{O}_1, \mathcal{A}) \; ,$$

as signature morphism σ is the identity we have

$$\beta^{\kappa^{\iota}}_{(\Sigma,I)}(\mathcal{M}, f) \models^{\mathcal{EL}+\iota}_{(\Sigma,I)} (\mathcal{O}_1, \mathcal{A})^* \; .$$

As

$$(\mathcal{O}_1, \mathcal{A}) \sqsubseteq^{1_{\mathcal{EL}+\iota}}_{(\Sigma,I)} (\mathcal{O}_2, \mathcal{A}) \; ,$$

we get

$$\beta^{\kappa^{\iota}}_{(\Sigma,I)}(\mathcal{M}, f) \models^{\mathcal{EL}+\iota}_{(\Sigma,I)} (\mathcal{O}_2, \mathcal{A})^* \; .$$

This implies

$$(\mathcal{M}, f) \models^{\mathcal{G}+\iota}_{(\Phi^{\kappa^{\iota}}(\Sigma,I))} \alpha^{\kappa^{\iota}}_{(\Sigma,I)}((\mathcal{O}_2, \mathcal{A})^*) \; .$$

By conservativity of κ^{ι} we get

$$\mathsf{Mod}^{\mathcal{G}}(\alpha^{\kappa^{\iota}}_{(\Sigma,I)}((\mathcal{O}_2, \mathcal{A})^*)) \subseteq \mathsf{Mod}^{\mathcal{G}}((\alpha^{\kappa^{\iota}}_{(\Sigma,I)}(\mathcal{O}_2, \mathcal{A}))^*) \; .$$

Since $\alpha^{\kappa^{\iota}}_{(\Sigma,I)}(\varphi) \in (\alpha^{\kappa^{\iota}}_{(\Sigma,I)}(\mathcal{O}_2, \mathcal{A}))^*$ we get $(\mathcal{M}, f) \models^{\mathcal{G}+\iota}_{(\Phi^{\kappa^{\iota}}(\Sigma,I))} \alpha^{\kappa^{\iota}}_{(\Sigma,I)}(\varphi)$. This gives us that $(\mathcal{O}_1, \mathcal{A}) \sqsubseteq^{\kappa^{\iota}}_{(\Sigma,I)} (\mathcal{O}_2, \mathcal{A})$ as required. □

130.

Chapter 5

Deciding the Σ-entailment Problem for $\mathcal{ELSH}$

5.1 Introduction

In the previous chapters we presented the notion of Σ-entailment in the institution independent way; now we present a particular case of that problem. In this chapter we consider ontologies formulated as general CBoxes (TBoxes together with RBoxes) in the description logic $\mathcal{ELSH}$ obeying some additional restrictions, we also allow for queries formulated in the same logic. As Proposition 3.2.11 suggests that in such a case employing frameworks does not give us any advantage, we will investigate that problem directly in $\mathcal{ELSH}$.

This chapter extends the result presented by Lutz and Wolter [61] for description logic $\mathcal{EL}$. The main result states that the Σ-entailment problem for such ontologies can be solved in ExpTime. Thus, this problem is no more complex than for plain $\mathcal{EL}$, which was shown to be ExpTime-complete [61]. For comparison, the computational complexity of this problem is 2ExpTime-complete for more expressive description logics such as $\mathcal{ALC}$, $\mathcal{ALCQ}$, and $\mathcal{ALCQI}$ [40,60], but even in such simple formalisms as acyclic propositional Horn Logic it is co-NP-complete [39].

It was proposed in [40,60] to provide ontology designers with a tool offering automated reasoning support for deciding Σ-entailment. Thanks to that they would be able to trace if the modifications made in the ontology, like refining or module extraction, had no impact on relationships between concepts of the original ontology.

The algorithm described in this chapter is based on and described along the lines of the algorithm deciding conservative extensions in $\mathcal{EL}$ [61].

5.2 Logical difference

In Definition 3.2.16 we set what does it mean that ontology $\mathcal{O}$ Σ-entails another ontology $\mathcal{O}'$. Now, for the convenience of notation, we introduce the notion of logical difference and define the notion of Σ-entailment in $\mathcal{ELSH}$ in terms of logical difference w.r.t. a signature Σ for TBoxes, RBoxes, and CBoxes.

Definition 5.2.1 (Σ-difference, Σ-entailment). *Let $\mathcal{C} = (\mathcal{O}, \mathcal{R})$ and $\mathcal{C}' = (\mathcal{O}', \mathcal{R}')$ be two $\mathcal{ELSH}$ CBoxes and Σ a signature. The Σ-**difference**, $\mathsf{Diff}_\Sigma(\mathcal{O},\mathcal{O}')$, **between** $\mathcal{O}$ **and** $\mathcal{O}'$ is defined as*

$$\mathsf{Diff}_\Sigma(\mathcal{O},\mathcal{O}') = \{C \sqsubseteq D \mid \mathcal{O} \not\models C \sqsubseteq D,\ \mathcal{O}' \models C \sqsubseteq D, \text{ and } Sig(C \sqsubseteq D) \subseteq \Sigma\}\ ,$$

*where C and D are $\mathcal{EL}$-concepts. $\mathcal{O} \sqsubseteq_\Sigma \mathcal{O}'$, if, and only if, $\mathsf{Diff}_\Sigma(\mathcal{O},\mathcal{O}') = \emptyset$. The Σ-**difference**, $\mathsf{Diff}_\Sigma(\mathcal{R},\mathcal{R}')$, **between** $\mathcal{R}$ **and** $\mathcal{R}'$ **is defined as***

$$\begin{aligned}\mathsf{Diff}_\Sigma(\mathcal{R},\mathcal{R}') &= \{r \sqsubseteq s \mid \mathcal{R} \not\models r \sqsubseteq s;\ \mathcal{R}' \models r \sqsubseteq s;\ r, s \in \Sigma\}\\ &\cup \{r \circ r \sqsubseteq r \mid \mathcal{R} \not\models r \circ r \sqsubseteq r,\ \mathcal{R}' \models r \circ r \sqsubseteq r, \text{ with } r \in \Sigma\}\ .\end{aligned}$$

*$\mathcal{R} \sqsubseteq_\Sigma \mathcal{R}'$, if, and only if, $\mathsf{Diff}_\Sigma(\mathcal{R},\mathcal{R}') = \emptyset$. The Σ-**difference between** $\mathcal{C}$ **and** $\mathcal{C}'$ **is** defined as*

$$\begin{aligned}\mathsf{Diff}_\Sigma(\mathcal{C},\mathcal{C}') &= \{C \sqsubseteq D \mid \mathcal{C} \not\models C \sqsubseteq D,\ \mathcal{C}' \models C \sqsubseteq D, \text{ and } Sig(C \sqsubseteq D) \subseteq \Sigma\}\\ &\cup\ \mathsf{Diff}_\Sigma(\mathcal{R},\mathcal{R}')\ .\end{aligned}$$

$\mathcal{C} \sqsubseteq_\Sigma \mathcal{C}'$, *if, and only if,* $\mathsf{Diff}_\Sigma(\mathcal{C}, \mathcal{C}') = \emptyset$.

As an illustration consider the following example:

Example 5.2.2. *Let* $\Sigma = \{A, B\}$ *be a signature* $\mathcal{C} = (\mathcal{O}, \mathcal{R})$ *and* $\mathcal{C}' = (\mathcal{O}', \mathcal{R})$ *be two CBoxes, where* $\mathcal{O} = \{A \sqsubseteq B\}$, $\mathcal{O}' = \{A \sqsubseteq \exists r.B', \exists s.B' \sqsubseteq B\}$ *and* $\mathcal{R} = \{r \sqsubseteq s\}$. *It is easy to see that* $\mathcal{O} \sqsubseteq_\Sigma \mathcal{O}'$, *but* $\mathcal{O}' \not\sqsubseteq_\Sigma \mathcal{O}$. *But if we take RBox into account then CBoxes* $\mathcal{C}$ *and* $\mathcal{C}'$ *are* Σ*-inseparable.*

Observe that, there exist $\mathcal{EL}$-ontologies $\mathcal{O}$ and $\mathcal{O}'$ such that all CIs $C \sqsubseteq D \in \mathsf{Diff}_\Sigma(\mathcal{O}, \mathcal{O}')$ have at least size doubly exponential in $\mathcal{O}$, $\mathcal{O}'$, where we define the size $\|\mathcal{O}\|$ of an ontology $\mathcal{O}$ as follows. The size $\|C\|$ of a concept C is the number of occurrences of symbols used to write C down and $\|\mathcal{O}\| = \sum_{C \sqsubseteq D \in \mathcal{O}} \|C\| + \|D\|$.

We also define the **outdegree** of C, as the maximum cardinality of any set $\mathcal{P}$ of pairs of the form (r, C'), with r a role name and C' a concept, such that $\sqcap_{(r,C') \in \mathcal{P}} \exists r.C' \in \mathsf{sub}(C)$. We use $\mathsf{sub}(C)$ and $\mathsf{sub}(\mathcal{O})$ to denote the set of subconcepts of a concept C and the set of subconcepts occurring in the TBox $\mathcal{O}$, respectively.

5.3 Canonical models and simulation relations

In this section, we construct canonical models for $\mathcal{EL}^+$ and describe relations between canonical models using a simulation relation.

Definition 5.3.1 (Σ-Simulation). *Let* $\mathcal{M}_1$ *and* $\mathcal{M}_2$ *be interpretations and* Σ *a signature. A relation* $S \subseteq \Delta^{\mathcal{M}_1} \times \Delta^{\mathcal{M}_2}$ *is a* **Σ-simulation from $\mathcal{M}_1$ to $\mathcal{M}_2$** *if the following holds:*

- *for all concept names* $A \in \Sigma$ *and all* $(d_1, d_2) \in S$ *with* $d_1 \in A^{\mathcal{M}_1}$ *we have* $d_2 \in A^{\mathcal{M}_2}$;
- *for all role names* $r \in \Sigma$, *all* $(d_1, d_2) \in S$, *and all* $e_1 \in \Delta^{\mathcal{M}_1}$ *with* $(d_1, e_1) \in r^{\mathcal{M}_1}$, *there exists* $e_2 \in \Delta^{\mathcal{M}_2}$ *such that* $(d_2, e_2) \in r^{\mathcal{M}_2}$ *and* $(e_1, e_2) \in S$.

If $d_1 \in \Delta^{\mathcal{M}_1}, d_2 \in \Delta^{\mathcal{M}_2}$, and there is an Σ-simulation S from $\mathcal{M}_1$ to $\mathcal{M}_2$ with $(d_1, d_2) \in S$, then $(\mathcal{M}_2, d_2)$ **Σ-simulates** $(\mathcal{M}_1, d_1)$, written $(\mathcal{M}_1, d_1) \leq_\Sigma (\mathcal{M}_2, d_2)$. If $\Sigma = P \cup R$, we write '$\leq$' instead of '$\leq_\Sigma$'.

Let $\mathcal{M}$ be an interpretation, Σ a signature, and $d \in \Delta^{\mathcal{M}}$. We define the abbreviation $d^{\Sigma,\mathcal{M}} := \{C \mid d \in C^{\mathcal{M}} \text{ and } \mathsf{Sig}(C) \subseteq \Sigma\}$. The **outdegree** of an interpretation is the maximum number of role successors at any point in its domain and for any role in R.

The following characterization of Σ-simulation establishes a connection between Σ-simulation and Σ-concepts.

Theorem 5.3.2 (Characterization of Σ-simulation). *If* $(\mathcal{M}_1, d_1) \leq_\Sigma (\mathcal{M}_2, d_2)$, *then* $d_1^{\Sigma,\mathcal{M}_1} \subseteq d_2^{\Sigma,\mathcal{M}_2}$. *Conversely, if* $\mathcal{M}_1, \mathcal{M}_2$ *have finite out-degree, and* $d_1^{\Sigma,\mathcal{M}_1} \subseteq d_2^{\Sigma,\mathcal{M}_2}$, *then* $(\mathcal{M}_1, d_1) \leq_\Sigma (\mathcal{M}_2, d_2)$.

Proof: "$\Rightarrow$": Let $(\mathcal{M}_1, d_1) \leq_\Sigma (\mathcal{M}_2, d_2)$ and $C \in d_1^{\Sigma,\mathcal{M}_1}$. We show that $C \in d_2^{\Sigma,\mathcal{M}_2}$. The proof is by induction on the structure of C. In the induction base, we have that $C = \top$ or

$C = A$ with $A \in \Sigma$. For the former, we trivially obtain $\top \in d_2^{\Sigma,\mathcal{M}_2}$. In the latter case, we have $A \in d_1^{\Sigma,\mathcal{M}_1}$. Thus, $d_1 \in A^{\mathcal{M}_1}$, which implies $d_2 \in A^{\mathcal{M}_2}$ by definition of $\leq_\Sigma$. Hence, $A \in d_2^{\Sigma,\mathcal{M}_2}$. Consider the induction step:

- $C = C_1 \sqcap C_2$. From $C_1 \sqcap C_2 \in d_1^{\Sigma,\mathcal{M}_1}$, we obtain $d_1 \in (C_1 \sqcap C_2)^{\mathcal{M}_1}$, i.e., $d_1 \in C_1^{\mathcal{M}_1}$ and $d_1 \in C_2^{\mathcal{M}_1}$. Then $C_1 \in d_1^{\Sigma,\mathcal{M}_1}$ and $C_2 \in d_1^{\Sigma,\mathcal{M}_1}$. By the induction hyposthesis, we have that $C_1 \in d_2^{\Sigma,\mathcal{M}_2}$ and $C_2 \in d_2^{\Sigma,\mathcal{M}_2}$. Thus, $d_2 \in C_1^{\mathcal{M}_2}$ and $d_2 \in C_2^{\mathcal{M}_2}$, and $d_2 \in (C_1 \sqcap C_2)^{\mathcal{M}_2}$. Hence, $C_1 \sqcap C_2 \in d_2^{\Sigma,\mathcal{M}_2}$.

- $C = \exists r.D'$. Suppose $\exists r.D' \in d_1^{\Sigma,\mathcal{M}_1}$. Then, $d_1 \in (\exists r.D')^{\mathcal{M}_1}$, i.e., there exists a $d_1' \in \Delta^{\mathcal{M}_1}$ with $(d_1, d_1') \in r^{\mathcal{M}_1}$ and $d_1' \in D'^{\mathcal{M}_1}$. Since $(\mathcal{M}_1, d_1) \leq_\Sigma (\mathcal{M}_2, d_2)$, there exists a $d_2' \in \Delta^{\mathcal{M}_2}$ with $(d_2, d_2') \in r^{\mathcal{M}_2}$ and $(\mathcal{M}_1, d_1') \leq_\Sigma (\mathcal{M}_2, d_2')$. Since $D' \in d_1'^{\Sigma,\mathcal{M}_1}$, it follows by the induction hypothesis that $D' \in d_2'^{\Sigma,\mathcal{M}_2}$, i.e., $d_2' \in D'^{\mathcal{M}_2}$. Thus, we have that $d_2 \in (\exists r.D')^{\mathcal{M}_2}$. Hence, $\exists r.D' \in d_2^{\Sigma,\mathcal{M}_2}$.

"$\Leftarrow$": Define $d_1 \leqslant d_2$ iff $d_1^{\Sigma,\mathcal{M}_1} \subseteq d_2^{\Sigma,\mathcal{M}_2}$.
Claim: '$\leqslant$' is a Σ-simulation.
Let $d_1 \leqslant d_2$. First assume $d_1 \in A^{\mathcal{M}_1}$. Then by definition $d_2 \in A^{\mathcal{M}_2}$. Now assume $x_1 \in \Delta^{\mathcal{M}_1}$, and $(d_1, x_1) \in r^{\mathcal{M}_1}$. We have to show that there exists $x_2 \in \Delta^{\mathcal{M}_2}$ with $(d_2, x_2) \in r^{\mathcal{M}_2}$ and $x_1 \leqslant x_2$. Assume that it is not. Let $D = \{y \in \Delta \mid (d_2, y) \in r^{\mathcal{M}_2}\}$. By assumption there is no $y \in D$, such that $x_1 \leqslant y$. Hence for every $y \in D$ there is at least one C with $x_1 \in C^{\mathcal{M}_1}$ but $y \notin C^{\mathcal{M}_2}$. Choose one - C_y. Thus $x_1 \in C_y^{\mathcal{M}_1}$, but $y \notin C_y^{\mathcal{M}_2}$. Then $d_1 \in \exists r.\bigsqcap_{y \in D} C_y$, but $d_2 \notin \exists r.\bigsqcap_{y \in D} C_y$. That gives us a contradiction. □

Definition 5.3.3 (Canonical model). *Let $\mathcal{C} = (\mathcal{O}, \mathcal{R})$ be a CBox in $\mathcal{EL}^+$, and D a concept. The* **canonical model** *$\mathcal{M}_{D,\mathcal{C}} = (\Delta^{\mathcal{M}_{D,\mathcal{C}}}, \cdot^{\mathcal{M}_{D,\mathcal{C}}})$ is defined as follows:*

- $\Delta^{\mathcal{M}_{D,\mathcal{C}}} = \{D\} \cup \{C \mid \exists r.C \in sub(D) \cup sub(\mathcal{O})\}$;
- $C \in A^{\mathcal{M}_{D,\mathcal{C}}}$ *iff* $\mathcal{C} \models C \sqsubseteq A$, *for all* $A \in P$;
- $(C, C') \in r^{\mathcal{M}_{D,\mathcal{C}}}$ *iff at least one of the following holds:*
  - *(a)* $\mathcal{C} \models C \sqsubseteq \exists r.C'$ *and* $C' \in sub(\mathcal{O})$,
  - *(b)* $C \leadsto^r_{\mathcal{R}} C'$,

where $C \leadsto^r_{\mathcal{R}} C'$ *iff there exists a sequence* $\exists r_0.D_0, \ldots, \exists r_n.D_n$ *with* $\exists r_0.D_0$ *a conjunct of* C, $\exists r_{i+1}.D_{i+1}$ *a conjunct of* D_i *for* $0 \leq i < n$ *such that* $D_n = C'$ *and* $\mathcal{R} \models r_0 \circ \cdots \circ r_n \sqsubseteq r$.

The model $\mathcal{M}_{D,\mathcal{C}}$ can be constructed in polynomial time in the size of $\mathcal{C}$ and D as subsumption w.r.t. CBoxes in $\mathcal{EL}^+$ can be decided in polynomial time [8].

Example 5.3.4. *For an illustration of canonical models, consider a TBox*

$$\mathcal{O} = \{\mathit{Toe} \sqsubseteq \exists \mathit{isPartOf.Foot}, \mathit{Foot} \sqsubseteq \exists \mathit{isPartOf.Leg}\}$$

together with the RBox

$$\mathcal{R} = \{\mathit{hasLocation} \circ \mathit{isPartOf} \sqsubseteq \mathit{hasLocation}\}.$$

Figure 5.1 shows the canonical model $\mathcal{M}_{D,\mathcal{C}}$, where $D = \exists hasLocation.Toe$ and $\mathcal{C} = (\mathcal{O}, \mathcal{R})$.

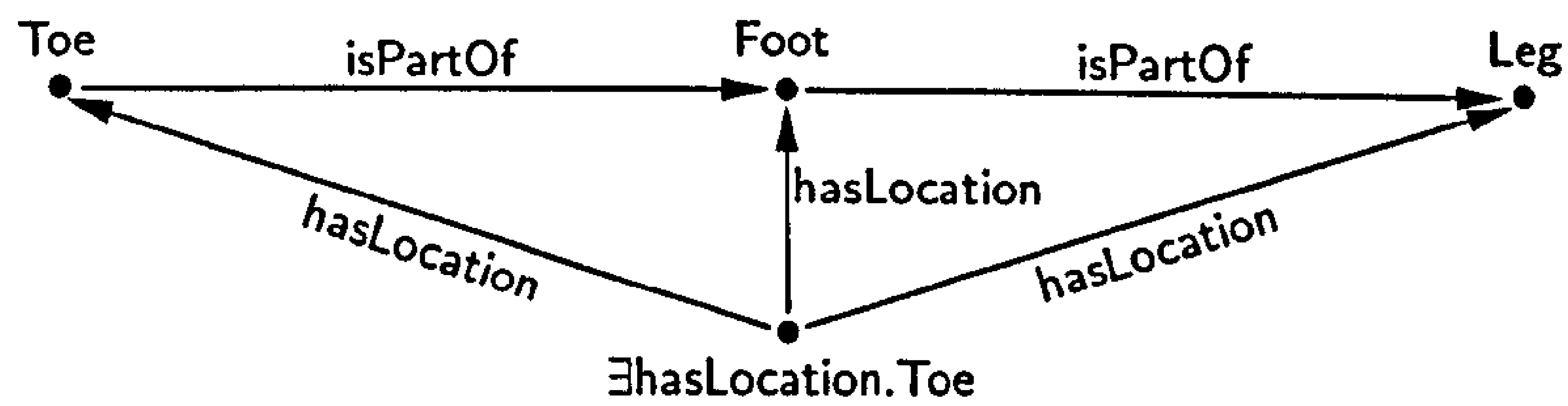


Figure 5.1: The canonical model $\mathcal{M}_{D,\mathcal{C}}$.

The following lemma summarizes the relevant properties of canonical models.

