
Workshop Notes

**The IJCAI-11 Workshop on
Automated Reasoning about Context and Ontology Evolution**

ARCOE-11

**July 18, 2011
Barcelona, Catalonia, Spain**

held at the
International Joint Conference on Artificial Intelligence

Co-Chairs

**Alan Bundy
Jos Lehmann
Ivan Varzinczak**

<http://www.arcoe.org>

Introduction to the Notes of the IJCAI-11 Workshop ARCOE-11

Automated Reasoning about Context and Ontology Evolution

Alan Bundy¹, Jos Lehmann¹, Ivan Varzinczak^{2,3}

¹ School of Informatics, University of Edinburgh

² CSIR Meraka Institute, Pretoria, South Africa

³ School of Computer Science, University of KwaZulu-Natal, Durban, South Africa

Methods of automated reasoning have solved a large number of problems in Computer Science by using formal ontologies expressed in logic. Over the years, though, each problem or class of problems has required a different ontology, and sometimes a different version of logic. Moreover, the processes of conceiving, controlling and maintaining an ontology and its versions have turned out to be inherently complex. All this has motivated much investigation in a wide range of disparate disciplines—from logic-based Knowledge Representation and Reasoning to Software Engineering, from Databases to Multimedia—about how to relate ontologies to one another.

Just like its previous editions, ARCOE-11 aims at bringing together researchers and practitioners from core areas of Artificial Intelligence (Knowledge Representation and Reasoning, Contexts, and Ontologies) to discuss these kinds of problems and relevant results. Historically, there have been at least three different, yet interdependent motivations behind this type of research: defining the relationship between an ontology and its context, providing support to ontology engineers, enhancing problem solving and communication for software agents.

ARCOE call for papers has been formulated against such historical background. Submissions to ARCOE-11 have been reviewed by two to three Chairs or PC members and ranked on relevance and quality. Seventy five percent of the submissions have been selected for presentation at the workshop and for inclusion in these Workshop Notes.

Thanks to the invaluable and much appreciated contributions of its program committee and its authors, ARCOE-11 provides participants with an opportunity to position various approaches with respect to one another. Hopefully, though the workshop and these Notes will also start a process of cross-pollination and consolidate the constitution of a truly interdisciplinary research-community dedicated to automated reasoning about contexts and ontology evolution.

(Edinburgh, Pretoria – June 2011)

ARCOE-11 Co-Chairs

Alan Bundy (University of Edinburgh, UK)

Jos Lehmann (University of Edinburgh, UK)

Ivan Varzinczak (CSIR Meraka Institute and University of KwaZulu-Natal, South Africa)

ARCOE-11 Invited Speaker

Chiara Ghidini (FBK Fondazione Bruno Kessler, Italy)

ARCOE-11 Program Committee

Franz Baader (TU Dresden, Germany)

Christoph Benzmueller (Articulate Software, USA)

Richard Booth (University of Luxembourg and Mahasarakham University, Thailand)

Paolo Bouquet (University of Trento, Italy)

Jim Delgrande (Simon Fraser University, Canada)

Jerome Euzenat (INRIA & LIG, France)

Nicola Fanizzi (University of Bari, Italy)

Giorgos Flouris (FORTH, Greece)

Chiara Ghidini (FBK Fondazione Bruno Kessler, Italy)

Deborah McGuinness (Rensselaer Polytechnic Institute, USA)

Thomas Meyer (CSIR Meraka Institute, South Africa)

Alessandra Mileo (Digital Enterprise Research Institute, Ireland)

Amedeo Napoli (LORIA CNRS, France)

Maurice Pagnucco (The University of New South Wales, Australia)

Valeria de Paiva (Cuil Inc., USA)

Jeff Pan (University of Aberdeen, UK)

Dimitris Plexousakis FORTH, Greece)

Guilin Qi (Southeast University, China)

Marcio Ribeiro (University of Sao Paulo, Brazil)

Luciano Serafini (FBK Fondazione Bruno Kessler, Italy)

Renata Wassermann (Universidade de Sao Paulo, Brazil)

Table of Contents

Alan Bundy <i>On the Evolution of Classifications</i>	7-12
Christophe Jouis, Jean-Gabriel Ganascia, Franck Guy and Claude Jouis <i>Emergence in scientific literature and dynamics in ontologies: application in paleoanthropology</i>	13-17
Valeria De Paiva and Natasha Alechina <i>Contextual Constructive Description Logics</i>	18-22
Sergio Alejandro Gomez, Carlos Chesevar and Guillermo Simari <i>Local-as-View Integration of Ontologies Through Argumentation</i>	23-27
Sine Zambach and Mai Ajspur <i>Reduction of composites of relations between classes within formal ontologies</i>	28-32
Alexander Demin, Denis Ponomaryov and Evgeny Vityaev <i>Probabilistic concepts in formal contexts</i>	33-37
William Gatens, Boris Konev, Michel Ludwig and Frank Wolter <i>Versioning Based on Logical Difference for Lightweight Description Logic Terminologies</i>	38-42
Montassar Ben Messaoud, Philippe Leray and Nahla Ben Amor <i>SemCaDo: a serendipitous causal discovery algorithm for ontology evolution</i>	43-47
Michael Chan, Jos Lehmann and Alan Bundy <i>GALILEO: A System for Automating Ontology Evolution</i>	48-52

On the Evolution of Classifications *

Alan Bundy

School of Informatics, University of Edinburgh
Edinburgh, Scotland, UK
A.Bundy@ed.ac.uk

Abstract

The *evolution* of ontologies has been identified by W3C as a significant issue. *Automatic* ontology evolution is increasingly important to allow runtime communication between vast collections of online agents. For this application, manual, offline evolution is both too slow and too expensive.

Very many ontologies are classifications of a set of objects. In order to gain insight into how the evolution of classifications might be automated, we discuss some historic examples of such evolution. From this discussion, we extract some general principles that might be employed to help automate the ontology evolution process.

1 Introduction

The 2004 W3C Recommendation on the “OWL Web Ontology Language: Use Cases and Requirements” justifies *ontology evolution* as follows:

Since the Web is constantly growing and changing, we must expect ontologies to change as well. Ontologies may need to change because there were errors in prior versions, because a new way of modeling the domain is preferred, or because new terminology has been created (e.g., as the result of the invention of new technology). A web ontology language must be able to accommodate ontology revision. [<http://www.w3.org/TR/webont-req/#goal-evolution>]

Many ontologies consist of classifications of a set of objects. A typical such classification will be a tree structure in which the nodes are subsets of the object set and arcs between them represent set inclusion. Given the importance assigned to ontology evolution by W3C and the ubiquity of classificatory ontologies, we thought it would be useful to investigate how some historic examples of how classifications have evolved. We hope that his analysis will reveal some general principles that might then be employed in the automation of classification evolution.

*Thanks to Fiona McNeill, Liwei Deng and two anonymous ARCOE-11 reviewers for feedback on earlier drafts and to my research group for the discussion at a seminar.

We take the stance that there is no perfect or final classification of a set of objects: *representation is a fluent* [Bundy and McNeill, 2006]. Classifications evolve with both our knowledge of the objects and the purpose for which they are employed. Neither our knowledge nor our applications are limited, so evolution must be seen as a non-terminating process. In a similar way, no definition may be regarded as final and complete, but will also evolve with our knowledge and applications. So all the definitions and classifications below must be regarded as snapshots in a process of flux.

2 The Classification of the Animals

From the earliest historical records, people have classified animals and plants, i.e., partitioned them into sets or arranged them in hierarchies. A typical example can be seen in Figure 1¹. It consists of a tree whose nodes each correspond to a set of animals. An arc between two nodes represents a subset relation between the lower and the upper node.

Notice that each node of the tree in Figure 1 has some associated text. This gives some properties common to the set of animals at that node. Some example animals are sometimes also listed.

The classification of animals has evolved significantly over time. The main driver of this evolution has been to review the defining properties of each node. For instance,

- Pre-scientific classifications were based on superficial properties, i.e., mainly matters of immediate appearance or human usage, the latter being culturally dependent.
 - An example of the former is that, because whales, dolphins and porpoises lived in the sea and were fish-shaped, they were classified as fish.
 - An example of the latter is the classification of all pests under ‘vermin’. An example from the classification of plants is the use of ‘vegetable’ to describe a selection of plant components, including some, but not all, fruits, leaves and roots, whose common property is that we can eat them².

¹<http://www.rapidonline.com/1/1/6239-classification-of-animals-wall-chart.html>

²I’m grateful to an anonymous reviewer for suggesting this example.

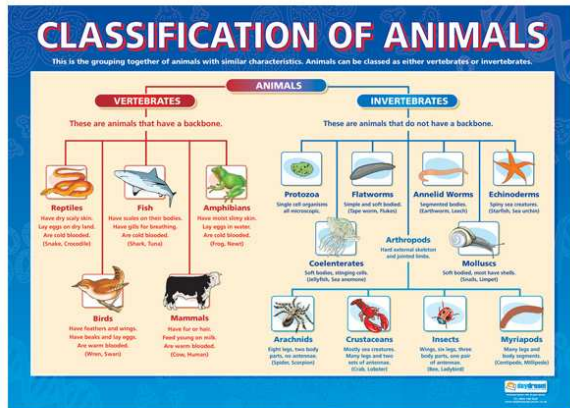


Figure 1: A Classification of Animals

- In 1778, Swedish biologist, Linnaeus started to classify whales as *mammals* on the grounds that they shared significant non-superficial properties with the other mammals³. These properties are some of those listed as defining ‘mammals’ in Figure 1, that is having hair or fur, feeding young on milk, being warm-blooded, etc.
- Darwin’s theory of evolution and the discovery of the fossil record led to both the inclusion of extinct animals in the classification and defining properties based on common ancestry and evolutionary descent. The hierarchical classification called *Cladistics* uses only such Darwinian properties⁴.
- Most recently, the discovery of DNA has led to a further evolution in which the defining properties are based on genetic similarities. This has led to the identification of previously unrecognised distinctions. For instance, recent genetic comparison of the forest-dwelling African elephant with its savannah-dwelling cousin shows them to have diverged as much as the Asian elephant and the woolly mammoth⁵. Similarly, recent DNA studies have discovered that the closest living relatives of the beaver are scaly-tailed flying squirrels, not gophers as was previously thought⁶.
- Note that this change of defining property also undermines the tree-structure of the classification. DNA dif-

³http://www.ecokids.ca/pub/eco_info/topics/whales/mammals.cfm

⁴<http://en.wikipedia.org/wiki/Cladistics>

⁵<http://www.newscientist.com/article/mg20927942.900-ancient-genomes-suggest-africa-has-two-elephant-species.html>

⁶<http://www.newscientist.com/article/mg20927984.100-60-seconds.html>

ferences are multi-dimensional: evolution can change any piece of DNA in multiple ways. Suppose, for instance, that gene A evolves to A1 to create one new sub-species and gene B to B1 to create another. These two sub-species may subsequently converge again in a new subspecies with *both* genes A1 and B1, making the tree into a graph. Moreover, it makes the boundaries between nodes more fuzzy. Some genetic variation has to be tolerated within a node, or each node would be singleton animal. But how much variation is necessary before a node needs to be split?

- Variant classifications may survive for a variety of reasons. For instance, simpler ones may be useful for elementary education, especially when the students cannot be expected to understand the potentially more sophisticated science informing the definitional properties of later classifications. Figure 1 is an example of such a simple classification aimed at primary school students. Some variant classifications may be aimed at specific applications, e.g., a guide to edible food, field identification for nature lovers, warnings of dangerous animals, historical interest.

From this example we can draw some conclusions.

1. We can’t hope to model the evolution of classifications unless we also know the defining properties of these classifications. In the animal classification, these defining properties change from first superficial to more fundamental ones and then to the underlying DNA. Each new advance in understanding may make new defining properties available, e.g., the discovery that some animals were warm blooded and some cold blooded, the evidence of common ancestry arising from the fossil record, the discovery of DNA.
 2. We have to be able to reason *about* these properties, i.e., to argue why one is a better basis for classification than another. The classification of the animals provides some possible criteria.
 - A classification may be preferred because it provides better explanatory power, e.g., DNA can potentially explain the similarities and differences between different animals.
 - A classification may be rejected because its defining properties may come to be seen as environmentally contingent rather than fundamental, e.g., fish shape is an efficient shape for animals swimming in water, which is why different kinds of animal (e.g., whales and sharks) sometimes have similar shape. Contingency may also arise because a property is defeasible, e.g., some birds can’t fly, an insect may lose one of its six legs, etc. Defeasible properties do not provide reliable definitions, e.g., penguins are still considered birds even though they are flightless.
- Note that what may be viewed as fundamental and final at one stage of classification evolution may be viewed as merely contingent at a later stage. For instance, the convergent evolution of the shape of

sharks and whales, and the efficiency of fish shape are both relatively recent discoveries. So we should not be seduced into believing that the current defining properties represent a *completion* of the evolutionary process. For instance, an animal's DNA can now change during its lifetime due to genetic engineering, so even this property is defeasible.

- A classification may be preferred because it leads to successful prediction, e.g., animals found to have some of the properties of mammals may be predicted to have others, even before these other properties are observed. Perhaps live birth and hair in whales are examples. Novel but viable bacterial cells have even been created from artificially constructed DNA, after analysis predicted what the minimal viable DNA was⁷.
- A classification may be preferred because it provides a finer granularity of distinction, e.g., the Linnean hierarchy contains kingdoms, classes, orders, genera and species, with wider branching and finer distinctions being frequently added, as new scientific discoveries are made. DNA provides an exponentially bigger space of possibilities yet.
- A new object may be discovered that does not fit into the existing classification, forcing a change to the classification so that it does fit, e.g., the duck-billed platypus has features of both mammals and birds. Its discovery led to the creation of a new order of mammals: the monotremes.
- Classifications may be tailored to particular purposes or usages, e.g., simplified versions for elementary education, field observation, edible *vs* inedible food, etc.

3. We have to be prepared for the shape of the classification to change if the properties do. For instance, Linnean trees may not be the best way to summarise DNA differences.

3 The Classification of the Elements

Similarly to the animals and plants, classifications of the elements go back to ancient history. The earliest classification was into four elements: earth, water, air and fire, and was due to the Greek philosopher Empedocles of Sicily⁸. The basis of this classification seems to be superficial physical appearance, although there also seems to have been a religious influence, with each element being associated with a god: Zeus, Hera, Nestis and Aidoneus.

Our modern classification originated with Mendeleev's periodic table⁹ (see Figure 2¹⁰). Firstly, the number of classified

elements changed from 4 to 92, with the original 4 elements now reclassified as mixtures, compounds or both, of the new 92 elements, or, in the case of fire, as being either a chemical reaction or photons, according to how you view it. Secondly, the new basis of Mendeleev's classification was chemical and physical properties of the elements. For instance, the elements line up in order of atomic weight: left/right, top/bottom. Each column contains elements of similar properties, e.g., the noble gases that line up in the rightmost column share the property of being chemically inert, while the alkali metals line up in the left-hand column.

The diagram shows Mendeleev's Periodic Table with elements arranged in rows (periods) and columns (groups). The table is color-coded: pink for Representative Elements (s-block), green for Transition Elements (d-block), orange for Inner Transition Elements (f-block), and light blue for Noble gases (p-block). The elements are labeled with their chemical symbols and atomic numbers. The table is divided into four main sections: Representative Elements (s-block), Transition Elements (d-block), Inner Transition Elements (f-block), and Noble gases (p-block).

Figure 2: Mendeleev's Periodic Table

Our modern understanding of atomic structure has given an explanation of why elements in the same column have similar properties. Electrons are arranged in nested shells around the nucleus¹¹. There is a limit to how many electrons can occupy each shell, but this limit increases quadratically. The number of electrons occupying the outermost shell determines the chemical bonds that the element can make. So, for instance, when the last shell is full, the element is chemically inert: a noble gas. The fact that the size of each shell increases, explains why the rows of the periodic table have gaps at the top and why two separate rows are required at the bottom: the lanthanides and the actinides.

A better way to represent this would be as concentric rings of increasing size, to reflect the growing size of the shells. There have been various attempts to produce such a circular periodic table¹², but the quadratic growth of shell size makes this difficult to do while retaining visibility in the outer rings. The best attempt I could find in a short search is in Figure 3¹³, but even here the lanthanides and actinides are squeezed into one entry, rather than using a bigger ring. Note that, once again, the defining properties of the classification have changed. Each element is placed in the ring that corresponds to the number of occupied shells in its electron cloud, whereas

⁷<http://www.newscientist.com/article/dn18942-immaculate-creation-birth-of-the-first-synthetic-cell.html>

⁸<http://en.wikipedia.org/wiki/Empedocles>

⁹<http://en.wikipedia.org/wiki/Mendeleev>

¹⁰<http://www.tutorvista.com/content/science/science-ii/periodic-classification-elements/modern-periodic.php>

¹¹http://en.wikipedia.org/wiki/Electron_shells#Shells

¹²http://en.wikipedia.org/wiki/Alternative_periodic_tables

¹³http://en.wikipedia.org/wiki/File:TROPE_777x777_AB_082009.png

in Figure 2 an element's row was determined by its chemical properties¹⁴.

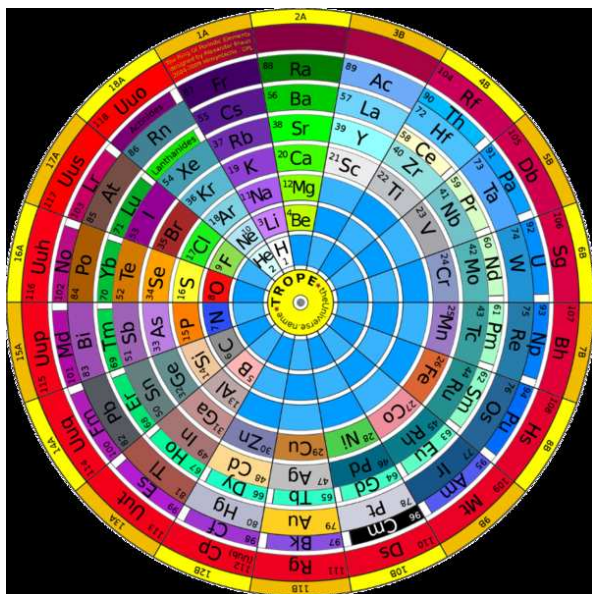


Figure 3: The Ring Of Periodic Elements (TROPE) (August 2009)

We can draw similar conclusions from the classification of the elements that we drew from the classification of the animals.

1. The change in defining properties of the elements, e.g., from physical appearance to chemical properties to atomic structure, drove the evolution of their classification. Scientific advances provided the candidate defining properties for the new classifications, e.g., separation between compounds and elements, discovery of their chemical properties and atomic weight, the quantum structure of the atom, etc.
2. To model this we'd have to reason why atomic structure is a better basis for a classification than chemical properties, which was a better basis than superficial appearance. The reasons echo some of those at work in the evolution of the classification of animals.
 - Better explanatory power, i.e., atomic structure explains chemical properties which, in turn, explains physical appearance.
 - Defining properties seen as contingent, e.g., whether an object appears at room temperature as solid (earth), liquid (water) or gas (air), depends only on its freezing and boiling points.
 - Prediction of new elements, by both Mendeleev's table and quantum mechanics.

¹⁴All these circular representations have two rings of 8 elements and two of 18, whereas a naive understanding of electron shells would suggest that there should be only one of each. It seems that some shells cannot be completely filled until the next one is started.

- A finer granularity of distinction, e.g., different kinds of 'air' in Mendeleev's table, different isotopes of the same element in quantum mechanics.
- The existence of objects that don't fit well into the existing classification, e.g., the lanthanides and actinides.

3. The move from 4 to 92+ elements was driven by chemical analysis as was the shape of Mendeleev's table, e.g., the discovery of multiple kinds of 'air' undermined its classification as a single element and led to the identification of new gaseous elements (and compounds). The move to a circular 'table' was driven by the understanding of electron shells.
4. Mendeleev's table is retained for educational purposes. Perhaps the quantum mechanical basis for a more sophisticated story would be hard to explain to students. Also, the majority of working quantum physicists subscribe to the 'Copenhagen interpretation', which rejects intuitive, non-mathematical accounts as unobtainable, misleading and undesirable, so they might not even *seek* a better classification.

4 Classification in Action in Astronomy

It's instructive to witness classifications being formed. We're historically lucky that several such classifications are currently being hotly debated, especially in Astronomy.

The case that has received the most public attention is whether Pluto should be classified as a planet¹⁵. What drove the debate was the discovery of similar objects to Pluto in the outer solar system, e.g., Chiron, and the expectation that many more such objects would be discovered in the future. It's interesting that the principal arguments on either side of the debate were more 'political' than scientific.

- Those arguing for the demotion of Pluto to a new classification of *dwarf planet* were mostly motivated by wanting to keep the designation 'planet' as something special. They did not want to open the door to huge numbers of objects being called planets.
- Those arguing for the retention of Pluto as a planet were mostly driven by tradition, i.e., Pluto had been classified as a planet since 1930, and they were reluctant to see this decision revoked.

The arbitrary nature of the final decision is emphasised by the method of decision: a vote, on August 24 2006, by the International Astronomical Union. However, what they voted for was some new defining properties of a planet, which would exclude Pluto, Chiron and other anticipated planetary candidates. These were:

1. The object must be in orbit around the Sun.
2. The object must be massive enough to be a sphere by its own gravitational force. More specifically, its own gravity should pull it into a shape of hydrostatic equilibrium.
3. It must have cleared the neighbourhood around its orbit.

¹⁵<http://en.wikipedia.org/wiki/Pluto>

Pluto (and Chiron *et al*) fails property 3. Its mass is too small to clear out other objects in its neighbourhood. One might speculate that property 3 was added *precisely* to exclude Pluto, since it seems a bit esoteric compared with the other two properties.