Lemma 5.3.5 (Properties of canonical models). *Let $\mathcal{C} = (\mathcal{O}, \mathcal{R})$ be a $\mathcal{EL}^+$ CBox and C a concept in $\mathcal{EL}^+$. Then the following holds:*

1. *$\mathcal{M}_{C,\mathcal{C}}$ is a model for $\mathcal{R}$.*
2. *For every $D \in \Delta^{\mathcal{M}_{C,\mathcal{C}}}$ we have that $D \in D^{\mathcal{M}_{C,\mathcal{C}}}$.*
3. *For every $D \in \Delta^{\mathcal{M}_{C,\mathcal{C}}}$ and every $C \in \mathsf{sub}(\mathcal{O})$ we have that*
$$\mathcal{C} \models D \sqsubseteq C \quad \text{iff} \quad D \in C^{\mathcal{M}_{C,\mathcal{C}}} .$$
In particular, $\mathcal{M}_{C,\mathcal{C}}$ is a model of $\mathcal{C}$.
4. *For all models $\mathcal{M}$ of $\mathcal{C}$ and all $d \in \Delta^{\mathcal{M}}$ and all $D \in \Delta^{\mathcal{M}_{C,\mathcal{C}}}$, the following conditions are equivalent:*
   - *(a) $d \in D^{\mathcal{M}}$;*
   - *(b) $(\mathcal{M}_{C,\mathcal{C}}, D) \leq (\mathcal{M}, d)$.*
5. *For all $D \in \Delta^{\mathcal{M}_{C,\mathcal{C}}}$, the following conditions are equivalent:*
   - *(a) $\mathcal{C} \models D \sqsubseteq C$;*
   - *(b) $D \in C^{\mathcal{M}_{C,\mathcal{C}}}$;*
   - *(c) $(\mathcal{M}_{C,\mathcal{C}}, C) \leq (\mathcal{M}_{D,\mathcal{C}}, D)$.*

Proof: (1) Let $\mathcal{R}$ be as in the lemma. To show that $\mathcal{M}_{C,\mathcal{C}}$ is a model of $\mathcal{R}$, let $r_1 \circ \cdots \circ r_n \sqsubseteq r \in \mathcal{R}$. We show that $r_1^{\mathcal{M}_{C,\mathcal{C}}} \circ \cdots \circ r_n^{\mathcal{M}_{C,\mathcal{C}}} \subseteq r^{\mathcal{M}_{C,\mathcal{C}}}$. Suppose that $(D_0, D_1) \in r_1^{\mathcal{M}_{C,\mathcal{C}}}$, $(D_1, D_2) \in r_2^{\mathcal{M}_{C,\mathcal{C}}}$, ... $(D_{n-1}, D_n) \in r_n^{\mathcal{M}_{C,\mathcal{C}}}$. By the construction of canonical model, we know that we have two sub-cases:

(i) $D_n \in \mathsf{sub}(\mathcal{O})$ and $D_n \notin \mathsf{sub}(C)$. (Note that this implies $D_i \in \mathsf{sub}(\mathcal{O})$ with $0 \leq i < n$.) Thus we have that $\mathcal{C} \models D_0 \sqsubseteq \exists r_1.D_1$, $\mathcal{C} \models D_1 \sqsubseteq \exists r_2.D_2$ ... $\mathcal{C} \models D_{n-1} \sqsubseteq \exists r_n.D_n$. Then from $\mathcal{R} \models r_1 \circ \cdots \circ r_n \sqsubseteq r$ it follows that $\mathcal{C} \models D_0 \sqsubseteq \exists r.D_n$. Since $D_n \in \mathsf{sub}(\mathcal{O})$, by condition (a) in Definition 5.3.3 we receive $(D_0, D_n) \in r^{\mathcal{M}_{C,\mathcal{C}}}$.

(ii) $D_n \notin \mathsf{sub}(\mathcal{O})$, note that this implies that $D_i \notin \mathsf{sub}(\mathcal{O})$, for $0 \leq i \leq n$. Then by Definition 5.3.3: $D_0 \rightsquigarrow^{r_1}_{\mathcal{R}} D_1, \ldots, D_{n-1} \rightsquigarrow^{r_n}_{\mathcal{R}} D_n$. Then, since $\mathcal{R} \models r_1 \circ \cdots \circ r_n \sqsubseteq r$, we obtain $D_0 \rightsquigarrow^{r}_{\mathcal{R}} D_n$. Then, by condition (b) in Definition 5.3.3 we receive $(D_0, D_n) \in r^{\mathcal{M}_{\mathcal{C},c}}$.

(2) We show that for every $D \in \Delta^{\mathcal{M}_{\mathcal{C},c}}$ we have $D \in D^{\mathcal{M}_{\mathcal{C},c}}$. The proof is by induction on the construction of D. Suppose:

(a) $D = A$, for a concept name A, thus we have $\mathcal{C} \models A \sqsubseteq A$ and by Definition 5.3.3 we have $A \in A^{\mathcal{M}_{\mathcal{C},c}}$, as required.

(b) $D = A_1 \sqcap \ldots \sqcap A_n \sqcap \exists r_1.D_1 \sqcap \ldots \sqcap \exists r_m.D_m$. So we have to show that $D \in A_1^{\mathcal{M}_{\mathcal{C},c}} \cap \ldots \cap A_n^{\mathcal{M}_{\mathcal{C},c}} \cap (\exists r_1.D_1)^{\mathcal{M}_{\mathcal{C},c}} \cap \ldots \cap (\exists r_m.D_m)^{\mathcal{M}_{\mathcal{C},c}}$. The part $D \in A_i^{\mathcal{M}_{\mathcal{C},c}}$ with $0 \leq i \leq n$ can be shown in the same way as (a), so we have to show only the part for $D \in (\exists r_j.D_j)^{\mathcal{M}_{\mathcal{C},c}}$ for $1 \leq j \leq m$. But we have $D \rightsquigarrow^{r_j}_{\emptyset} D_j$ and so $D \rightsquigarrow^{r_j}_{\mathcal{R}} D_j$. Hence $(D, D_j) \in r_j^{\mathcal{M}_{\mathcal{C},c}}$ and $D_j \in D_j^{\mathcal{M}_{\mathcal{C},c}}$. Thus $D \in (\exists r_j.D_j)^{\mathcal{M}_{\mathcal{C},c}}$, as required.

(3) For "⇒" assume $\mathcal{C} \models D \sqsubseteq C$, the proof is by induction on the construction of C. Suppose:

(a) $C = A$, for a concept name A, then we have $\mathcal{C} \models D \sqsubseteq A$ and by definition $D \in A^{\mathcal{M}_{\mathcal{C},c}}$.

(b) $C = A_1 \sqcap \ldots \sqcap A_n \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_m.C_m$. For every A_i, with $1 \leq i \leq n$, we can show that $D \in A_i^{\mathcal{M}_{\mathcal{C},c}}$ in the same way as (a). So we only have to show that $D \in (\exists r_j.C_j)^{\mathcal{M}_{\mathcal{C},c}}$, for $1 \leq j \leq m$. By the assumption, for every conjunct $\exists r_j.C_j$ we have that $\mathcal{C} \models D \sqsubseteq \exists r_j.C_j$. This, together with the fact that $C_j \in \mathsf{sub}(\mathcal{O})$ and with use of point (a) of Definition 5.3.3 gives us that $(D, C_j) \in r^{\mathcal{M}_{\mathcal{C},c}}$. As $\mathcal{C} \models C_j \sqsubseteq C_j$ and $C_j \in \mathsf{sub}(\mathcal{O})$, by induction hypothesis we get that $C_j \in C_j^{\mathcal{M}_{\mathcal{C},c}}$. This in turn gives us that $D \in (\exists r_j.C_j)^{\mathcal{M}_{\mathcal{C},c}}$, as required.

Now we show the direction "⇐". Assume that $D \in C^{\mathcal{M}_{\mathcal{C},c}}$, again, the proof is by induction on the construction of C. Suppose:

(a) $C = A$, for a concept name A, then we have $D \in A^{\mathcal{M}_{\mathcal{C},c}}$ and by definition $\mathcal{C} \models D \sqsubseteq A$.

(b) $C = A_1 \sqcap \ldots \sqcap A_n \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_m.C_m$. For every A_i, with $1 \leq i \leq n$, we can show that $\mathcal{C} \models D \sqsubseteq A_i$ in the same way as (a). So we have to show only that for every conjunct $\exists r_j.C_j$, with $1 \leq j \leq m$ we get $\mathcal{C} \models D \sqsubseteq \exists r_j.C_j$. As we have that $D \in (\exists r_j.C_j)^{\mathcal{M}_{\mathcal{C},c}}$, there is $C'_j \in C_j^{\mathcal{M}_{\mathcal{C},c}}$, such that $(D, C'_j) \in r_j^{\mathcal{M}_{\mathcal{C},c}}$. As $C_j \in \mathsf{sub}(\mathcal{O})$ it follows by the induction hypothesis that $\mathcal{C} \models C'_j \sqsubseteq C_j$ and $\mathcal{C} \models \exists r.C'_j \sqsubseteq \exists r.C_j$. Since $(D, C'_j) \in r_j^{\mathcal{M}_{\mathcal{C},c}}$ we have that $\mathcal{C} \models D \sqsubseteq \exists r_j.C'_j$, by transitivity of '$\sqsubseteq$' we have that $\mathcal{C} \models D \sqsubseteq \exists r_j.C_j$, as required.

By (1) and the fact that $\mathcal{M}_{\mathcal{C},c}$ is a model of $\mathcal{C}$ we have that $\mathcal{C} \models D \sqsubseteq C$.

(4) The direction (b) $\Rightarrow$ (a) follows from Theorem 5.3.2 and the fact that $D \in D^{\mathcal{M}_{\mathcal{C},c}}$. For opposite direction let $\mathcal{M}$ be a model of $\mathcal{C}$ and $d \in D^{\mathcal{M}}$. Define relation $S \subseteq \Delta^{\mathcal{M}_{\mathcal{C},c}} \times \Delta^{\mathcal{M}}$ by setting $(D,e) \in S$ iff $e \in D^{\mathcal{M}}$, for all $D \in \Delta^{\mathcal{M}_{\mathcal{C},c}}$. We have to show that S is a simulation. Assume that $D \in A^{\mathcal{M}_{\mathcal{C},c}}$, where A is a concept name. From that follows that $\mathcal{C} \models D \sqsubseteq A$, and by the fact that $\mathcal{M}$ is a model of $\mathcal{C}$ together with $e \in D^{\mathcal{M}}$ we receive $e \in A^{\mathcal{M}}$. Now assume that $(D, D') \in r^{\mathcal{M}_{\mathcal{C},c}}$. Thus we receive that $\mathcal{C} \models D \sqsubseteq \exists r.D'$, which implies $e \in \exists r.D'^{\mathcal{M}}$. Hence, there exists $e' \in \Delta^{\mathcal{M}}$ such that $e' \in D'^{\mathcal{M}}$ and $(e,e') \in r^{\mathcal{M}}$. From that follows that $(D', e') \in S$. From this follows that relation S is a simulation. By definition, we receive that $(D, d) \in S$.

(5) For arbitrary D, consider the following:

- (a) implies (b). Assume that $\mathcal{C} \models D \sqsubseteq C$. Then $D \in D^{\mathcal{M}_{\mathcal{C},c}}$. Since $\mathcal{M}_{\mathcal{C},\mathcal{C}}$ is a model of $\mathcal{C}$ we have that $D \in C^{\mathcal{M}_{\mathcal{C},c}}$.
- (b) implies (c). Follows from Point 4 of Lemma 5.3.5.
- (c) implies (a). Assume $(\mathcal{M}_{C,\mathcal{C}}, C) \leq (\mathcal{M}_{D,\mathcal{C}}, D)$. Let $\mathcal{M}$ be a model of $\mathcal{C}$ and $d \in D^{\mathcal{M}}$. To show $\mathcal{C} \models D \sqsubseteq C$, we have to show $d \in C^{\mathcal{M}}$. By Point 4 of Lemma 5.3.5, we receive that $(\mathcal{M}_{D,\mathcal{C}}, D) \leq (\mathcal{M}, d)$. By $(\mathcal{M}_{C,\mathcal{C}}, C) \leq (\mathcal{M}_{D,\mathcal{C}}, D)$ and transitivity of '$\leq$', we receive that $(\mathcal{M}_{C,\mathcal{C}}, C) \leq (\mathcal{M}, d)$. Again by Point 4 of Lemma 5.3.5, we receive that $d \in C^{\mathcal{M}}$. □

5.4 Characterization of Σ-entailment

In this section, we provide a characterization of Σ-entailment w.r.t. CBoxes in terms of canonical models.

Lemma 5.4.1. *Let $\mathcal{C} = (\mathcal{O}, \mathcal{R})$ be a CBox in $\mathcal{EL}^+$. Suppose $\mathcal{C} \models C \sqsubseteq \exists r.D$. Then one of the following holds:*

(a) there exists a $C' \in sub(\mathcal{O})$ such that $\mathcal{C} \models C \sqsubseteq \exists r.C'$ and $\mathcal{C} \models C' \sqsubseteq D$;

(b) there exists a C' such that $C \rightsquigarrow^r_{\mathcal{R}} C'$ and $\mathcal{C} \models C' \sqsubseteq D$.

Proof: Let $\mathcal{C} = (\mathcal{O}, \mathcal{R})$ be as in the lemma and let $\mathcal{C} \models C \sqsubseteq \exists r.D$. By Point 5 of Lemma 5.3.5, we have $C \in (\exists r.D)^{\mathcal{M}_{\mathcal{C},c}}$. Thus there is a $C' \in D^{\mathcal{M}_{\mathcal{C},c}}$ such that $(C, C') \in r^{\mathcal{M}_{\mathcal{C},c}}$. By Definition 5.3.3 of the canonical model $\mathcal{M}_{\mathcal{C},\mathcal{C}}$, it holds that:

(i) $\mathcal{C} \models C \sqsubseteq \exists r.C'$ and $C' \in \mathsf{sub}(\mathcal{O})$, or

(ii) $C \rightsquigarrow^r_{\mathcal{R}} C'$.

In both cases, it remains to show that $\mathcal{C} \models C' \sqsubseteq D$, which follows from $C' \in D^{\mathcal{M}_{\mathcal{C},c}}$ by Point 5 of Lemma 5.3.5. □

This lemma is essential for characterizing Σ-entailment. Intuitively, it states that, given an arbitrary large concept C, we can always find a possibly shorter concept with bounded outdegree that expresses the same "relevant" information. What information is considered relevant, is made precise by a set of consequences $K_{\mathcal{C}}(D)$ of a concept D in the presence of a CBox $\mathcal{C}$. The set $K_{\mathcal{C}}(D)$ is given as:

$$K_{\mathcal{C}}(D) = \{E \in \mathsf{cl}(\mathcal{C}) \mid \mathcal{C} \models D \sqsubseteq E\},$$

where $\mathsf{cl}(\mathcal{C}) = \mathsf{sub}(\mathcal{O}) \cup \{\exists r.C \mid C \in \mathsf{sub}(\mathcal{O}), r \text{ a role in } \mathcal{C}\}$.

Lemma 5.4.2 (Bounded outdegree). *For all $\mathcal{EL}^+$ CBoxes $\mathcal{C} = (\mathcal{O}, \mathcal{R})$ and concepts C in $\mathcal{EL}^+$, there is a concept D such that the following conditions are satisfied:*

1. $\emptyset \models C \sqsubseteq D$;
2. $K_{\mathcal{C}}(C) = K_{\mathcal{C}}(D)$;
3. $\|D\| \leq \|C\|$;
4. *the outdegree of D is bounded by $\|\mathcal{C}\|$.*

Proof: Let $\mathcal{C} = (\mathcal{O}, \mathcal{R})$ be a CBox and C a concept. If the outdegree of C is bounded by $\|\mathcal{C}\|$, then C itself is the wanted concept D. Assume that this is not the case. Then there is a subconcept C_0 of C, with

$$\|\mathcal{P}\| > \|\mathcal{C}\| \quad \text{for} \quad \mathcal{P} = \{(r, E) \mid \exists r.E \text{ is a conjunct of } C_0\},$$

such that there is a sequence (possibly empty) $r_1, ..., r_m$ of roles that occur in C and a sequence $E_0, ..., E_m$ of subconcepts of C such that:

- $E_m = C, E_0 = C_0$, and
- $\exists r_i.E_{i-1}$ is a conjunct of E_i for all i with $1 \leq i \leq m$.

Thus $C_0 = F \sqcap \bigsqcup_{(r,E)\in\mathcal{P}} \exists r.E$ where F is a conjunction of the concept names in C_0, let Q be a minimal subset of $\mathcal{P}$ such that for all $\exists s.G \in \mathsf{cl}(\mathcal{C})$, if there is a $(r, E) \in \mathcal{P}$ such that $\mathcal{C} \models \exists r_i. \cdots \exists r.E \sqsubseteq \exists s.G$, with $i = 0, ..., m$ (where by $i = 0$ we mean that this part of the path is empty, so we have $\mathcal{C} \models r.E \sqsubseteq \exists s.G$), then there is a $(r', E') \in Q$ such that $\mathcal{C} \models \exists r_i. \cdots \exists r'.E' \sqsubseteq \exists s.G$. Notice that the cardinality of Q is bounded by $\|\mathcal{C}\|$. Now, replace in C the subconcept C_0 with $C_1 := F \sqcap \bigsqcap_{(r',E')\in Q} \exists r'.E'$ and call the result C'. Obviously $\|C'\| \leq \|C\|$. To obtain the desired concept D, it thus suffices to execute the described contraction procedure until the outdegree is bounded by $\|\mathcal{C}\|$. Clearly, $\emptyset \models C \sqsubseteq C'$. In what follows, we show that for all $E \in \mathsf{cl}(\mathcal{C})$:

$$\mathcal{C} \models C \sqsubseteq E \quad \text{iff} \quad \mathcal{C} \models C' \sqsubseteq E.$$

"$\Leftarrow$". Immediate consequence of $\emptyset \models C \sqsubseteq C'$.
"$\Rightarrow$". We show this direction by contraposition. Suppose $\mathcal{C} \not\models C' \sqsubseteq H$ for some $H \in \mathsf{cl}(\mathcal{C})$. We have to show that $\mathcal{C} \not\models C \sqsubseteq H$. There is a model $\mathcal{M}$ of $\mathcal{C}$ with $d_0 \in C'^{\mathcal{M}} \setminus H^{\mathcal{M}}$. For each $(r, E) \in \mathcal{P} \setminus Q$, take a copy $\mathcal{M}_{r,E}$ of the canonical model $\mathcal{M}_{E,\mathcal{C}}$ in which $E = d_{r,E}$,

such that all these copies have disjoint domains, and their domains are disjoint from that of $\mathcal{M}$. We now define a model $\mathcal{M}'$ of $\mathcal{C}$ that refutes $C \sqsubseteq H$.
First, we introduce some auxiliary notion. Given $x \in \Delta^{\mathcal{M}}$, $y \in \Delta^{\mathcal{M}_{r,E}}$ and $d \in C_1^{\mathcal{M}}$, we say that x and y are connected via d and induce s, if there are two sequences of roles $r_1, ..., r_n$ and $s_1, ..., s_m$ such that $(x, d) \in r_1^{\mathcal{M}} \circ ... \circ r_n^{\mathcal{M}}$, $(d_{r,E}, y) \in s_1^{\mathcal{M}_{r,E}} \circ ... \circ s_m^{\mathcal{M}_{r,E}}$, and $\mathcal{R} \models r_1 \circ ... \circ r_n \circ r \circ s_1 \circ ... s_m \sqsubseteq s$.
Define the interpretation $\mathcal{M}' = (\Delta^{\mathcal{M}'}, \{A^{\mathcal{M}'}\}_{A \in P}, \{r^{\mathcal{M}'}\}_{r \in R})$ as follows:

- $\Delta^{\mathcal{M}'} := \Delta^{\mathcal{M}} \uplus \biguplus_{(s,E) \in \mathcal{P} \setminus Q} \Delta^{\mathcal{M}_{s,E}}$
- $A^{\mathcal{M}'} := A^{\mathcal{M}} \cup \bigcup_{(s,E) \in \mathcal{P} \setminus Q} A^{\mathcal{M}_{s,E}}$, for all $A \in P$
- $r^{\mathcal{M}'} := r^{\mathcal{M}} \cup \bigcup_{(s,E) \in \mathcal{P} \setminus Q} r^{\mathcal{M}_{s,E}} \cup \bigcup \{(x, y) \in \Delta^{\mathcal{M}} \times \Delta^{\mathcal{M}_{s,E}} \mid (s, E) \in \mathcal{P} \setminus Q$ and $\exists d \in C_1^{\mathcal{M}}$, such that x and y are connected via d and induce $r\}$

For illustration of $\mathcal{M}'$ see Figure 5.2.
It is possible to prove the following:

1. $d_0 \in C^{\mathcal{M}'}$;
2. for all $(r, E) \in \mathcal{P} \setminus Q$, all $d \in \Delta^{\mathcal{M}_{r,E}}$, and all concepts D_0, $d \in D_0^{\mathcal{M}'}$ iff $d \in D_0^{\mathcal{M}_{r,E}}$;
3. for all $d \in \Delta^{\mathcal{M}}$ and $D_0 \in \mathsf{cl}(\mathcal{C})$, $d \in D_0^{\mathcal{M}}$ iff $d \in D_0^{\mathcal{M}'}$;
4. we have $\mathcal{M}' \models \mathcal{R}$.

Point (1) is clear by definition.
Point (2) follows from the fact that model $\mathcal{M}_{r,E}$ is a generated submodel of $\mathcal{M}'$, i.e. if $(e, e') \in r^{\mathcal{M}'}$ and $e \in \Delta^{\mathcal{M}_{r,E}}$, then $(e, e') \in r^{\mathcal{M}_{r,E}}$.
Point (3). The proof is by the induction on the structure of D_0, the only interesting case is that with $D_0 = \exists r.D_0'$.
"$\Rightarrow$": Let $d \in (\exists r.D_0')^{\mathcal{M}}$. Then there is a $d' \in D_0'^{\mathcal{M}}$ such that $(d, d') \in r^{\mathcal{M}}$. By definition of $\mathcal{M}'$, we have $(d, d') \in r^{\mathcal{M}'}$. By the induction hypothesis, $d' \in D_0'^{\mathcal{M}'}$. Hence, $d \in (\exists r.D_0')^{\mathcal{M}'}$.
"$\Leftarrow$": Let $d \in (\exists r.D_0')^{\mathcal{M}'}$. Then there is a $d' \in D_0'^{\mathcal{M}'}$ such that $(d, d') \in r^{\mathcal{M}'}$. Distinguish two cases: First, $d' \in \Delta^{\mathcal{M}}$. Then $(d, d') \in r^{\mathcal{M}}$ by definition of $\mathcal{M}'$. By the induction hypothesis, $d' \in D_0'^{\mathcal{M}}$ and thus $d \in (\exists r.D_0')^{\mathcal{M}}$.
Second, $d' \in \Delta^{\mathcal{M}'} \setminus \Delta^{\mathcal{M}}$. By construction of $\mathcal{M}'$, we have $d' \in \Delta^{\mathcal{M}_{s,E}}$ for some $(s, E) \in \mathcal{P} \setminus Q$. It follows by Point (2) above that $d' \in D_0'^{\mathcal{M}'}$ implies $d' \in D_0'^{\mathcal{M}_{s,E}}$. Since $\mathcal{M}_{s,E}$ is a copy of the canonical model $\mathcal{M}_{E,\mathcal{C}}$, we have $d' = E'$ for some $E' \in \Delta^{\mathcal{M}_{E,\mathcal{C}}}$ and $E' \in D_0'^{\mathcal{M}_{E,\mathcal{C}}}$. By Point 3 of Lemma 5.3.5, we get $\mathcal{C} \models E' \sqsubseteq D_0'$.

By definition of $\mathcal{M}'$, d and d' are connected via d_m, i.e., there are $d_m \in C_1^{\mathcal{M}}$ and $d_{s,E} \in \Delta^{\mathcal{M}_{s,E}}$ and sequences $r_1, \cdots, r_m$, $s_1, \cdots, s_n$ of roles with $m, n \geq 0$ such that $(d, d_m) \in r_1^{\mathcal{M}'} \circ \cdots \circ r_m^{\mathcal{M}'}$, $(d_m, d_{s,E}) \in s^{\mathcal{M}'}$ and $(d_{s,E}, d') \in s_1^{\mathcal{M}'} \circ \cdots \circ s_n^{\mathcal{M}'}$ and $\mathcal{R} \models r_1 \circ \cdots \circ r_m \circ s \circ s_1 \circ \cdots \circ s_n \sqsubseteq r$. Thus, $\mathcal{C} \models \exists r_1. \cdots \exists r_m. \exists s. \exists s_1. \cdots \exists s_n.E' \sqsubseteq \exists r.E'$. Since $d_{s,E} = E$, we get $\mathcal{C} \models E \sqsubseteq \exists s_1.\dots.\exists s_n.E'$, thus $\mathcal{C} \models \exists r_1.\dots.\exists r_m.\exists s.E \sqsubseteq \exists r.E'$. Since $\mathcal{C} \models E' \sqsubseteq D_0'$, we receive $\mathcal{C} \models \exists r_1.\dots.\exists r_m.\exists s.E \sqsubseteq \exists r.D_0'$. By definition of Q, there is an $(s', E'') \in Q$ such that $\mathcal{C} \models \exists r_1.\dots.\exists r_m.\exists s'.E'' \sqsubseteq \exists r.D_0'$. Since $d_m \in C_1^{\mathcal{M}}$ and $\exists s'.E''$ is a conjunct of $C_1^{\mathcal{M}}$, we have $d_m \in (\exists s'.E'')^{\mathcal{M}}$ and, thus, $d \in (\exists r_1.\cdots\exists r_m.\exists s'.E'')^{\mathcal{M}}$. And again by Lemma 5.3.5, we get $d \in (\exists r.D_0')^{\mathcal{M}}$.

Point (4). To show that $\mathcal{M}' \models \mathcal{R}$ assume that $r_1 \circ \cdots \circ r_n \sqsubseteq s \in \mathcal{R}$ and $(d', d'') \in r_1^{\mathcal{M}'} \circ \cdots \circ r_n^{\mathcal{M}'}$, we show that $(d', d'') \in s^{\mathcal{M}'}$. We have to consider two cases:

1. $d' \in \Delta^{\mathcal{M}_{r,E}}$, this implies $(d', d'') \in r_1^{\mathcal{M}_{r,E}} \circ \cdots \circ r_n^{\mathcal{M}_{r,E}}$. Since $\mathcal{M}_{r,E} \models \mathcal{R}$, we receive that $(d', d'') \in s^{\mathcal{M}_{r,E}}$, this implies that $(d', d'') \in s^{\mathcal{M}'}$.

2. $d' \in \Delta^{\mathcal{M}}$, then we have to consider two further cases:

   a) $d'' \in \Delta^{\mathcal{M}}$, then $(d', d'') \in r_1^{\mathcal{M}} \circ \cdots \circ r_n^{\mathcal{M}}$. Since $\mathcal{M} \models \mathcal{R}$, we receive that $(d', d'') \in s^{\mathcal{M}}$, this implies that $(d', d'') \in s^{\mathcal{M}'}$.

   b) $d'' \in \Delta^{\mathcal{M}_{r,E}}$, then d' and d'' are connected via d, i.e. there are $d \in C_1^{\mathcal{M}}$ and $d_{r,E} \in \Delta^{\mathcal{M}_{r,E}}$ together with sequences of roles $r_1, \dots r_k$ and $r_m, \dots r_n$, such that $(d', d) \in r_1^{\mathcal{M}} \circ \dots \circ r_k^{\mathcal{M}}$, $(d_{r,E}, d'') \in r_m^{\mathcal{M}_{r,E}} \circ \dots \circ r_n^{\mathcal{M}_{r,E}}$ and $r = r_l$ with $k < l < m$. This together with the assumption induces role s such that $(d', d'') \in s^{\mathcal{M}'}$.

Since $\mathcal{M}$ and every $\mathcal{M}_{r,E}$ are models of $\mathcal{C}$ and by points (2), (3) and (4) above, it follows that $\mathcal{M}'$ is a model of $\mathcal{C}$. Since $d_0 \in C'^{\mathcal{M}} \setminus H^{\mathcal{M}}$, Point (3) implies that $d_0 \notin H^{\mathcal{M}'}$. By Point (1) we have $d_0 \in C^{\mathcal{M}'}$ which implies $\mathcal{C} \not\models C \sqsubseteq H$. □

For the following characterization of Σ-entailment, we use a relation '$\Rightarrow_1$' on concepts. Let $\mathcal{C}_1, \mathcal{C}_2$ be CBoxes, C a Σ-concept, and D a Sig($\mathcal{C}_2$)-concept. We write $C \Rightarrow_1 D$ if, and only if, for all Σ-concepts E, $\mathcal{C}_2 \models D \sqsubseteq E$ implies $\mathcal{C}_1 \models C \sqsubseteq E$.

Lemma 5.4.3 (Characterization of non-Σ-entailment). *Let $\mathcal{C}_1 = (\mathcal{O}_1, \mathcal{R}_1)$ and $\mathcal{C}_2 = (\mathcal{O}_2, \mathcal{R}_2)$ be two $\mathcal{EL}^+$ CBoxes, and $\Sigma \subseteq$ Sig($\mathcal{C}_2$) a signature and assume $\mathcal{R}_1 \sqsubseteq_\Sigma \mathcal{R}_2$. Then* $\mathsf{Diff}_\Sigma(\mathcal{C}_1, \mathcal{C}_2) \neq \emptyset$ *if, and only if, there is a Σ-concept C and a Sig($\mathcal{C}_2$)-concept $D \in \mathsf{cl}(\mathcal{C}_2)$ such that:*

(a) $\mathcal{C}_2 \models C \sqsubseteq D$;

(b) $C \not\Rightarrow_1 D$;

(c) the outdegree of C is bounded by $\|\mathcal{C}_2\|$.

Proof: "⇐". Assume that (a) to (c) are satisfied. By (b), there is a Σ-concept E with $\mathcal{C}_2 \models D \sqsubseteq E$ and $\mathcal{C}_1 \not\models C \sqsubseteq E$. From $\mathcal{C}_2 \models D \sqsubseteq E$ and (a), it follows $\mathcal{C}_2 \models C \sqsubseteq E$, which implies that $C \sqsubseteq E \in \mathsf{Diff}_\Sigma(\mathcal{C}_1, \mathcal{C}_2)$.

"⇒". Suppose $C \sqsubseteq D \in \mathsf{Diff}_\Sigma(\mathcal{C}_1, \mathcal{C}_2)$. We first show (a) and (b). Note that C and D are Σ-concepts. If $D \in \mathsf{sub}(\mathcal{C}_2)$, we are done: we have $\mathcal{C}_2 \models C \sqsubseteq D$ and $\mathcal{C}_1 \not\models C \sqsubseteq D$, therefore $C \not\Rightarrow_1 D$. Otherwise, assume that for all $C \sqsubseteq D \in \mathsf{Diff}_\Sigma(\mathcal{C}_1, \mathcal{C}_2)$, $D \notin \mathsf{sub}(\mathcal{C}_2)$.

Let $C \sqsubseteq D$ be minimal in the sense that there is no $C' \sqsubseteq D' \in \mathsf{Diff}_\Sigma(\mathcal{C}_1, \mathcal{C}_2)$ with D' shorter than D. Then D is of the form $\exists r.D'$:

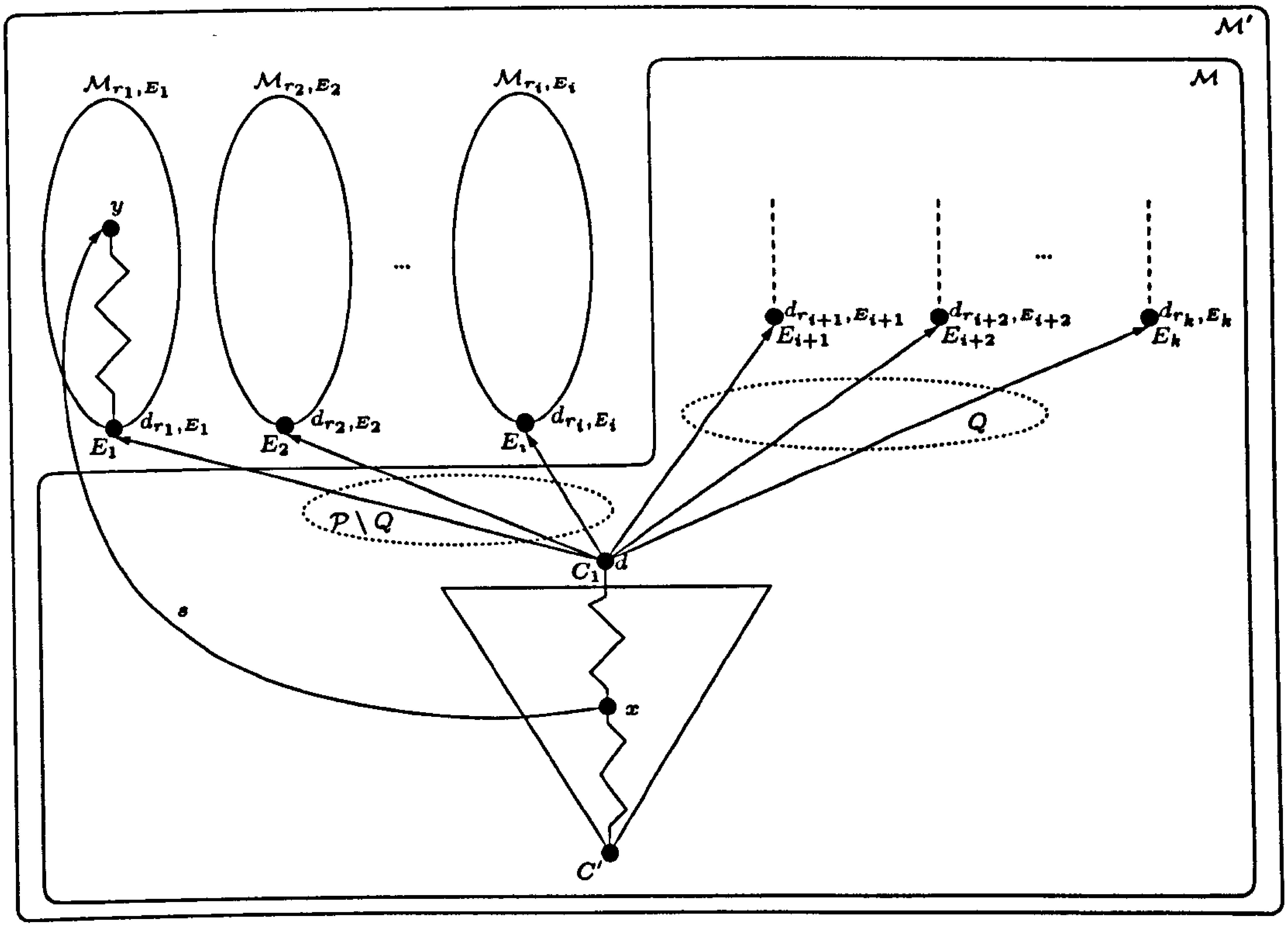


Figure 5.2: The interpretation $\mathcal{M}'$.

- If $D = \top$, then $\mathcal{C}_1 \models C \sqsubseteq D$, contradicting the fact that $C \sqsubseteq D \in \mathsf{Diff}_\Sigma(\mathcal{C}_1, \mathcal{C}_2)$.
- If D is an atomic concept, then $D \in \mathsf{sub}(\mathcal{C}_2)$, which we have assumed not be the case.
- If D is a conjunction $D_1 \sqcap D_2$, then $\mathcal{C}_2 \models C \sqsubseteq D_i$ for all $i \in \{1,2\}$, and $\mathcal{C}_1 \not\models C \sqsubseteq D_i$ for some $i \in \{1,2\}$. Thus, one of $C \sqsubseteq D_1$ and $C \sqsubseteq D_2$ is in $\mathsf{Diff}_\Sigma(\mathcal{C}_1, \mathcal{C}_2)$. Thus, D is not minimal.

By Lemma 5.4.1, $\mathcal{C}_2 \models C \sqsubseteq \exists r.D'$ implies that one of the following holds:

1. there is a C' such that $C \leadsto^r_{\mathcal{R}_2} C'$ and $\mathcal{C}_2 \models C' \sqsubseteq D'$;
2. there is a $C' \in \mathsf{sub}(\mathcal{O}_2)$ such that $\mathcal{C}_2 \models C \sqsubseteq \exists r.C'$ and $\mathcal{C}_2 \models C' \sqsubseteq D'$.

Observe that, in Case (1), we have $\mathcal{C}_1 \models C' \sqsubseteq D'$. Suppose not, i.e., $\mathcal{C}_1 \not\models C' \sqsubseteq D'$. Together with $\mathcal{C}_2 \models C' \sqsubseteq D'$, this implies that $C' \sqsubseteq D' \in \mathsf{Diff}_\Sigma(\mathcal{C}_1, \mathcal{C}_2)$; contradicting the minimality of D.

Now suppose that Case (1) applies. From $C \leadsto^r_{\mathcal{R}_2} C'$ together with the fact that we assumed $\mathcal{R}_1 \sqsubseteq_\Sigma \mathcal{R}_2$, it follows $\mathcal{C}_1 \models C \sqsubseteq \exists r.C'$. Since in Case (1) it holds $\mathcal{C}_1 \models C' \sqsubseteq D'$, we get $\mathcal{C}_1 \models \exists r.C' \sqsubseteq \exists r.D'$. This implies $\mathcal{C}_1 \models C \sqsubseteq \exists r.D'$. As $D = \exists r.D'$, we have $\mathcal{C}_1 \models C \sqsubseteq D$, contradicting the fact that $C \sqsubseteq D \in \mathsf{Diff}_\Sigma(\mathcal{C}_1, \mathcal{C}_2)$.

Consequently, Case (2) holds. Substitute $\exists r.C'$ for concept D. We show that the Conditions (a) and (b) hold for the concept C and $\exists r.C'$. Condition (a) follows from $\mathcal{C}_2 \models C \sqsubseteq \exists r.C'$ by Case (2). For Condition (b), recall that $\mathcal{C}_1 \not\models C \sqsubseteq \exists r.D'$ since we assumed $C \sqsubseteq \exists r.D' \in \mathsf{Diff}_\Sigma(\mathcal{C}_1, \mathcal{C}_2)$. Note that, by (2), we also have $\mathcal{C}_2 \models C' \sqsubseteq D'$, which implies $\mathcal{C}_2 \models \exists r.C' \sqsubseteq \exists r.D'$ and, then, $\mathcal{C}_2 \models C \sqsubseteq \exists r.D'$. By the fact that $\exists r.D'$ is a Σ-concept, we have $C \not\Rightarrow_1 \exists r.C'$.

It remains to show that Condition (c) is satisfied. Suppose the concepts C and D satisfy Conditions (a) and (b), $\mathsf{Sig}(C) \subseteq \Sigma$, and D in $\mathsf{sub}(\mathcal{O}_2)$. Take a concept C' that satisfies the four conditions of Lemma 5.4.2 for $\mathcal{C}_2$. By Condition 4, C' satisfies Point (c). Condition 2 implies that C' and D satisfy (a). Since $C \not\Rightarrow_1 D$, there is a Σ-concept E such that $\mathcal{C}_2 \models D \sqsubseteq E$ and $\mathcal{C}_1 \not\models C \sqsubseteq E$. By Condition 1, it follows $\mathcal{C}_1 \models C \sqsubseteq C'$. This implies $\mathcal{C}_1 \not\models C' \sqsubseteq E$. Then we have that $C' \not\Rightarrow_1 D$. Hence, C' and D satisfy Point (b). □

The next lemma characterizes the relation "$\Rightarrow_1$" semantically in terms of Σ-simulation between canonical models. Moreover, it states that membership in "$\Rightarrow_1$" can be decided in polynomial time.

Lemma 5.4.4 (Semantic characterization). *Let $\mathcal{C}_1, \mathcal{C}_2$ be $\mathcal{EL}^+$ CBoxes and C, D concepts in $\mathcal{EL}^+$. Then we have $C \Rightarrow_1 D$ if, and only if, $(\mathcal{M}_{D,\mathcal{C}_2}, D) \leq_\Sigma (\mathcal{M}_{C,\mathcal{C}_1}, C)$. Hence, the problem $C \Rightarrow_1 D$ is decidable in polynomial time in the size of C, D, $\mathcal{C}_1$ and $\mathcal{C}_2$.*

Proof: "⇒". Suppose $C \not\Rightarrow_1 D$. Then there is a Σ-concept E such that $\mathcal{C}_2 \models D \sqsubseteq E$ and $\mathcal{C}_1 \not\models C \sqsubseteq E$. By Point 3 of Lemma 5.3.5, this yields $D \in E^{\mathcal{M}_{D,\mathcal{C}_2}}$ and $C \notin E^{\mathcal{M}_{C,\mathcal{C}_1}}$. Hence, by Theorem 5.3.2, we receive that $(\mathcal{M}_{D,\mathcal{C}_2}, D) \not\leq_\Sigma (\mathcal{M}_{C,\mathcal{C}_1}, C)$.

"⇐". Let $(\mathcal{M}_{D,\mathcal{C}_2}, D) \not\leq_\Sigma (\mathcal{M}_{C,\mathcal{C}_1}, C)$. By Theorem 5.3.2, there exists a Σ-concept E with $D \in E^{\mathcal{M}_{D,\mathcal{C}_2}}$. By Point 2 of Lemma 5.3.5, we have $\mathcal{C}_2 \models D \sqsubseteq E$ and $\mathcal{C}_1 \not\models C \sqsubseteq E$. Hence, $C \not\Rightarrow_1 D$. It is well-known that computing the largest Σ-simulation between two finite graphs can be done in polynomial time [26]. □

5.5 Algorithm

While previously we have presented results for $\mathcal{EL}^+$, in this section we present an algorithm for deciding Σ-entailment for its restriction, description logic $\mathcal{ELSH}$. The reason for that is the fact that in our approach the problem for $\mathcal{EL}^+$ cannot be decided, as Lemma 5.5.3 does not hold in $\mathcal{EL}^+$. Therefore for description logic $\mathcal{EL}^+$ the problem remains open.

Before we introduce the algorithm we need some additional lemmas.

Lemma 5.5.1. *Let $\mathcal{C} = (\mathcal{O}, \mathcal{R})$ be a $\mathcal{ELSH}$ CBox, then the following holds for $n \geq 2$:*

$$\mathcal{C} \models r_1 \circ \cdots \circ r_n \sqsubseteq r \quad \text{iff} \quad \mathcal{R} \models r_1 \sqsubseteq r, \ldots, \mathcal{R} \models r_n \sqsubseteq r \text{ and } \mathcal{R} \models r \circ r \sqsubseteq r\,.$$

Proof: This follows from the construction of canonical models $\mathcal{EL}$ as in [8]. □

As in this section we consider description logic $\mathcal{ELSH}$, we take into account a simplified version of relation "⇝" from Definition 5.3.3. This is presented in the following lemma.

Lemma 5.5.2. *If $\mathcal{R}$ is an $\mathcal{ELSH}$ role box, then $C \leadsto^{r}_{\mathcal{R}} C'$ iff:*

(a) there exists a sequence $\exists r_0.D_0, \ldots, \exists r_n.D_n$ with $n \geq 1$, where $\exists r_0.D_0$ a conjunct of C, $\exists r_{i+1}.D_{i+1}$ a conjunct of D_i for $0 \leq i < n$ such that $D_n = C'$ and for every r_i with $0 \leq i \leq n$ we have $\mathcal{R} \models r_i \sqsubseteq r$ and $\mathcal{R} \models r \circ r \sqsubseteq r$, or

(b) there exists $\exists r'.C'$ a conjunct of C and $\mathcal{R} \models r' \sqsubseteq r$.