A similar debate may be brewing over the separation between star clusters and galaxies¹⁶. Star clusters are collections of stars that formed simultaneously from the same gas cloud, whereas galaxies contain enough gas to form several generations of stars, but neither have a formal definition. Recent observations of the star cluster Omega Centauri, however, suggest that it might contain multiple generations of stars. Two astronomers have recently launched an informal poll to ask astronomers what they think the defining properties of a galaxy should be.

Again, we see voting techniques being used to settle a classification problem. This recognises that, ultimately, although scientific properties are used to define a classification, the choice of those properties is essentially a political decision. We see similar drivers as in the case of Pluto:

- A galaxy is something special. We don't want to grant its status too readily.
- We should respect tradition and not change the status of an object too readily.

In this case, however, both these arguments are pushing in the same direction: to retain the classification of Omega Centauri as a star cluster. I expect the polled astronomers to agree on an additional defining property of galaxies that excludes Omega Centauri, in the way that property 3 above excludes Pluto as a planet¹⁷.

5 Conclusion

We have looked at a few examples of how classifications have evolved historically. This has lessons for us in the automation of classification evolution.

Firstly, classifications are defined by properties of the objects being classified. Objects with similar properties are placed into the same class. Moreover, objects with only slight differences in their properties are classified closer together than those with major differences. Advances in science may make new kinds of defining property available.

Secondly, to emulate classification automation, a system will have to reason about rival defining properties, i.e., deciding that one is better than another. Common preference criteria are:

- Better explanatory power of the new classification.
- Defining properties initially seen as fundamental are subsequently seen as contingent and, therefore, rejected in favour of more fundamental ones.
- Successful prediction from the new classification.
- Finer granularity of distinctions in the new classification.

¹⁶<http://www.newscientist.com/article/dn20026-when-is-a-group-of-stars-not-a-galaxy.html>

¹⁷This prediction has been subsequently confirmed by the outcome of the vote.

- The discovery of objects that don't fit into the old classification.

Sometimes, however, defining properties are chosen not by these scientific criteria but on 'political' grounds, e.g., a preference for reserving some classes to a few privileged objects or a conservative attitude to change. We saw these at work in the astronomical examples of §4, but they can also be detected in other examples. For instance, the hierarchical structure of the animal classification is partly driven by a desire not to have too many or too few animals in each class. We also see the power of tradition in the continued use of Mendeleev's periodic table, even though a classification based on quantum mechanics might better fit the scientific criteria. Another example of tradition can be seen in classification of star temperatures. The letters O, B, A, F, G, K, and M classify stars from hottest (O) to coldest (M). The choice of letters arises from an earlier, incorrect assignment, in which they were in alphabetic order, ABCDEF..., where A was the hottest¹⁸.

Sometimes, classifications are devised for specific purposes, e.g., education, particular usage, field observation in the absence of sophisticated instruments, etc. In these cases the reasons for preferring particular defining properties will also be informed by those specific purposes.

Thirdly, the types of the defining properties used will determine the structure of the classification. We have seen partitions, hierarchies, tables and concentric rings, for instance, and the evolution from one to another. [Kemp and Tenenbaum, 2008] describes some interesting work automating the form of a classification from the defining properties. Such a system might serve a useful role in classification evolution, i.e., by automatically constructing a new classification once its defining properties have been determined.

Fourthly, the evolution of a classification may also require an enrichment of the representation language. This can arise not only because of an increased complexity of the structure of the classification, e.g., from a mere naming of the elements to the periodic table. It can also arise from the need to express more complex defining properties. The defining properties used in the simple animal classification of Figure 1 are unary predicates, but Cladistics requires binary and ternary relations of descent and common ancestry. A classification based on DNA requires functions returning a complex structure of chromosomes, genes and double helices of nucleotides. Similar remarks can be made for the classification of the elements. The representation of the defining properties required in Astronomy requires complex geometric concepts, such as a sphere and an orbit and its neighbourhood. Note that an orbit, for instance, is naturally represented as a function from time to 3D position, which takes us into higher-order logic. Note that the representation language used to express the defining properties is frequently more sophisticated than that used to describe the classification itself. This suggests that ontology evolution will itself require a more sophisticated language than the one used to describe the ontology being evolved.

Fifthly, what drives the process of classification evolution? It may arise from within the classification, e.g., the failure to

¹⁸I'm indebted to an anonymous reviewer for this example.

classify a newly discovered object or to be useful for a new application. More likely, however, is that it will arise from an evolution of knowledge leading to better defining properties, i.e., properties that are more explanatory, fundamental, predictive and/or finer grained.

References

- [Bundy and McNeill, 2006] A. Bundy and F. McNeill. Representation as a fluent: An AI challenge for the next half century. *IEEE Intelligent Systems*, 21(3):85–87, 2006.
- [Kemp and Tenenbaum, 2008] C. Kemp and J.B. Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Sciences of the United States of America*, 105(31):10687–10692, August 2008.

Emergence in scientific literature and dynamics in ontologies: application in paleoanthropology

Christophe Jouis

Université Paris-Sorbonne Nouvelle & LIP6 (UPMC-CNRS),

Paris, France

Christophe.jouis@lip6.fr

Jean-Gabriel Ganascia

LIP6 (UPMC-CNRS),

Paris, France

Jean-gabriel.ganascia@lip6.fr

Franck Guy

Université de Poitiers - Faculté des Sciences - CNRS UMR 6046 - IPHEP,

Poitiers, France

Franck.guy@univ-poitiers.fr

Claude Jouis

Ecole Polytechnique

Palaiseau, France

jouis.claude@gmail.com

Abstract

In artificial intelligence, ontologies help to document a domain of knowledge with classes, instances and logical relations between classes. Usually, it might be considered as classes organized in connected graphs of which arrows correspond to semantic relations. Whenever these graphs are fixed, ontologies are static. In paleontology, systematics, that is the inventory and the organization of the diversity of species over time, is complicated by the evolution: the graph of relations may no longer be static; it has to take into account the time variable. The morphological changes lead, by natural selection, to the emergence of new morphs (taxa) differing from the ancestral and contemporaneous species (taxa) of the same class hierarchy. Therefore, discovering new taxa implies the rearrangement of the class hierarchy or the definition of new classes, based on the degree of atypicality of new morphs. Note that this phenomenon occurs in many domains such as physics, biology, linguistics, for example.

1 Introduction

In artificial intelligence, an ontology is a structured set of words that represent concepts of a domain of knowledge. The ontology itself is “a model” corresponding to a set of concepts (or classes) in a field, and the relations between these concepts. It is used to reason about items and concepts. Concepts are organized in a graph the relations of which may be semantic, e.g. topological, or simply logical, for instance hyponymy. The primary objective of an ontology is to model a body of knowledge in a given field. Usual-

ly, an ontology is static in its classes, because classes and class hierarchy are fixed. At the class level, it describes a state of the art in a field, that is to say, situations where neither the beginning nor the end are considered. There is no consideration of changes over time (events, processes). Transitions concern only instances of classes (individual entities) that will emerge, fade or see their properties change.

Besides, *emergence* is a general phenomenon that is found in many fields such as physics, biology, paleontology or linguistics and that many scientists attempt to model with dynamical systems. In this paper, we define emergence as the appearance of new properties in a subset of a class. From an existing class, these new properties allow to derive a new class that may be *typical* or a sub-class that may be *atypical*.

1.1 Typicality and Atypicality

We consider that an individual entity is *typical* of a class if it checks all the properties of this class. Similarly, we can consider that a class is a *typical* subclass of another class if all its typical elements check the properties of the more general class. We consider a sub-class as *atypical* if all its elements don't own all the properties of its super-ordinate classes, i.e. of its hypernyms. To specify whether a subclass is atypical or not, we borrow topological concepts of interior, border, closure and outside. Then, we may represent atypical entities in ontologies, by defining a system of relations of inclusion and belonging that are inspired from standard topological operators. In previous publications, the authors formalized the topological relations of inclusion and belonging by using the mathematical properties of topological operators. Those properties are defined as combinations of operators like interior, border, closure and outside, which

allow the definition of an algebra. These mathematical properties are seen as a set of axioms that enables us to establish the properties of inclusion and belonging relationship. A model is implemented in AnsProlog [Baral, 2003], a recent logic programming language that allows in particular the use of negative predicates in the inference rules.

1.2 taxa

In zoology, biological entities or taxa (i.e. sets of organisms established on the basis of the common morphological characters they share – and that come from a common ancestor – that is to say classes) are referenced in hierarchical class system, each class being defined by a set of morphological characters common to all the individuals that belong to it. The zoological categories include therefore individuals that are alike and, from an evolutionary perspective, they infer the phylogenetic relationship between these individuals (the emergence of characters). However, a taxon not always exhibits all the morphological criteria of a given class hierarchy. For example, in mammals (taxon Class), primates (ordinal taxon) are defined by a set of characters carried by soft tissues and mineralized tissues (unguiculate, pectoral breasts, pendulous penis, opposable thumb, orbits with bar and post-orbital septum), but all primates do not have all the characteristics of the group and there are no characters shared by all the primates.

1.3 Paleoanthropology

In paleoanthropology, the assignment of new fossil taxa to a zoological class is complicated by pre-existing associations of morphological characters often unprecedented and by taking into account the time variable, i.e. the modification of characters over time and the emergence of new characters in the class hierarchy. Therefore, systematics incorporates both a field of static data (morphological variation among taxa) and a field of dynamic data (character change over time) when establishing a coherent classification that reflects kinship between taxa. Hierarchies of class are established upon the principle of monophyly: a monophyletic (natural) group includes the ancestral morph and all its descendants.

Morphological change over time through the establishment of original set of characters induces the appearance of new taxa distinct from the ancestral form and from contemporary taxa of the same class hierarchy. Therefore, the discovery of these new taxa implies, redefining the hierarchy of classes or the invention of new classes, the rank of which is inferred by the degree of atypicality (i.e., the sum of the newly observed morphological changes) of specimens compared to reference morphs. The resulting class hierarchy becomes the standard to resolve the inclusion of new atypical specimens. In this context, the resolution of atypical cases is a major issue in evolutionary science.

2 Combined use of a non-monotonic logics and general topology to represent a scientific or technical text

2.1. Non-monotonic logics

One of the main problems that the logic-based artificial intelligence has to face is to conciliate the specificity of singular cases with general rules. This problem has been recognized as crucial for many years now and many logicians spent a lot of time to solve it. Depending on the domain of application, it has got different names: “frame problem”, “common sense reasoning”, etc. For the sake of clarity, let us give a classical example illustrating the problem. Usually birds fly. One would like to take advantage of such a general knowledge by having a rule that can be easily translated in first order logic “ $\forall X$ if bird(X) then fly(X)”. But, if a particular bird, named Tweety for instance, is an ostrich, it may not fly: it would contradict the general rule stating that birds fly. It would then be suitable to modify the initial rule and to write: “ $\forall X$ if bird(X) & not ostrich(X) then fly(X)”. Nevertheless, this rule should also be rewritten for penguins and other non-flying birds. As a result, we have both a lengthy rule (translating a very common knowledge) and an inefficient rule, since, to be triggered, the computer has to check that the bird under investigation is neither an ostrich, nor a penguin, etc.

One of possible solutions that is envisaged to solve this problem, is to have rules saying, for instance, that “ $\forall X$ if bird(X) & not inconsistent fly(X) then fly(X)”. The formalisms that may support such rules, may not be reduced to first order predicate logic, since predicate “inconsistent” is clearly a second-order predicate, i.e. a predicate the arguments of which are predicates. Classical mechanical proof procedures used in artificial intelligence, for instance Robinson resolution rule, are not appropriate to deal with such rules. Moreover, usual properties of formal systems are not conserved. For instance, they are not monotonic: in the past, many Artificial Intelligence researchers tried to simulate non-monotonic reasoning, i.e. reasoning based on general rules and accepting exceptions. Several formalisms have been developed, for instance, default logic [Reiter, 1980]. However, mechanical solvers based on those formalisms were quite inefficient.

2.2. Answer Set Programming

Recently, a new general formalism called Answer Set Programming (ASP) [Baral, 2003] has been developed to simulate non-monotonic reasoning. It has been designed to unify previous non-monotonic reasoning formalisms. ASP formalism is also fully operational. More precisely, ASP proposes both a clear formalization with a well define semantics and efficient operational solvers, which renders automate demonstrations possible.

Within this formalization, it is possible to specify logical properties of objects with programs Π that are sets of expressions p of the following form:

$$\rho : L_0 \text{ or } L_1 \text{ or } \dots L_k \leftarrow L_{k+1}, L_{k+2}, \dots L_m, \text{not } L_{m+1}, \dots, \text{not } L_n$$

where L_i are literals, i.e. atoms or atom negations, and “not” is a logical connector called “negation as failure”. The intuitive meaning of such a rule is that, for all Herbrand interpretations that render true all literals in $\{L_{k+1}, L_{k+2}, \dots, L_m\}$ while not satisfying any literals in $\{L_{m+1}, \dots, L_n\}$, one can derive at least one literal in $\{L_0, L_1, \dots, L_k\}$. More precisely, a close model of the rule ρ where not $k=0$ and $L_0 \neq \perp$ is a Herbrand interpretation S such that:

if $\{L_{k+1}, L_{k+2}, \dots, L_m\} \subset S$ and $\{L_{m+1}, \dots, L_n\} \cap S = \{\}$ then $\{L_0, L_1, \dots, L_k\} \neq \{\}$. A *stable model* of a program Π is a minimal close model of all the rules ρ belonging to Π .

Let us first remark that ASP formalism contains two negations that need to be distinguished: a classical negation noted “ $\neg$ ” and a negation by failure noted “not”, which means that a literal may not be proved in the absence of sufficient information. As we shall see, the non-monotonic properties are mainly due to this “negation as failure” connector.

The second remark concerns the strict declarative character of this formalism: the order in which are given the different rules and the order in which are written the literals in rules do not influence the results. In this respect, it has to be distinguished from classical logic programming techniques, results of which largely depend on the order in which clauses are written. Moreover, there are neither “cut” as in PROLOG, nor any procedural predicate in programs.

Being given a program Π , an Answer Set (or a stable model) is a minimal subset of the Herbrand base of Π , which satisfies all rules of Π . Each subset describes a possible world that renders true the rules of Π . Let us note that this intuitive meaning of the programs may be easily formalized, which provides a formal semantics of ASP.

For instance, let us consider the following rules:

```
fly(x) ← bird(x), not ¬fly(x)
¬fly(x) ← ostrich(x), not ab_ostrich(x)
bird(rocky) ←
bird(tweety) ←
ostrich(tweety) ←
```

There is an unique answer set for this program. $S = \{\text{fly(rocky)}, \neg\text{fly(tweety)}, \text{bird(rocky)}, \text{bird(tweety)}, \text{ostrich(tweety)}\}$

Let us now suppose that *Tweety* is in a plane; we can add the three following facts that translate that *Tweety* is in a plane, that one flies when, for an ostrich, it is an abnormal situation to be in a plane:

```
in_plane(tweety) ←
fly(x) ← in_plane(x)
ab_ostrich(x) ← in_plane(x)
```

Then, the only answer set is: $S = \{\text{fly(rocky)}, \text{fly(tweety)}, \text{bird(rocky)}, \text{bird(tweety)}, \text{ostrich(tweety)}, \text{in_plane(tweety)}, \text{ab_ostrich(tweety)}\}$

Without going into technical details, efficient solvers able to compute all the stable models for a given program act in two steps. At the first step, the program variables are grounded, i.e. they are instantiated by all Herbrand universe terms;

through the second step, a “sat” solver generates all the interpretations that satisfy the instantiated rules.

The principal advantage of such formalism is that it renders possible the expression of default rules by the use of the “negation as failure”.

Moreover, ASP formalism has a clear semantics, i.e. a well defined mathematical meaning, and there exist solvers that automate the computation of Answer Sets¹. It is then possible to get specific formalizations and efficient simulations that fully validate the formalizations, because they generate correct mathematical proofs.

2.3. Why using general topology?

Literally, topology means the study of the area and defines the nature of an area (also called a space E) and its properties. We postulate that networks of concepts and semantic relations between concepts may be represented on a plan. Instances are points of the plan, while classes are demarcated areas of the plan, which consist of: (a) an interior (the typical elements belonging to the class), (b) an exterior (the elements that are not in the class), (c) a border (atypical elements that do not check all the properties of the class, i.e. atypical elements that are neither within nor outside the class).

2.4. Integration of non-monotonic logics

Non-monotonic logics may be seen as an illustration of the general topology described above, and may be used to represent exceptions in ontologies. Given an element X and an ASP expression C (or class, that is to say a part of E), we have the following assertions:

- If $C(X)$ is true, unless otherwise indicated, then this corresponds in topology to say that X is at the interior of C . From the perspective of ontology, it is also to say that X is a typical element of C .
- If $C(X)$ is false, unless otherwise indicated, then this corresponds in topology to say that X is at the exterior of C . From the perspective of ontology, it is also to say that X is an unrelated C .
- Finally, if the $C(X)$ is “partially” true, that is to say if it does not check all the properties of C , then this corresponds in topology to say that X is on the border C . From the perspective of ontology, it is also equivalent to say that X is an atypical C .

We have the same three equivalences with classes and typical or atypical subclasses.

3. Paleoanthropology: A brief introduction to human evolution

The human family (i.e., the hominin), including modern humans and their fossil relatives, emerges from a group of African apes. The dichotomy, from a common ancestor, of the two lineages: one leading to the modern chimpanzee

¹ The interested reader can visit the URL <http://www.tcs.hut.fi/Software/smodels> to download an efficient ASP solver.

(panin) and one leading to our species (*Homo sapiens*) occurs, at least, 7 million years ago (Late Miocene); this age corresponding to the oldest evidence of an African hominin representative.

The first steps of the evolutionary history of our family take place in Africa and remain limited to the African continent from 7 to about 1.8 My. Numerous hominin (contemporaneous and successive) species are known from this period. The genus *Homo* emerges a short time before 2 My, exhibiting a series of major morphological novelties along with new practices and strategies (e.g., tool manufacture). Unfortunately, the late Miocene African sites are still uncommon and irregularly distributed. Besides, the fossil remains that support the paleoanthropological studies, belong to various individuals, from spatially (east, central and south Africa) and chronologically (several million years) distant sites. This situation complicates the interpretation of the evolutionary processes and pathways within our family.

The paleontology describes, analyzes and interprets fossil remains (bone material, environmental indicators, stone tools) on which are based the evolutionary hypothesis. One of its objectives is to reveal common characteristics (e.g., illustrating common ancestry) among diverse and disparate fossil materials, i.e. representative of individuals that exhibit different ages, sexes, geographical and temporal origins. Basically, individuals that show common morphological features are gathered within a same class defined from a reference morph called holotype. The holotype represents a morphological type of a zoological class (usually species). To be integrated in a class, the individuals must exhibit most of the features of the class-holotype. Atypical individuals that do not correspond to any given type, may become themselves a reference for a new class. The zoological classes are then organized in a hierarchy based on the displayed morphological features within each class and their evolutionary significance.

We present here the morphological features [Maclatchy et al., 2010] – a group of innovations or characters inherited from the ancestor – as literals (in the sense of predicate logics):

- I10 Body size
- I11 Body weight
- I12 Endocranial volume relative to body mass
- I13 Bipedality
- I14 Teeth morphology and relative teeth size
- I15 Occlusal morphology, enamel thickness and molar wear patterns
- I16 Skeletal morphology linked to bipedal locomotion;
- ...
- I20 Arboreal locomotion
- I21 Diet
- I22 Cranial vault thickness
- I23 Post-orbital morphology
- I24 Cranial vault profile
- I25 Prognathism

The literals listed above document characters or group of characters which are classically used in paleoanthropology. Of course, they provide only a partial illustration of the

complete set of criteria used for defining and classifying zoological classes. Most of them admit a normal variation within and between classes and they are presented only as an example.

For instance, this set of morphological characters is coded below for the fossil hominin *Australopithecus afarensis*:

- I10 Average size, close to a chimpanzee
- I11 Average body mass, close to a chimpanzee
- I12 Endocranial volume increased, relative size reduced
- I13 Bipedal at least occasional
- I14 Reduced anterior dentition. Honing complex lost
- I15 Thicker enamel
- I16 Hind limbs adapted to bipedality
- I20 Forelimbs retain arboreal features
- I21 Canine reduced. Hard food is consumed.
- I22 Mosaic pattern, primitive characters retained
- I25 Variable, usually prognathism expressed.

New discoveries periodically call into question evolutionary hypothesis formulated in hominin evolution. In fact, morphological novelties (in its evolutionary sense) generate atypical individuals; when discovered, these individuals imply the reinterpretation of the species relationships within our family and might require the invention of original classes within the hierarchy defining our group. In zoology, the class hierarchy is based on homogenous systems: the taxa. These biological classes gather individuals which present a comparable set of characters inherited from a common ancestor (taxonomy). To understand the evolutionary relationships (phylogeny) between taxa (and then establish a coherent hierarchy), paleontologists sort the morphological features following two categories: (1) the primitive elements are present in the ancestor morphology and remain unchanged in the descendant(s); (2) the derived elements correspond to morphological novelties in the descendant morphology relative to the ancestor state. The combination of topology and non-monotonic logics allows assessing both taxonomic and phylogenetic processes in order to facilitate the interpretation of atypical individuals in an objective manner.