We present an algorithm for deciding Σ-entailment for $\mathcal{ELSH}$. For CBoxes $\mathcal{C}_1 = (\mathcal{O}_1, \mathcal{R}_1)$ and $\mathcal{C}_2 = (\mathcal{O}_2, \mathcal{R}_2)$, to check whether $\mathcal{C}_1 \sqsubseteq_\Sigma \mathcal{C}_2$, the algorithm enters two stages. In the first stage the algorithm checks if $\mathcal{R}_1 \sqsubseteq_\Sigma \mathcal{R}_2$. In other words, the algorithm first computes sets of "relevant" role inclusions entailed by $\mathcal{R}_1$ and $\mathcal{R}_2$, defined for any $\mathcal{R}$ by

$$\mathfrak{R} = \{r \sqsubseteq s \mid \mathcal{R} \models r \sqsubseteq s \text{ and } r, s \in \Sigma\} \cup \{r \circ r \sqsubseteq r \mid r \circ r \sqsubseteq r \in \mathcal{R} \text{ and } r \in \Sigma\} .$$

This set is computed in polynomial time. After computing sets $\mathfrak{R}_1$ and $\mathfrak{R}_2$ the algorithm computes $\mathfrak{R}_2 \setminus \mathfrak{R}_1$. If $\mathfrak{R}_2 \setminus \mathfrak{R}_1 \neq \emptyset$, then $\mathcal{R}_1 \not\sqsubseteq_\Sigma \mathcal{R}_2$ and thus $\mathcal{C}_1 \not\sqsubseteq_\Sigma \mathcal{C}_2$. In case $\mathfrak{R}_2 \setminus \mathfrak{R}_1 = \emptyset$ the algorithm proceeds to the second stage. In the second stage the algorithm searches for a Σ-concept C such that for some $D \in \mathsf{sub}(\mathcal{O}_2)$, the Points (a)–(c) of Lemma 5.4.3 are satisfied. The algorithm proceeds in rounds. In the first round, Points (a) and (b) are checked for all conjunctions C of concept names from Σ and all $D \in \mathsf{sub}(\mathcal{O}_2)$. Each check can be done in polynomial time by Lemma 5.4.4. In case, no suitable C is found in round one, the algorithm proceeds to the second round in which concepts C of role depth one are considered. Here C is a conjunction of concept names from Σ and concepts of the form $\exists r.E$, where r is a role from Σ and E is a candidate for C from the previous round, i.e., E is a conjunction of concept names. By Point (c) of Lemma 5.4.3, we only have to consider those Cs with no more than $\|\mathcal{O}_2\|$ many conjuncts of the form $\exists r.E$. For checking Points (a) and (b), we make use of the information we have gained about the Es in the previous round. If still no suitable C is found, the algorithm starts round three that checks concepts C of role depth two in which we reuse the Cs from the second round as role successors. If again no suitable concept C was found, the algorithm proceeds to the next round, etc.

To avoid constructing doubly exponentially large concepts C, the algorithm uses a succinct data structure that represents the relevant information about C. Which information about C is relevant can be read of the characterization of Σ-entailment in Lemma 5.4.3: For every C, take the quintuple

$$C^\sharp = (\mathcal{Q}_0, \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3, \mathcal{Q}_4),$$

where the set $\mathcal{Q}_0$ contains all concept names occurring in the top-level conjunction of C,

$$\begin{aligned}
\mathcal{Q}_1 &= K_{\mathcal{C}_1}(C),\\
\mathcal{Q}_2 &= K_{\mathcal{C}_2}(C),\\
\mathcal{Q}_3 &= \{(r, D') \in (\Sigma \cap R) \times \mathsf{sub}(\mathcal{O}_2) \mid C' \Rightarrow_1 D' \text{ and}\\
&\qquad \mathcal{C}_1 \models C \sqsubseteq \exists r.C' \text{ with } C' \in \mathsf{sub}(\mathcal{O}_1) \text{ or } C \leadsto^{r}_{\mathcal{R}_1} C'\}, \text{ and}\\
\mathcal{Q}_4 &= \{D \in \mathsf{sub}(\mathcal{O}_2) \mid C \Rightarrow_1 D\}.
\end{aligned}$$

The quintuple $C^\sharp$ is said to be **determined** by C. Intuitively, the components $\mathcal{Q}_1$ and $\mathcal{Q}_2$ contain concepts that are implied by C in the context of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, and $\mathcal{Q}_4$ contains concepts which, while being implied by D in the context of $\mathcal{C}_2$, can be Σ-simulated by C in the context of $\mathcal{C}_1$.

According to Lemma 5.4.3, the quintuple $C^\sharp$ determined by a concept C contains sufficient information to decide whether C is the left-hand side of a CI witnessing the logical difference between two CBoxes. Moreover, the information in $C^\sharp$ enables the recursive search described above and to formulate a termination condition for the algorithm to run in exponential time.

Figure 5.3 presents the algorithm for deciding Σ-entailment for $\mathcal{ELSH}$. Observe that the termination condition $\mathcal{Q}_2 \setminus \mathcal{Q}_4 \neq \emptyset$ corresponds to satisfaction of Points (a) and (b) in Lemma 5.4.3. Note that Point (a) in the definition of the set $\mathcal{F}_3$ uses canonical models, which are constructed on demand in polynomial time.

Before we continue to show correctness of the algorithm, in Lemma 5.5.4 we explicitly state the concepts that determine the quintuples constructed in Step 3 of Figure 5.3. But first we need an auxiliary lemma.

Lemma 5.5.3. *Let $\mathcal{C} = (\mathcal{O}, \mathcal{R})$ be a CBox in $\mathcal{ELSH}$. Let the concepts C and D be given as*

$$C = F \sqcap \bigsqcap_{(r,E)\in Q} \exists r.E, \qquad D = F \sqcap \bigsqcap_{(r,E)\in Q} \exists r.(\bigsqcap_{G\in K_{\mathcal{C}}(E)} G),$$

where F is a conjunction of concept names. Then $K_{\mathcal{C}}(C) = K_{\mathcal{C}}(D)$.

Proof: We show set-entailment in both directions.
"$\supseteq$". This follows from $\mathcal{C} \models C \sqsubseteq D$, which follows from $\mathcal{C} \models E \sqsubseteq \bigsqcap_{G\in K_{\mathcal{C}}(E)} G$.
"$\subseteq$". We show the contrapositive. Let $H \in \mathsf{cl}(\mathcal{C}) \setminus K_{\mathcal{C}}(D)$. We show $H \notin K_{\mathcal{C}}(C)$. Consider a model $\mathcal{M}$ of $\mathcal{C}$ with $d_0 \in D^{\mathcal{M}} \setminus H^{\mathcal{M}}$. We may assume that d_0 has no predecessor (i.e., $\{d \mid (d, d_0) \in r^{\mathcal{M}}\} = \emptyset$ for all r). For each $(r, E) \in Q$, take a copy $\mathcal{M}_{r,E}$ of the canonical model $\mathcal{M}_{E,\mathcal{C}}$ such that all these copies have mutually disjoint domains and are disjoint with $\mathcal{M}$. Denote the copy of E in $\mathcal{M}_{r,E}$ by $d_{r,E}$. Define a new interpretation $\mathcal{M}'$ as follows:

- $\Delta^{\mathcal{M}'} := \Delta^{\mathcal{M}} \uplus \biguplus_{(r,E)\in Q} \Delta^{\mathcal{M}_{r,E}}$,
- $A^{\mathcal{M}'} := A^{\mathcal{M}} \cup \bigcup_{(r,E)\in Q} A^{\mathcal{M}_{r,E}}$, for all $A \in P$,
- $s^{\mathcal{M}'} := s^{\mathcal{M}} \cup \bigcup_{(r,E)\in Q} s^{\mathcal{M}_{r,E}} \cup \{(d_0, d_{r,E}) \mid \mathcal{R} \models r \sqsubseteq s, (r, E) \in Q\}$
  $\cup \{(d_0, d') \mid \exists (r, E) \in Q, \exists\, r' : (d_{r,E}, d') \in r' \text{ and } \mathcal{R} \models r \circ r' \sqsubseteq s\}$.

The following can be shown by induction on the structure of D_0:

(i) for all $(r, E) \in Q$, all $d \in \Delta^{\mathcal{M}_{r,E}}$, and all concepts D_0: $d \in D_0^{\mathcal{M}'}$ iff $d \in D_0^{\mathcal{M}_{r,E}}$;

(ii) for all $d \in \Delta^{\mathcal{M}}$ and $D_0 \in \mathsf{cl}(\mathcal{C})$: $d \in D_0^{\mathcal{M}}$ iff $d \in D_0^{\mathcal{M}'}$.

Input: CBoxes $\mathcal{C}_1 = (\mathcal{O}_1, \mathcal{R}_1)$ and $\mathcal{C}_2 = (\mathcal{O}_2, \mathcal{R}_2)$ and signature $\Sigma \subseteq \mathsf{Sig}(\mathcal{C}_2)$.

Stage 1:

1. Compute $\mathfrak{R}_1$ and $\mathfrak{R}_2$.
2. Compute the set $\mathfrak{R}_2 \setminus \mathfrak{R}_1$.

If $\mathfrak{R}_2 \setminus \mathfrak{R}_1 \neq \emptyset$, then output '$\mathcal{C}_1 \not\sqsupseteq_\Sigma \mathcal{C}_2$', otherwise proceed to Stage 2.

Stage 2:

1. Compute the set $\mathcal{N}_0$ of quintuples determined by conjunctions of concept names from Σ.
2. If $\mathcal{N}_0$ contains a quintuple $(\mathcal{Q}_0, \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3, \mathcal{Q}_4)$ such that $\mathcal{Q}_2 \setminus \mathcal{Q}_4 \neq \emptyset$, then output '$\mathcal{C}_1 \not\sqsupseteq_\Sigma \mathcal{C}_2$'.
3. Generate the sequence $\mathcal{N}_1, \mathcal{N}_2, \ldots$ of sets of quintuples such that $\mathcal{N}_{i+1} = \mathcal{N}_i \cup \mathcal{N}_i'$, where $\mathcal{N}_i'$ is the set of quintuples $(\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4)$ which can be obtained from a set $\mathcal{F}_0$ of concept names from Σ and a set $\mathcal{Q} \subseteq (\Sigma \cap R) \times \mathcal{N}_i$ of cardinality not exceeding $\|\mathcal{O}_2\|$ in the following way:

   - $\mathcal{F}_1 = K_{\mathcal{C}_1}(\bigsqcap_{A \in \mathcal{F}_0} A \sqcap \bigsqcap_{(r,(\mathcal{Q}_0,\mathcal{Q}_1,\mathcal{Q}_2,\mathcal{Q}_3,\mathcal{Q}_4)) \in \mathcal{Q}} \exists r.(\bigsqcap_{D \in \mathcal{Q}_1} D))$;
   - $\mathcal{F}_2 = K_{\mathcal{C}_2}(\bigsqcap_{A \in \mathcal{F}_0} A \sqcap \bigsqcap_{(r,(\mathcal{Q}_0,\mathcal{Q}_1,\mathcal{Q}_2,\mathcal{Q}_3,\mathcal{Q}_4)) \in \mathcal{Q}} \exists r.(\bigsqcap_{D \in \mathcal{Q}_2} D))$;
   - $\mathcal{F}_3 = \{(r, D) \in (\Sigma \cap R) \times \mathsf{sub}(\mathcal{C}_2) \mid$
     (a) there is a $\exists r.C' \in \mathcal{F}_1$ such that $(\mathcal{M}_{D,\mathcal{C}_2}, D) \leq_\Sigma (\mathcal{M}_{C',\mathcal{C}_1}, C')$; or
     (b) there is a $(s, (\mathcal{Q}_0, \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3, \mathcal{Q}_4)) \in \mathcal{Q}$ such that:
       (i) $D \in \mathcal{Q}_4$ and $\mathcal{R}_1 \models s \sqsubseteq r$; or
       (ii) $(t, D) \in \mathcal{Q}_3$, $\mathcal{R}_1 \models s \sqsubseteq r$, $\mathcal{R}_1 \models t \sqsubseteq r$, and $r \circ r \sqsubseteq r \in \mathcal{R}_1\}$;
   - $\mathcal{F}_4 = \{D \in \mathsf{sub}(\mathcal{C}_2) \mid$
     (a) for all $A \in \Sigma$, $A \in K_{\mathcal{C}_2}(D)$ implies $A \in \mathcal{F}_1$; and
     (b) for all $r \in \Sigma$, $(D, D') \in r^{\mathcal{M}_{D,\mathcal{C}_2}}$ implies $(r, D') \in \mathcal{F}_3\}$.

Stage 2 is repeated until $\mathcal{N}_i$ contains a quintuple $(\mathcal{Q}_0, \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3, \mathcal{Q}_4)$ such that $\mathcal{Q}_2 \setminus \mathcal{Q}_4 \neq \emptyset$, or $\mathcal{N}_{i+1} = \mathcal{N}_i$. Output '$\mathcal{C}_1 \not\sqsupseteq_\Sigma \mathcal{C}_2$' if the first condition applies; otherwise, output '$\mathcal{C}_1 \sqsupseteq_\Sigma \mathcal{C}_2$'.

Figure 5.3: Algorithm for deciding Σ-entailment in $\mathcal{ELSH}$.

Point (i) follows from the fact that model $\mathcal{M}_{r,E}$ is a generated submodel of $\mathcal{M}'$, i.e. if $(e, e') \in r^{\mathcal{M}'}$ and $e \in \Delta^{\mathcal{M}_{r,E}}$, then $(e, e') \in r^{\mathcal{M}_{r,E}}$.

For point (ii) the only interesting case is that with $D_0 = \exists r.D_0'$.
"$\Rightarrow$": Let $d \in (\exists r.D_0')^{\mathcal{M}}$. Then there is a $d' \in D_0'^{\mathcal{M}}$ such that $(d, d') \in r^{\mathcal{M}}$. By definition of $\mathcal{M}'$, we have $(d, d') \in r^{\mathcal{M}'}$. By the induction hypothesis, $d' \in D_0'^{\mathcal{M}'}$. Hence, $d \in (\exists r.D_0')^{\mathcal{M}'}$.
"$\Leftarrow$": Let $d \in (\exists r.D_0')^{\mathcal{M}'}$. Then there is a $d' \in D_0'^{\mathcal{M}'}$ such that $(d, d') \in r^{\mathcal{M}'}$. Distinguish two cases:

(a) First, $d' \in \Delta^{\mathcal{M}}$. Then $(d, d') \in r^{\mathcal{M}}$ by definition of $\mathcal{M}'$. By the induction hypothesis, $d' \in D_0'^{\mathcal{M}}$ and thus $d \in (\exists r.D_0')^{\mathcal{M}}$.

(b) Second, $d' \in \Delta^{\mathcal{M}'} \setminus \Delta^{\mathcal{M}}$. Then $d = d_0$. By construction of $\mathcal{M}'$, we have $d' \in \Delta^{\mathcal{M}_{s,E}}$ for some $(s, E) \in Q$. It follows by Point (i) above that $d' \in D_0'^{\mathcal{M}'}$ implies $d' \in D_0'^{\mathcal{M}_{s,E}}$. Since $\mathcal{M}_{s,E}$ is a copy of the canonical model $\mathcal{M}_{E,\mathcal{C}}$, we have $d' = E'$ for some $E' \in \Delta^{\mathcal{M}_{E,\mathcal{C}}}$ and $E' \in D_0'^{\mathcal{M}_{E,\mathcal{C}}}$. By Point 3 of Lemma 5.3.5, we get $\mathcal{C} \models E' \sqsubseteq D_0'$.

Now we distinguish two subcases:

Case 1) We have that $E' = d_{s,E}$, and $\mathcal{C} \models s \sqsubseteq r$. Then $E' = E$. By $\mathcal{C} \models E \sqsubseteq D_0'$, we have $D_0' \in K_{\mathcal{C}}(E)$. Since $d_0 \in D^{\mathcal{M}}$, we have $d_0 \in (\exists s. \bigsqcap_{G \in K_{\mathcal{C}}(E)} G)^{\mathcal{M}}$. Hence $d_0 \in (\exists s.D_0')^{\mathcal{M}}$. By $\mathcal{C} \models s \sqsubseteq r$ we obtain $d_0 \in (\exists r.D_0')^{\mathcal{M}}$, as required.

Case 2) $(d_{s,E}, d') \in r'$ and $\mathcal{R}' \models s \circ r' \sqsubseteq r$. Then $\mathcal{R} \models s \sqsubseteq r$, $r' \sqsubseteq r$, $r \circ r \sqsubseteq r$ and $\mathcal{C} \models E \sqsubseteq \exists r'.D_0'$. So $\mathcal{C} \models E \sqsubseteq \exists r.D_0'$. Hence $\exists r.D_0' \in K_{\mathcal{C}}(E)$. Hence $d_0 \in (\exists s.\exists r.D_0')^{\mathcal{M}}$. So $d_0 \in (\exists r.D_0')^{\mathcal{M}}$, as required.

Since $\mathcal{M}$ and all $\mathcal{M}_{r,E}$ are models of $\mathcal{C}$ and by Points (i) and (ii), it follows that $\mathcal{M}'$ is a model of $\mathcal{O}$. We only have to show that $\mathcal{M}' \models \mathcal{R}$. We consider two cases:

Case 1) We have that $(d, d_{s,E}) \in s^{\mathcal{M}'}$ and $\mathcal{R} \models s \sqsubseteq r$. Then by definition we receive that $(d, d_{s,E}) \in r^{\mathcal{M}'}$.

Case 2) We have that $(d, d_{s,E}) \in s^{\mathcal{M}'}$, $(d_{s,E}, d') \in r'^{\mathcal{M}_{s,E}}$ and $\mathcal{R} \models r \circ r \sqsubseteq r$ as well as $\mathcal{R} \models s \sqsubseteq r$ and $\mathcal{R} \models r' \sqsubseteq r$. By definition we receive $(d, d') \in r^{\mathcal{M}'}$.

By Point (ii), $d_0 \notin H^{\mathcal{M}'}$. $d_0 \in C^{\mathcal{M}}$ is trivial. This proves $H \notin K_{\mathcal{C}}(C)$. □

Lemma 5.5.4. *Let $(\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4)$ be the quintuple computed from $\mathcal{F}_0$ and Q in Figure 5.3. For each $(r, q) \in Q$, let $C_{r,q}$ be the concept which determines the quintuple q. Then $C = \bigsqcap_{A \in \mathcal{F}_0} A \sqcap \bigsqcap_{(r,q) \in Q} \exists r.C_{r,q}$ determines $(\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4)$.*

Proof: Let $(\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4)$ and C be as in the lemma, let $(\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4)$ be the quintuple determined by C. We show that $\mathcal{G}_i = \mathcal{F}_i$ for $0 \leq i \leq 4$.

$\mathcal{F}_0 = \mathcal{G}_0$ is trivial, $\mathcal{F}_1 = \mathcal{G}_1$ and $\mathcal{F}_2 = \mathcal{G}_2$ follow from Lemma 5.5.3.

We now have to show that $\mathcal{F}_3 = \mathcal{G}_3$.

"$\supseteq$". Let $(r, D') \in \mathcal{G}_3$, then $(r, D') \in (\Sigma \cap \mathcal{R}_1) \times \mathsf{sub}(\mathcal{O}_2)$ and $C' \Rightarrow_1 D'$, where at least one of the following holds:

1. $\mathcal{C}_1 \models C \sqsubseteq \exists r.C'$, and $C' \in \mathsf{sub}(\mathcal{O}_1)$, or
2. $C \rightsquigarrow^{r}_{\mathcal{R}_1} C'$.

If (1) holds then $\exists r.C' \in K_{\mathcal{C}_1}(C)$, then by Lemma 5.4.4, $(\mathcal{M}_{D',\mathcal{C}_2}, D') \leq_{\Sigma} (\mathcal{M}_{C',\mathcal{C}_1}, C')$. But then by point (a) of definition of $\mathcal{F}_3$ we have $(r, D') \in \mathcal{F}_3$.

If (2) holds then, by Lemma 5.5.2, we distinguish two further cases:

Case 1) There exists a sequence $\exists r_0.C_0, \ldots, \exists r_n.C_n$ with $n \geq 1$, where $\exists r_0.C_0$ is a conjunct of C, $\exists r_{i+1}.C_{i+1}$ a conjunct of C_i for $0 \leq i < n$ such that $C_n = C'$ and for every r_j with $0 \leq j \leq n$ we have $\mathcal{R}_1 \models r_j \sqsubseteq r$ and $\mathcal{R}_1 \models r \circ r \sqsubseteq r$. Then $\exists r_0.C_0 = \exists r'.C_{r',q}$ for some $(r', q) \in \mathcal{Q}$, for q the quintuple representing $C_{r',q}$. As now we have $C_{r',q} \rightsquigarrow^{r}_{\mathcal{R}_1} C'$ and $C' \Rightarrow_1 D'$, we have $(r, D') \in q_3$. As $\mathcal{R}_1 \models r' \sqsubseteq r$ (since $r' = r_0$ and $\mathcal{R}_1 \models r_0 \sqsubseteq r$) and $\mathcal{R}_1 \models r \sqsubseteq r$ and $\mathcal{R}_1 \models r \circ r \sqsubseteq r$, and so by (b)(ii), we have $r, D' \in \mathcal{F}_3$.

Case 2) There exists $\exists r'.C'$ a conjunct of C and $\mathcal{R} \models r' \sqsubseteq r$. Then $\exists r'.C' = \exists r'.C_{r',q}$ for some $(r', q) \in \mathcal{Q}$. Then $C_{r',q} \Rightarrow_1 D'$ and so $D' \in q_4$ for q the quintuple representing $C_{r',q}$. Since $\mathcal{R}_1 \models r' \sqsubseteq r$ we obtain $(r, D') \in \mathcal{F}_3$ by (b)(i).