4. Representation of Atypical Entities in our model: Implementation in AnsProlog*

4.1. Representation of a taxon in AnsProlog*

For simplicity, AnsProlog* is a program that interprets expressions in the formalism of ASP. In AnsProlog*, one way among others to describe a taxon is to directly translate the taxon as a rule. For example, the taxon (or the class) *Australopithecus afarensis* is translated directly as a rule below. We have added a set of facts about an object named Joe, and verify all properties of the rule:

```
#domain object (0).
object (joe ; jean).
australopithecus_afarensis(0) :-
    average_body_size(0),
```

```
average_body_weight(0),
average_endocranial_volume(0),biped(0),
reduced_anterior_dentition(0),
thick_enamel(0),
skeletal_adaptations_to_bipedality(0),
not arboricolism(0),
not canine_reduction(0),
not derived_cranial_morphology(0),
not primitive_characters_retained(0),
not prognatism_variable(0).
```

By adding the following facts, we get that "Joe" is an *australopithecus_afarensis*:

```
average_body_size(joe).
average_body_weight(joe).
average_endocranial_volume(joe).
biped(joe).
reduced_anterior_dentition(joe).
thick_enamel(joe).
skeletal_adaptations_to_bipedality(joe).
```

i.e. there answer sets that satisfy all the preceding facts contain *australopithecus_afarensis(joe)*.

4.2. When topology, paleoanthropology and ASP meet defeasible logics

We note that the negative literals:

```
"not arboricolism(0),
not canines_reduction(0),
not derived_cranial_morphology(0),
not primitive_characters_retained(0),
not prognatism_variable(0)"
```

have been validated as true by default (no information: "negation as failure").

Now we can imagine that the paleontologist discovers, in a prehistoric site, Joe jaw. However, the paleontologist finds that Joe has the property "canine-reduction". In AnsProlog*, this corresponds to adding the new fact:

```
canine_reduction(joe).
```

But the literal:

```
not canine_reduction(joe)
```

becomes false. Hence the rule:

```
australopithecus_afarensis(0):-.... is no longer
verified. Accordingly, AnsProlog* can no longer deduct the
new fact:
```

```
australopithecus_afarensis(joe).
```

Thus, we have a revisable reasoning. The object "Joe" is no longer classified as class *australopithecus_afarensis*. The paleontologist must invent a new taxon to classify "Joe." Note the new taxon *A_nov_taxon*, which has the properties defined by the rule:

```
a_nov_taxon(0):-
average_body_size(0),
average_body_weight(0),
average_endocranial_volume(0),biped(0),
reduced_anterior_dentition(0),
thick_enamel(0),
skeletal_adaptations_to_bipedality(0),
not arboricolism(0),
canine_reduction(0),
not derived_cranial_morphology(0),
```

```
not primitive_characters_retained(0),
not prognatism_variable(0).
```

AnsProlog may deduce a new fact:

```
a_nov_taxon(joe).
```

These procedures parallelize methods used in paleoanthropology. The description of a fossil taxon is based, by nature, upon incomplete data such as cranial fragments, post-cranial elements, etc. Missing data might be inferred but usually remain unresolved until the discoveries of new materials. The identification of a fossil specimen and its allocation to a pre-existing or a new class constitute a first step of the paleontological work. This "taxonomic" work is the basis of the phylogenetic analyses which aim organizing taxa in natural groups, i.e. establishing ancestor / descendant relationships and defining kinship (sister groups).

5. Conclusion

We aim to present a brief evaluation of an application of Topology and Non-Monotonic logics to practical issues in Paleoanthropology. The implementation in AnsProlog of typicality/atypicality concepts based on zoological groups and morphological criteria leads to propose new solutions for resolving topologic trees in a systematics framework. This first step assessment of the method may be tested and extended in several ways including: (1) a formalization of the concept of typicality scale as a base for a "neo-topology" with boundary thicknesses and (2) a larger and a more detailed set of morphological characters, than introduced into our presentation, in order to better assess our method in a complex and exhaustive zoological framework.

References

- [Baral, C. 2003]. C. Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press.
- [Brunet et al., 2005] Brunet M., Guy F. et al. 2005. "A new hominid from the Upper Miocene of Chad, Central Africa". *Nature* 418: 145-151.
- [Jouis, C. et al, 2008] Jouis, C. & Bourdaillet, "Exceptions in Ontologies: when Topology meets Typicality". In *Proceeding of the 21st International Florida Artificial Intelligence Research Society Conference, (FLAIRS-21)*, Coconut Grove, Florida, USA, May 14-16, 2008, published by the AAAI.
- [Kuratowski, C., 1958] Kuratowski, C. *Topologie*. 2 vol. Panstwowe Wydawnie two Naukowe, Varsovie,
- [Maclatchy L.M., 2010] Maclatchy L.M., Desilva J., Sanders W. J. & Wood, B. "Hominini". In L. Werdelin & W. J. Sanders (eds), *Cenozoic Mammals of Africa*. University of California Press, Berkeley. p. 471-540.
- [Reiter, R., 1980] Reiter, R. 1980. "A logic for default reasoning". *Artificial Intelligence*, 13, p. 81-132.

Contextual Constructive Description Logics

Valeria de Paiva¹ and Natasha Alechina²

¹ Rearden Commerce, CA, valeria.depaiva@gmail.com

² University of Nottingham, nza@cs.nott.ac.uk

Abstract

Constructive modal logics come in several different flavours and constructive *description* logics, while much more recent and less studied, not surprisingly, do the same. After all, it is a well-known result of Schild that description logics are simply variants of n -ary basic modal logic. There are several extensions of classical description logics, with modalities, temporal assertions, etc. As far as we know there are no such extensions for constructive description logics. Hence this note is a formal description of the extension of a constructive description logic $cALC$ [MS08] with contexts as modalities, as described in [deP03a], following the blueprint of [WZ99].

Keywords: constructive logic, description logic, contexts

1 Introduction

Description Logics are a knowledge representation formalism, much used in Artificial Intelligence. They are logic-based formalisms intended for representing knowledge about concept hierarchies, supplied with effective reasoning procedures and a declarative semantics. Description logics are very popular nowadays, perhaps due to their proposed applications in the Semantic Web. Most uses of description logics consider classical systems. However, considering versions of *constructive* description logics makes sense, both from a theoretical and from a practical viewpoint, as discussed in [deP03].

Description logics tend to be bundled in families of logical systems, depending on which concept *constructors* you allow in the logic. Since description logics came into existence as fragments of first-order logic, chosen to find the best trade-off possible between expressiveness and tractability of the fragment, several systems were discussed and eventually a taxonomy of systems emerged. In this taxonomy, the system called ALC (for *Attributive Language with Complements*) has come to be known as the canonical basic one. As far as *constructive* description logics are concerned, Mendler and Scheele have worked out a very compelling system, which they call $cALC$ ([MS08], based on the constructive modal logic CK[BPR01]). A different constructive version of ALC, based on the framework for constructive modal logics developed by Simpson in

his phd thesis [Sim95] was developed by Hausler, Rademaker and de Paiva. Their system, called *iALC* for *Intuitionistic ALC*, was described in [HRP10] and it is the reduced version of Braüner and de Paiva's system of *Intuitionistic Hybrid logic*, IHL [BdP06]. The systems $cALC$ and *iALC* are alternative formalizations of constructive description logics, and the main difference between these systems is whether they satisfy (or not) distribution of possibility over disjunction.

In this note, we start by recalling the description logic $cALC$. We then consider one extra modality on top of $cALC$, following the blueprint of [WZ99], and prove decidability of the resulting system. In previous work [deP03a] one of us suggested the use of constructive modalities as MacCarthy-style contexts in AI. On that work, the application envisaged was constructive modalities for formalising natural language 'microtheories' and it was regretted that the system obtained only described constructive modalities over a propositional basis. The work in this paper gets closer to the desired ultimate system, as we can now talk about context as a constructive modality over a constructive description logic basis. But most of the hard work is still to be done, as it concerns the interactions of the finite (but very large) collection of linguistics based contexts/modalities. Here we briefly discuss this as future work.

2 Constructive description logic $cALC$

The basic building blocks of description logics are *concepts*, *roles* and *individuals*. We think of concepts as unary predicates in usual first-order logic and of roles as binary predicates, used to modify the concepts. Like classical ALC [DL03] the intuitionistic version $cALC$ is a basic description language whose concept constructors are described by the following grammar:

$$C, D ::= A \mid \perp \mid \top \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid C \sqsubseteq D \mid \\ \exists R.C \mid \forall R.C$$

where A stands for an atomic concept and R for an atomic role. Note however that as in [MS08] we treat $\sqsubseteq$ as a concept forming operator, unlike in ALC. In the classical setting, $C \sqsubseteq D$ would have been definable as $\neg C \sqcup D$, but we are in the constructive setting and connectives are not interdefinable.

Following Mendler and Scheele we say a constructive interpretation of $cALC$ is a structure $\mathcal{I} = (\Delta^{\mathcal{I}}, \preceq^{\mathcal{I}}, \perp^{\mathcal{I}}, \cdot^{\mathcal{I}})$

consisting of a non-empty set $\Delta^{\mathcal{I}}$ of entities in which each entity represents a partially defined individual; a refinement pre-ordering $\preceq^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, i.e., a reflexive and transitive relation; $\perp^{\mathcal{I}}$ is a subset of fallible entities satisfying $\perp$ (fallible entities are over-defined and hence self-contradictory. This set is closed under refinement, that is, $x \in \perp^{\mathcal{I}}$ and $x \preceq^{\mathcal{I}} y$ implies $y \in \perp^{\mathcal{I}}$); and an interpretation function $\cdot^{\mathcal{I}}$ mapping each role name R to a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and each atomic concept A to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ which is closed under refinement, i.e., $x \in A^{\mathcal{I}}$ and $x \preceq^{\mathcal{I}} y$ implies $y \in A^{\mathcal{I}}$. The interpretation $\mathcal{I}$ is lifted from atomic $\perp, A$ to arbitrary concepts, where $\Delta_c^{\mathcal{I}} =_{df} \Delta^{\mathcal{I}} \setminus \perp^{\mathcal{I}}$ is the set of non-fallible elements, via:

$$\begin{aligned}
\top^{\mathcal{I}} &=_{df} \Delta^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &=_{df} \{x \mid \forall y \in \Delta_c^{\mathcal{I}}. x \preceq y \Rightarrow y \notin C^{\mathcal{I}}\} \\
(C \sqcap D)^{\mathcal{I}} &=_{df} C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &=_{df} C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(C \sqsubseteq D)^{\mathcal{I}} &=_{df} \{x \mid \forall y \in \Delta_c^{\mathcal{I}}. (x \preceq y \text{ and } y \in C^{\mathcal{I}}) \Rightarrow y \in D^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &=_{df} \{x \mid \forall y \in \Delta_c^{\mathcal{I}}. x \preceq y \Rightarrow \\
&\quad \exists z \in \Delta^{\mathcal{I}}. (y, z) \in R^{\mathcal{I}} \text{ and } z \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &=_{df} \{x \mid \forall y \in \Delta_c^{\mathcal{I}}. x \preceq y \Rightarrow \\
&\quad \forall z \in \Delta^{\mathcal{I}}. (y, z) \in R^{\mathcal{I}} \Rightarrow z \in C^{\mathcal{I}}\}
\end{aligned}$$

Semantic validity can be introduced as follows: say “ x satisfies C in the interpretation $\mathcal{I}$ ”, written as $\mathcal{I}, x \models C$, if x is in the interpretation of C , $x \in C^{\mathcal{I}}$. Say $\mathcal{I} \models C$ if this happens for all x in $\Delta^{\mathcal{I}}$. Finally say $\models C$ if for all interpretations $\mathcal{I}$ we have $\mathcal{I} \models C$. These definitions are usually extended to sets of concepts.

Typical reasoning in description logics is done via TBoxes and ABoxes. If we use Θ for a TBox, i.e., a collection of concepts and subsumptions¹ and Γ for an ABox, a collection of instantiations of concepts then we can say $\Theta, \Gamma \models C$ if for all interpretations $\mathcal{I}$, which are models of all the concepts in Θ it is the case that every x in $\mathcal{I}$ which satisfy the axioms in Γ must also satisfy C , or

$$\forall \mathcal{I}. \forall x \in \Delta^{\mathcal{I}}. (\mathcal{I} \models \Theta \text{ and } \mathcal{I}, x \models \Gamma) \text{ implies } \mathcal{I}, x \models C$$

Note that if we only consider TBox reasoning, that is if the ABox is empty, the definition above gives $\Theta, \emptyset \models C$ iff

$$\forall \mathcal{I}. \forall x \in \Delta^{\mathcal{I}}. \mathcal{I} \models \Theta \text{ implies } \mathcal{I}, x \models C$$

A Hilbert-style axiomatization of TBox reasoning in $c\mathcal{ALC}$ consists of the axioms and rules given in Figure 1. We denote derivability in this calculus as $\vdash_H$.

Mendler and Scheele([MS08] p.7) proved:

Theorem 1 (Mendler-Scheele). *The Hilbert calculus given in Figure 1 is sound and complete for TBox reasoning, that is $\Theta, \emptyset \models C$ if and only if $\Theta \vdash_H C$.*

¹This is a somewhat non-traditional usage, since usually a TBox would contain subsumptions, rather than concepts. However recall that following [MS08], we express subsumptions as concepts.

$$\begin{aligned}
(\sqsubseteq_1) \quad & C \sqsubseteq (D \sqsubseteq C) \\
(\sqsubseteq_2) \quad & (C \sqsubseteq (D \sqsubseteq E)) \sqsubseteq ((C \sqsubseteq D) \sqsubseteq (C \sqsubseteq E)) \\
(\sqcap_1) \quad & C \sqcap D \sqsubseteq C \\
(\sqcap_2) \quad & C \sqcap D \sqsubseteq D \\
(\sqcap_3) \quad & C \sqsubseteq (D \sqsubseteq (C \sqcap D)) \\
(\sqcup_1) \quad & C \sqsubseteq C \sqcup D \\
(\sqcup_2) \quad & D \sqsubseteq C \sqcup D \\
(\sqcup_3) \quad & (C \sqsubseteq E) \sqsubseteq ((D \sqsubseteq E) \sqsubseteq (C \sqcup D \sqsubseteq E)) \\
(\perp) \quad & \perp \sqsubseteq C \\
(\forall K) \quad & \forall R. (C \sqsubseteq D) \sqsubseteq (\forall R.C \sqsubseteq \forall R.D) \\
(\exists K) \quad & \forall R. (C \sqsubseteq D) \sqsubseteq (\exists R.C \sqsubseteq \exists R.D) \\
(Nec) \quad & \text{If } \vdash_H C \text{ then } \vdash_H \forall R.C \\
(MP) \quad & \text{If } \vdash_H C \text{ and } \vdash_H C \sqsubseteq D, \text{ then } \vdash_H D
\end{aligned}$$

Figure 1: The System $c\mathcal{ALC}$: Hilbert-style

A sequent calculus version of $c\mathcal{ALC}$ is given by Mendler and Scheele, who also prove cut-elimination for their calculus. But their calculus is a tableau style calculus with positive and negative information about concepts and a simpler one is available, see Figure 2.

Note that our version, which is constructive, has restrictions to a single conclusion formula in the rules for subsumption and universal-quantification-role on the right, which are essential to keep the system intuitionistic. It is reassuring to see the same rules for roles in Straccia’s 4-valued Description Logic [Str97]. The rules for the propositional connectives ($\sqcap, \sqcup$) are basically the same as for classical $\mathcal{ALC}$, and the rules for subsumption $\sqsubseteq$ are just the rules for intuitionistic implication.

The system $c\mathcal{ALC}$ [MS08] is related to constructive CK ([BPR01] and [MdP05]) in the same way classical multimodal K is related to $\mathcal{ALC}$ [Sch91]. In the system $c\mathcal{ALC}$, the classical principles of the excluded middle $C \sqcup \neg C = \top$, double negation elimination $\neg\neg C = C$ and the definitions of the modalities $\exists R.C = \neg\forall R.\neg C$ and $\forall R.C = \neg\exists R.\neg C$ are no longer tautologies, but simply non-trivial TBox statements used to axiomatize specific application scenarios.

Soundness and completeness of a sequent calculus version of $c\mathcal{ALC}$ is indicated in page 10 of [MS08], although not exactly for the sequent calculus we proposed in Figure 2. Our sequents are simpler than theirs, as we do not insist in carrying negative information along derivations, as they do. Nonetheless we have:

Theorem 2. *The sequent calculus for $c\mathcal{ALC}$ in Figure 2 and the Hilbert calculus described in Figure 1 are equivalent. For any TBox Θ and concept C , we have that $\Theta, \emptyset \vdash_H C$ if and only if the sequent $\Theta \Rightarrow C$ has a derivation using the rules in Figure 2.*

The proof of soundness and completeness of the sequent calculus for $c\mathcal{ALC}$ does not come straight from Straccia’s work, as our rules for roles are the same, but our semantics

$\frac{}{\Gamma, C \Rightarrow C, \Delta}$ $\frac{\Gamma \Rightarrow C, \Delta \quad \Gamma, D \Rightarrow \Delta}{\Gamma, C \sqsubseteq D \Rightarrow \Delta} \sqsubseteq\text{-l}$ $\frac{\Gamma, C, D \Rightarrow \Delta}{\Gamma, (C \sqcap D) \Rightarrow \Delta} \sqcap\text{-l}$ $\frac{\Gamma, C \Rightarrow \Delta \quad \Gamma, D \Rightarrow \Delta}{\Gamma, (C \sqcup D) \Rightarrow \Delta} \sqcup\text{-l}$ $\frac{\Gamma, C \Rightarrow \Delta}{\Gamma, \forall R.C \Rightarrow \Delta} \forall\text{-l}$ $\frac{\Gamma, C \Rightarrow \Delta}{\Gamma, \exists R.C \Rightarrow \Delta} \exists\text{-l}$	$\frac{}{\Gamma, \perp \Rightarrow C, \Delta}$ $\frac{\Gamma, C \Rightarrow D}{\Gamma \Rightarrow C \sqsubseteq D} \sqsubseteq\text{-r}$ $\frac{\Gamma \Rightarrow C, \Delta \quad \Gamma \Rightarrow D, \Delta}{\Gamma \Rightarrow (C \sqcap D), \Delta} \sqcap\text{-r}$ $\frac{\Gamma \Rightarrow C, D, \Delta}{\Gamma \Rightarrow (C \sqcup D), \Delta} \sqcup\text{-r}$ $\frac{\Gamma \Rightarrow C}{\Gamma \Rightarrow \forall R.C} \forall\text{-r}$ $\frac{\Gamma \Rightarrow C, \Delta}{\Gamma \Rightarrow \exists R.C, \Delta} \exists\text{-r}$
--	--

Figure 2: The System $c\mathcal{ALC}$: Sequent calculus

are different. Straccia insists on 4-valuedness, we only want constructiveness.

Theorem 3. *The sequent calculus described in Figure 2 is sound and complete for TBox reasoning, that is $\Theta, \emptyset \models C$ if and only if $\Theta \vdash_S C$.*

3 Extending $c\mathcal{ALC}$

The main idea of the extension of $c\mathcal{ALC}$ with a constructive context operator $\square$ from the logic CK [MdP05] is similar to the approach of Wolter and Zakharyashev to modalizing classical description logics. Intuitively, we introduce a Kripke-style model to interpret the $\square$ where each possible world is a $c\mathcal{ALC}$ model. We treat $c\mathcal{ALC}$ formulas as atomic formulas of the extended logic $c\mathcal{ALC}_\square$.

If ϕ^{at} is a formula of $c\mathcal{ALC}$, then formulas of $c\mathcal{ALC}_\square$ are defined as follows:

$$\phi ::= \top \mid \perp \mid \phi^{at} \mid \square\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \rightarrow \psi$$

The logic CK (see for example [MdP05]) is interpreted on models $(W, \leq, R, I)$ where W is a non-empty set of possible worlds, $\leq$ is a reflexive transitive binary relation on W , R is an arbitrary binary relation on W , and I is an interpretation function (p^I is a subset of W satisfying p). Inconsistent worlds are allowed, namely $\perp^I$ is not necessarily empty, so we have fallible worlds. The conditions on models are as follows:

- $\leq$ is hereditary with respect to atomic formulas, that is for every atomic p , if $w \in p^I$ and $w \leq w'$, then $w' \in p^I$. In particular, if $w \in \perp^I$ and $w \leq w'$, then $w' \in \perp^I$.

- if $w \in \perp^I$, then $w \in p^I$ for every atomic p .

A model of $c\mathcal{ALC}_\square$, $\mathcal{M}$, is a CK-model $(W, \leq, R, I)$ where in addition each W is a $c\mathcal{ALC}$ model and for every formula ϕ^{at} of $c\mathcal{ALC}$, $w \in (\phi^{at})^I$ iff $w \models \phi^{at}$.

Definition 1 (satisfaction in $\mathcal{M}$). *The relation “the $c\mathcal{ALC}_\square$ -model $\mathcal{M}$ and the world $w \in W$ satisfy a formula ϕ ” (in symbols $\mathcal{M}, w \models \phi$) is defined inductively as follows:*

$$\begin{aligned} \mathcal{M}, w \models \phi^{at} &\text{ iff } w \in (\phi^{at})^I \\ \mathcal{M}, w \models \phi \wedge \psi &\text{ iff } \mathcal{M}, w \models \phi \text{ and } \mathcal{M}, w \models \psi \\ \mathcal{M}, w \models \phi \vee \psi &\text{ iff } \mathcal{M}, w \models \phi \text{ or } \mathcal{M}, w \models \psi \\ \mathcal{M}, w \models \phi \rightarrow \psi &\text{ iff for all } w' \text{ with } w \leq w' \text{ if } \mathcal{M}, w' \models \phi, \\ &\text{ then } \mathcal{M}, w' \models \psi \\ \mathcal{M}, w \models \square\phi &\text{ iff for all } w' \text{ with } w \leq w', \forall u (R(w', u) \Rightarrow \\ &\mathcal{M}, u \models \phi) \end{aligned}$$

Theorem 4. *Satisfiability for $c\mathcal{ALC}_\square$ is decidable.*

Proof sketch. The decision procedure builds on decidability results for $c\mathcal{ALC}$ [MS08] and CK [MdP05], respectively. We formulate the procedure as a non-deterministic algorithm (guess a model and check that it is indeed a satisfying model) but it can also be described as a deterministic algorithm working by exhaustive enumeration of all models of fixed bounded size.