"$\subseteq$". Let $(r, D') \in \mathcal{F}_3$. We distinguish two cases:

Case 1) (a) holds, thus there is a $\exists r.C' \in \mathcal{F}_1$ such that $(\mathcal{M}_{D',C_2}, D') \leq_\Sigma (\mathcal{M}_{C',C_1}, C')$. Then we have $C' \Rightarrow_1 D'$ by Lemma 5.4.4 and $\exists r.C' \in K_{C_1}(C)$ by the definition of $\mathcal{F}_1$. Hence $(r, D') \in \mathcal{G}_3$; or

Case 2) (b) holds, thus there is a $(s, (\mathcal{Q}_0, \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3, \mathcal{Q}_4)) \in \mathcal{Q}$, here we distinguish two sub-cases:

(1) we have that (i) holds, then there is $C_{s,q}$, such that $C_{s,q} \Rightarrow_1 D'$ and $\mathcal{R}_1 \models s \sqsubseteq r$ thus we have $C \rightsquigarrow^{r}_{\mathcal{R}_1} C_{s,q}$, and so $(r, D') \in \mathcal{G}_3$,

(2) we have that (ii) holds, then there is $C_{s,q}$, such that $C_{s,q} \Rightarrow_1 D'$ and $C \rightsquigarrow^{t}_{\mathcal{R}_1} C_{s,q}$ we also have that $\mathcal{R}_1 \models s \sqsubseteq r$, $\mathcal{R}_1 \models t \sqsubseteq r$ and $\mathcal{R}_1 \models r \circ r \sqsubseteq r$. Thus we have $C \rightsquigarrow^{r}_{\mathcal{R}_1} C_{s,q}$, and so $(r, D') \in \mathcal{G}_3$.

Now we show $\mathcal{F}_4 = \mathcal{G}_4$.

"$\supseteq$". Let $D \in \mathcal{G}_4$ then $D \in \mathsf{sub}(\mathcal{O}_2)$ and $C \Rightarrow_1 D$. By Lemma 5.4.4, $C \Rightarrow_1 D$ iff $(\mathcal{M}_{D,C_2}, D) \leq_\Sigma (\mathcal{M}_{C,C_1}, C)$. From this together with Definition 5.3.1 it follows that $C \Rightarrow_1 D$ iff both of the following hold:

1. for all concept names $A \in \Sigma$, $A \in K_{C_2}(D)$ implies $A \in K_{C_1}(C)$, i.e. $A \in \mathcal{F}_1$;
2. for all role names $r \in \Sigma$ and all concepts D' with $(D, D') \in r^{\mathcal{M}_{D,C_2}}$, there exists a concept C' with $(C, C') \in r^{\mathcal{M}_{C,C_1}}$ and $(\mathcal{M}_{D,C_2}, D') \leq_\Sigma (\mathcal{M}_{C,C_1}, C')$.

"$\subseteq$". Let $D \in \mathcal{F}_4$. Then:

1. for all $A \in \Sigma$, $A \in K_{C_2}(D)$ implies $A \in \mathcal{F}_1$, i.e. $A \in K_{C_1}(C)$; and
2. for all role names $r \in \Sigma$ and all concepts D' with $(D, D') \in r^{\mathcal{M}_{D,C_2}}$ implies $(r, D') \in \mathcal{F}_3$. But as shown above, this implies that $(r, D') \in \mathcal{G}_3$, i.e. for all role names $r \in \Sigma$ and all concepts D' with $(D, D') \in r^{\mathcal{M}_{D,C_2}}$, there exists a concept C' with $(C, C') \in r^{\mathcal{M}_{C,C_1}}$ and $(\mathcal{M}_{D,C_2}, D') \leq_\Sigma (\mathcal{M}_{C,C_1}, C')$.

By Definition 5.3.1 points (1) and (2) give us that $(\mathcal{M}_{D,C_2}, D) \leq_\Sigma (\mathcal{M}_{C,C_1}, C)$. By Lemma 5.4.4, $(\mathcal{M}_{D,C_2}, D) \leq_\Sigma (\mathcal{M}_{C,C_1}, C)$ iff $C \Rightarrow_1 D$. Thus $D \in \mathcal{G}_4$ as required. □

Theorem 5.5.5 (Correctness and Complexity). *The algorithm for deciding Σ-entailment for $\mathcal{ELSH}$ is sound, complete, and runs in exponential time.*

Proof: Soundness follows from Lemmas 5.4.3 and 5.5.4. For completeness, assume $\mathcal{R}_1 \sqsubseteq_\Sigma \mathcal{R}_2$ and $\mathsf{Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2) \neq \emptyset$. By Lemma 5.4.3 there exists C, D, with C of outdegree bounded by $\|\mathcal{C}_2\|$ and $D \in \mathsf{sub}(\mathcal{O}_2)$ such that $\mathcal{C}_2 \models C \sqsubseteq D$ and $C \not\Rightarrow_1 D$. If C is a conjunction of concept names, then the algorithm outputs '$\mathcal{C}_1 \not\sqsubseteq_\Sigma \mathcal{C}_2$' in Step 2. Suppose C has role depth $n \geq 1$. One can show by induction on i using Lemma 5.5.4 that, for all $i \geq 0$, the set $\mathcal{N}_i$ contains all quintuples determined by subconcepts C' of C of role depth smaller than i. Hence, after computing the set $\mathcal{N}_i$ for some $i \leq n$, the algorithm outputs '$\mathcal{C}_1 \not\sqsubseteq_\Sigma \mathcal{C}_2$'.

For termination and time complexity consider the following. To see that Steps 1 and 2 of the algorithm run in polynomial time notice that, by Lemma 5.4.4, the algorithm can compute any quintuple determined by a conjunction of concept names from Σ in polynomial time. Consider Step 3. For each quintuple $(\mathcal{Q}_0, \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3, \mathcal{Q}_4)$, we have $\mathcal{Q}_0 \subseteq \Sigma$ and $\mathcal{Q}_i \subseteq \mathsf{cl}(\mathcal{C}_2)$, for $1 \leq i \leq 4$. That is, the total number of possible quintuples is bounded by $2^{5\|\mathcal{C}_2\|}$. Consequently, the algorithm terminates since $\mathcal{N}_j = \mathcal{N}_{j+1}$, for some $j \leq 2^{5\|\mathcal{C}_2\|}$. For showing that the algorithm runs in exponential time, we now show that $\mathcal{N}_{i+1}$ can be computed from $\mathcal{N}_i$ in exponential time. The number of pairs $(\mathcal{F}_0, \mathcal{Q})$ in Figure 5.3, where $\mathcal{F}_0 \subseteq \Sigma \cap P$ and $Q \subseteq (\Sigma \cap R) \times \mathcal{N}_i$ with $\|Q\| \leq \|\mathcal{C}_2\|$, is exponential in $\|\mathcal{C}_2\|$. Moreover, given a pair $(\mathcal{F}_0, \mathcal{Q})$, computing the quintuple $(\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4)$ in Figure 5.3 only takes polynomial time in $\|\mathcal{C}_2\|$. □

Chapter 6

Conclusion

In this thesis we propose a solution to problems related to the use of multiple ontologies. We focused our attention on these problems due to the fact that in the recent years we can observe an increasing interest in using ontologies in different branches of science and commerce, combined with the increasing number of formal languages used for ontology formulation. On the other hand, we can also observe the need to use ontologies in new and evolving applications, and this requires ontologies to evolve. As a result, users are often interested in using multiple ontologies, but the variety of formal languages used for ontology formulation is a potential source of problems. In our work we approached that problem with the aim to present a construction allowing for answering queries with use of ontologies even if they are formulated in distinct formalisms, this construct was also intended to allow for comparing and combining ontologies formulated in arbitrary formal languages.

As the base for our work we proposed a new view on ontologies, called the functional approach. As opposed to the standard approach to ontologies, in the functional approach the focus is not on the way ontologies are built or what formalisms are used to construct them, but on their function. In other words, we adopt an abstract view of an ontology as a black box providing answers to queries about some vocabulary of interest.

The next step towards providing a construct allowing for working with ontologies in a logic independent way required the use of institutions [41]. The use of institutions gives us an abstract view of logical systems and allows to formulate the consequence relation in a way that does not depend on a particular formal language. This fits very well with the functional approach to ontologies and with the aim to work with multiple ontologies even if presented in distinct formalisms. In addition the theory of institutions offers us truth-preserving translations from one logical system to another, these are institution morphisms and comorphisms. Finally, the theory of institutions allowed us to formulate our results in an institution independent way. For reasons of convenience (presented in Chapter 3) we choose to use comorphisms in our constructions. After introducing the notion of institution we showed how logical systems can be represented as institutions, we presented examples for PL, FOL, $\mathcal{EL}$, $\mathcal{EL}^+$, $\mathcal{ALC}$ and $\mathcal{CH}$, and investigated the relations between them (morphisms and comorphisms). In fact, we show that the institutions of interest are inclusive, as this property addresses the problem of different signatures. Using the theory of institutions we presented notions of query basis and framework which provides a language in which both the ontology and the query languages are translated by means of institution comorphisms. Then we proposed an institution independent formulation of the notion of consequence relation in a framework. This notion allows us to use an ontology for answering a query even if they are formulated in different formal languages and use different signatures. We also introduced a notion of binary framework. With the use of binary frameworks we presented an institution independent formulation of Σ-entailment and Σ-inseparability of ontologies. This allows us to compare and to combine arbitrary ontologies despite the fact that they may be formulated in different formal languages and signatures. Among the results of that section we have that the consequence relation in framework $1_{\mathcal{I}} : \mathcal{I} \to \mathcal{I}$ over itself as query basis is equivalent to the consequence relation in $\mathcal{I}$. We also show that if an ontology language can be translated directly into the query language, then entailment can be reduced to showing that each sentence in one ontology is a consequence of the other. We show that for framework $\mu : \mathcal{L} \to \mathcal{G}$ over query basis μ itself for any Λ-ontology $\mathcal{O}$ for μ, and any $\varphi \in \mathsf{Sen}^{\mathcal{L}}(\Lambda')$ if $\mathcal{O} \models^{\mathcal{L}}_{\Lambda \cup \Lambda'} \varphi$ then $\mathcal{O} \models^{\mu}_{\Lambda'} \varphi$, moreover, if β^{μ} is surjective on models, then the converse implication also holds. We also show that moving to a richer language preserves the consequence relation. These results show that frameworks behave in the expected way and do not affect the consequence relation.

In our work we also investigated three types of robustness for binary frameworks, namely robustness under vocabulary extension, robustness under joins and robustness under replacement in a framework, and investigated how these types of robustness are related to the Craig interpolation property and the notion of conservative extension. To a great extent this is a generalization of results by Konev et al. presented in [50]. As one of the results of that section we show a close relation between robustness under joins and the conservative extension property. Namely we show that for a binary frameworks (μ, μ_1), (μ, μ_2) over η, both robust under joins, given a Λ-ontology $\mathcal{O}$ for μ, Λ_1-ontology $\mathcal{O}_1$ for μ_1, and Λ_2-ontology $\mathcal{O}_2$ for μ_2, with signatures satisfying $\Phi^{\mu}(\Lambda) \cap \Phi^{\mu_1}(\Lambda_1) \subseteq \Phi^{\eta}(\Sigma)$, and $\Phi^{\mu}(\Lambda) \cap \Phi^{\mu_2}(\Lambda_2) \subseteq \Phi^{\eta}(\Sigma)$, we have that if $\mathcal{O} \cup \mathcal{O}_i$ is a conservative extension of $\mathcal{O}$ (for $i = 1, 2$), then also $\mathcal{O} \cup \mathcal{O}_1 \cup \mathcal{O}_2$ is a conservative extension of $\mathcal{O}$. This property is important for ontology refinement. Another result shows the relation between weak interpolation and robustness under vocabulary extension. More precisely, the result states that in a binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis $\eta : \mathcal{Q} \to \mathcal{G}$, where η is conservative and for every signature Σ in $\mathsf{Sig}^{\mathcal{Q}}$, β^{η}_{Σ} is surjective and there is a comorphism $\rho : \mathcal{L}_2 \to \mathcal{Q}$ such that $\mu_2 = \rho; \eta$: if $\mathcal{G}$ has weak interpolation, then $\mathfrak{F}$ is robust under vocabulary extension. Another result of that chapter tells us that if framework $1_{\mathcal{I}} : \mathcal{I} \to \mathcal{I}$ over query basis $1_{\mathcal{I}}$ is robust under vocabulary extension, then $\mathcal{I}$ has weak interpolation. We also show for a binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis η, with η conservative, and surjective β^{η} on models, with comorphisms $\rho_i : \mathcal{L}_i \to \mathcal{Q}$ such that $\mu_i = \rho_i; \eta$ for $i = 1, 2$: if $\mathcal{Q}$ is closed under Boolean operators, and $\mathcal{G}$ has weak interpolation, then $\mathfrak{F}$ is robust under joins for finite ontologies. We also show that any framework over query basis $1_{FOL} : FOL \to FOL$ is robust under vocabulary extensions, joins, and under replacement for finite ontologies. We also show that for $\mathcal{L}$ an institution of any fragment of first-order logic closed under Boolean operators, robustness under joins for any framework $\mu : \mathcal{L} \to \mathcal{G}$ over itself as query basis implies interpolation in $\mathcal{L}$.

We also investigated the problem of using ontologies together with ABoxes and determining Σ-entailment and Σ-inseparability of ontologies in the presence of ABoxes. But to make it possible first we had to introduce individuals into the signatures and present how assertions are built. To do that first we provided a definition of description logic, which is based on the notion of slice category. Namely, a description logic is an institution $\mathcal{I}$ together with a morphism $\mu : \mathcal{I} \to \overline{\mathcal{CH}}$. This definition allows us to treat description logics in a systematic way and moreover it is used to show how to introduce individuals into the signature and how to construct assertions with individuals. Then we presented how an institution of a description logic extends to an institution of a description logic with individuals, and similarly how morphisms and comorphisms between those description logics extend to morphisms and comorphisms between description logics with individuals. We also investigated the relations between institutions of description logics and their counterparts with individuals. Among the results we include that given an institution $\mathcal{I}$ and $\mathcal{I}{+}\iota$ there is a morphism $\mu^- : \mathcal{I}{+}\iota \to \mathcal{I}$ and another one $\mu^+ : \mathcal{I} \to \mathcal{I}{+}\iota$. This is graphically represented in Figure 6.1. We also show that for signatures $\Sigma \in \mathsf{Sig}^{\mathcal{I}}$ and $(\Sigma, I) \in \mathsf{Sig}^{\mathcal{I}+\iota}$ the

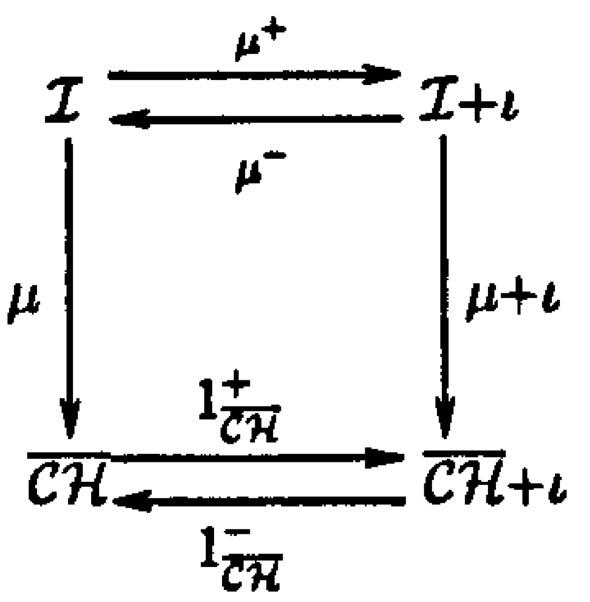


Figure 6.1.

functor $\Psi^{\mu^-} : (\Sigma, I) \mapsto \Sigma$ has left adjoint $\Psi^{\mu^+} : \Sigma \mapsto (\Sigma, \emptyset)$. This, together with the results of Arrais and Fiadeiro presented in [5], gives rise to comorphism $\vec{\mu} : \mathcal{I} \to \mathcal{I}{+}\iota$. We also introduced the notions of query conservativity, query expansion and concept interpolation. We could say that concept interpolation 'splits' the consequence relation for assertions into two different logics. This in turn means that determining if assertion $\varphi(i)$ is a consequence of $(\mathcal{O}, \mathcal{A})$ reduces to determining if there is ψ such that $\psi(i)$ is a consequence of $\mathcal{A}$ and $\psi \sqsubseteq \varphi$ is a consequence of $\mathcal{O}$. Extending results of [62] showing that $\mathcal{EL}{+}\iota$ has concept interpolation we show that $\mathsf{e}_3^+\overline{\mathsf{c}}{+}\iota$ has concept interpolation. Which in that case means that two conditions are satisfied:

- if φ is a concept name C and we have $(\mathcal{O}, \mathcal{R}, \mathcal{A}) \models^{\mathsf{e}_3^+\overline{\mathsf{c}}}_{(\Sigma,I)} C(i)$, then we can find a concept C', such that $\mathcal{A} \models^{\mathsf{e}_3^+\overline{\mathsf{c}}}_{(\Sigma,I)} C'(i)$ and $\mathcal{O}^{\mathcal{R}} \models^{\mathcal{EL}_3^+}_{\Sigma} C' \sqsubseteq C$,
- if φ is of the form $r_1 \circ \cdots \circ r_n$, and we have that pair $(\mathcal{R}, \mathcal{A})$ entails role assertion $r_1 \circ \cdots \circ r_n(i_1, i_2)$, then there is a sequence of role assertions $r_{1,1} \circ \cdots \circ r_{1,m}(i_1, i'), r_{2,1} \circ \cdots \circ r_{2,l}(i', i''), \ldots, r_{k,1} \circ \cdots \circ r_{k,j}(i, i_2)$ in $\mathcal{A}$, such that for $\psi : r_{1,1} \circ \cdots \circ r_{1,m} \circ \cdots \circ r_{k,1} \circ \cdots \circ r_{k,j}$, we have $\mathcal{A} \models \psi(i_1, i_2)$ and $\mathcal{R} \models \psi \sqsubseteq r_1 \circ \cdots \circ r_n$.

We also show that if μ has query expansion, then it is query conservative. This material allowed us to show how a framework built from description logics extends to a framework allowing for use of individuals. Having this, we were able to formulate the notion of Σ-entailment and Σ-inseparability based on instance checking rather than based on concept subsumption, which is usually too weak when using ontologies together with ABoxes. In our work we considered the case when an ontology and an ABox are formulated in the same formal language.

In the final part of our work we presented a particular application of frameworks; we investigated Σ-entailment in a framework $1_{\mathcal{ELSH}} : \mathcal{ELSH} \to \mathcal{ELSH}$ over itself as a query basis. This part of our research extends the result presented by Lutz and Wolter in [61] by considering ontologies formulated as general CBoxes (TBoxes together with RBoxes) in the description logic $\mathcal{ELSH}$ obeying some additional restrictions. The main result states that the Σ-entailment problem for such ontologies can be solved in ExpTime. Thus, this problem is no more complex than for plain $\mathcal{EL}$, which was shown to be ExpTime-complete [61].

Future work

We realize that our research does not cover the area completely, therefore below we suggest some of the possible directions for future work.

In our work we show that for general CBoxes formulated in the description logic $\mathcal{ELSH}$ the Σ-entailment problem can be solved in ExpTime and is no more complex than for plain $\mathcal{EL}$, which was shown to be ExpTime-complete. It is also known that the computational complexity of this problem is 2ExpTime-complete for more expressive description logics like $\mathcal{ALC}$, $\mathcal{ALCQ}$, and $\mathcal{ALCQI}$, but even in such simple formalisms as acyclic propositional Horn Logic it is co-NP-complete. On the other hand, there are many logical systems for which the Σ-entailment problem has not been investigated. For instance, it was already mentioned in Chapter 5, the problem is still open for description logic $\mathcal{EL}^{+}$. Also investigating the problem for other extensions of $\mathcal{EL}$ could be interesting.

Another possible area of future research is to investigate further the problem of deciding Σ-entailment for knowledge bases in the framework setting. As mentioned in Section 4.3, when using ontologies together with ABoxes one has to determine how to include the ABox into the framework. In our work we have investigated the case when the ABox is a part of the ontology, i.e., the ontology and the ABox are closely related, for instance, if they are designed and maintained together, and the ABox does not change significantly more often than the ontology. But the ABox can be a part of the query language or even be formulated in yet another language, in such a case the ontology and the ABox have a different status, which usually is the case (the usual situation is when an ontology designer does not know the ABox and that the Abox changes more often than the ontology). Of course in that scenario the notion of Σ-entailment for knowledge bases is not sufficient and we need a definition of Σ-entailment quantifying over all possible ABoxes, i.e. taking ABoxes as unknown "black boxes". To determine inseparability of ontologies with all possible ABoxes we will have to additionally assume that we can formulate the deduction theorem in $\mathcal{G}+\iota$. But this will involve additional definitions. One way of assuring that we can formulate the deduction theorem in $\mathcal{G}+\iota$ is to make Boolean operators available, so we can say what the conjunct $\bigwedge \mathcal{A}$ is and what "$\rightarrow$" means in $\mathcal{Q}+\iota$.

Appendix A

A.1 Σ-entailment and inseparability in morphism frameworks

In section 3.2 we have introduced frameworks, which were used for studying the fundamental notions of description logics, Σ-entailment, Σ-inseparability w.r.t. a query language and Σ-conservative extension. For framework construction in section 3.2 we have used comorphisms, while arguing for use of comorphisms we have mentioned that it is possible to introduce similar construct with use of morphisms. In this section we provide a formulation of a framework with the use of morphism, we also point what are the difficulties of that formulation.