Given a formula ϕ of $c\mathcal{ALC}_\square$, we first use the result from [MS08] to guess a bounded size CK model M for ϕ (ignoring the extra conditions for the $c\mathcal{ALC}_\square$ models). Note that for each world of M , by construction from [MdP05], only the interpretation of subformulas of ϕ in M matters. Let us call this set of subformulas $Sf(\phi)$. Note that $Sf(\phi)$ is finite. Now

we check, for each world w of M , whether the set $\{\phi^{at} : M, w \models \phi^{at} \text{ and } \phi^{at} \in Sf(\phi)\}$ has a $c\mathcal{ALC}$ model. By the result of [MS08], this is decidable. If every world in M has a corresponding $c\mathcal{ALC}$ model, we are done: we found a $c\mathcal{ALC}_\square$ model for ϕ . $\square$

Having obtained decidability for $c\mathcal{ALC}_\square$ we conjecture that the system can be extended with many non-interacting boxes, to provide a system $c\mathcal{ALC}_{ctx}$ for Artificial Intelligence contexts, the application to Natural Language semantics that we are after, which is briefly described below.

In a series of papers ([deP05],[Bal07],[Bal07a]) the PARC team has described a formalization of the semantics produced by their system **Bridge**, that automatically creates semantics for natural language sentences from the sentences themselves, using symbolic and statistical methods. The logic that emerges from their representations (which they called TIL for textual inference logic) is a description logic of concepts and contexts, determined by the lexical semantics of the words used. Their description corresponds intuitively to a system of constructive description logic, where the linguistic contexts are induced by intensional concepts, such as propositional attitude verbs and negations. For a trivial example consider the sentence *Ed knew that he closed the door*, in the logic TIL the representation of this sentence will have two contexts, one corresponding to the world of the things that Ed knew, one corresponding to how the author of the sentence conceives the real world to be. These linguistically induced contexts were also called ‘microtheories’, by analogy to CYC’s contexts, which correspond to first-order logic theories. The work in this paper started from a desire to give a firm logical footing to the system TIL.

However, the PARC team also did some work in the interactions amongst their linguistic contexts [NCK06]. The ultimate goal of the research reported in this paper is to formalize that latter work on linguistics contexts interaction. As discussed by Nairn et al these interactions are predictable, but not trivial to formalize. Much work remains to be done towards this goal and we are barely starting it. However, it must be noted that we do not envisage adding any work along the lines of the work of Klarman and Gutierrez [KG10]. Despite the superficial similarity between our paper and their work (both papers add contexts MacCarthy-style to the description logic $\mathcal{ALC}$), the resulting systems are very different. First because what we take as contexts MacCarthy-style is different and secondly because of our commitment to make our logic constructive, both in the basis and in the context structure over it. It is hoped that the constructivity of the logic will ‘pay’ for itself via the usefulness of the Curry-Howard correspondence, as argued, *inter alia*, by Mendler and Scheele [MS08].

4 Conclusions

We extended the constructive description logic $c\mathcal{ALC}$ with a modality box operator and proved that the resulting logic is decidable. This extension is motivated by a proposed application to modelling contexts in AI, as described, in the propositional setting, in [deP03a]. Much remains to be done, in particular we want to investigate the complexity price of con-

structivity in our setting and we also must check the adaptation needs of our application.

References

- [DL03] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, 2003.
- [BPR01] Bellin, G., de Paiva, V. and E. Ritter. *Extended Curry-Howard Correspondence for a Basic Constructive Modal Logic*. In Proc of Methods for the Modalities 2, 2001.
- [Bal07] Bobrow, D., C. Condoravdi, R. Crouch, L. Karttunen, T. H. King, R. Nairn and A. Zaenen, *Textual Inference Logic: Take Two* Proceedings of the Workshop on Contexts and Ontologies, Representation and Reasoning, CONTEXT 2007.
- [Bal07a] Bobrow, D. G., C. Condoravdi, L. Karttunen, T. H. King, L. Price, R. Nairn, L. Price, A. Zaenen. *Precision-focused Textual Inference*. Proceedings of ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 16-21, 2007.
- [Boz09] Bozzato, L., M. Ferrari, P. Villa *A note on constructive semantics for description logics*. Accepted at CILC09 - 24-esimo Convegno Italiano di Logica Computazionale.
- [BdP06] Bräuner, T. and de Paiva, V. *Intuitionistic hybrid logic*. J. Applied Logic (JAPLL) 4(3):231-255 (2006)
- [HRP10] Haeusler, E. H., Rademaker, A. and de Paiva, V. *Using Intuitionistic Logic as a basis for Legal Ontologies* in 4th Workshop on Legal Ontologies and Artificial Intelligence Techniques, 2010, Firenze. CEUR Workshop Proceedings (CEUR-WS.org), 2010. v. 605. p. 69-76.
- [HRP10b] Haeusler, E. H., Rademaker, A. and de Paiva, V. *Dealing with Conflict of Laws in Private International Law by Means of Intuitionistic Description Logic: A Sequent Calculus based case-study*. Short Paper in JURIX 2010. The 23rd International Conference on Legal Knowledge and Information Systems. University of Liverpool, UK, 16th-17th December 2010.
- [Hof05] Hofmann, M. *Proof-theoretic Approach to Description-Logic*. In Proc. of Logic in Computer Science (LICS-05), 229–237, 2005.
- [KG10] Klarman, S. and Gutiérrez-Basulto, V. *ALC_{ALC}: a context description logic*. Proceedings of the 12th European Conference on Logics in Artificial Intelligence (JELIA’10), Springer-Verlag Berlin, Heidelberg 2010.

- [MdP05] Michael Mendler, Valeria de Paiva. *Constructive CK for Contexts* Proc. of the Workshop on Context Representation and Reasoning, Paris, France, July 2005.
- [MS08] Michael Mendler, Stephan Scheele. *Towards Constructive DL for Abstraction and Refinement*. Description Logics 2008.
- [Neg05] Sara Negri. *Proof analysis in Modal Logic*. Journal of Philosophical Logic, vol 34, pp. 507–544, 2005.
- [NCK06] Nairn, R., C. Condoravdi and L. Karttunen. *Computing Relative Polarity for Textual Inference*. Inference in Computational Semantics 5 (ICoS-5), 2006.
- [deP03] de Paiva, V. *Constructive description logics: what, why and how*. Technical report, Xerox Parc, 2003.
- [deP03a] de Paiva, V. *Natural Deduction and Context as (Constructive) Modality*. Modeling and Using Context, Proceedings of the 4th International and Interdisciplinary Conference CONTEXT 2003, Stanford, CA, USA, Springer Lecture Notes in Artificial Intelligence, vol 2680, June 2003.
- [deP05] de Paiva, Bobrow, Condoravdi, Crouch, Kaplan, Karttunen, King and Zaenen, *A Basic Logic for Textual inference*, In Proceedings of the AAAI Workshop on Inference for Textual Question Answering, Pittsburgh PA, July 2005.
- [RHP09] Rademaker, A., Haeusler, E. H., Pereira, L. C. *On the Proof Theory of ALC*. In: Walter Carnielli; Marcelo Coniglio; Itala D’Ottaviano. (Org.). The Many Sides of Logic. London: College Publications, 2009, v. 21, p. 273-285.
- [Ran09] Ranalter, K. *Embedding Constructive K into Intuitionistic K* in Electr. Notes Theor. Comput. Sci. 262: 205-219 (2010).
- [Sch91] Schilds, K. *A correspondence theory for terminological logics: Preliminary report* Proc. of the 12th IJCAI, Sydney Australia, 1991.
- [Sim95] Simpson, A. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD Thesis, University of Edinburgh, December 1993, revised September 1994.
- [Str97] Straccia, U.A. *sequent calculus for reasoning in four-valued description logics* TABLEAUX 1997. LNAI 1227, Pont-a-Mousson, France, 1997.
- [WZ99] Wolter, F. and M. Zakharyashev. *Modal Description Logics: modalizing roles*. Fundamenta Informaticae. 39 (1999), pp.411-438.

Local-as-View Integration of Ontologies Through Argumentation*

†Sergio Alejandro Gómez and ‡Carlos Iván Chesñevar and †Guillermo Ricardo Simari

†Artificial Intelligence Research Lab, Dept. of Computer Science and Engineering,
Universidad Nacional del Sur, Argentina

‡Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina

Abstract

In this article we show how to perform local-as-view integration of possibly inconsistent and incomplete ontologies in terms of Defeasible Logic Programming (DeLP). We show how to reason on Description Logics (DL) ontologies in an ontology integration system by performing a dialectical analysis in order to determine the membership of individuals to a concepts. We present formal definitions of a framework for ontology integration of DL ontologies based on DeLP along with a case study and review some of the properties of the approach.

1 Introduction

The *Semantic Web* [Berners-Lee *et al.*, 2001] (SW) is a vision of the Web where agents can reason about resources whose meaning is assigned in terms of ontologies [Gruber, 1993]. The OWL language is the current standard for defining ontologies and its underlying semantics is based on *Description Logics* (DL) [Baader *et al.*, 2003], for which specialized reasoners exist [Haarslev and Möller, 2001]. *Description Logic Programming* (DLP) is an alternative approach to reason with DL ontologies that proposes translating them into the language of logic programming (LP) [Grosz *et al.*, 2003]. Although DLP offers several advantages in terms of efficiency and reuse of existing LP tools (such as Prolog environments), that approach is incapable of reasoning in the presence of inconsistent ontologies. In this regard we developed a framework called *δ -ontologies* [Gómez *et al.*, 2010] based on *Defeasible Logic Programming* (DeLP) [García and Simari, 2004], an approach which allows to assess defeasibly the membership of an individual to a concept description in the presence of potential inconsistent ontology.

The semantic integration of potentially inconsistent information sources in the SW is complicated because the knowledge engineer usually has no authority to correct foreign ontologies or even worse the domains modeled are inherently inconsistent. The resources and their data are modeled in

terms of ontologies whose terms can differ, so the ontologies must be aligned to put their terms into mutual agreement [Klein, 2001]. There are two kinds of ontology integration approaches, *viz.* global-as-view (GAV) and local-as-view (LAV) [Calvanese *et al.*, 2001]. When performing LAV integration, concepts of the local ontologies are mapped to queries over a global ontology.

In this article, we present an approach for modeling LAV ontology integration when the involved ontologies can be potentially inconsistent. The ontologies are expressed in the language of DL but their semantics is expressed in terms of DeLP. The alignments between the local and global ontologies are expressed as DL inclusion axioms that are also interpreted as DeLP sentences. As the ontologies are potentially inconsistent, a dialectical analysis is performed on the interpretation of both the ontologies and the mappings from the local to the global ontology.

Outline: In Section 2 we present the fundamentals of Defeasible Logic Programming along with a brief introduction to the δ -ontologies framework for reasoning with possibly inconsistent ontologies. In Section 3, we extend δ -ontologies for performing local-as-view integration. Finally Section 4 concludes the paper.

2 Knowledge Representation and Reasoning with δ -Ontologies

2.1 Fundamentals of Defeasible Logic Programming

Defeasible Logic Programming (DeLP) [García and Simari, 2004] provides a language for knowledge representation and reasoning that uses *defeasible argumentation* [Chesñevar *et al.*, 2000] to decide between contradictory conclusions through a *dialectical analysis*. In a DeLP program $\mathcal{P} = (\Pi, \Delta)$, a set Π of strict rules $P \leftarrow Q_1, \dots, Q_n$, and a set Δ of defeasible rules $P \multimap Q_1, \dots, Q_n$ can be distinguished. An *argument* $\langle \mathcal{A}, H \rangle$ is a minimal non-contradictory set of ground defeasible clauses $\mathcal{A}$ of Δ that allows to derive a ground literal H possibly using ground rules of Π . Since arguments may be in conflict (concept captured in terms of a logical contradiction), an attack relationship between arguments can be defined. Generalized specificity [Stolzenburg *et al.*, 2003] is one possible criterion used to decide between two conflicting arguments. If the attacking argument is strictly

*This research is funded by Projects PIP 112-200801-02798 (CONICET, Argentina), PGI 24/N006 (SGCyT, UNS, Argentina) and Universidad Nacional del Sur.

preferred over the attacked one, then it is called a *proper defeater*. If no comparison is possible, or both arguments are equi-preferred, the attacking argument is called a *blocking defeater*. In order to determine whether a given argument $\mathcal{A}$ is ultimately undefeated (or *warranted*), a dialectical process is recursively carried out, where defeaters for $\mathcal{A}$, defeaters for these defeaters, and so on, are taken into account. The answer to a query H w.r.t. a DeLP program $\mathcal{P}$ takes such dialectical analysis into account and can be one of *yes*, *no*, *undecided*, or *unknown*.

2.2 Reasoning with Inconsistent DL Ontologies in DeLP

In the presence of inconsistent ontologies, traditional DL reasoners (such as RACER [Haarslev and Möller, 2001]) issue an error message and stop further processing. Thus, the burden of repairing the ontology (*i.e.*, making it consistent) is on the knowledge engineer. For defining δ -ontologies in [Gómez *et al.*, 2010] we use the *SHOTQ* DL but (following [Grosz *et al.*, 2003]) distinguish three kinds of inclusion axioms for defining classes: (i) having no disjunctions in the right-hand side of inclusion statements (called $\mathcal{L}_b$ -classes), (ii) having no disjunctions in the left-hand side (known as $\mathcal{L}_h$ -classes), and (iii) having no disjunctions at all (called $\mathcal{L}_{hb}$ -classes).

Definition 2.1 Let C be an $\mathcal{L}_b$ -class, D an $\mathcal{L}_h$ -class, A, B $\mathcal{L}_{hb}$ -classes, P, Q properties, a, b individuals. Let T be a set of inclusion and equality sentences in $\mathcal{L}_{DL}$ of the form $C \sqsubseteq D$, $A \equiv B$, $\top \sqsubseteq \forall P.D$, $\top \sqsubseteq \forall P^-.D$, $P \sqsubseteq Q$, $P \equiv Q$, $P \equiv Q^-$, or $P^+ \sqsubseteq P$ such that T can be partitioned into two disjoint sets T_S and T_D . Let A be a set of assertions disjoint with T of the form $a : D$ or $\langle a, b \rangle : P$. A δ -ontology Σ is a tuple (T_S, T_D, A) . The set T_S is called the *strict terminology* (or *Sbox*), T_D the *defeasible terminology* (or *Dbox*) and A the *assertional box* (or *Abox*).

Example 2.1 Consider the δ -ontology $\Sigma_1 = (T_S^1, T_D^1, A^1)$ presented in Fig. 1. The strict terminology T_S^1 says every Highlander always carries a sword. The defeasible terminology T_D^1 expresses that usually a man is mortal unless he is a Highlander. The set A^1 asserts that it is known that Socrates is a man and McLeod is both a man and a Highlander. Notice that the traditional (in the sense of [Baader *et al.*, 2003]) DL ontology $(T_S^1 \cup T_D^1, A^1)$ is incoherent since somebody who is both a man and a Highlander then belongs both to the “Mortal” concept and to its complement, rendering the concept empty.

Strict terminology T_S^1:	Assertional box A^1:
Highlander $\sqsubseteq$ CarriesSword	SOCRATES : Man
Defeasible terminology T_D^1:	MCLEOD : Man
Man $\sqsubseteq$ Mortal	MCLEOD : Highlander
Man $\sqcap$ Highlander $\sqsubseteq$ $\neg$ Mortal	

Figure 1: Ontology $\Sigma_1 = (T_S^1, T_D^1, A^1)$

For assigning semantics to a δ -ontology two translation functions T_Δ and T_Π from DL to DeLP based on the work of [Grosz *et al.*, 2003] are defined. The basic premise for

achieving the translation of DL ontologies into DeLP is based on the observation that a DL inclusion axiom “ $C \sqsubseteq D$ ” is regarded as a First-Order Logic statement “ $(\forall x)(C(x) \rightarrow D(x))$ ”, which in turn is regarded as a Horn-clause “ $d(X) \leftarrow c(X)$ ”.¹ Naturally “ $C \sqcap D \sqsubseteq E$ ” is treated as “ $e(X) \leftarrow c(X), d(X)$ ”. Lloyd-Topor transformations are used to handle special cases as conjunctions in the head of rules and disjunctions in the body of rules; so “ $C \sqsubseteq D \sqcap E$ ” is interpreted as two rules “ $d(X) \leftarrow c(X)$ ” and “ $e(X) \leftarrow c(X)$ ” while “ $C \sqcup D \sqsubseteq E$ ” is transformed into “ $e(X) \leftarrow c(X)$ ” and “ $e(X) \leftarrow d(X)$ ”. Likewise axioms of the form “ $\exists r.C \sqsubseteq D$ ” are treated as “ $d(X) \leftarrow r(X, Y), c(Y)$ ”. Dbox axioms are treated as *defeasible* and are transformed using the T_Δ function (*e.g.*, $T_\Delta(C \sqsubseteq D)$ is interpreted as $d(X) \multimap c(X)$); Sbox axioms are considered *strict* and are transformed using T_Π (*e.g.*, $T_\Pi(C \sqsubseteq D)$ is interpreted as $\{(d(X) \leftarrow c(X)), (\sim c(X) \leftarrow \sim d(X))\}$).² Abox assertions are always considered strict (*e.g.*, $T_\Pi(a : C)$ is regarded as a fact $c(a)$ and $T_\Pi(\langle a, b \rangle : r)$ as $r(a, b)$). Formally:

Definition 2.2 Let $\Sigma = (T_S, T_D, A)$ be a δ -ontology. The interpretation of Σ is a DeLP program $\mathcal{P} = (T_\Pi(T_S) \cup T_\Pi(A), T_\Delta(T_D))$.

Notice that in order to keep consistency within an argument, some internal coherence between the Abox and the Tbox must be enforced; namely given a δ -ontology $\Sigma = (T_S, T_D, A)$, it must not be possible to derive two complementary literals from $T_\Pi(T_S) \cup T_\Pi(A)$. We recall how the reasoning task of *instance checking* [Baader *et al.*, 2003, p. 19] is interpreted in δ -ontologies:

Definition 2.3 Let $\Sigma = (T_S, T_D, A)$ be a δ -ontology, C a class name, a an individual, and $\mathcal{P} = (T_\Pi(T_S) \cup T_\Pi(A), T_\Delta(T_D))$.

1. The individual a potentially belongs to class C iff there exists an argument $\langle \mathcal{A}, C(a) \rangle$ w.r.t. $\mathcal{P}$;
2. the individual a justifiedly belongs to class C iff there exists a warranted argument $\langle \mathcal{A}, C(a) \rangle$ w.r.t. $\mathcal{P}$, and,
3. the individual a strictly belongs to class C iff there exists an argument $\langle \emptyset, C(a) \rangle$ w.r.t. $\mathcal{P}$.

Example 2.2 (Continues Ex. 2.1) Consider again the δ -ontology Σ_1 , which is interpreted as the DeLP program $\mathcal{P}_1$ according to Def. 2.2 as shown in Fig. 2. From $\mathcal{P}_1$, we can determine that Socrates justifiedly belongs to the concept “Mortal” in Σ_1 as there exists an undefeated (and thus warranted) argument structure $\langle \{mortal(socrates) \multimap man(socrates)\}, mortal(socrates) \rangle$. McLeod belongs potentially to the concept “Mortal” as he is also a man, but this argument is defeated by another (undefeated) argument $\langle \mathcal{A}, \sim mortal(mcleod) \rangle$, where $\mathcal{A} = \{\sim mortal(mcleod) \multimap man(mcleod), highlander(mcleod)\}$. Therefore McLeod justifiedly belongs to the concept “ $\neg$ Mortal”. It can be also inferred that McLeod strictly

¹Following standard logic programming notation, in DeLP rules we note constant and predicate names with an initial lowercase and variable names with an initial uppercase.

²The function T_Π computes transposes of rules to allow for reasoning with *modus tollens*.

belongs to the concept “CarriesSword” as there exists an argument $\langle \emptyset, \text{carriessword}(\text{mcleod}) \rangle$.

DeLP program $\mathcal{P}_1 = (\Pi_1, \Delta_1)$ obtained from Σ_1 :

Facts and strict rules Π_1 :

$\text{carriessword}(x) \leftarrow \text{highlander}(x).$
 $\sim \text{highlander}(x) \leftarrow \sim \text{carriessword}(x).$
 $\text{man}(\text{socrates}).$
 $\text{man}(\text{mcleod}).$
 $\text{highlander}(\text{mcleod}).$

Defeasible rules Δ_1 :

$\text{mortal}(x) \prec \text{man}(x).$
 $\sim \text{mortal}(x) \prec \text{man}(x), \text{highlander}(x).$

Figure 2: DeLP program $\mathcal{P}_1$ interpreting ontology Σ_1

3 Local-as-View Integration of δ -Ontologies

Data integration is the problem of combining data residing at different sources, and providing the user with a unified view of those data [Lenzerini, 2002]. There are two main approaches to data integration, namely *global-as-view* (GAV) and *local-as-view* (LAV). In the LAV approach, we assume a global ontology $\mathcal{G}$, a set $\mathcal{S}$ of local/source ontologies, and the mapping between the global and the local ontologies is given by associating each term in the local ontologies with a *view* $V_{\mathcal{G}}$ over the global ontology [Calvanese *et al.*, 2001, Sect. 4]. The intended meaning of associating with a term C in $\mathcal{S}$ a view $V_{\mathcal{G}}$ over $\mathcal{G}$ is that such a view represents the best way to characterize the instances of C using the concepts in $\mathcal{G}$. The correspondence between C and the associated view can be *sound* (all the individuals satisfying C satisfy $V_{\mathcal{G}}$), *complete* (if no individual other than those satisfying C satisfies $V_{\mathcal{G}}$), and/or *exact* (the set of individuals that satisfy C is exactly the set of individuals that satisfy $V_{\mathcal{G}}$).

In the GAV and LAV approaches to data integration, the queries w.r.t. the target ontology are reformulated w.r.t. the sources. Hasse & Motik [Haase and Motik, 2005] (referring to [Lenzerini, 2002]) explain that in the GAV systems the problem is simply reduced to *unfolding* the views, since the reformulation is explicit in the mappings. In the LAV case, the problem requires more complex reasoning steps as in the case of sound mappings is not clear how to reformulate the concepts of a source ontology in terms of a global ontology. Therefore, in this work, we will restrict the case of LAV integration to complete views.

Definition 3.1 An ontology integration system $\mathcal{I}$ is a triple $(\mathcal{G}, \mathcal{S}, \mathcal{M})$ where:

- $\mathcal{G}$ is a global ontology expressed as a δ -ontology over an alphabet $\mathcal{A}_{\mathcal{G}}$.
- $\mathcal{S}$ is a set of n source ontologies $\mathcal{S}_1, \dots, \mathcal{S}_n$ expressed as δ -ontologies over alphabets $\mathcal{A}_{\mathcal{S}_1}, \dots, \mathcal{A}_{\mathcal{S}_n}$, resp. Each alphabet $\mathcal{A}_{\mathcal{S}_i}$ includes a symbol for each concept or role name of the source $\mathcal{S}_i$, $i = 1, \dots, n$.
- $\mathcal{M}$ is a set of n mappings $\mathcal{M}_1, \dots, \mathcal{M}_n$ between $\mathcal{G}$ and $\mathcal{S}_1, \dots, \mathcal{S}_n$, resp. Each mapping $\mathcal{M}_i$ is constituted by a set of inclusion axioms of the form $q_{\mathcal{S}_i} \sqsubseteq q_{\mathcal{G}}$, where $q_{\mathcal{G}}$ and $q_{\mathcal{S}_i}$ are concept descriptions defined over the global

ontology $\mathcal{G}$ and $\mathcal{S}_i$, $i = 1, \dots, n$, resp. The concepts $q_{\mathcal{G}}$ are expressed over the alphabet $\mathcal{A}_{\mathcal{G}}$ and $q_{\mathcal{S}_i}$ are expressed over the alphabet $\mathcal{A}_{\mathcal{S}_i}$. The sets $\mathcal{M}_1, \dots, \mathcal{M}_n$ are called bridge ontologies.

An ontology integration system will be interpreted as a DeLP program.

Definition 3.2 Let $\mathcal{I} = (\mathcal{G}, \mathcal{S}, \mathcal{M})$ be an ontology integration system such that $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_n\}$ and $\mathcal{M} = \{\mathcal{M}_1, \dots, \mathcal{M}_n\}$, where $\mathcal{G} = (T_{\mathcal{S}}^{\mathcal{G}}, T_{\mathcal{D}}^{\mathcal{G}}, \mathcal{A}_{\mathcal{G}})$; $\mathcal{S}_i = (T_{\mathcal{S}}^{\mathcal{S}_i}, T_{\mathcal{D}}^{\mathcal{S}_i}, \mathcal{A}_{\mathcal{S}_i})$, and, $\mathcal{M}_i = (T_{\mathcal{S}}^{\mathcal{M}_i}, T_{\mathcal{D}}^{\mathcal{M}_i})$, with $i = 1, \dots, n$. The system $\mathcal{I}$ is interpreted as the DeLP program $\mathcal{I}_{DeLP} = (\Pi, \Delta)$, with:

$$\begin{aligned} \Pi &= (\tau_{\Pi}(T_{\mathcal{S}}^{\mathcal{G}})) \cup (\tau_{\Pi}(\mathcal{A}_{\mathcal{G}})) \cup (\bigcup_{i=1}^n \tau_{\Pi}(T_{\mathcal{S}}^{\mathcal{S}_i})) \cup \\ &\quad (\bigcup_{i=1}^n \tau_{\Pi}(T_{\mathcal{S}}^{\mathcal{M}_i})), \text{ and} \\ \Delta &= (\tau_{\Delta}(T_{\mathcal{D}}^{\mathcal{G}})) \cup (\bigcup_{i=1}^n \tau_{\Delta}(T_{\mathcal{D}}^{\mathcal{S}_i})) \cup (\bigcup_{i=1}^n \tau_{\Delta}(T_{\mathcal{D}}^{\mathcal{M}_i})). \end{aligned}$$

Possible inferences in the integrated ontology $\mathcal{I}_{DeLP}$ are modeled by means of a dialectical analysis in the DeLP program that is obtained when each DL sentence of the ontology is mapped into DeLP clauses. Thus conclusions supported by warranted arguments will be the valid consequences that will be obtained from the original ontology, provided the strict information in $\mathcal{I}_{DeLP}$ is consistent. Formally:

Definition 3.3 Let $\mathcal{I} = (\mathcal{G}, \mathcal{S}, \mathcal{M})$ be an ontology integration system. Let a be an individual name, and C a concept name defined in $\mathcal{G}$.

1. Individual a is a potential member of C iff there exists an argument $\mathcal{A}$ for the literal $C(a)$ w.r.t. DeLP program $\mathcal{I}_{DeLP}$.
2. Individual a is a justified member of C iff there exists a warranted argument $\mathcal{A}$ for the literal $C(a)$ w.r.t. DeLP program $\mathcal{I}_{DeLP}$.
3. Individual a is an strict member of C iff there exists an empty argument for the literal $C(a)$ w.r.t. DeLP program $\mathcal{I}_{DeLP}$.

We will illustrate the above notions with an example. Notice that we label a concept C with the name of the ontology $\mathcal{S}_i$ to which it belongs (as in $\mathcal{S}_i : C$) following the XML name-space convention (This notation follows the one used in [Haase and Motik, 2005]).

Example 3.1 Let us consider the problem of assigning reviewers for papers. In Fig. 3, we present a global ontology $\mathcal{G}_{3.1}$ interpreted as: a professor with a postgraduate degree can be a reviewer; someone should not be a reviewer unless they are either a professor or have a graduate degree; however, a professor, despite not having a postgraduate degree, is accepted as a reviewer if he is an outstanding researcher. We also present local ontologies $\mathcal{L}_1$ and $\mathcal{L}_2$. $\mathcal{L}_1$ expresses that John is a professor who has a PhD, Paul is also a professor but has neither a PhD nor a MSc, Mary just has a MSc, and Steve is not a professor but has a MSc. The mapping $\mathcal{M}_{\mathcal{L}_1, \mathcal{G}}$ expresses that the terms MSc and PhD from local ontology $\mathcal{L}_1$ are contained in the term postgraduated in the global ontology, and that someone have neither a MSc nor a PhD is not a postgraduate. Ontology $\mathcal{L}_2$ expresses that a is an article, b a

book, c a chapter and that Paul has published a , b and c . The mapping $\mathcal{M}_{\mathcal{L}_2, \mathcal{G}}$ expresses that the view corresponding to the individuals who have published an article, a chapter and a book corresponds to the set of outstanding researchers.

The interpretation of above ontologies in DeLP yields the code presented in Fig. 4. We show next the dialectical analyses that have to be performed to compute the justified membership of John, Paul, Mary and Steve to the concept “Reviewer” w.r.t. the ontology integration system $(\mathcal{G}, \{\mathcal{L}_1, \mathcal{L}_2\}, \{\mathcal{M}_{\mathcal{L}_1, \mathcal{G}}, \mathcal{M}_{\mathcal{L}_2, \mathcal{G}}\})$.

The individual John is a justified member of the concept “Reviewer” because the argument $\langle \mathcal{A}, \text{reviewer}(\text{john}) \rangle$ has no defeaters and is thus warranted (see Fig. 5.(a)),³ with:

$$\mathcal{A} = \left\{ \begin{array}{l} (\text{reviewer}(\text{john}) \multimap \text{postgrad}(\text{john}), \text{prof}(\text{john})), \\ (\text{postgrad}(\text{john}) \multimap \text{phd}(\text{john})) \end{array} \right\}.$$

In Paul’s case, we conclude that he is a possible reviewer as he is also a justified member of the concept “Reviewer.” Notice that Paul is a potential member of the concept “¬Reviewer” as there is an argument $\langle \mathcal{B}_1, \sim \text{reviewer}(\text{paul}) \rangle$, with:

$$\mathcal{B}_1 = \left\{ \begin{array}{l} (\sim \text{reviewer}(\text{paul}) \multimap \sim \text{postgrad}(\text{paul})), \\ (\sim \text{postgrad}(\text{paul}) \multimap \sim \text{msc}(\text{paul}), \sim \text{phd}(\text{paul})) \end{array} \right\}.$$

However, we see that there is another argument $\langle \mathcal{B}_2, \text{reviewer}(\text{paul}) \rangle$ that defeats $\mathcal{B}_1$, where:

$$\mathcal{B}_2 = \left\{ \begin{array}{l} (\text{reviewer}(\text{paul}) \multimap \text{prof}(\text{paul}), \sim \text{postgrad}(\text{paul}), \\ \text{outstanding}(\text{paul})), \\ (\text{outstanding}(\text{paul}) \multimap \\ \text{published}(\text{paul}, a), \text{article}(a), \\ \text{published}(\text{paul}, b), \text{book}(b), \\ \text{published}(\text{paul}, c), \text{chapter}(c)), \\ (\sim \text{postgrad}(\text{paul}) \multimap \sim \text{msc}(\text{paul}), \sim \text{phd}(\text{paul})) \end{array} \right\}.$$

As $\mathcal{B}_2$ is undefeated, we conclude that the literal $\text{reviewer}(\text{paul})$ is warranted (see Fig. 5.(b)-(c)).

Steve is not a reviewer as he is a justified member of the concept “¬Reviewer” (see Fig. 5.(d)). In this case, there exists a unique (undefeated) argument $\langle \mathcal{C}, \sim \text{reviewer}(\text{steve}) \rangle$. On the other hand, it is not possible to assess Mary’s membership to the concept “Reviewer” as no arguments for $\text{reviewer}(\text{mary})$ nor $\sim \text{reviewer}(\text{mary})$ can be built.

We now introduce some properties of the proposed approach to ontology integration presented above. Notice that the validity of the properties is strongly related to DeLP reasoning dynamics.

Property 3.1 Let $\mathcal{I} = (\mathcal{G}, \mathcal{S}, \mathcal{M})$ be an ontology integration system. It cannot be the case that an individual a is a justified member of concepts C and $\neg C$ simultaneously.

Proof: Suppose that both a is a justified member of both C and $\neg C$. Then it must be the case that there exist two warranted arguments $\langle \mathcal{A}, C(a) \rangle$ and $\langle \mathcal{B}, \sim C(a) \rangle$. But this is impossible as DeLP cannot warrant two complementary literals (see [García and Simari, 2004]).

³In a dialectical tree nodes are labeled as either defeated (D) or undefeated (U). Leaves are always labeled as undefeated; a node is labeled as undefeated iff all of its children are labeled as defeated, otherwise a node is labeled as defeated.

Global ontology $\mathcal{G}_{3,1} = (\emptyset, T_D^{\mathcal{G}}, \emptyset)$:	
$T_D^{\mathcal{G}} = \left\{ \begin{array}{l} \text{Prof} \sqcap \text{Postgrad} \sqsubseteq \text{Reviewer}; \\ \neg \text{Postgrad} \sqcup \neg \text{Prof} \sqsubseteq \neg \text{Reviewer}; \\ \text{Prof} \sqcap \neg \text{Postgrad} \sqcap \text{Outstanding} \sqsubseteq \text{Reviewer} \end{array} \right\}$	
Local ontology $\mathcal{L}_1 = (\emptyset, \emptyset, A^{\mathcal{L}_1})$:	
$A^{\mathcal{L}_1} = \left\{ \begin{array}{lll} \text{JOHN} : \text{Prof}; & \text{JOHN} : \text{Phd}; & \text{PAUL} : \text{Prof}; \\ \text{PAUL} : \neg \text{Msc}; & \text{PAUL} : \neg \text{Phd}; & \text{MARY} : \text{Msc}; \\ \text{STEVE} : \neg \text{Prof}; & \text{STEVE} : \text{Msc} & \end{array} \right\}$	
Mapping $\mathcal{M}_{\mathcal{L}_1, \mathcal{G}}$ between $\mathcal{L}_1$ and $\mathcal{G}_{3,1}$:	
$\mathcal{M}_{\mathcal{L}_1, \mathcal{G}} = \left\{ \begin{array}{l} \mathcal{L}_1 : \text{Msc} \sqcup \mathcal{L}_1 : \text{Phd} \sqsubseteq \mathcal{G} : \text{Postgrad}; \\ \mathcal{L}_1 : \neg \text{Msc} \sqcap \mathcal{L}_1 : \neg \text{Phd} \sqsubseteq \mathcal{G} : \neg \text{Postgrad} \end{array} \right\}$	
Local ontology $\mathcal{L}_2 = (\emptyset, \emptyset, A^{\mathcal{L}_2})$:	
$A^{\mathcal{L}_2} = \left\{ \begin{array}{lll} a : \text{Article}; & b : \text{Book}; & \\ c : \text{Chapter}; & \langle \text{PAUL}, a \rangle : \text{published}; & \\ \langle \text{PAUL}, b \rangle : \text{published}; & \langle \text{PAUL}, c \rangle : \text{published} & \end{array} \right\}$	
Mapping $\mathcal{M}_{\mathcal{L}_2, \mathcal{G}}$ between $\mathcal{L}_2$ and $\mathcal{G}_{3,1}$:	
$\mathcal{M}_{\mathcal{L}_2, \mathcal{G}} = \left\{ \begin{array}{l} \mathcal{L}_2 : (\exists \text{published.Article} \sqcap \exists \text{published.Book} \sqcap \\ \exists \text{published.Chapter}) \sqsubseteq \mathcal{G} : \text{Outstanding} \end{array} \right\}$	

Figure 3: LAV ontology integration system

Property 3.2 Let $\mathcal{I} = (\mathcal{G}, \mathcal{S}, \mathcal{M})$ be an ontology integration system. It cannot be the case that an individual a is an strict member of concepts C and $\neg C$ simultaneously.

Proof: Suppose that $\mathcal{G} = (T_S^{\mathcal{G}}, T_D^{\mathcal{G}}, A^{\mathcal{G}})$; $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_n\}$ with $\mathcal{S}_i = (T_i^{\mathcal{S}}, T_i^{\mathcal{D}}, A_i)$, and $\mathcal{M}_i = (T_{\mathcal{M}_i}^{\mathcal{S}}, T_{\mathcal{M}_i}^{\mathcal{D}}, A_{\mathcal{M}_i})$, with $i = 1, \dots, n$. Suppose that a is an strict member of both C and $\neg C$. Then there must exist two arguments $\langle \emptyset, C(a) \rangle$ and $\langle \emptyset, \sim C(a) \rangle$ w.r.t. $\mathcal{I}_{DeLP}$. Therefore $\mathcal{T}_{\Pi}(T_S^{\mathcal{G}}) \cup \bigcup_{i=1}^n (\mathcal{T}_{\Pi}(T_i^{\mathcal{S}}) \cup \mathcal{T}_{\Pi}(T_{\mathcal{M}_i}^{\mathcal{S}}) \cup \mathcal{T}_{\Pi}(A_i) \cup \mathcal{T}_{\Pi}(A_{\mathcal{M}_i}))$ will be inconsistent contradicting the internal coherence assumption (see Sect. 2.2).

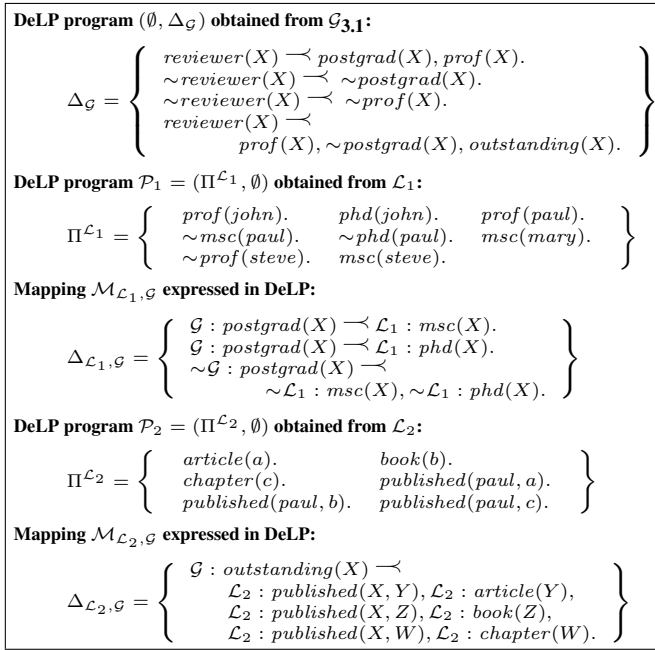
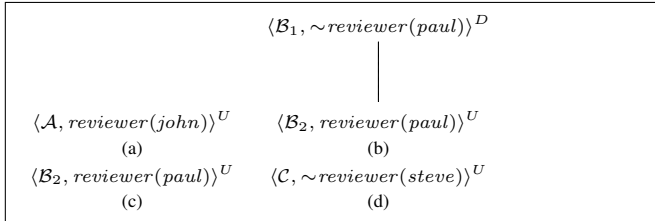
Property 3.3 The running time for determining the strict, potential and justified membership of an individual to a class in an ontology integration system is finite.

Proof sketch: Let $\mathcal{I} = (\mathcal{G}, \mathcal{S}, \mathcal{M})$ be an ontology integration system. As the δ -ontologies in $\mathcal{I}$ have a finite number of both named concepts and individual constants, the DeLP program $\mathcal{I}_{DeLP}$ is finite. Cecchi *et al.* [2006] have shown that determining if there exists an argument for a literal is NP; besides, as the warrant procedure always builds a finite dialectical tree [García and Simari, 2004], then proceses for determining the strict, potential and justified membership always terminate.

4 Conclusions and Future Work

We have presented an approach for performing local-as-view integration of Description Logic ontologies when these ontologies can be potentially inconsistent. We have adapted the notion of ontology integration system of [Calvanese *et al.*, 2001] for making it suitable for the δ -ontology framework, presenting both formal definitions and a case study.

For reasoning with an inconsistent ontology, it can be repaired manually by the knowledge engineer or automatically (*e.g.*, using Belief Revision [Ribeiro and Wassermann, 2009]). Other approaches involve using some non-monotonic

Figure 4: Ontologies $\mathcal{G}_{3,1}$, $\mathcal{L}_1$ and $\mathcal{L}_2$ expressed in DeLPFigure 5: Dialectical trees for $\text{reviewer}(\text{john})$, $\text{reviewer}(\text{paul})$ and $\text{reviewer}(\text{steve})$

inference procedure to obtain meaningful answers (e.g., para-consistent logics [Huang *et al.*, 2005] which consider reasoning with a consistent subset of the inconsistent ontology). Our approach directly copes with inconsistency by reasoning within the framework of DeLP, thus relying in defeasible argumentation for determining the membership status of individuals to concepts. It must be noted that the proposed approach is only useful in the case of complete mappings, and therefore the case for local-as-view integration with sound and exact mappings remains as an open problem and is part of our current research efforts.

References

- [Baader *et al.*, 2003] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook – Theory, Implementation and Applications*. Cambridge University Press, 2003.
- [Berners-Lee *et al.*, 2001] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scient. American*, 2001.
- [Calvanese *et al.*, 2001] D. Calvanese, G. De Giacomo, and M. Lenzerini. A framework for ontology integration. In *First Semantic Web Working Symposium*, pages 303–316, 2001.
- [Cecchi *et al.*, 2006] L. Cecchi, P. Fillottrani, and G. Simari. On Complexity of DeLP through Game Semantics. In J. Dix and A. Hunter, editors, *11th. Intl. Workshop on Non-monotonic Reasoning*, pages 386–394, 2006.
- [Chesñevar *et al.*, 2000] C. Chesñevar, A. Maguitman, and R. Loui. Logical Models of Argument. *ACM Computing Surveys*, 32(4):337–383, December 2000.
- [García and Simari, 2004] A. García and G. Simari. Defeasible Logic Programming an Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.
- [Gómez *et al.*, 2010] S.A. Gómez, C. Chesñevar, and G. Simari. Reasoning with Inconsistent Ontologies Through Argumentation. *Applied Artificial Intelligence*, 1(24):102–148, 2010.
- [Grosz *et al.*, 2003] B. Grosz, I. Horrocks, R. Volz, and S. Decker. Description Logic Programs: Combining Logic Programs with Description Logics. *WWW2003, May 20-24, Budapest, Hungary*, 2003.
- [Gruber, 1993] T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.
- [Haarslev and Möller, 2001] V. Haarslev and R. Möller. RACER System Description. Technical report, University of Hamburg, Computer Science Department, 2001.
- [Haase and Motik, 2005] P. Haase and B. Motik. A Mapping System for the Integration of OWL-DL Ontologies. In A. Hahn, S. Abels, and L. Haak, editors, *IHIS 05: Proceedings of the first international workshop on Interoperability of heterogeneous information systems*, pages 9–16. ACM Press, NOV 2005.
- [Huang *et al.*, 2005] Zhisheng Huang, Frank van Harmelen, and Annette ten Teije. Reasoning with Inconsistent Ontologies. In Leslie Pack Kaelbling and Alessandro Safiotti, editors, *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI’05)*, pages 454–459, Edinburgh, Scotland, August 2005.
- [Klein, 2001] M. Klein. Combining and relating ontologies: an analysis of problems and solutions. In A. Gomez-Perez, M. Gruninger, H. Stuckenschmidt, and M. Uschold, editors, *Workshop on Ontologies and Information Sharing, IJCAI’01*, Seattle, USA, August 4–5, 2001.
- [Lenzerini, 2002] M. Lenzerini. Data integration: A theoretical perspective. *Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2002, Madison, Winsconsin, USA*, 2002.
- [Ribeiro and Wassermann, 2009] Márcio Moretto Ribeiro and Renata Wassermann. Base Revision for Ontology Debugging. *J. Log. Comput.*, 19(5):721–743, 2009.
- [Stolzenburg *et al.*, 2003] F. Stolzenburg, A. García, C. Chesñevar, and G. Simari. Computing Generalized Specificity. *J. of N.Classical Logics*, 13(1):87–113, 2003.