Definition A.1.1. *A **morphism query basis** is an inclusive morphism $\eta : \mathcal{G} \to \mathcal{Q}$. A **morphism framework over a morphism query basis** $\eta : \mathcal{G} \to \mathcal{Q}$ is an inclusive morphism $\mu : \mathcal{G} \to \mathcal{L}$.*

Convention A.1.2. *As in this section we talk about morphism frameworks and morphism query bases only, we will refer to them simply as frameworks and query bases, which should not cause any confusion.*

The intuition behind this construct is that given an ontology and a query represented in two institutions, $\mathcal{L}$ and $\mathcal{Q}$ respectively, we chose a global institution $\mathcal{G}$ such that there are morphisms $\mathcal{G} \xrightarrow{\mu} \mathcal{L}$ and $\mathcal{G} \xrightarrow{\eta} \mathcal{Q}$. Using these morphisms we can translate the ontology and the query into $\mathcal{G}$, then we can check whether the query is a consequence of the ontology. In more detail, institution $\mathcal{G}$ provides us with a signature, this signature is translated down to $\mathcal{L}$ and $\mathcal{Q}$. This translated signature is then used for constructing an ontology and a query respectively. Then both, the ontology and the query are translated back to $\mathcal{G}$ where we are able to check if the ontology entails the query.

As previously, we allow more than one framework over a query basis. Figure A.1 is a graphical representation of frameworks μ_1 and μ_2 over query basis η.

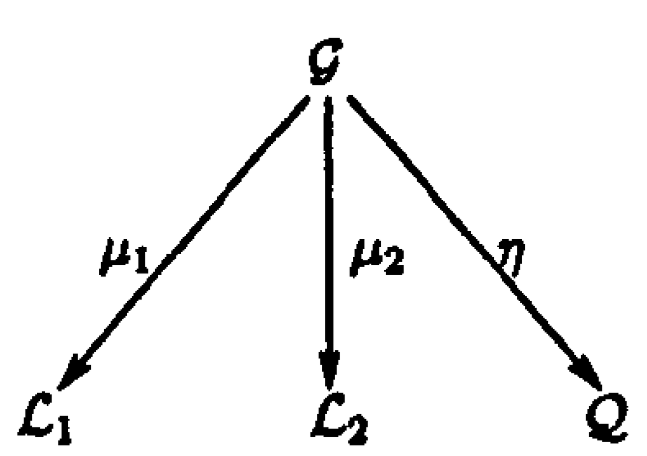


Figure A.1: Frameworks μ_1 and μ_2 over a query basis η

As for framework using comorphisms we define the consequence relation.

Definition A.1.3. *Let $\mu : \mathcal{L} \to \mathcal{G}$ be a framework over η, and let Σ be a $\mathcal{G}$-signature, $\mathcal{O}$ be a $\Psi^{\mu}(\Sigma)$-ontology for μ and $\varphi \in Sen^{\mathcal{Q}}(\Psi^{\eta}(\Sigma))$ be a query. We say that φ is a **consequence of $\mathcal{O}$ with respect to η** (written $\mathcal{O} \models^{\eta}_{\Sigma} \varphi$) iff*

$$\gamma^{\mu}_{\Sigma}(\mathcal{O}) \models^{\mathcal{G}}_{\Sigma} \gamma^{\eta}_{\Sigma}(\varphi).$$

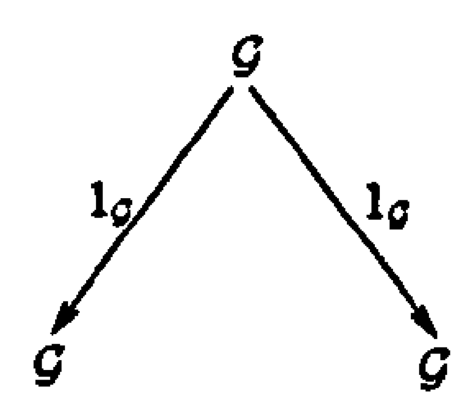
Just like in Definition 3.2.5 this is to say that φ is a consequence of $\mathcal{O}$ in framework μ iff $\mathcal{O}$ translated along morphism μ into $\mathcal{G}$ entails in $\mathcal{G}$, with respect to the signature Σ, the translation of φ along η.

Compared to Definition 3.2.5 we can see that Definition A.1.3 is less intuitive. Namely it suggests that the signature was introduced in the global language and then translated to ontology and query languages. In that case one could argue that homonyms and synonyms should have been avoided already in the global language, moreover it would be not clear why different natural languages should be used within one signature. Alternatively, if in the original signature homonyms and synonyms were not introduced and the signature was over one natural language only, but ontologies or queries would still use homonyms, synonyms or different natural languages it would mean that morphisms are responsible for that. In that case one could argue that the morphisms should be chosen in more careful way, and avoiding introducing any ambiguities in the signatures.

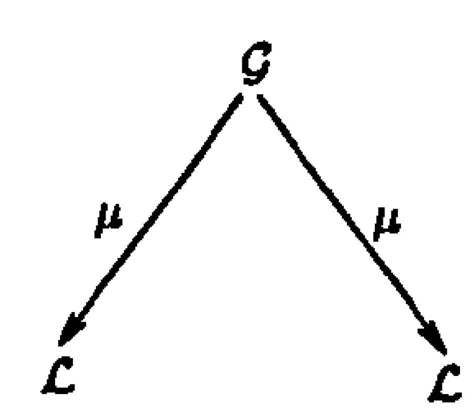
Basic framework structures

As for the case with comorphisms, we discuss six special cases of morphism frameworks. Here we do not repeat the examples introduced in Chapter 3.

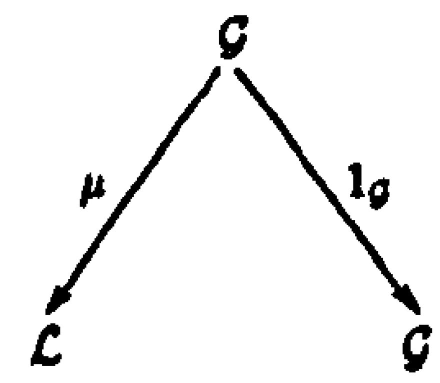
1. Let $\mathcal{L} = \mathcal{G} = \mathcal{Q}$, in this case morphisms are identities. Proposition A.1.4 below shows that the entailment in framework $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$ over the same query basis is the same as entailment in $\mathcal{G}$.



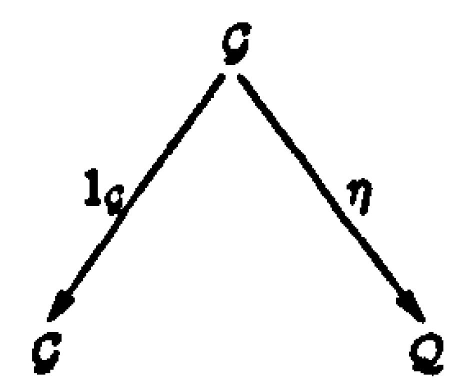
2. Let $\mathcal{L} = \mathcal{Q}$, i.e. an ontology $\mathcal{O}$ and a query φ are expressed in the same language. Proposition A.1.5 states that for this framework there may be entailments that arise from the greater power of $\mathcal{G}$, and that the ontology $\mathcal{O}$ in this framework entails exactly the same consequences as in $\mathcal{L}$ only if β^{μ} is surjective.



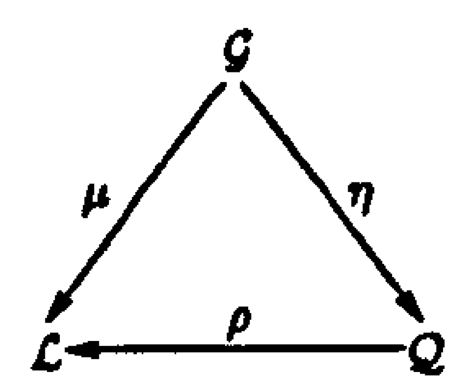
3. For the case where $\mathcal{G} = \mathcal{Q}$ with a framework $\mu : \mathcal{G} \to \mathcal{L}$ over a query base $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$, we translate an $\mathcal{L}$-ontology into a richer language $\mathcal{G}$, and then, in $\mathcal{G}$, we check whether a query is a consequence of the ontology. Proposition A.1.6, states that entailment in this framework is the same as entailment in $\mathcal{G}$.



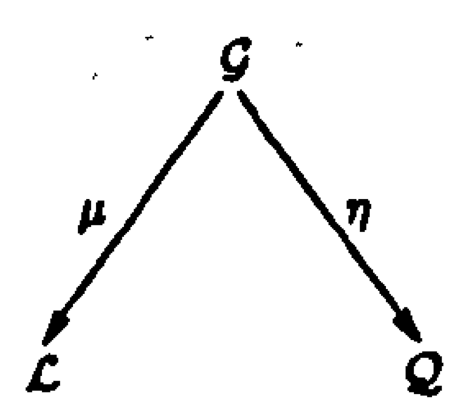
4. For the case where $\mathcal{G} = \mathcal{L}$ and a framework $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$ over a query base $\eta : \mathcal{G} \to \mathcal{Q}$, we translate a $\mathcal{Q}$-query into $\mathcal{G}$, and then, in $\mathcal{G}$, we check whether the query is a consequence of the ontology. Proposition A.1.7 shows that entailment in this framework is the same as entailment in $\mathcal{G}$.



5. Consider a scenario with distinct institutions $\mathcal{L}$, $\mathcal{G}$ and $\mathcal{Q}$, together with a framework $\mu : \mathcal{G} \to \mathcal{L}$ over a query basis $\eta : \mathcal{G} \to \mathcal{Q}$ and a morphism $\rho : \mathcal{L} \to \mathcal{Q}$. We can see μ as a composition $\rho; \eta$. So, in fact we are translating an $\mathcal{L}$-ontology into $\mathcal{G}$ via $\mathcal{Q}$. Properties of this framework are presented in Corollary A.1.18 and Proposition A.1.19.



6. Let $\mu : \mathcal{G} \to \mathcal{L}$ be a framework over a query basis $\eta : \mathcal{G} \to \mathcal{Q}$ with incomparable $\mathcal{L}$ and $\mathcal{Q}$. In this case we have to translate an ontology $\mathcal{O}$ and a query φ into a language in which we can check whether φ is a consequence of $\mathcal{O}$, i.e. we have to find a $\mathcal{G}$ such that there are comorphisms from $\mathcal{L}$ and $\mathcal{Q}$ to $\mathcal{G}$.



Now we present counterparts of properties presented in Chapter 3, this time we preset them using morphisms instead of comorphisms. First we present a proposition which is a direct consequence of how the frameworks are constructed and applies to the case 1 of framework. This property is a counterpart of Proposition 3.2.11 presented in Section 3.2

Proposition A.1.4. *For framework $1_\mathcal{G} : \mathcal{G} \to \mathcal{G}$ over query basis $1_\mathcal{G} : \mathcal{G} \to \mathcal{G}$, consequence in the framework is just consequence in $\mathcal{G}$; i.e., $\mathcal{O} \models^{1_\mathcal{G}}_\Sigma \varphi$ iff $\mathcal{O} \models^{\mathcal{G}}_\Sigma \varphi$ for any Σ-ontology $\mathcal{O}$ and Σ-sentence φ.*

Proof: The proof is similar to that of Proposition 3.2.11.

Proposition A.1.5. *For any framework $\mu : \mathcal{G} \to \mathcal{L}$ over query basis μ itself, for any $\Psi^\mu(\Sigma)$-ontology $\mathcal{O}$ for μ, and any $\varphi \in Sen^{\mathcal{L}}(\Psi^\mu(\Sigma))$ we have:*

$$\mathcal{O} \models^{\mathcal{L}}_\Sigma \varphi \quad \textit{implies} \quad \mathcal{O} \models^{\mu}_\Sigma \varphi\ .$$

Moreover, if β^μ is surjective on models, then the converse implication also holds.

The proof is similar to the proof of Proposition 3.2.12.

Proposition A.1.6. *For any framework $\mu : \mathcal{G} \to \mathcal{L}$ over query basis $1_\mathcal{G} : \mathcal{G} \to \mathcal{G}$, for any $\Psi^\mu(\Sigma)$-ontology $\mathcal{O}$ for $1_\mathcal{G}$, and any $\varphi \in Sen^{\mathcal{G}}(\Sigma)$ we have:*

$$\mathcal{O} \models^{1_\mathcal{G}}_\Sigma \varphi \quad \textit{iff} \quad \gamma^\mu_\Lambda(\mathcal{O}) \models^{\mathcal{G}}_\Sigma \varphi\ .$$

Proof: This follows directly from Definition A.1.3. □

Proposition A.1.7. *For any framework $1_\mathcal{G} : \mathcal{G} \to \mathcal{G}$ over query basis $\eta : \mathcal{G} \to \mathcal{Q}$, for any Σ-ontology $\mathcal{O}$ for η, and any $\varphi \in Sen^{\mathcal{Q}}(\Psi^\eta(\Sigma))$ we have:*

$$\mathcal{O} \models^{\eta}_\Sigma \varphi \quad \textit{iff} \quad \mathcal{O} \models^{\mathcal{G}}_\Sigma \gamma^\eta_\Sigma(\varphi)\ .$$

Proof: This follows directly from Definition A.1.3. □

Notation A.1.8. *In the following definition we use expressions 'Σ-entailment', 'Σ-inseparability' and 'Σ-conservative extension' as abbreviations of '$\Psi^\eta(\Sigma)$-entailment', '$\Psi^\eta(\Sigma)$-inseparability' and '$\Psi^\eta(\Sigma)$-conservative extension' respectively, this abbreviation is set to be a notational convention in the remaining part of the text.*

As in Section 3.2, we define the notion of Σ-entailment and closely related notions of **inseparability** and **conservative extension.**

Definition A.1.9 (Σ-entailment and inseparability). *For query basis η and frameworks μ_1, μ_2 over η, $\Sigma \in Sig^{\mathcal{G}}$, $\Psi^{\mu_1}(\Sigma)$-ontology $\mathcal{O}_1$ for μ_1 and $\Psi^{\mu_2}(\Sigma)$-ontology $\mathcal{O}_2$ for μ_2, we say that*

- *$\mathcal{O}_1$ **Σ-entails** $\mathcal{O}_2$ wrt η, and write $\mathcal{O}_1 \sqsubseteq^\eta_\Sigma \mathcal{O}_2$, iff for all $\varphi \in Sen^{\mathcal{Q}}(\Psi^\eta(\Sigma))$ we have:*

$$\mathcal{O}_2 \models^{\eta}_\Sigma \varphi \quad \textit{implies} \quad \mathcal{O}_1 \models^{\eta}_\Sigma \varphi\ .$$

- *$\mathcal{O}_1$ and $\mathcal{O}_2$ are **Σ-inseparable** wrt η, written $\mathcal{O}_1 \approx^\eta_\Sigma \mathcal{O}_2$, iff:*

$$\mathcal{O}_1 \sqsubseteq^\eta_\Sigma \mathcal{O}_2 \quad \textit{and} \quad \mathcal{O}_2 \sqsubseteq^\eta_\Sigma \mathcal{O}_1 \,.$$

- *$\mathcal{O}_2$ is a **Σ-conservative extension** of $\mathcal{O}_1$ wrt η iff $\gamma^{\mu_2}_\Sigma(\mathcal{O}_2) \supseteq \gamma^{\mu_1}_\Sigma(\mathcal{O}_1)$ and $\mathcal{O}_1$ and $\mathcal{O}_2$ are Σ-inseparable wrt η.*

- *$\mathcal{O}_2$ is a **conservative extension** of $\mathcal{O}_1$ wrt η iff $\mathcal{O}_2$ is a Σ-conservative extension of $\mathcal{O}_1$ wrt η for all $\Sigma \in |Sig^{\mathcal{G}}|$.*

In the situation of the previous Definition, we say that φ separates $\mathcal{O}_1$ and $\mathcal{O}_2$ iff $\mathcal{O}_1 \models^\eta_\Sigma \varphi$ and $\mathcal{O}_2 \not\models^\eta_\Sigma \varphi$ or vice versa.

For any query basis η and a signature $\Psi^\eta(\Sigma) \in Sig^{\mathcal{Q}}$ the relation $\approx^\eta_\Sigma$ is an equivalence relation.

In the situation of the previous definition, we say that φ separates $\mathcal{O}_1$ and $\mathcal{O}_2$ iff $\mathcal{O}_1 \models^{\mathfrak{I}}_\Sigma \varphi$ and $\mathcal{O}_2 \not\models^{\mathfrak{I}}_\Sigma \varphi$, or vice versa.

The following Lemma states that if the square in Figure A.2 has CIP in the global institution, then the framework has a kind of conservativity property: extending the

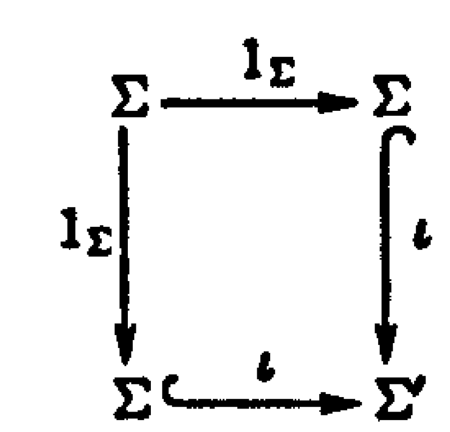


Figure A.2.

$\mathcal{Q}$-signature (i.e. the part of $\mathcal{G}$-signature that is translated to $\mathcal{Q}$ only) with fresh symbols has no impact on the consequence relation in the framework for the queries formulated in the original signature.

This lemma is also an example of problems related to use of morphisms for framework construction.

As already mentioned in Section 3.2, for the case of frameworks and query bases with morphisms, signatures used for ontology and query formulation originate in the global language and are only translated into ontology and query languages respectively. This issue becomes problematic if we want to manipulate only the signature used for the query formulation, but leave the ontology signature untouched (or vice versa). For instance, in the case of comorphism framework $\mu : \mathcal{L} \to \mathcal{G}$ over a query $\eta : \mathcal{Q} \to \mathcal{G}$ given $\mathcal{L}$-signature Λ and $\mathcal{Q}$-signature Σ we can easily express that we have another signature which extends one of them only, for example $\Sigma \hookrightarrow \Sigma'$. It is clear that this inclusion is preserved along functor Φ^μ. But when we want to express the same fact for morphism framework $\kappa : \mathcal{G} \to \mathcal{L}$ over a query basis $\lambda : \mathcal{G} \to \mathcal{Q}$ we encounter some difficulties. Now we have two $\mathcal{G}$-signatures Σ, Σ', such that $\Sigma \subseteq \Sigma'$, but as we want to have $\mathcal{L}$-signatures untouched we have to introduce further constraints, and say that $\Psi^\kappa(\Sigma) = \Psi^\kappa(\Sigma')$ but $\Psi^\lambda(\Sigma) \subseteq \Psi^\lambda(\Sigma')$. Alternatively, we could say that $\mathcal{G}$-signature is a union $\Gamma = (\Lambda \cup \Sigma)$ and add a constraint that Ψ^{κ} forgets about Σ-part of the signature, whereas Ψ^{λ} forgets about the Λ-part (in the approach described before it was implicit). Both we find to be non-intuitive and inconvenient if we have to deal with multiple signatures.

Lemma A.1.10. *For any framework $\mu : \mathcal{G} \to \mathcal{L}$ over query basis $\eta : \mathcal{G} \to \mathcal{Q}$ and signatures Σ, Σ' in $Sig^{\mathcal{G}}$, such that $\Sigma \subseteq \Sigma'$ and $\Psi^{\mu}(\Sigma) = \Psi^{\mu}(\Sigma')$ while $\Psi^{\eta}(\Sigma) \subseteq \Psi^{\eta}(\Sigma')$, any $\Psi^{\mu}(\Sigma)$-ontology $\mathcal{O}$ for μ, and any query $\varphi \in Sen^{\mathcal{Q}}(\Psi^{\eta}(\Sigma))$, the following property holds:*

$$\mathcal{O} \models^{\eta}_{\Sigma} \varphi \quad \text{implies} \quad \mathcal{O} \models^{\eta}_{\Sigma'} \varphi \,.$$

Moreover, if Figure A.2 is a CIP-square, then the converse implication also holds.

The proof is similar to the proof of Lemma 3.2.18.

This lemma says that extending the part of the signature used for query formulation has no impact on the consequence relation in framework for queries formulated in the original signature. This property also extends to entailment:

Proposition A.1.11. *For any binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis η, signatures $\Sigma, \Sigma' \in Sig^{\mathcal{G}}$, such that $\Psi^{\eta}(\Sigma) \subseteq \Psi^{\eta}(\Sigma')$ and $\Psi^{\mu_1}(\Sigma)$-ontology $\mathcal{O}$ for μ_1, $\Psi^{\mu_2}(\Sigma)$-ontology $\mathcal{O}_2$ for μ_2, if Figure 3.2 has CIP, then:*

$$\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma'} \mathcal{O}_2 \quad \text{implies} \quad \mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2 \,.$$

Proof is similar to the proof of Proposition 3.2.19

Note that the opposite direction does not hold because it would imply extending the signature over which queries may be expressed.

Investigating frameworks gives us an insight into properties of entailment also in more complex situations. For instance we can consider frameworks with attached morphisms. Example A.1.12 represents two situations where frameworks have attached morphisms.

So we may have a morphism attached on the query language side, intuitively it is a situation when we already have a framework with $\mathcal{L}$ and $\mathcal{Q}$ and we have a query formulated in a language $\mathcal{Q}'$, such that there is a morphism $\xi : \mathcal{Q} \to \mathcal{Q}'$, i.e. $\mathcal{Q}'$ is a weaker language than $\mathcal{Q}$. We may also have a situation when we already have a framework with $\mathcal{L}$ and $\mathcal{Q}$ and we have an ontology formulated in a language $\mathcal{L}'$, such that there is a morphism $\zeta : \mathcal{L} \to \mathcal{L}'$, i.e. $\mathcal{L}'$ is a weaker language than $\mathcal{L}$. We also present the consequences of these two situations.