Reduction of composites of relations between classes within formal ontologies

Mai Ajspur and Sine Zambach

Roskilde University, Computer Science

Roskilde, Denmark

ajspur, sz@ruc.dk

1 Introduction

In knowledge based systems, class-relation has played an expanding role for the representation of knowledge. For example in biomedical ontologies, the OBO-foundry operates with relations among classes based on the relations among the individuals of these classes [Smith and Rosse, 2004; Smith *et al.*, 2005]. In e.g. [Blonde *et al.*, 2011], a framework is presented on the formal and informal properties of the class-relations as well as other ontological relations within a semantic web-approach to e.g. biomedical ontologies.

The reason for using relations among classes¹ rather than relations among individuals is that classes, if you see them as reflecting the real world, are really the main matters in e.g. biomedical ontologies rather than the individuals. The terminology with which we talk about classes or concepts reflects that we always try to abstract from individual to class when modeling ontologies.

Additionally, the reason for being able to reason over class-relations is that it can be useful within drug discovery as noted in e.g. [Qu *et al.*, 2009]. For example the link between a drug and a disease might be deduced by a chain of events, which can be formed logically as compositions of (different) class-relations.

In the popular ontology language, Description Logic [Baader *et al.*, 2003] the universal and existential restrictions are by our definition also relations among classes.

In this work we reduce the potential class-relation composites to create a logical foundation for complex role inclusions on class-relationships. This is based on relationships among individuals belonging to these classes of interest.

We consider a total of nine possible different binary class-relationships, namely those in table 1 which is partly based on [Nilsson, 2011]. The class-relationships and their motivation will be introduced in section 2, and the degree of knowledge, that the different class-relations represent are discussed in section 2.1 adding of examples from biomedicine. In section 3, the notation and formal definitions are given, while we go on to study the possible combinations of these relations as compositions of class-relationships in section 4.

¹Note that we use *relation between classes* and *class-relations* although the latter is not strictly mathematically correct without a second order definition.

2 Class-relationships and role inclusions

When we talk about a class in this paper, we refer to what is often called concepts, types or kinds, representing *reality*, which is mainly in line with e.g. [Smith *et al.*, 2005].

With regards to individuals, complex role inclusions based on compositions are well studied and formalized, e.g. in Description Logic ($\mathcal{DL}$) [Baader *et al.*, 2003] where they can be modeled as reasoning rules in e.g. the $\mathcal{DL}$ -language $\mathcal{EL}+$ language [Baader *et al.*, 2006] using the adopted notation: $\mathbf{R}^3 \sqsubseteq \mathbf{R}^1 \circ \mathbf{R}^2$ for relations among individuals.

Introducing a $\odot_B$ -connective, our subject of investigation is thus combinations of the form $A (R_{\forall\exists}^1 \odot_B R_{\forall\forall}^2) C$, for given class-relationships $A R_{\forall\exists}^1 B$ and $B R_{\forall\forall}^2 C$; the formal definition will be given in section 3.

Our goal is then to determine which type the resulting class-relationships will have, given the type of the two relations in the composite, i.e. given any combination of the quantifiers specifying the type of the relations R^1 and R^2 between classes A and B , and B and C respectively (i.e. $\forall\exists$ and $\forall\forall$ in the example above). We will examine whether a class-relationship between A and C of any of the considered types in table 1 exists, and if so, which it is.

2.1 Degrees of knowledge

The actual application of the reasoning rules in table 2 and definitions of meaningful inferences on a certain domain is a task of knowledge engineers. These modeled rules are based on domain specific knowledge and complement the pure logic inferences. Thus, it can be useful to consider the levels of abstractions that can be used within the domain.

Knowledge using class-relationship representation can come in many forms and are based on the *degree of knowledge* your knowledge base contains. Some examples are discussed as follows:

The $\exists\exists$ relation represents the least degree of knowledge. In this case one only knows that an instance from the first class has a relation to an instance from the second class. This is not of much use if one is presented with a concrete instance of either of the classes; given an amount of insulin and a glucose transport one cannot infer anything about them from the knowledge that “*Insulin stimulates $\exists\exists$ Glucose_transport*”. An appearance of knowledge on this form could be a semantic extraction from for instance a biomedical text. The relation has

Table 1: The different class-relationship types in First-order Logic using the six possible combinations of quantifiers.

$CR\mathcal{L}$	FOL formulation	Natural language formulation
$A R_{\exists\exists} B$	$\exists x(A(x) \wedge \exists y(xRy \wedge B(y)))$	There exists an element in A that is related to some element in B
$A R_{\exists\forall} B$	$\exists x(A(x) \wedge \forall y(B(y) \rightarrow xRy))$	There exists an element in A that is related to every element in B
$A R_{\exists\forall_o} B$	$\exists x(A(x) \wedge \forall y(xRy \rightarrow B(y)))$	There exists an element in A that is <i>only</i> related to elements inside B
$A R_{\forall\exists} B$	$\forall x(A(x) \rightarrow \exists y(xRy \wedge B(y)))$	Every element in A is related to some element in B
$A R_{\forall\forall} B$	$\forall x(A(x) \rightarrow \forall y(B(y) \rightarrow xRy))$	Every element in A is related to every element in B
$A R_{\forall\forall_o} B$	$\forall x(A(x) \rightarrow \forall y(xRy \rightarrow B(y)))$	Any element in A is <i>only</i> related to elements inside B
$A R_{\forall_o\exists} B$	$\exists y(B(y) \wedge \forall x(xRy \rightarrow A(x)))$	There exists an element in B that no element outside A is related to
$A R_{\forall_o\forall} B$	$\forall y(B(y) \rightarrow \forall x(xRy \rightarrow A(x)))$	There are no elements outside A that is related to any element in B
$A R_{\forall_o\forall_o} B$	$\forall x\forall y(xRy \rightarrow (B(y) \wedge A(x)))$	It is <i>only</i> elements in A and B that are related

been seen, but nothing general should be concluded from just one (uncertain) example.

A next natural step in the degrees of knowledge may be the $\forall\exists$ relation. This knowledge might be obtained by observing a larger sample of instances from the first class. The knowledge “*Insulin stimulates $_{\forall\exists}$ Glucose_transport*” lets us infer something about insulin, namely that if we have any amount of insulin we can find a glucose transport that it stimulates. However, this knowledge does not really tell us anything about glucose transport. This relation type is typically found in canonical ontologies and the subsumption-relation can also be seen as this relation type.

Continuing, the $\forall\forall_o$ relation will intuitively represent a higher degree of knowledge. But in the case of insulin and glucose transport it still only tells us something about insulin. It tells us that whatever insulin stimulates it must be a glucose transport. But it does not tell us that we can always stimulate glucose transports with insulin. It is thus typically used to create constraints for the first class, e.g. “*Insulin IsSecretedBy $_{\forall\forall_o}$ Beta_cells*”, saying that whenever insulin is secreted, it is by a beta-cell.

Likewise, the inverse relation behaves. If R is $\forall\forall_o$, then R^{-1} is $\forall_o\forall$ and the statement would be “*Beta_cells Secretes $_{\forall_o\forall}$ Insulin*”.

The “final” degree of knowledge as represented by the $\forall\forall$ relation gives us information about both insulin and glucose transports. This is often the really useful information, namely that given an arbitrary glucose transport we can always stimulate it with any given amount of insulin. This knowledge is typically found in interaction databases such as [Kanehisa and Goto, 2000]².

Notice, however, that the logical strength of $\forall\forall$ is not necessarily larger than the one of $\forall\forall_o$; there is also information contained in knowing, that e.g. insulin *cannot* be secreted by anything *else* than beta cells. $\forall_o\forall_o$ is “stronger” since it restricts both the relata classes.

²However, this does not prevent the $\forall\exists$ in being the underlying relationships in e.g. popular presentations of the knowledge in texts and speech in which people tend to generalize.

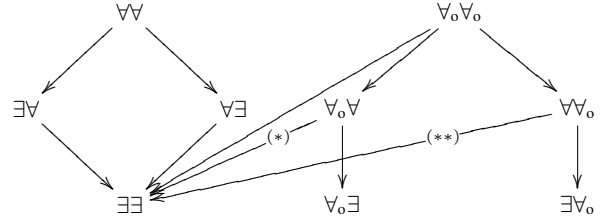


Figure 1: Ordering of the different relation types, assuming the relation is non-empty. (*) = $(M \times B) \cap R \neq \emptyset$, (**) = $(A \times M) \cap R \neq \emptyset$

These considerations show that when specifying formal relations for knowledge representation in biomedicine for instance, one should not only consider how the world actually is, i.e. what the ontology looks like, but also what kind of knowledge can be useful to represent in an application in the first place.

Mathematically speaking, we can order the relations according to their degree of knowledge following the diagram in Figure 1, assuming that all concerned classes and relations are non-empty. This should be read such that if R e.g. is a $\forall\forall$ -relation, then it is also a $\forall\exists$ -relation as well as an $\exists\forall$ -relation, and (due to transitivity) it is also an $\exists\exists$ -relation, given that the relation is non-empty. This is easily seen by the (first order) definitions of the relationship types (see table 1). As described above, we consider a $\forall\forall$ -relation to be more informative than a $\forall\forall_o$ -relation. Meanwhile, a $\forall\forall$ -relation is not a $\forall\forall_o$ -relation; though, under certain (natural) conditions described below, it is a $\exists\exists$ -relation. In the example above with the $\forall\forall_o$ -relation *Stimulate* between *Insulin* and *Glucose_transport*, we are not told that there is not anything *else* besides insulin that could stimulate glucose transport. But if we make the (further) assumption that insulin indeed stimulates *something* at all, then we can infer that *Stimulates* is an $\exists\exists$ -relation. This requirement is that $(Insulin \times M) \cap Stimulates \neq \emptyset$, where M denotes

the entire domain.

This requirement corresponds to the intuition behind our knowledge representation, though this is often an implicit assumption; when we say that insulin stimulates glucose transport, indeed we are implicitly excluding the case where something that is not insulin stimulates something that is not a glucose transport, while no insulin stimulate any glucose transport, even though this, mathematically speaking, would still qualify as a valid $Stimulates_{\forall\forall_0}$ -relation between *Insulin* and *Glucose_transport*.

As a final remark to this diagram, one might think that if R is a non-empty $\exists\forall_0$ -relation, then it would likewise be an $\exists\exists$ -relation, but this is not the case as can be seen from the example $A = \{a, b\}$, $B = \{b\}$ and $R = \{(b, a)\}$. Unlike the example above, the extra restriction on R that would ensure it being an $\exists\exists$ -relation would, in itself, imply that R is an $\exists\exists$ -relation. This could also indicate that the usage of the $\exists\forall_0$ -relation is limited within Knowledge Representation.

3 Definitions and notation

We use " $x, y, z, \dots$ " as variables ranging over arbitrary individuals and " $A, B, C, \dots$ " to range over classes. Relations among individuals will be in boldface and capitalized, e.g. " $\mathbf{R} \ y$ " whereas the corresponding, *lifted* relations among classes will be italicized and capitalized, e.g. " $A \ R \ B$ ". Notationwise, we will allow ourselves to write " $\exists x \in X(\dots)$ " for " $\exists x(X(x) \wedge \dots)$ " and " $\forall x \in X(\dots)$ " for " $\forall x(X(x) \rightarrow \dots)$ ". Formally we consider the situation where we are given previously defined relations $\mathbf{R}^1$ and $\mathbf{R}^2$ on $M \times M$ (where M is the entire domain), that give rise to two class-relations, each of one of the nine types $\exists\exists, \exists\forall, \exists\forall_0, \forall\exists, \forall\forall, \forall\forall_0, \forall_0\exists, \forall_0\forall$ and $\forall_0\forall_0$ as defined in table 1. I.e. we have that $A \ R_{HI}^1 \ B$ and $B \ R_{JK}^2 \ C$, where H, I, J, K and K are quantifiers with $H, I, J, K \in \{\forall, \exists, \forall_0\}$.

The composition of these class relations, denoted by $\odot_B$, is then:

$$(A \ R_{HI}^1 \ B) \odot_B (B \ R_{JK}^2 \ C) \equiv A \ (R_{HI}^1 \odot_B \ R_{JK}^2) \ C. \quad (1)$$

Our task at hand is now to determine which type, if any, this class-relation $R_{HI}^1 \odot_B \ R_{JK}^2$ between A and C is: Given (1), we will investigate whether there exist quantifiers $L, M \in \{\forall, \exists, \forall_0\}$, such that $A \ R_{LM}^3 \ C$, where R^3 is the "lifted" relation of $\mathbf{R}^3 = \mathbf{R}^1 \odot_B \mathbf{R}^2$. $\odot_B$ is here a modified version of the conventional composition of relations, where we restrict the mediating element m to be in a specified, non-arbitrary subset of M :

$$\mathbf{R}^1 \odot_B \mathbf{R}^2 =$$

$$\{(m_1, m_2) \in M \times M \mid \exists m \in B(m_1 \mathbf{R}^1 m \wedge m \mathbf{R}^2 m_2)\}.$$

This B -composition corresponds to the intuition behind the knowledge representation using classes which are based on complex role inclusions using individuals.

An example of such B -compositions is:

$$A \text{ Regulates}_{\forall\forall} B \odot_B B \text{ Isa}_{\forall\exists} D \sqsubseteq A \text{ Regulates}_{\forall\exists} D,$$

whereas a non- B -composition as $A \ R_{\forall\forall}^1 \ B \odot C \ R_{\forall\exists}^2 \ D \sqsubseteq A \ R_{\forall\exists}^3 \ D$ often is an un-acquired composition for knowledge modelling purposes.

To illustrate our task, we consider the example given in section 2, i.e. we consider given relations $A \ R_{\forall\exists}^1 \ B$ and $B \ R_{\forall\forall}^2 \ C$. We now want to find out whether $\mathbf{R}^3$ defines a class-relationship between classes A and C , where

$$\mathbf{R}^3 = \left\{ (x, z) \mid \left(\exists y (B(y) \wedge x \mathbf{R}^1 y \wedge y \mathbf{R}^2 z) \right) \wedge \left(\forall x (A(x) \rightarrow \exists y (x \mathbf{R}^1 y \wedge B(y))) \right) \wedge \left(\forall y (B(y) \rightarrow \forall z (C(z) \rightarrow y \mathbf{R}^2 z)) \right) \right\}$$

If so, the given class-relationship will then be $A \ (R_{\forall\exists}^1 \odot_B \ R_{\forall\forall}^2) \ C = R_{LM}^3$. According to Table 2, in this case we have that $L = M = \forall$, i.e. $A \ (R_{\forall\exists}^1 \odot_B \ R_{\forall\forall}^2)_{\forall\forall} \ C$.

4 Obtainable compositions

The possible combination of types of two class-relations $A \ R_{HI}^1 \ B$ and $B \ R_{JK}^2 \ C$, together with the type of their composite $A \ R_{LM}^3 \ C$ are shown in table 2. We note, that the stated relation-types are the most expressive relation with respect to the degree of knowledge it expresses, cf. Figure 1, though the proofs of these claims are left out. "/" means that the relation is one of the two stated relation types, while "&" means that the relation is both of the stated relation types. The results with a "/" is - besides the minus - the "weakest" ones, since we cannot say a priori which it is without examining the relations further. Also, these results could be split in an other fashion, e.g. the composition of a $\forall_0\exists$ -relation and a $\forall\forall$ -relation is either a $\exists\forall$ - or a $\forall\forall_0$ -relation, but it is also either a $\forall\forall$ -relation or a $\exists\forall_0$ -relation. Since the ordering of the relation types is not total, we cannot say which is the most expressive in this case.

The possible combinations are argued for in this section and is also tested with the higher order theorem prover using the TPTP-platform [Sutcliffe, 2009].

It should be noted that if $\mathbf{R}^1 \odot_B \mathbf{R}^2 = \emptyset$ then $R^1 \odot_B R^2$ is trivially a $\forall_0\forall_0$ -relation. In the following we have therefore assumed, that $\mathbf{R}^1 \odot_B \mathbf{R}^2 \neq \emptyset$, and further, that all involved classes and relations - $A, B, C, \mathbf{R}^1$ and $\mathbf{R}^2$ - are non-empty, even though an empty relation as such is a valid $\forall_0\forall_0$ -relation. The results in the table reflect these assumptions.

This is mostly a reasonable assumption within knowledge representation; it is required from the modeler that a class or set in a knowledge base is a part of the world rather than a meta-class that is potentially empty.

When we in the following give counter examples for some of the possible combinations of relations, we will use the three classes A, B and C of the domain M with the individuals and different relations between the classes. The

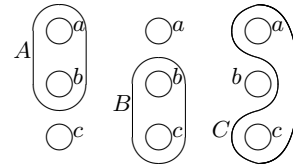


Figure 2: Our standard example consisting of the domain $M = \{a, b, c\}$ with classes A, B and C

Table 2: The obtainable type of the composite of two class-relationships depending on their types. The numbers refer to the cases where the claims are proven.

		Type of R^2								
Type of R^1		$\exists\exists$	$\forall\exists$	$\exists\forall$	$\forall\forall$	$\exists\forall_o$	$\forall\forall_o$	$\forall_o\exists$	$\forall_o\forall$	$\forall_o\forall_o$
	$\exists\exists$	— (1)	$\exists\exists$ (4)	— (1)	$\exists\forall$ (4)	— (1)	$\forall\forall_o$ (2)	— (1)	— (1)	$\forall\forall_o$ (2)
	$\forall\exists$	— (1)	$\forall\exists$ (4)	— (1)	$\forall\forall$ (4)	— (1)	$\forall\forall_o$ (2)	— (1)	— (1)	$\forall\forall_o$ (2)
	$\exists\forall$	$\exists\exists$ (5)	$\exists\exists$ (5)	$\exists\forall$ (5)	$\exists\forall$ (5)	$\exists\exists/\forall_o\forall$ (6)	$\exists\exists\&\forall\forall_o$ (10)	$\exists\exists/\forall_o\forall$ (6)	$\exists\exists/\forall_o\forall$ (6)	$\exists\exists\&\forall\forall_o$ (10)
	$\forall\forall$	$\forall\exists$ (5)	$\forall\exists$ (5)	$\forall\forall$ (5)	$\forall\forall$ (5)	$\exists\forall/\forall_o\forall$ (6)	$\forall\exists\&\forall\forall_o$ (10)	$\forall\exists/\forall_o\forall$ (6)	$\forall\exists/\forall_o\forall$ (6)	$\forall\exists\&\forall\forall_o$ (10)
	$\exists\forall_o$	— (1)	$\exists\exists/\forall\forall_o$ (3)	— (1)	$\exists\forall/\forall\forall_o$ (3)	— (1)	$\forall\forall_o$ (2)	— (1)	— (1)	$\forall\forall_o$ (2)
	$\forall\forall_o$	— (1)	$\exists\exists/\forall\forall_o$ (3)	— (1)	$\exists\forall/\forall\forall_o$ (3)	— (1)	$\forall\forall_o$ (2)	— (1)	— (1)	$\forall\forall_o$ (2)
	$\forall_o\exists$	— (1)	$\exists\exists/\forall\forall_o$ (3)	— (1)	$\exists\forall/\forall\forall_o$ (3)	— (1)	$\forall\forall_o$ (2)	— (1)	— (1)	$\forall\forall_o$ (2)
	$\forall_o\forall$	$\forall_o\forall$ (8)	$\exists\exists\&\forall_o\forall$ (7)	$\forall_o\forall$ (8)	$\exists\forall\&\forall_o\forall$ (7)	$\forall_o\forall$ (8)	$\exists\exists\&\forall_o\forall_o$ (9)	$\forall_o\forall$ (8)	$\forall_o\forall$ (8)	$\exists\exists\&\forall_o\forall_o$ (9)
	$\forall_o\forall_o$	$\forall_o\forall$ (8)	$\exists\exists\&\forall_o\forall$ (7)	$\forall_o\forall$ (8)	$\exists\forall\&\forall_o\forall$ (7)	$\forall_o\forall$ (8)	$\exists\exists\&\forall_o\forall_o$ (9)	$\forall_o\forall$ (8)	$\forall_o\forall$ (8)	$\exists\exists\&\forall_o\forall_o$ (9)

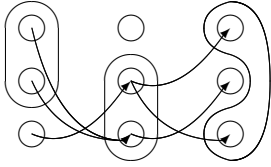


Figure 3: $R^1 \in \{\exists\exists, \forall\exists, \exists\forall_o, \forall\forall_o, \forall_o\exists\}$, $R^2 \in \{\exists\exists, \exists\forall, \exists\forall_o, \forall_o\exists, \forall_o\forall\}$. $R^3 \neq \emptyset$ is none of the nine types.

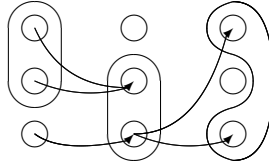


Figure 4: $R^1 \in \{\exists\exists, \exists\forall_o, \forall\exists, \forall\forall_o, \forall_o\exists\}$ and $R^2 \in \{\forall\forall_o, \forall_o\forall_o\}$. $R^3 \neq \emptyset$ is a $\forall\forall_o$ -relation but neither an $\exists\exists$ - nor a $\forall_o\exists$ -relation.

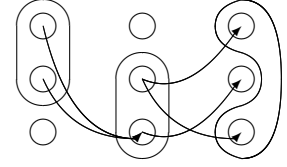


Figure 5: $R^1 \in \{\forall_o\forall, \forall_o\forall_o\}$ and $R^2 \in \{\exists\exists, \exists\forall, \exists\forall_o, \forall_o\exists, \forall_o\forall\}$. $R^3 \neq \emptyset$ is a $\forall_o\forall$ -relation but neither an $\exists\exists$ - nor a $\exists\forall_o$ -relation.

setting can be seen in Figure 2.

Case 1 – $R^1 \in \{\exists\exists, \forall\exists, \exists\forall_o, \forall\forall_o, \forall_o\exists\}$ and $R^2 \in \{\exists\exists, \exists\forall, \exists\forall_o, \forall_o\exists, \forall_o\forall\}$: In Figure 3, R^1 is an $\exists\exists$ -, $\forall\exists$ -, $\exists\forall_o$ -, $\forall\forall_o$ - and a $\forall_o\exists$ -relation, while R^2 is both an $\exists\exists$ -, $\exists\forall$ -, $\exists\forall_o$ -, $\forall_o\exists$ - and a $\forall_o\forall$ -relation. This example gives a counter example that $R^1 \odot_B R^2$ will be of any of the considered nine types unless, as mentioned earlier, $R^1 = \emptyset$, $R^2 = \emptyset$, or just $R^1 \odot_B R^2 = \emptyset$, in which case $R^1 \odot_B R^2$ is a $\forall_o\forall_o$ -relation; this is easily seen by inspection.

Case 2 – $R^1 \in \{\exists\exists, \forall\exists, \exists\forall_o, \forall\forall_o, \forall_o\exists\}$ and $R^2 \in \{\forall\forall_o, \forall_o\forall_o\}$: Let $x \in A$, and assume $x(R^1 \odot_B R^2)z$ for an arbitrary z . I.e. there is a $y \in B$, such that xR^1y and yR^2z . According to the $\forall\forall_o$ - or $\forall_o\forall_o$ -property of R^2 , necessarily $z \in C$, i.e. R^3 is a $\forall\forall_o$ -relation. Figure 4 shows that in general, R^3 cannot be a $\exists\exists$ - nor a $\forall_o\exists$ -relation either. Accordingly, it cannot be of any more expressive type, nor can it be of any other type, that cannot be compared via the ordering among types. Thus it is far to say that in general, R^3 is at most a $\forall\forall_o$ -relation.

Case 3 – $R^1 \in \{\forall_o\exists, \exists\forall_o, \forall\forall_o\}$ and $R^2 \in \{\forall\exists, \forall\forall\}$: Either there exists an $x \in A$ and a z s.t. xR^3z , or for all $x \in A$ and all z , $\neg(xR^3z)$ holds. In the first case, there exists $y \in B$ s.t. xR^1y and yR^2z , which in the case where R^2 is a $\forall K$ -relation gives that $Kz' \in C$ (yR^2z'), i.e. R^3 is an $\exists K$ -relation. In the latter case, $\forall x \in A \forall z (xR^3z \rightarrow z \in C)$

is true, i.e. R^3 is a $\forall\forall_o$ -relation.

It should be noted, however, that this is not the only way of "dividing" the cases. It is also the case that either $\forall x \in A \exists y \in B (xR^1y)$ or $(\exists x \in A \forall y \in B (\neg(xR^1y)))$, which leads to the conclusion, that either R^3 is a $\forall K$ -relation or a $\exists\forall_o$ -relation.

Case 4 – $R^1 \in \{\exists\exists, \forall\exists\}$ and $R^2 \in \{\forall\exists, \forall\forall\}$: Assume R^1 is a $H\exists$ -relation. If $H = \exists$, we let $x \in A$ satisfy the $\exists\exists$ -property of R^1 . If $H = \forall$, we let $x \in A$ be arbitrary. Since R^1 is an $H\exists$ -relation, there exists an $y \in B$, such that xR^1y . Since R^2 is an $\forall K$ -relation, we know that the defining $\forall K$ -property of R^2 is met by y , i.e.:

- If $K = \exists$: This means that there exists a z in C such that yR^2z , and thus by definition $x(R^1 \odot_B R^2)z$. Therefore, if x was given, there exists $x \in A$ and $z \in C$ such that $x(R^1 \odot_B R^2)z$. If x was arbitrary it would be a $\forall\exists$ -relation, i.e. an $H\exists$ -relation.
- If $K = \forall$: Similarly this means that for any z in C , yR^2z . Thus $x(R^1 \odot_B R^2)z$ for all $z \in C$.

Case 5 – $R^1 \in \{\exists\forall, \forall\forall\}$ and $R^2 \in \{\exists\exists, \exists\forall, \forall\exists, \forall\forall\}$: Assume R^1 is a $H\forall$ -relation and R^2 is a JK -relation. If $H = \forall$, we let $x \in A$ be arbitrary, and if $H = \exists$ we let x be such that $\forall y \in B (xR^1y)$. Likewise, we let $y \in B$ be arbitrary if $J = \forall$ and otherwise we let $y \in B$ be such that $Kz \in C (yR^2z)$ is true. From the defining $H\forall$ -property of the R^1 -relation

we now have that $x\mathbf{R}^1y$ holds, and from the defining JK -property of the R^2 -relation we get that $Kz \in C$ ($y\mathbf{R}^2z$). That is, we have that

$$Hx \in A (\exists y \in B (Kz \in C (y\mathbf{R}^2z)) \wedge x\mathbf{R}^1y). \quad (2)$$

When $K = \exists$ we get that

$$\begin{aligned} \exists y \in B (\exists z (z \in C \wedge y\mathbf{R}^2z) \wedge x\mathbf{R}^1y) \\ \equiv \exists z \in C (\exists y \in B (x(\mathbf{R}^1 \circ_B \mathbf{R}^2)z)) \end{aligned}$$

I.e. we now have that $A(R^1 \circ_B R^2)_{H\exists}C$.

On the other hand, if $K = \forall$ we get that

$$\begin{aligned} \exists y \in B (\forall z \in C (y\mathbf{R}^2z)) \wedge x\mathbf{R}^1y \\ \equiv \exists y \in B \forall z ((z \notin C \vee y\mathbf{R}^2z) \wedge x\mathbf{R}^1y) \\ \stackrel{*}{\Rightarrow} \forall z \exists y \in B ((z \notin C \vee y\mathbf{R}^2z) \wedge x\mathbf{R}^1y) \\ \Rightarrow \forall z \exists y \in B (z \notin C \vee (y\mathbf{R}^2z \wedge x\mathbf{R}^1y)) \\ \equiv \forall z (z \notin C \vee \exists y \in B (y\mathbf{R}^2z \wedge x\mathbf{R}^1y)) \\ \equiv \forall z \in C (x(\mathbf{R}^1 \circ_B \mathbf{R}^2)y) \end{aligned}$$

where $*$ is true since in general $\exists y \forall z f(y, z)$ implies $\forall z \exists y f(y, z)$. Therefore we now have that (2) implies that $Hx \in A (\forall z \in C (x(\mathbf{R}^1 \circ_B \mathbf{R}^2)z))$, that is, $A(R^1 \circ_B R^2)_{H\forall}C$.

In both cases we have that $R^1 \circ_B R^2$ is a HK -relation.

Case 6 – $R^1 \in \{\exists\forall, \forall\forall\}$ and $R^2 \in \{\exists\forall_0, \forall_0\exists, \forall_0\forall\}$: Either $\exists z \in C \exists x (x\mathbf{R}^3z)$, or $\forall z \in Z \forall x (\neg(x\mathbf{R}^3z))$. The first case gives there is a $y \in B$ s.t. $x\mathbf{R}^1y$ and $y\mathbf{R}^2z$. Since R^1 is an $H\forall$ -relation, $Hx' \in A (x'\mathbf{R}^1y)$ holds, i.e. R^3 is a $H\exists$ -relation. In the latter case, we have that $\forall z \in C \forall x (x\mathbf{R}^3z \rightarrow x \in A)$, i.e. R^3 is a $\forall_0\forall$ -relation.

Case 7 – $R^1 \in \{\forall_0\forall, \forall_0\forall_0\}$ and $R^2 \in \{\forall\exists, \forall\forall\}$: For all x and z , if $x\mathbf{R}^3z$, there exists $y \in B$ s.t. $x\mathbf{R}^1y$ and $y\mathbf{R}^2z$. Since R^1 is a $\forall_0\forall$, $x \in A$. Thus, for all $z \in C$, if $x\mathbf{R}^3z$, $x \in A$ holds. Thus R^3 is an $\forall_0\forall$ -relation. Given that R^2 is a $\forall K$ -relation, we also have that if $x\mathbf{R}^1y$ and $y\mathbf{R}^2z$, $Kz' \in C (y\mathbf{R}^2z')$, i.e. R^3 is a $\exists K$ -relation.

Case 8 – $R^1 \in \{\forall_0\forall, \forall_0\forall_0\}$ and $R^2 \in \{\exists\exists, \exists\forall, \exists\forall_0, \forall_0\exists, \forall_0\forall\}$: For all x and z , if $x\mathbf{R}^1y$ and $y\mathbf{R}^2z$ for a $y \in B$, then since R^1 is a $\forall_0\forall$, $x \in A$. Thus, for all $z \in C$, if $x\mathbf{R}^3z$, $x \in A$ holds, and R^3 is an $\forall_0\forall$ -relation. Figure 5 show that in general, R^3 cannot be an $\exists\exists$ - nor a $\exists\forall_0$ -relation.

Case 9 – $R^1 \in \{\forall_0\forall, \forall_0\forall_0\}$ and $R^2 \in \{\forall\forall_0, \forall_0\forall_0\}$: Assume $x\mathbf{R}^3z$. Then there exists $y \in B$ s.t. $x\mathbf{R}^1y$ and $y\mathbf{R}^2z$. Since R^1 is a $\forall_0\forall$ - or $\forall_0\forall_0$ -relation, $x \in A$, and since R^2 is a $\forall\forall_0$, $z \in C$. That is, R^3 is a $\forall_0\forall_0$ -relation. Since $R^3 \neq \emptyset$, there exists x and z such that $x\mathbf{R}^3z$, and from the above $x \in A$ and $z \in C$, i.e. R^3 is a $\exists\exists$ -relation too.

Case 10 – $R^1 \in \{\exists\forall, \forall\forall\}$ and $R^2 \in \{\forall\forall_0, \forall_0\forall_0\}$: Since $R^3 \neq \emptyset$, there exists x, z and a $y \in B$ such that $x\mathbf{R}^1y$ and $y\mathbf{R}^2z$. Since R^2 is a $\forall\forall_0$ - or a $\forall_0\forall_0$ -relation, necessarily $z \in C$. Since R^1 is a $H\forall$ -relation, $Hx' \in A (x'\mathbf{R}^1y)$, i.e. $Hx' \in A \exists z \in C (x'\mathbf{R}^3z)$, that is, R^3 is a $H\exists$ -relation.

Likewise, if $x \in A$ and $x\mathbf{R}^3z$, then the defining $\forall\forall_0$ - or $\forall_0\forall_0$ -property of R^2 gives that $z \in C$, i.e. R^3 is a $\forall\forall_0$ -relation, too.

5 Conclusion

In this study we present the notion of compositions of relation among classes. We present why class-relations are important within knowledge representation and reasoning and discuss the different level of knowledge they represent.

Finally, we present the possible obtainable compositions and discuss the cases where a composition cannot lead to a logical deduction.

Acknowledgement

Thanks to Jørgen Fischer Nilsson for technical discussions, to Jens Ulrik Hansen for contributions to section 2.1 and to Christoph Benzmueller for support on the automated theorem proving.

References

- [Baader *et al.*, 2003] F. Baader, D. Calvanese, and D. McGuinness, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
- [Baader *et al.*, 2006] F. Baader, C. Lutz, and B. Suntisrivaraporn. *CEL - a polynomial-time reasoner for life science ontologies*, pages 287–291. Springer Berlin/Heidelberg, 2006.
- [Blonde *et al.*, 2011] W. Blonde, V. Mironov, A. Venkatesan, E. Antezana, B. De Baets, and M. Kuiper. Reasoning with bio-ontologies: using relational closure rules to enable practical querying. *Bioinformatics*, 2011.
- [Kanehisa and Goto, 2000] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, 28:27–30, 2000.
- [Nilsson, 2011] J. Fischer Nilsson. Class relationship logic for ontologies. *Manuscript*, 2011.
- [Qu *et al.*, 2009] X. A. Qu, R. C. Gudivada, A. G. Jegga, E. K. Neumann, and B. J. Aronow. Inferring novel disease indications for known drugs by semantically linking drug action and disease mechanism relationships. *BMC Bioinformatics*, 10 Suppl 5:S4, 2009.
- [Smith and Rosse, 2004] B. Smith and C. Rosse. The role of foundational relations in the alignment of biomedical ontologies. *MEDINFO*, pages 444–448, 2004.
- [Smith *et al.*, 2005] B. Smith, W. Ceusters, B. Klagges, J. Kohler, A. Kumar, J. Lomax, C. Mungall, F. Neuhaus, A. L. Rector, and C. Rosse. Relations in biomedical ontologies. *Genome Biol.*, 6:R46, 2005.
- [Sutcliffe, 2009] G. Sutcliffe. The TPTP Problem Library and Associated Infrastructure: The FOF and CNF Parts, v3.5.0. *Journal of Automated Reasoning*, 43(4):337–362, 2009.

Probabilistic concepts in formal contexts*

Alexander Demin¹ Denis Ponomaryov¹ Evgeny Vityaev²
alexandredemin@yandex.ru ponom@iis.nsk.su vityaev@math.nsc.ru

Abstract

We generalize the main notions of Formal Concept Analysis with the ideas of the semantic probabilistic inference. We demonstrate that under standard restrictions, our definitions completely correspond to the original notions of Formal Concept Analysis. From the point of view of applications, we propose a method of recovering concepts in formal contexts in presence of noise on data.

1 Introduction

Assume that a scientist needs to classify some finite set of objects with respect to n attributes. The objects are observed in a number of experiments, where each of them is assigned a certain set of attributes. The results of each experiment can be represented as a table, with rows labelled by the object names, columns labelled by the attribute names, and each cell (i, j) filled iff object i has attribute j . Having results of one particular experiment it is reasonable to classify the objects in the following way: put those objects in groups which have a common set of attributes and no object out of this group has these attributes. It is well known that the pairs (object set, attribute set) of this kind can be naturally ordered and represented in a convenient way as studied in Formal Concept Analysis [2; 3] (FCA). Now assume that we know the results of the whole body of the experiments and we would like to build a classification of the objects with respect to the whole collection of data. Typically, an object may have some attribute in a number of experiments and lack this attribute in the remaining number of them. To cope with this ambiguity when building classifications, we employ the method of semantic probabilistic inference introduced in [9; 10; 11]. In this paper, we generalize the standard notion of truth of an implication on data by means of a truth valuation based on a probability measure. We define an analog of the classification unit studied in FCA in terms of fixed points of implications which hold on data

with respect to this valuation. To the best of our knowledge, there are no published papers describing similar probabilistic approaches. For instance, in [1] the main notions of Formal Concept Analysis are reformulated in terms of a probability logic, but their definitions are not generalized in the scope of FCA. By this work, we aim at establishing connections between Formal Concept Analysis and the method of semantic probabilistic inference. The contribution of this paper is the generalization of the key notions of FCA in terms of the semantic probabilistic inference.

2 Preliminaries

Let us start with basic definitions and results from Formal Concept Analysis.

Definition 1. A formal context is a triple (G, M, I) , where G and M are sets and $I \subseteq G \times M$ is a relation between the elements of G and M . The elements of G are called objects and the elements of M are called attributes of the context. We call a context finite if G and M are finite sets.

For brevity, we omit the word “formal” and call the triples (G, M, I) from the definition above contexts. Every context can be naturally represented in a tabular form, as noted in the introduction. For a context (G, M, I) , we define the operation $'$ on the subsets $A \subseteq G$ and $B \subseteq M$ as follows:

$$\begin{aligned} A' &= \{m \in M \mid \forall g \in A (g, m) \in I\}, \\ B' &= \{g \in G \mid \forall m \in B (g, m) \in I\}. \end{aligned}$$

For $g \in G$, the set $\{g\}'$ will be abbreviated by the notation g' .

Definition 2. A concept in context (G, M, I) is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$. The set A is called the extent and B the intent of the concept (A, B) .

In fact, a concept can be viewed as the classification unit which groups objects and attributes of a context.

Definition 3. A (partial) order $\leq$ on concepts is defined as follows: if (A_1, B_1) and (A_2, B_2) are concepts of a context then $(A_1, B_1) \leq (A_2, B_2)$ if $A_1 \subseteq A_2$ (or, equivalently, if $B_2 \subseteq B_1$).

Theorem. The relation $\leq$ induces a complete lattice on the set of concepts of a context, with the infimum and supremum of subsets given, respectively, by:

$$\bigwedge_{j \in J} (A_j, B_j) = \left(\bigcap_{j \in J} A_j, \left(\bigcup_{j \in J} B_j \right)' \right)$$

*This work was funded by the grant of the President of Russian Federation (grant No. MK-2037.2011.9), the Russian Foundation for Basic Research (grant No. 11-07-00560a), and Integration Projects of the Siberian Division of the Russian Academy of Sciences (grant No. 47, 111, 119).

¹ A.P. Ershov Institute of Informatics Systems, Novosibirsk, Russia

² S.L. Sobolev Institute of Mathematics, Novosibirsk, Russia

$$\bigvee_{j \in J} (A_j, B_j) = ((\bigcup_{j \in J} A_j)'', \bigcap_{j \in J} B_j).$$

The procedures of computing the complete concept lattice for a given finite context [7; 8] are one of the basic algorithms in Formal Concept Analysis. In fact, they provide a classification of objects of a context with respect to their attributes and allow for finding all possible classes.

If $K = (G, M, I)$ is a context, we may speak about the truth of the following statements on K : “all objects having attribute set $B_1 \subseteq M$ also have attributes $B_2 \subseteq M$ ”. As all properties of a context are in some sense symmetric with respect to the sets G and M , we can formulate the similar statements about subsets of G : “all attributes having the set $A_1 \subseteq G$ as their objects also have the set $A_2 \subseteq G$ ”. W.l.o.g. we consider the statements only of the first kind. In fact, they define a monotone operator, an implication, on the boolean algebra of subsets of M . If a context K is finite, then clearly the set of all such statements true on K is also finite.

Let us formalize the notion of an implication true on a context by the definitions from Chapter 2.3 in [2].

Definition 4. An implication on a set M is an ordered pair of subsets $A, B \subseteq M$ denoted as $A \rightarrow B$. The set A is called the premise and B the conclusion of the implication $A \rightarrow B$. A subset $T \subseteq M$ respects an implication $A \rightarrow B$ if $A \not\subseteq T$ or $B \subseteq T$. A family of subsets of M respects an implication $A \rightarrow B$ if every set of this family respects $A \rightarrow B$.

An implication $A \rightarrow B$ holds on a context $K = (G, M, I)$ (notation $K \models A \rightarrow B$) if $A, B \subseteq M$ and the family of sets $\{g' \mid g \in G\}$ respects $A \rightarrow B$.

The premise of an implication $A \rightarrow B$ is said to be false on a context $K = (G, M, I)$ if there is no element $g \in G$ such that $A \subseteq g'$. An implication $A \rightarrow B$ is called a tautology if $B \subseteq A$.

For a context $K = (G, M, I)$, we denote by $\text{Imp}(K)$ the set of all implications on M which hold on K . It is easy to verify that the tautologies and the set of implications whose premise is false in K are subsets of $\text{Imp}(K)$. When ambiguity does not arise, we will use the same symbol $\models$ to denote that a set or a family of sets respects an implication.

Every family L of implications on a set M defines the monotone operator $f_L : 2^M \rightarrow 2^M$ given by

$$f_L(X) = X \cup \{B \mid A \rightarrow B \in L, A \subseteq X\}.$$

Clearly, for each $X \subseteq M$, it holds $f_L(X) = X \Leftrightarrow X \models L$.

Remark 1. Let L be a family of implications on a set M . Then for each $X \subseteq M$, there exists a minimal set $Y \subseteq M$ such that $X \subseteq Y$ and $f_L(Y) = Y$.

Therefore, any family L of implications on a set M defines the operator $f_L : 2^M \rightarrow 2^M$ which for every $X \subseteq M$ gives the minimal subset $Y \subseteq M$ satisfying the conditions of the remark. Clearly, for each $X \subseteq M$, we have $f_L(X) = X \Leftrightarrow f_L(X) = X$.

Remark 2. If $K = (G, M, I)$ is a context and $A \rightarrow B$ is an implication on M then $K \models A \rightarrow B \Leftrightarrow \forall m \in B (K \models A \rightarrow \{m\})$.

In the following, we consider implications only of the form $A \rightarrow \{m\}$ and use the notation $A \rightarrow m$ for them.

If K is a context, then for every implication $A \rightarrow m \in \text{Imp}(K)$, there exists a set $\{A_0 \rightarrow m \in \text{Imp}(K) \mid A_0 \subseteq A \text{ and for each } A_1 \subseteq A, \text{ if } A_1 \subset A_0 \text{ then } A_1 \rightarrow m \notin \text{Imp}(K)\}$. For a context K , let us denote by $\text{MinImp}(K)$ the set of all implications of the form $A_0 \rightarrow m \in \text{Imp}(K)$ in which the premise A_0 is minimal in the above mentioned sense. We note that this definition is a variant of the notion of a law in [9; 10; 11].

Below, we give a slightly modified formulation of Proposition 20 from [2] which is central for results in this paper.