Example A.1.12. *First we consider a framework $\mu : \mathcal{G} \to \mathcal{L}$ over a query basis $\eta : \mathcal{G} \to \mathcal{Q}$ with an additional morphism $\xi : \mathcal{Q} \to \mathcal{Q}'$ (see Figure A.3). Using morphisms composition $\eta;\xi = \eta'$ we receive a framework $\mu : \mathcal{G} \to \mathcal{L}$ over query basis $\eta' : \mathcal{G} \to \mathcal{Q}'$ (see Figure A.4).*

Note that $\mathcal{O}$ has the same set of consequences in μ over both η and η'. This is the statement of Lemma A.1.13.

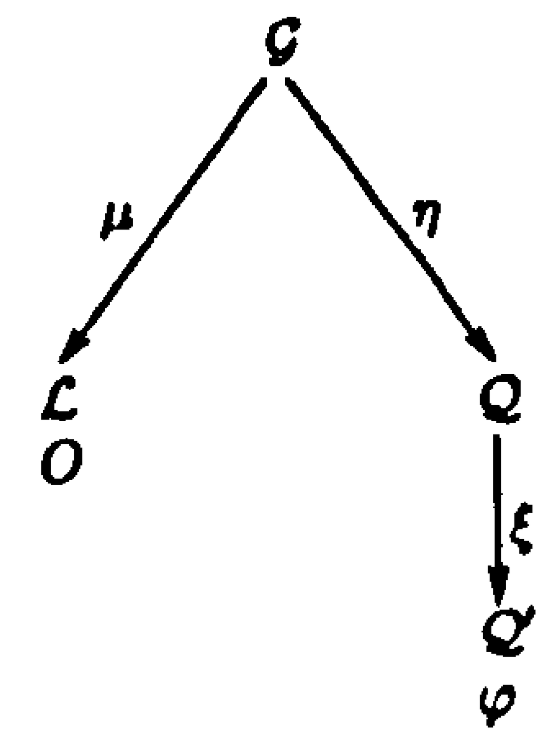


Figure A.3.

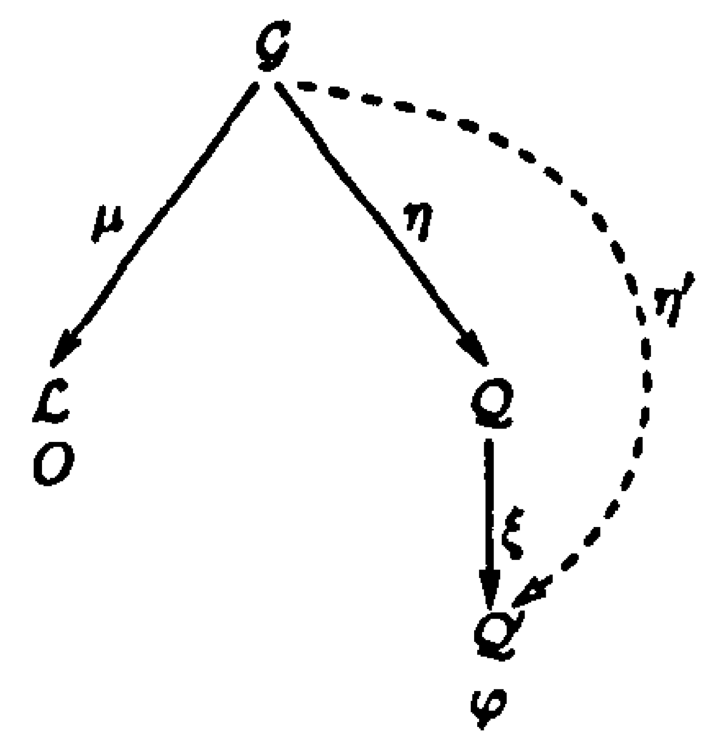


Figure A.4.

Lemma A.1.13. *Given a framework $\mu : \mathcal{G} \to \mathcal{L}$ over a query basis $\eta : \mathcal{G} \to \mathcal{Q}$ with an attached morphism $\xi : \mathcal{Q} \to \mathcal{Q}'$, we get a framework $\mu : \mathcal{G} \to \mathcal{L}$ over query basis $\eta' : \mathcal{G} \to \mathcal{Q}'$, with $\eta' = \eta; \xi$, and an ontology $\mathcal{O} \subseteq Sen^{\mathcal{L}}(\Psi^{\mu}(\Sigma))$, and a query $\varphi \in Sen^{\mathcal{Q}'}(\Psi^{\eta}(\Psi^{\xi}(\Sigma)))$, the following holds:*

$$\mathcal{O} \models^{\eta'}_{\Sigma} \varphi \quad \textit{iff} \quad \mathcal{O} \models^{\eta}_{\Sigma} \gamma^{\xi}_{\Psi^{\eta}(\Sigma)}(\varphi).$$

Proof is similar to the one given for Lemma 3.2.20.

In other words Lemma A.1.13 states that given a framework $\mu : \mathcal{G} \to \mathcal{L}$ over query basis $\eta : \mathcal{G} \to \mathcal{Q}$ and a query formulated in $\mathcal{Q}'$, such that there is a morphism $\mathcal{Q} \to \mathcal{Q}'$ then we can safely lift the query to $\mathcal{Q}$ and then translate it into $\mathcal{G}$. That gives us exactly the same results as creating a framework μ over query basis $\eta' : \mathcal{G} \to \mathcal{Q}'$ (using composition of morphisms) for answering the query.

Next proposition shows the relation between morphisms of query languages and the problem of inseparability of ontologies.

Proposition A.1.14. *For any binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis $\eta : \mathcal{G} \to \mathcal{Q}$ with an attached morphism $\xi : \mathcal{Q} \to \mathcal{Q}'$, and the same binary framework over $\eta' : \mathcal{G} \to \mathcal{Q}'$, where $\eta' = \eta; \xi$, signature $\Sigma \in Sig^{\mathcal{G}}$ and ontologies $\mathcal{O}_1 \subseteq Sen^{\mathcal{L}_1}(\Psi^{\mu_1}(\Sigma))$ and $\mathcal{O}_2 \subseteq Sen^{\mathcal{L}_2}(\Psi^{\mu_2}(\Sigma))$ the following implication holds:*

$$\textit{if} \quad \mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2 \quad \textit{then,} \quad \mathcal{O}_1 \sqsubseteq^{\eta'}_{\Sigma} \mathcal{O}_2.$$

The proof is similar to the proof of Proposition 3.2.22

In other words, Proposition A.1.14 states that if two ontologies are inseparable relative to query $\varphi \in \mathsf{Sen}^{\mathcal{Q}'}(\Psi^{\xi}(\Psi^{\eta}(\Sigma)))$, translated to $\mathcal{G}$ along morphisms ξ and then η via $\mathcal{Q}$, then these ontologies are inseparable relative to φ, translated to $\mathcal{G}$ along morphism η'. This result extends to inseparability.

Corollary A.1.15. *For any binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis $\eta : \mathcal{G} \to \mathcal{Q}$ with an attached morphism $\xi : \mathcal{Q} \to \mathcal{Q}'$, and the same binary framework over $\eta' : \mathcal{G} \to \mathcal{Q}'$, where $\eta' = \eta;\xi$, for signature $\Sigma \in Sig^{\mathcal{G}}$, and ontologies $\mathcal{O}_1 \subseteq Sen^{\mathcal{L}_1}(\Psi^{\mu_1}(\Sigma))$ and $\mathcal{O}_2 \subseteq Sen^{\mathcal{L}_2}(\Psi^{\mu_2}(\Sigma))$, we have that $\mathcal{O}_1 \approx^{\eta}_{\Sigma} \mathcal{O}_2$ implies $\mathcal{O}_1 \approx^{\eta'}_{\Sigma} \mathcal{O}_2$.*

As promised above now we consider a framework with an attached morphism on the ontology language side.

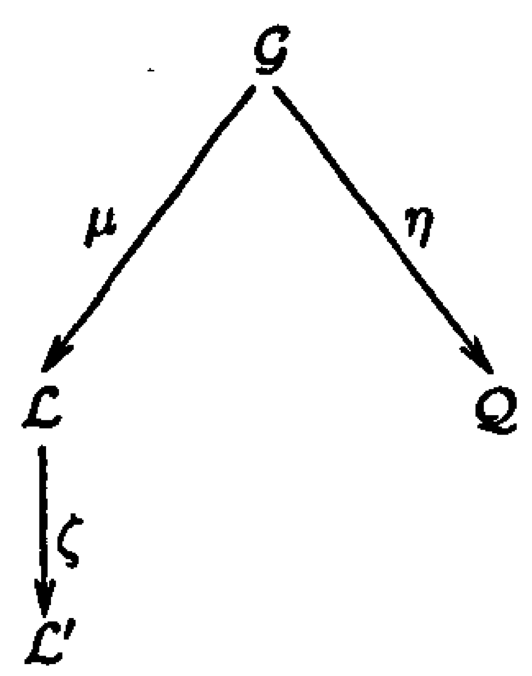


Figure A.5.

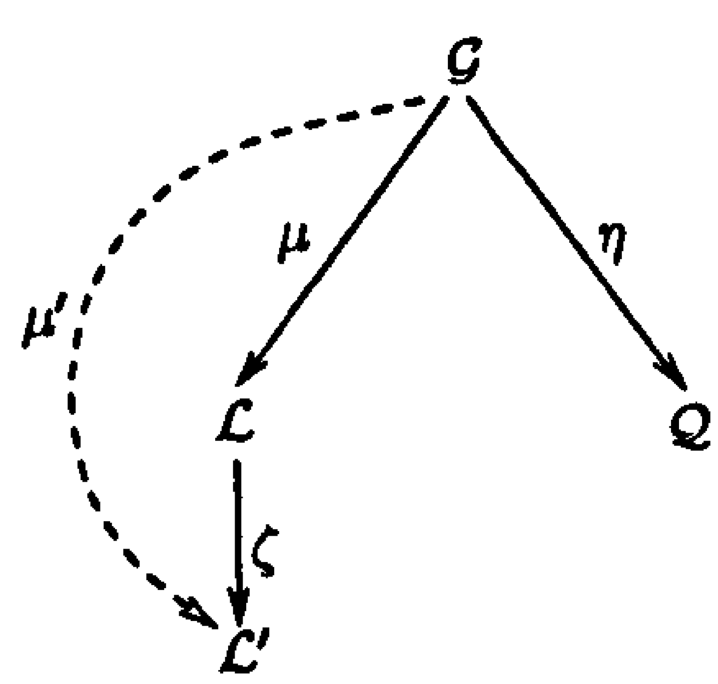


Figure A.6.

Lemma A.1.16. *For a framework $\mu : \mathcal{G} \to \mathcal{L}$ over query basis $\eta : \mathcal{G} \to \mathcal{Q}$ with an attached morphism $\zeta : \mathcal{L} \to \mathcal{L}'$, we can create a framework $\mu' : \mathcal{G} \to \mathcal{L}'$ over query basis η with $\mu' = \mu;\zeta$, such that:*

$$\mathcal{O} \models^{\eta}_{\Sigma} \varphi \quad \textit{iff} \quad \gamma^{\zeta}_{\Psi^{\mu}(\Sigma)}(\mathcal{O}) \models^{\eta}_{\Sigma} \varphi$$

for any $\Psi^{\zeta}(\Psi^{\mu}(\Sigma))$-ontology $\mathcal{O}$, and a query $\varphi \in Sen^{\mathcal{Q}}(\Psi^{\eta}(\Sigma))$.

The proof is similar to the proof of Lemma 3.2.24.

This is to say, that given a framework $\mu : \mathcal{G} \to \mathcal{L}$ over query basis $\eta : \mathcal{G} \to \mathcal{Q}$ and an ontology formulated in $\mathcal{L}'$, then, provided there is a morphism $\zeta : \mathcal{L} \to \mathcal{L}'$ we can safely lift the ontology to $\mathcal{L}$ and after translating it into $\mathcal{G}$ answer the query. That gives us exactly the same results as creating framework $\mu' : \mathcal{G} \to \mathcal{L}'$ over query basis η using composition of morphisms and then answering the query.

The following proposition shows for a framework with a morphism attached to $\mathcal{L}$ (see Figure A.6) choosing a morphism μ' or a composition of morphisms $\mu;\zeta$ has no impact on the inseparability result. Which is expected behavior of frameworks.

Proposition A.1.17. *For a binary framework $\mathfrak{F} = (\mu_1, \mu_2)$ over query basis $\eta : \mathcal{G} \to \mathcal{Q}$ with attached morphisms $\zeta_i : \mathcal{L}_i \to \mathcal{L}'_i$, and a binary framework $\mathfrak{F} = (\mu'_1, \mu'_2)$ over query basis η with $\mu'_i = \mu_i;\zeta_i$ (for $i = 1,2$), we have that:*

$$\gamma^{\zeta}_{\Psi^{\mu}(\Sigma)}(\mathcal{O}_1) \sqsubseteq^{\eta}_{\Sigma} \gamma^{\zeta}_{\Psi^{\mu}(\Sigma)}(\mathcal{O}_2) \quad \textit{iff} \quad \mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2$$

for any signature $\Sigma \in |Sig^{\mathcal{G}}|$ and ontologies $\mathcal{O}_i \subseteq Sen^{\mathcal{L}'_i}(\Psi^{\zeta_i}(\Psi^{\mu_i}(\Sigma)))$ the following holds:

The proof is similar to the proof given for Proposition 3.2.26

The next corollary is a consequence of Proposition A.1.17, it shows that for frameworks $\eta : \mathcal{G} \to \mathcal{Q}$ over query basis η, with an attached morphism $\rho : \mathcal{Q} \to \mathcal{L}$ and framework $\mu : \mathcal{G} \to \mathcal{L}$ over query basis η, where $\mu = \eta;\rho$, Σ-entailment of $\mathcal{L}$-ontologies $\mathcal{O}_1, \mathcal{O}_2$ over translated signature Σ coincides (see Figure A.7).

Corollary A.1.18. *Let $\mu : \mathcal{G} \to \mathcal{L}$ be a framework over a query basis $\eta : \mathcal{G} \to \mathcal{Q}$, such that there is morphism $\rho : \mathcal{Q} \to \mathcal{L}$ and $\mu = \eta;\rho$. Let $\Sigma \in |Sig^{\mathcal{G}}|$ and $\mathcal{O}_1$ and $\mathcal{O}_2$ be $\Psi^{\mu}(\Sigma)$-ontologies for μ then:*

$$\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2 \quad \textit{iff} \quad \gamma^{\rho}_{\Psi^{\mu}(\Sigma)}(\mathcal{O}_1) \sqsubseteq^{\eta}_{\Sigma} \gamma^{\rho}_{\Psi^{\mu}(\Sigma)}(\mathcal{O}_2).$$

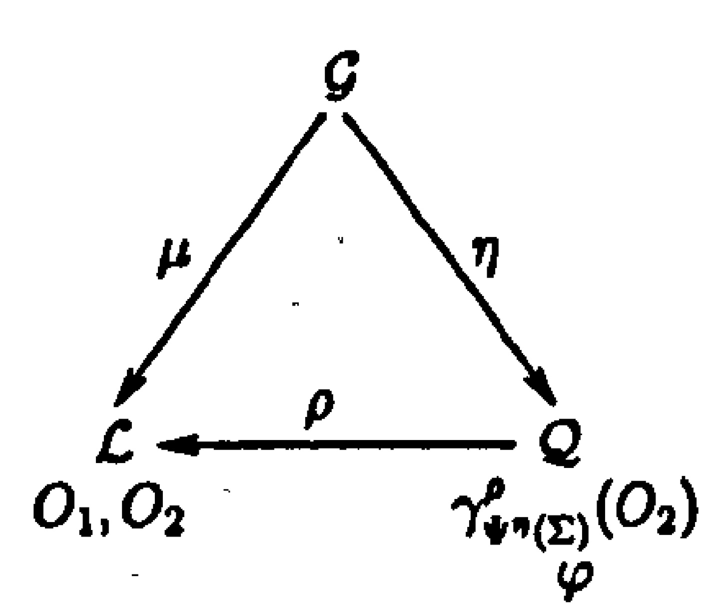


Figure A.7.

If an ontology language can be translated directly into the query language, then entailment can be reduced to showing that each sentence in one ontology is a consequence of the other:

Proposition A.1.19. *Given a framework $\mathcal{G} \xrightarrow{\mu} \mathcal{L}$ over a query basis $\mathcal{G} \xrightarrow{\eta} \mathcal{Q}$ with a morphism $\mathcal{Q} \xrightarrow{\rho} \mathcal{L}$, such that $\mu = \eta;\rho$, and $\Psi^{\mu}(\Sigma)$-ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, we have: $\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2$ iff $\mathcal{O}_1 \models^{\eta}_{\Sigma} \varphi$, for all $\varphi \in \gamma^{\rho}_{\Psi^{\eta}(\Sigma)}(\mathcal{O}_2)$.*

The proof is similar to the proof of Proposition 3.2.28

Moving to a richer global language preserves consequences:

Lemma A.1.20. *For framework μ over query basis $\eta : \mathcal{G} \to \mathcal{Q}$, if we have a morphism $\lambda : \mathcal{G}' \to \mathcal{G}$, there is a framework $\mu' = \lambda;\mu$ over query basis $\eta' = \lambda;\eta$, and we have:*

$$\mathcal{O} \models^{\eta}_{\Sigma} \varphi \quad \textit{implies} \quad \mathcal{O} \models^{\eta'}_{\Sigma} \varphi$$

for any $\Psi^{\mu}(\Sigma)$-ontology for μ and any query $\varphi \in Sen^{\mathcal{Q}}(\Psi^{\eta}(\Sigma))$, with $\Sigma \in Sig^{\mathcal{G}}$. Moreover, if δ^{λ} is surjective, then the converse implication also holds.

The proof is similar to the proof of Lemma 3.2.29

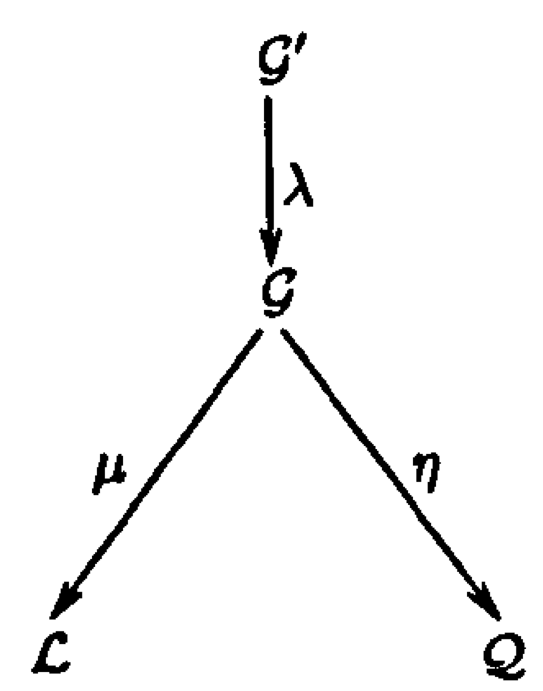


Figure A.8.

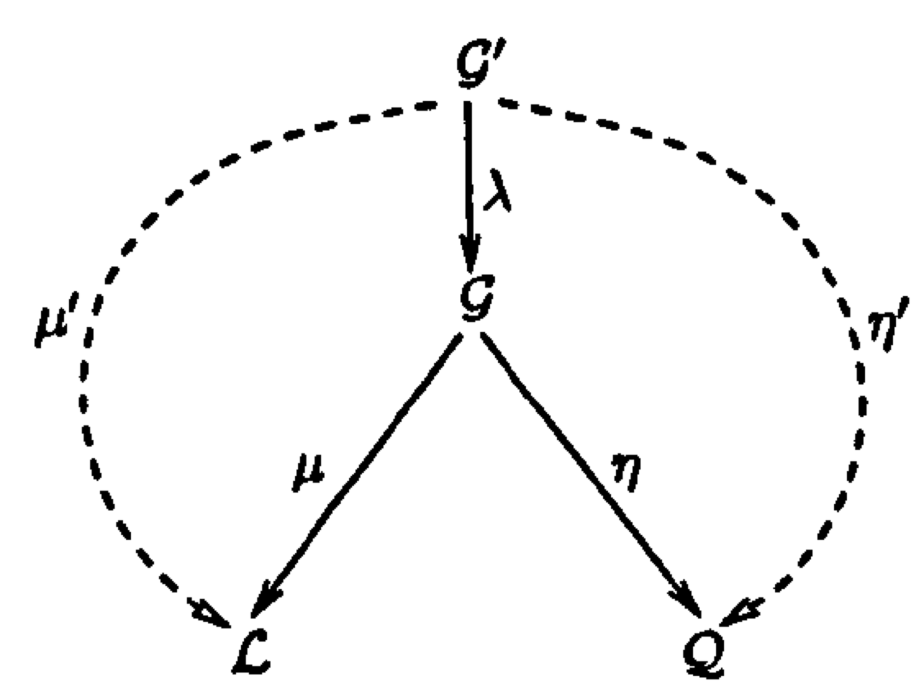


Figure A.9.

Now we compare two frameworks presented in Figure A.10 and in Figure A.11. The former is simply the case when the ontology language is a sublanguage of the query language, but the query is expressed in $\mathcal{Q}$ over a translated $\mathcal{L}$-signature. Whereas the latter is the case when the ontology language is the same as the query language, but we translate both into $\mathcal{Q}$, which in this case is the global language. Additionally, in both cases morphism $\mu : \mathcal{G} \to \mathcal{L}$ is conservative. We show a correlation of conservativeness of morphism $\mu : \mathcal{G} \to \mathcal{L}$ and coincidence of $\mathcal{O}_1 \approx^{\mu}_{\Psi^{\mu}(\Sigma)} \mathcal{O}_2$ and $\mathcal{O}_1 \approx^{1_{\mathcal{G}}}_{\Sigma} \mathcal{O}_2$. To do that we need an auxiliary lemma. The statement of this auxiliary lemma is that given two signatures $\Sigma, \Sigma' \in \mathsf{Sig}^{\mathcal{G}}$, such that there is a signature morphism $\sigma : \Sigma \to \Sigma'$ and an ontology $\mathcal{O} \subseteq \mathsf{Sen}^{\mathcal{L}}(\Psi^{\mu}(\Sigma'))$ we are guaranteed that $\mathcal{O}$ itself and the set of consequences of $\mathcal{O}$, restricted to these sentences that were originally expressed in $\Psi^{\mu}(\Sigma)$ and then translated into $\Psi^{\mu}(\Sigma')$ using $\mathsf{Sen}^{\mathcal{L}}(\sigma)$ (i.e. $\mathcal{O}_{\sigma}$), give us exactly the same set of consequences over sentences expressed in $\Psi^{\mu}(\Sigma)$.

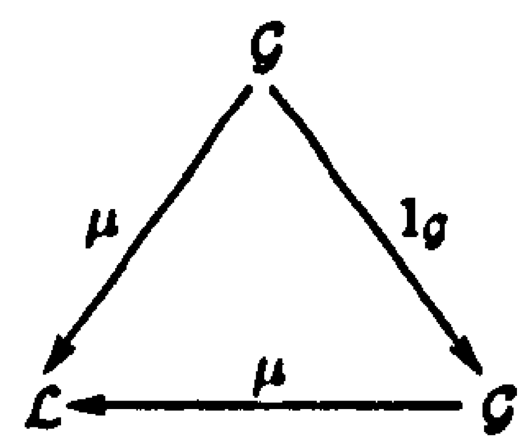


Figure A.10: Framework μ over query basis $1_{\mathcal{G}}$

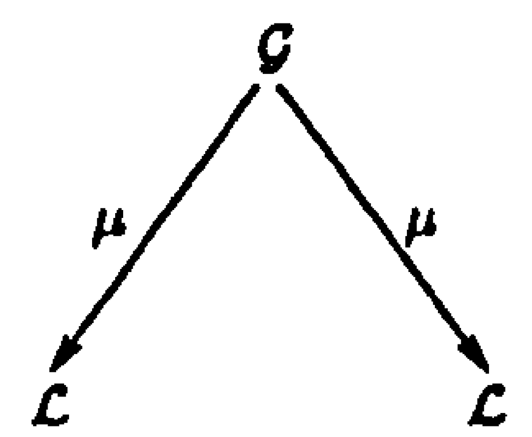


Figure A.11: Framework μ over itself as a query basis

Lemma A.1.21. *For any framework $\mu : \mathcal{G} \to \mathcal{L}$ over itself as a query basis and all signatures $\Sigma, \Sigma' \in Sig^{\mathcal{G}}$, s.t. there is a signature morphism $\sigma : \Lambda \to \Lambda'$ and an ontology $\mathcal{O} \subseteq Sen^{\mathcal{L}}(\Psi^{\mu}(\Lambda'))$ the following holds:*

$$\mathcal{O} \approx^{1_{\mathcal{G}}}_{\Sigma} Sen^{\mathcal{L}}(\sigma)(\mathcal{O}_{\sigma}),$$

i.e. for every $\varphi \in Sen^{\mathcal{L}}(\Psi^{\mu}(\Lambda))$ we have that $\mathcal{O} \models^{1_{\mathcal{G}}}_{\Sigma} \varphi$ iff $Sen^{\mathcal{L}}(\sigma)(\mathcal{O}_{\sigma}) \models^{\mu}_{\Sigma} \varphi$.

Proof is similar to the proof of Lemma 3.2.31.

Next proposition presents close correlation between the inseparability problem in two types of frameworks presented above and conservativity of morphism $\mu : \mathcal{G} \to \mathcal{L}$.

Proposition A.1.22. *For framework $\mu : \mathcal{G} \to \mathcal{L}$ over query basis $1_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$ and framework μ over query basis μ, morphism μ is conservative iff Σ-inseparability w.r.t. μ, coincides with Σ-inseparability w.r.t. $1_{\mathcal{G}}$ for any signature $\Sigma \in |Sig^{\mathcal{G}}|$.*

Note that $\mathcal{O}_1 \approx^{1_{\mathcal{G}}}_{\Sigma} \mathcal{O}_2$ means that $\mathcal{O}_1$ and $\mathcal{O}_2$ are indistinguishable relative to the sentences from the set $Sen^{\mathcal{G}}(\Sigma)$, whereas $\mathcal{O}_1 \approx^{\mu}_{\Sigma} \mathcal{O}_2$ means that $\mathcal{O}_1$ and $\mathcal{O}_2$ are indistinguishable relative to the sentences from the set $Sen^{\mathcal{L}}(\Psi^{\mu}(\Sigma))$.

Proof is similar to the proof of Proposition 3.2.32.

A.2 Robustness properties

In Section 3.3 we discussed robustness properties in the comorphism framework setting. Now we define robustness for morphism frameworks. This formulation is also suggests that the use of comorphisms is more intuitive. Similarly to the above argument the fact that signatures originate in the global language makes the definition less intuitive.

Definition A.2.1. *For any binary framework* $\mathfrak{F} = (\mu_1, \mu_2)$ *over query basis* η *we say that* $\mathfrak{F}$ *is* ***robust under:***

- ***vocabulary extension*** *if for all signatures* $\Lambda_1, \Lambda_2, \Sigma, \Sigma'$ *in* $Sig^{\mathcal{G}}$*, such that* $\Sigma' \cap (\Lambda_1 \cup \Lambda_2) \subseteq \Sigma$*, all ontologies* $\mathcal{O}_1 \subseteq Sen^{\mathcal{L}_1}(\Psi^{\mu_1}(\Lambda_1))$ *and* $\mathcal{O}_2 \subseteq Sen^{\mathcal{L}_2}(\Psi^{\mu_2}(\Lambda_2))$*, the following holds:*
$$\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2 \quad \textit{implies} \quad \mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma'} \mathcal{O}_2,$$
- ***joins*** *if for all signatures* $\Lambda_1, \Lambda_2, \Sigma$ *in* $Sig^{\mathcal{G}}$*, such that* $\Lambda_1 \cap \Lambda_2 \subseteq \Sigma$ *and all ontologies* $\mathcal{O}_1 \subseteq Sen^{\mathcal{L}_1}(\Psi^{\mu_1}(\Lambda_1))$ *and* $\mathcal{O}_2 \subseteq Sen^{\mathcal{L}_2}(\Psi^{\mu_2}(\Lambda_2))$*, the following holds for* $i = 1, 2$*:*
$$\mathcal{O}_1 \approx^{\eta}_{\Sigma} \mathcal{O}_2 \quad \textit{implies} \quad \mathcal{O}_i \approx^{\eta}_{\Sigma} \mathcal{O}_1 \cup \mathcal{O}_2,$$
- ***replacement in framework*** $\mu : \mathcal{L} \to \mathcal{G}$ *if for all signatures* $\Lambda_1, \Lambda_2, \Lambda, \Sigma$ *in* $Sig^{\mathcal{G}}$*, such that* $\Lambda \cap (\Lambda_1 \cup \Lambda_2) \subseteq \Sigma$*, for all ontologies* $\mathcal{O}_1 \subseteq Sen^{\mathcal{L}_1}(\Psi^{\mu_1}(\Lambda_1)), \mathcal{O}_2 \subseteq Sen^{\mathcal{L}_2}(\Psi^{\mu_2}(\Lambda_2))$*,* $\mathcal{O} \subseteq Sen^{\mathcal{L}}(\Psi^{\mu}(\Lambda))$*, the following holds:*
$$\mathcal{O}_1 \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2 \quad \textit{implies} \quad \mathcal{O}_1 \cup \mathcal{O} \sqsubseteq^{\eta}_{\Sigma} \mathcal{O}_2 \cup \mathcal{O}\,.$$

From the above definition it is easy to see that even though expressing inclusion conditions for signatures becomes easier to read if we use morphisms, it remains less intuitive as all the concerns presented while discussing Definition A.1.3 remain in power.

168

Bibliography

[1] SNOMEDClinical Terms® Reference Sets - Technical specification. `http://www.ihtsdo.org/fileadmin/user_upload/Docs_01/Technical_Docs/reference_sets.pdf`, 2006.

[2] J. Adamek, H. Herrlich, and G. Strecker. *Abstract and concrete categories: The joy of cats.* John Wiley and Sons, New York, 1990.

[3] M. Aiguier and F. Barbier. An Institution-independent Proof of the Beth Definability Theorem. *Studia Logica*, 85(3):333–359, 2007.

[4] G. Antoniou, , and F. V. Harmelen. Web ontology language: OWL. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 67–92. Springer, 2003.

[5] M. Arrais and J. L. Fiadeiro. Unifying theories in different institutions. In M. Haveraaen, O. Owe, and O.-J. Dahl, editors, *Recent Trends in Data Type Specifications. 11th Workshop on Specification of Abstract Data Types*, volume 1130 of *Lecture Notes in Computer Science*, pages 81–101. Springer Verlag, 1996.

[6] A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. DL-lite in the light of first-order logic. In *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pages 361–366. AAAI Press, 2007.

[7] S. Awodey. *Category Theory.* Oxford University Press, Oxford, 2006.

[8] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In L. P. Kaelbling and A. Saffiotti, editors, *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 364–369. Professional Book Center, 2005.

[9] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.

[10] F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL—a polynomial-time reasoner for life science ontologies. In U. Furbach and N. Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR'06)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 287–291. Springer-Verlag, 2006.

[11] F. Baader, R. Peñaloza, and B. Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In *Proceedings of the 30th German Conference on Artificial Intelligence (KI2007)*, LNAI, Osnabrück, Germany, 2007. Springer.

[12] M. Barr and C. Wells. *Toposes, Triples, and Theories.* Number 278 in Grundlehren der mathematischen Wissenschaften. Springer-Verlag, 1985.

[13] M. Barr and C. Wells. *Category theory for computing science.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1990.

[14] C. Batini, S. Ceri, and S. B. Navathe. *Conceptual Database Design: An Entity-Relationship Approach.* Benjamin/Cummings, 1992.

[15] D. Beckett. RDF/xml syntax specification (revised). Technical report, W3C, 2004. http://www.w3.org/TR/rdf-syntax-grammar/.

[16] J.-Y. Béziau. 13 Questions About Universal Logic. *Bulletin of the Section of Logic*, 35:133–150, 2006.

[17] W. N. Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse.* PhD thesis, Universiteit Twente, Enschede, September 1997.

[18] T. Borzyszkowski. Generalized interpolation in casl. In *Information Processing Letter*, volume 76, page 19ï£¡24, 2001.

[19] T. Borzyszkowski. Logical systems for structured specifications. *Theoretical Computer Science*, 286:197–245, 2002.

[20] R. Brachman, R. Fikes, and H. Levesque. KRYPTON: A functional approach to knowledge representation. *IEEE Computer*, 16(10):67–73, 1983.

[21] S. Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In R. L. de Mantáras and L. Saitta, editors, *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-2004)*, pages 298–302. IOS Press, 2004.

[22] D. Brickley and G. R.V. RDF Vocabulary Description Language 1.0: RDF Schema. Technical report, W3C, 2004. http://www.w3.org/TR/rdf-schema/.

[23] R. Burstall and J. Goguen. The semantics of Clear, a specification language. In *Lecture Notes in Computer Science*, volume 86, pages 292–332. Springer, 1980.

[24] D. Calvanese, D. Lembo, M. Lenzerini, and R. Rosati. DL-lite: Tractable description logics for ontologies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, pages 602–607, 2005.

[25] P. P. Chen. The entity-relationship model - toward a unified view of data. *ACM Trans. Database Syst.*, 1(1):9–36, 1976.

[26] E. Clarke and H. Schlingloff. Model checking. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 24, pages 1635–1790. Elsevier Science, 2001.

[27] J. R. B. Cockett. Introduction to distributive categories. *Mathematical Structures in Computer Science*, 3(3):277–307, 1993.

[28] M. Codescu and T. Mossakowski. Heterogeneous colimits. In *ICSTW '08: Proceedings of the 2008 IEEE International Conference on Software Testing Verification and Validation Workshop*, pages 131–140, Washington, DC, USA, 2008. IEEE Computer Society.

[29] V. E. Căzănescu and G. Roşu. Weak inclusion systems. *Mathematical. Structures in Comp. Sci.*, 7(2):195–206, 1997.

[30] V. E. Căzănescu and G. Roşu. Weak inclusion systems: Part two. *Journal of Universal Computer Science*, 6(1):5–21, 2000.

[31] R. Diaconescu. Grothendieck institutions. *Applied Categorical Structures*, 10(4):383–402, 2002. Preliminary version appeared as IMAR Preprint 2-2000, ISSN 250-3638, February 2000.

[32] R. Diaconescu. An Institution-independent Proof of Craig Interpolation Theorem. *Studia Logica*, 77(1):59–79, 2004.

[33] R. Diaconescu. Institutions, Madhyamaka and universal model theory. In J.-Y. Béziau and A. Costa-Leite, editors, *Perspectives on Universal Logic*, pages 41–65. Polimetrica, 2007.

[34] R. Diaconescu. *Institution-independent Model Theory*. Birkhäuser Basel, 2008.

[35] R. Diaconescu, J. Goguen, and P. Stefaneas. Logical support for modularisation. In G. Huet and G. Plotkin, editors, *Logical Environments*, pages 83–130. Cambridge, 1993. Proceedings of a Workshop held in Edinburgh, Scotland, May 1991.

[36] T. Dimitrakos and T. Maibaum. On a generalised modularisation theorem, 2000.

[37] S. Eilenberg and S. Mac Lane. General Theory of Natural Equivalences. *Transactions of American Mathematical Society*, 58(2):231–294, 1945.

[38] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

[39] A. Flögel, H. K. Büning, and T. Lettmann. On the restricted equivalence of subclasses of propositional logic. *ITA*, 27(4):327–340, 1993.

[40] S. Ghilardi, C. Lutz, and F. Wolter. Did I Damage my Ontology? A Case for Conservative Extensions in Description Logics. In P. Doherty, J. Mylopoulos, and C. Welty, editors, *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 187–197. AAAI Press, 2006.

[41] J. Goguen and R. Burstall. Introducing institutions. In *Proceedings, Logics of Programming Workshop*, volume 164, pages 221–256. Springer, 1984.

[42] J. A. Goguen and R. M. Burstall. Institutions: abstract model theory for specification and programming. *J. ACM*, 39(1):95–146, 1992.

[43] J. A. Goguen and G. Roşu. Institution morphisms. *Formal Asp. Comput.*, 13(3-5):274–307, 2002.

[44] B. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):309–322, 2008.

[45] T. R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowl. Acquis.*, 5(2):199–220, June 1993.

[46] I. Helene. Presentation rate and the representation of briefly glimpsed pictures in memory. In *Journal of Experimental Psychology [Human Learning and Memory]*, volume 6, pages 1–12. 1980.

[47] P. Hitzler, R. Sebastian, and M. Krötzsch. *Foundations of Semantic Web Technologies.* Chapman & Hall/CRC, London, 2009.

[48] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The Making of a Web Ontology Language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[49] G. Klyne and J. J. Carroll. Resource Description Framework (RDF):Concepts and Abstract Syntax: Technical report, W3C, 2004. http://www.w3.org/TR/rdf-concepts/.

[50] B. Konev, C. Lutz, D. Walther, and F. Wolter. Formal properties of modularisation. In C. Parent, S. Spaccapietra, and H. Stuckenschmidt, editors, *Ontology Modularisation*, LNCS. Springer, 2008.

[51] B. Konev, D. Walther, and F. Wolter. The logical difference problem for description logic terminologies. In *IJCAR '08: Proceedings of the 4th international joint conference on Automated Reasoning*, pages 259–274. Springer, 2008.

[52] R. Kontchakov, C. Lutz, D. Toman, F. Wolter, and M. Zakharyaschev. Combined FO Rewritability for Conjunctive Query Answering in DL-Lite. In *Proceedings of the 22nd International Workshop on Description Logics (DL2009)*, volume 477 of *CEUR-WS*, 2009.

[53] R. Kontchakov, F. Wolter, and M. Zakharyaschev. Modularity in DL-lite. In D. Calvanese, E. Franconi, V. Haarslev, D. Lembo, B. Motik, A.-Y. Turhan, and S. Tessaris, editors, *Description Logics*, volume 250 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[54] R. Kontchakov, F. Wolter, and M. Zakharyaschev. Can you tell the difference between DL-lite ontologies? In *KR*, pages 285–295, 2008.

[55] O. Kutz, D. Lücke, and T. Mossakowski. Heterogeneously structured ontologies - integration, connection, and refinement. In T. Meyer and M. A. Orgun, editors, *Knowledge Representation Ontology Workshop (KROW 2008)*, volume 90 of *CRPIT*, pages 41–50, Sydney, Australia, 2008. ACS.

[56] O. Kutz and T. Mossakowski. Modules in transition - conservativity, composition, and colimits. In B. C. Grau, V. Honavar, A. Schlicht, and F. Wolter, editors, *WoMO*, volume 315 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[57] O. Kutz and T. Mossakowski. Conservativity in structured ontologies. In M. Ghallab, C. D. Spyropoulos, N. Fakotakis, and N. M. Avouris, editors, *ECAI*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 89–93. IOS Press, 2008.

[58] S. M. Lane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971. Graduate Texts in Mathematics, Vol. 5.

[59] H. J. Levesque. Foundations of a functional approach to knowledge representation. *Artif. Intell.*, 23(2):155–212, 1984.

[60] C. Lutz, D. Walther, and F. Wolter. Conservative extensions in expressive description logics. In M. Veloso, editor, *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 453–458. AAAI Press, 2007.

[61] C. Lutz and F. Wolter. Conservative extensions in the lightweight description logic $\mathcal{EL}$. In F. Pfenning, editor, *Proceedings of the 21th Conference on Automated Deduction (CADE-21)*, Lecture Notes in Artificial Intelligence, pages 84–99. Springer, 2007.

[62] C. Lutz and F. Wolter. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *Journal of Symbolic Computation*, 2009 (To appear).

[63] F. Manola and E. Miller. RDF primer. Technical report, W3C, 2004. http://www.w3.org/TR/rdf-primer/.

[64] J. Meseguer. General logics. In H. D. Ebbinghaus, F. J. Prida, M. Garrido, D. Lascar, and R. M. Artalejo, editors, *Logic Colloquium*, pages 275 – 329. NorthHolland, 1989.

[65] N. F. Noy and M. Musen. Promptdiff: A fixed-point algorithm for comparing ontology versions. In *Proceedings of AAAI*, pages 744–750, 2002.

[66] S. R.N. Recognition memory for words sentences and pictures. In *Journal of Verbal Learning and Verbal Behavior*, volume 6, pages 156–163. 1967.

[67] G. Roşu. Axiomatizability in inclusive equational logic. *Mathematical Structures in Computer Science*, To appear. Submitted January 1997.

[68] J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual.* Addsion Wesley, 2006.

[69] A. Salibra and G. Scollo. Interpolation and compactness in categories of pre-institutions. *Mathematical Structures in Computer Science*, 6(3):261–286, 1996.

[70] D. Sannella and A. Tarlecki. Specifications in an arbitrary institution. *Information and Contorol*, 76(2/3):165–210, 1988.

[71] M. Schorlemmer and Y. Kalfoglou. Institutionalising ontology-based semantic integration. *Appl. Ontol.*, 3(3):131–150, 2008.

[72] K. A. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. *Journal of the American Medical Informatics Association*, Fall Symposium Special Issue, 2000.

[73] S. Staab and R. Studer, editors. *Handbook on Ontologies.* International Handbooks on Information Systems. Springer, 2004.

[74] L. J. Stockmeyer and A. K. Chandra. Provably difficult combinatorial games. *SIAM Journal on Computing*, 8(2):151–174, 1979.

[75] R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *Data Knowl. Eng.*, 25(1-2):161–197, 1998.

[76] A. Tarlecki. On the existence of free models in abstract algebraic institutuons. *Theoretical Computer Science*, 37:269–304, 1986.

[77] A. Tarlecki. Moving between logical systems. In M. Haveraaen, O. Owe, and O.-J. Dahl, editors, *Recent Trends in Data Type Specifications. 11th Workshop on Specification of Abstract Data Types*, volume 1130 of *Lecture Notes in Computer Science*, pages 478–502. Springer Verlag, 1996.

[78] A. Tarlecki. Towards heterogeneous specifications. In *In Frontiers of Combining Systems FroCoS'98, Applied Logic Series*, pages 337–360. Kluwer Academic Publishers, 1998.

[79] A. Tarlecki. Institutions: An Abstract Framework for Formal Specifications. In *Algebraic Foundations of Systems Specification*, pages 105–130. Springer-Verlag New York, Inc., 1999.

[80] T. J. Teorey. *Database modeling and design: the entity-relationship approach.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[81] B. Thalheim. Foundations of entity - relationship modeling. *Ann. Math. Artif. Intell.*, 7(1-4):197–256, 1993.

[82] B. Thalheim. *Entity-Relationship Modeling: Foundations of Database Technology.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2000.

[83] M. Uschold and M. Grüninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11:93–136, 1996.

[84] R. F. C. Walters. *Categories and Computer Science*. Cambridge University Press, 1991.

[85] A. Zimmermann, M. Krötzsch, J. Euzenat, and P. Hitzler. Formalizing ontology alignment and its operations with category theory. In *Proceeding of the 2006 conference on Formal Ontology in Information Systems*, pages 277–288, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.