Proposition 1. Let $K = (G, M, I)$ be a context, $T \subseteq \text{Imp}(K)$ be the set of tautologies on M , and $F \subseteq \text{Imp}(K)$ be the set of implications whose premise is false on K . Then for every subset $B \subseteq M$, we have:

1. $f_{\text{MinImp}(K) \setminus T}(B) = B \Leftrightarrow B'' = B$;
2. if $B' \neq \emptyset$ then $f_{\text{MinImp}(K) \setminus \{F \cup T\}}(B) = B \Leftrightarrow B'' = B$.

It is straightforward by Definition 2 that for each context $K = (G, M, I)$, a subset $B \subseteq M$ is an intent of some concept in context K iff $B'' = B$. Therefore, as soon as a context $K = (G, M, I)$ is given, we have the set $\text{Imp}(K)$ of all implications which hold on K and the fixed points of the operator $f_{\text{MinImp}(K) \setminus T} : 2^M \rightarrow 2^M$ correspond exactly to the intents of the concepts of K . If we omit the set F of implications from $\text{MinImp}(K) \setminus T$ whose premise is false on K then the fixed points of $f_{\text{MinImp}(K) \setminus \{F \cup T\}} : 2^M \rightarrow 2^M$ correspond to the intents of the concepts of K excluding the single concept $(\emptyset, M)$. Because for each $B \subseteq M$, the condition $B'' \neq M$ obviously yields $B' \neq \emptyset$.

3 Probabilistic concepts on classes of contexts

In Section 2, we have defined the notion of truth of an implication on a given context. Let us now demonstrate how this notion can be generalized with a truth valuation wrt a class of contexts. In this section, we proceed to ideas of the method of semantic probabilistic inference in application to FCA. As described in [9; 10; 11], the regularities on data (in particular, implications) are formalized in this method as universal formulas of the first order language of a countable signature consisting of predicates and constants. Thus, the standard notion of implication defined in [2] is far more specific than the concept of regularity on data considered in the semantic probabilistic inference (we note that there have been studied implications in papers on FCA which also go far beyond the definitions in [2]). However, in order to show this method useful in the case of FCA, it will be convenient to stay within the standard algebraic definitions. For this reason, we further present some restriction of the method of semantic probabilistic inference in terms common in Formal Concept Analysis.

Definition 5. A class of contexts over sets G and M is a family $\mathcal{K} = \{(G, M, I_j)\}_{j \in J, j \neq \emptyset}$, where for each $j \in J$, the triple (G, M, I_j) is a context. We use the notation $\mathcal{K}(G, M)$ for a class $\mathcal{K}$ of contexts over sets G and M . A probability model of type I is a pair $\mathcal{M} = (\mathcal{K}(G, M), \rho)$, where $G \neq \emptyset$ and ρ is a probability measure on the set $\mathcal{K}$ satisfying the condition: $\forall S_1, S_2 \subseteq G \times M \quad \forall (G, M, I) \in \mathcal{K}$

$$S_1 \not\subseteq I \text{ or } S_2 \subseteq I \iff \rho(\{(G, M, I_j) \mid S_1 \cup S_2 \subseteq I_j\}) = \rho(\{(G, M, I_j) \mid S_1 \subseteq I_j\}).$$

For a subset $S \subseteq G \times M$, we call the value of the function $\nu_{\mathcal{M}}(S) = \rho(\{(G, M, I) \in \mathcal{K} \mid S \subseteq I\})$ the probability of the set S on $\mathcal{M}$.

For brevity, in this section we call the pair $(\mathcal{K}(G, M), \rho)$ from the definition above the *probability model* or simply, *model*.

Let $\mathcal{M} = (\mathcal{K}(G, M), \rho)$ be a probability model and $A \rightarrow m$ be an implication on the set M . An instantiation of $A \rightarrow m$ on the model $\mathcal{M}$ is a pair $\langle g, A \rightarrow m \rangle$, where $g \in G$.

The value of the function

$$\mu_{\mathcal{M}}(\langle g, A \rightarrow m \rangle) = \begin{cases} \frac{\nu_{\mathcal{M}}(S \cup \{\langle g, m \rangle\})}{\nu_{\mathcal{M}}(S)} & \text{if } \nu_{\mathcal{M}}(S) \neq 0, \\ & \text{where } S = \{\langle g, a \rangle \mid a \in A\}; \\ \text{undefined,} & \text{otherwise} \end{cases}$$

is called the probability of the instantiation $\langle g, A \rightarrow m \rangle$ on the model $\mathcal{M}$.

If $\mathcal{M} = (\mathcal{K}(G, M), \rho)$ is a probability model and $A \rightarrow m$ is an implication on the set M then the value of the function

$$\eta_{\mathcal{M}}(A \rightarrow m) = \begin{cases} \text{undefined} & \text{if } \forall g \in G \\ & \mu_{\mathcal{M}}(\langle g, A \rightarrow m \rangle) \\ & \text{is undefined;} \\ \inf_{g \in G} \mu_{\mathcal{M}}(\langle g, A \rightarrow m \rangle), & \text{otherwise} \end{cases}$$

is called the probability of the implication $A \rightarrow m$ on the model $\mathcal{M}$.

Remark 3. Let $\mathcal{M} = (\mathcal{K}(G, M), \rho)$ be a probability model and $A \rightarrow m$ be an implication on the set M whose probability is defined on $\mathcal{M}$. Then $\eta_{\mathcal{M}}(A \rightarrow m) = 1$ iff $\forall K \in \mathcal{K} (A \rightarrow m \in \text{Imp}(K))$.

Definition 6. Let $\mathcal{M} = (\mathcal{K}(G, M), \rho)$ be a probability model and $\text{imp}(M)$ be the set of all those implications on M whose probability is defined on $\mathcal{M}$. We call an implication $A \rightarrow m \in \text{imp}(M)$ a probabilistic law on $\mathcal{M}$, if the following conditions hold:

- $\eta_{\mathcal{M}}(A \rightarrow m) \neq 0$;
- if $A_0 \rightarrow m \in \text{imp}(M)$ and $A_0 \subset A$ then $\eta_{\mathcal{M}}(A_0 \rightarrow m) < \eta_{\mathcal{M}}(A \rightarrow m)$.

An implication $A \rightarrow m \in \text{imp}(M)$ is called a *maximally specific probabilistic law* on $\mathcal{M}$ if it is a probabilistic law on $\mathcal{M}$, $A \neq \{m\}$, and there is no probabilistic law $A_0 \rightarrow m$ on $\mathcal{M}$ such that $A \subset A_0$ and $A_0 \rightarrow m$ is not a tautology.

Remark 4. If an implication is a maximally specific probabilistic law on $\mathcal{M}$ then it is not a tautology.

Definition 7. Let $\mathcal{M} = (\mathcal{K}(G, M), \rho)$ be a probability model and $S(\mathcal{M})$ be the set of all maximally specific probabilistic laws on $\mathcal{M}$. An implication $A \rightarrow m \in S(\mathcal{M})$ is called the *strongest probabilistic law* on $\mathcal{M}$ if its probability value on $\mathcal{M}$ is maximal among all implications $B \rightarrow m \in S(\mathcal{M})$. We use the notation $D(\mathcal{M})$ for the set of all strongest probabilistic laws on $\mathcal{M}$.

Due to the the minor restrictions on the function ρ in the definition of the probability model, the existence of the maximum in the sense of Definition 7 is not guaranteed. Thus, the existence of the strongest probabilistic laws is not guaranteed

either. However, we describe further in this section a way to define a probability model (based on a finite class of finite contexts), which gives a large class of models guaranteeing the existence of such implications. Note that in general for a given m , there may exist several strongest probabilistic laws of the form $A \rightarrow m$.

Informally, every implication on a probability model can be seen as a “prediction” (wrt some measure of truth) of the fact that each object having the attributes from the premise will also have the attribute from the conclusion. Similarly to Formal Concept Analysis (recall Proposition 1), implications in the method of semantic probabilistic inference are directly related to the process of grouping objects and attributes into classification units. If data are represented by a class $\mathcal{K}$ of contexts then the choice of implications wrt their probability on a model $(\mathcal{K}, \rho)$ becomes central for generating classes on the basis of the provided data. The definition of a minimal implication (as given by the set $\text{MinImp}(K)$), probabilistic law, maximally specific and the strongest probabilistic law are adopted from the corresponding definitions in [9; 10; 11; 13] to the case of FCA. Such implications have a number of useful theoretical and practical properties which justify their application:

- the set of all minimal implications which hold on every context from a class $\mathcal{K}$ gives, in some sense, an axiomatization of this class of contexts: the implicational theory of $\mathcal{K}$ (restricted to implications with non-false premises) semantically follows from it [9; 11] (the analogue of the Duquenne-Guigues theorem on implication base [4]);
- a probabilistic law excludes the possibility that an attribute in the conclusion can be “predicted” by a proper subset of the premise with probability greater than the probability of the law itself; together with the requirement of maximal specificity, this leads in practice to grouping attributes into smaller classes, with greater probability;
- it is proved in [9; 10] that in case negative information is allowed in implications, the set of maximally specific probabilistic laws is consistent (i.e. there can not be a situation when the presence and absence of some attribute are “predicted” simultaneously);
- the strongest probabilistic laws lead to assigning an attribute to a class which “predicts” it with maximal probability; at the same time, this does not rule out situations when the same attribute can belong to different classes [12];
- the **Discovery** software tool is implemented which allows to find the above mentioned types of implications on tabular data and compute the corresponding object-attribute classes; this software has proved successful in a large number of applications [5; 9; 13].

Definition 8. Let $\mathcal{M} = (\mathcal{K}(G, M), \rho)$ be a probability model of type I. A pair of sets (A, B) is called a *probabilistic concept* of a context $(G, M, I) \in \mathcal{K}$ in model $\mathcal{M}$ if it satisfies the following conditions:

- $A \subseteq G, B \subseteq M$,

- $f_{D(\mathcal{M})}(B) = B$,
- $\exists E \subseteq B (\bar{f}_{D(\mathcal{M})}(E) = B \text{ and } E \neq \emptyset \neq E')$,
- $A = \bigcup \{E' \mid \emptyset \neq E \subseteq B, \bar{f}_{D(\mathcal{M})}(E) = B\}$,

where $'$ is the operation in context (G, M, I) . The set A is called the extent and B the intent of the probabilistic concept (A, B) .

Therefore, given a probability model $\mathcal{M} = (\mathcal{K}(G, M), \rho)$, the set of the fixed points of the operator $f_{D(\mathcal{M})}$ restricts the set of all possible probabilistic concepts of contexts from the class $\mathcal{K}$ in the model $\mathcal{M}$.

Theorem 1. Consider a context $K = (\emptyset \neq G, M, I)$ and a probability model $\mathcal{M} = (\{K\}, \rho)$. For all non-empty subsets $A \subseteq G$ and $B \subseteq M$, the pair (A, B) is a concept in context K iff (A, B) is a probabilistic concept of context K in model $\mathcal{M}$.

Let $\mathcal{K} = \{(\emptyset \neq G, M, I_j)\}_{j \in J \neq \emptyset}$ be a finite class consisting of finite contexts. We now describe a natural way to define a probability model $(\mathcal{K}, \rho)$ on the class $\mathcal{K}$. For each context $K \in \mathcal{K}$, we set $\rho(\{K\}) = 1/|J|$ and for a subset $\mathcal{C} \subseteq \mathcal{K}$ define $\rho(\mathcal{C}) = \sum_{K \in \mathcal{C}} \rho(\{K\})$.

Then ρ is a discrete probability measure on $\mathcal{K}$ and for every $S \subseteq G \times M$, we have $\nu_{\mathcal{M}}(S) = |\tilde{J}|/|J|$, where $\tilde{J}$ is the maximal subset of J satisfying the condition $\forall j \in \tilde{J} (S \subseteq I_j)$. It is easy to verify that $(\mathcal{K}, \rho)$ is indeed, a probability model. We call a model defined in this way the *frequency probability model* (of type I).

Let us illustrate the given definitions by an example.

Example 1. Consider the sets $G = \{g_1, g_2\}$, $M = \{m_1, m_2, m_3\}$, and the class $\mathcal{K} = \{(G, M, I_j)\}_{j \in \{1,2,3\}}$ consisting of three contexts given in the tabular form below.

I_1	m_1	m_2	m_3	I_2	m_1	m_2	m_3
g_1	×		×	g_1		×	×
g_2		×		g_2	×	×	
I_3	m_1	m_2	m_3				
g_1	×		×				
g_2	×	×					

Then the pairs $(\{g_1\}, \{m_1, m_2, m_3\})$ and $(\{g_1, g_2\}, \{m_1, m_2\})$ are the only probabilistic concepts of context (G, M, I_1) in the frequency probability model $\mathcal{M} = (\mathcal{K}, \rho)$.

4 Probabilistic concepts on one context

In Section 3, we have considered the notion of a probability model of type I defined on a class of contexts. In fact, every class $\mathcal{K}$ of contexts which allows to define a probability measure, raises a set of probability models and thus defines possible families of the strongest probabilistic laws. Based on such implications, we made a “prediction” of the existence of attributes for objects in an arbitrary chosen context from class $\mathcal{K}$. Similarly to this approach, we may define the strongest probabilistic laws on the basis of only one given context. For this, we need only to slightly modify Definition 5 of a probability model.

Definition 9. A probability model of type II (a probabilistic context) is a pair $\mathcal{M} = (K, \rho)$, where $K = (G, M, I)$ is a context and ρ is a probability measure on the set G satisfying the condition

$$\forall B, C \subseteq M (B' \subseteq C' \Leftrightarrow \rho((B \cup C)') = \rho(B')).$$

If $B \rightarrow m$ is an implication on the set M then the value of the function

$$\eta_{\mathcal{M}}(B \rightarrow m) = \begin{cases} \frac{\rho((B \cup \{m\})')}{\rho(B')} & \text{if } \rho(B') \neq 0 \\ \text{undefined,} & \text{otherwise} \end{cases}$$

is called the probability of $B \rightarrow m$ on the model $\mathcal{M}$.

For brevity, in this section we call the pair (K, ρ) from the definition above the *probability model* or simply, *model* and use the notation $K(G, M)$ for a context K over the set of objects G and the set of attributes M .

For a finite context $K = (\emptyset \neq G, M, I)$, a model $\mathcal{M} = (K, \rho)$ is called a *frequency probability model* (of type II) if the function ρ is defined as $\rho(\{g\}) = 1/|G|$ for every $g \in G$ and $\rho(A) = \sum_{g \in A} \rho(\{g\})$ for each subset $A \subseteq G$. We have $\forall B \subseteq M (\rho(B') = |B'|/|G|)$. Note that $\mathcal{M}$ is indeed, a model, since for all subsets $B, C \subseteq M$ it holds that $B' \subseteq C' \Leftrightarrow (B \cup C)' = B' \Leftrightarrow |(B \cup C)'| = |B'|$.

Remark 5. If $\mathcal{M} = (K(G, M), \rho)$ is a probability model and $B \rightarrow m$ is an implication on M then $\eta_{\mathcal{M}}(B \rightarrow m) = 1$ iff $B \rightarrow m \in \text{Imp}(K)$ and $B' \neq \emptyset$ (where $'$ is the operation in context K).

Let us define the notions of a probabilistic law, maximally specific probabilistic law, and the strongest probabilistic law on a model of type II in full accordance with Definitions 6 and 7. In the following, we use the same notation $D(\mathcal{M})$ as in Section 3 for the set of all strongest probabilistic laws on a model $\mathcal{M}$ of type II.

Proposition 2. Let $\mathcal{M} = (K(G, M), \rho)$ be a probability model and $S \subseteq \text{Imp}(K)$ be the set of all tautologies on M and all the implications whose premise is false on K . Then we have $\text{MinImp}(K) \setminus S \subseteq D(\mathcal{M})$.

Definition 10. Let $\mathcal{M} = (K(G, M), \rho)$ be a probability model of type II. A pair of sets (A, B) is called a *probabilistic concept* in model $\mathcal{M}$ (a concept in probabilistic context $\mathcal{M}$) if it satisfies the conditions of Definition 8.

Let $\mathcal{M} = (K, \rho)$ be a model with $K = (G, M, I)$. Consider the context $\bar{K} = (G, M, \bar{I})$, where $\bar{I} = \{< g, m > \mid g \in G, m \in \bar{f}_{D(\mathcal{M})}(g')\}$, $'$ is the operation in the K . In other words, we have $I \subseteq \bar{I}$ and the relation $\bar{I}$ is obtained from I by adding the pairs $< g, m >$ “predicted” by the implications in $D(\mathcal{M})$. To clarify the connection between the concepts in context K and probabilistic concepts in model $\mathcal{M}$, we need to note that the following statement is false in both directions: for all non-empty subsets $A \subseteq G$ and $B \subseteq M$, the pair (A, B) is a probabilistic concept in model $\mathcal{M}$ iff (A, B) is a concept in context $\bar{K}$.

Nevertheless, the following property is guaranteed which characterizes the connection between concepts in a context K and probabilistic concepts in the model $\mathcal{M} = (K, \rho)$.

Theorem 2. Every probability model $\mathcal{M} = (K(G, M), \rho)$ has the following properties:

1. for each concept (A, B) in context K with $A \neq \emptyset \neq B$, there exists a probabilistic concept (A_1, B_1) in model $\mathcal{M}$ such that $A \subseteq A_1$ and $B \subseteq B_1$;
2. if (A_1, B_1) is a probabilistic concept in model $\mathcal{M}$ then there exists a concept (A, B) in context K with $\emptyset \neq A \subseteq A_1$ and $\emptyset \neq B \subseteq B_1$. Moreover, the set A_1 is the union of the extents of some of these concepts.

Due to space limitations, we are unable to provide here schemas of computation procedures for finding the set of probabilistic laws and probabilistic concepts on a given frequency probability model $\mathcal{M} = (K, \rho)$. The algorithms, proofs, and experimental results will be given in the full version of the paper at <http://www.iis.nsk.su/persons/ponom/papers/>

5 Conclusions

It easy to note from Definition 5 and 9 that the distinction between the probability model of type I and II is rather subtle. In particular, for each model $\mathcal{M}_2 = (K, \rho_2)$ of type II with $K = (\emptyset \neq G, M, I)$, it is possible to define a model $\mathcal{M}_1 = (K, \rho_1)$ of type I such that $D(\mathcal{M}_1) = D(\mathcal{M}_2)$. It suffices to set $\mathcal{K} = \{K_g \mid g \in G, K_g = (\{h\}, M, I_g), I_g = \{< h, m > \mid < g, m > \in I\}\}$ and define $\forall \mathcal{C} \subseteq \mathcal{K} \rho_1(\mathcal{C}) = \rho_2(\{g \mid K_g \in \mathcal{C}\})$. Then for every implication $B \rightarrow m$ on M , we have $\eta_{\mathcal{M}_1}(B \rightarrow m) = \mu_{\mathcal{M}_1}(\langle h, B \rightarrow m \rangle) = \frac{\rho_1(\{K_g \mid \{< h, n > \mid n \in B \cup \{m\}\} \subseteq I_g\})}{\rho_1(\{K_g \mid \{< h, n > \mid n \in B\} \subseteq I_g\})} = \frac{\rho_2((B \cup \{m\})')}{\rho_2(B')}$ and thus $\eta_{\mathcal{M}_1}(B \rightarrow m) = \eta_{\mathcal{M}_2}(B \rightarrow m)$ which clearly, yields $D(\mathcal{M}_1) = D(\mathcal{M}_2)$. However, it is important in practice to make a distinction between the analysis of data represented by a class of contexts and analysis of data on the basis of a single given context. In the first case, we have a problem of classification of objects which are observed in a number of experiments in which every object is assigned a certain set of attributes. In the second case, the classification of objects is based on a single context which represents the whole body of experimental data on these objects. A context uniquely determines whether an object has a particular attribute and Formal Concept Analysis provides tools for building precise classification of objects on the basis of a given context. On the other hand, discovering probabilistic laws on a model over a given context allows to obtain classification units which are stable with respect to noise.

Preliminary experiments evidence that the noise of a certain level does not change the set of concepts in a context, i.e. the set of concepts with a non-empty extent and intent in a given context is equal to the set of probabilistic concepts in a new context obtained by adding noise into the initial one. There exist types of noise (a formal definition is given in [9]) such that any level of a noise of this kind does not change the set of concepts in a context; such noise is called *concept preserving* [9]. This raises the problem of characterization of these types of noise.

In the definitions of implications and probabilistic laws in this paper, the notion of negation was not present. Due to this, the formulation of Theorem 2 appeared to be weaker than expected. This is because the negation was not present in the fundamentals of Formal Concept Analysis and we

aimed at giving the most simple generalization of the basic notions of this method. The generalization of FCA according to the given ideas will allow to formalize the notions of “natural classification” and “idealization” as defined in [9]. The semantic probabilistic inference which is central in the definitions of probabilistic concepts has been first introduced for first order logic and provides a method for discovering rather complicated regularities on data in comparison to those considered in this paper. Moreover, in the relational approach described in monographs [5; 9], it is argued that the formalization of regularities in the language of first order logic is essential for analyzing the whole body of information contained in data. Some examples of such regularities are given at the Web page [13] at http://math.nsc.ru/AP/ScientificDiscovery/pages/Examples_of_rules.html

References

- [1] J. Deogun, L. Jiang, Y. Xie, V. Raghavan. Probability logic modeling of knowledge discovery in databases. *In Proc. ISMIS'2003*, pp. 402–407, 2003.
- [2] B. Ganter, R. Wille. Formal Concept Analysis: Mathematical Foundations. Springer Verlag, 1999.
- [3] B. Ganter, G. Stumme, R. Wille (ed.). Formal Concept Analysis: Foundations and Applications. Springer Verlag, 2005.
- [4] J.- L. Guigues, V. Duquenne. Familles minimales d'implications informatives resultant d'un tableau de données binaires. *Math. Sci. Humaines*, 95, pp. 5–18, 1986.
- [5] B. Kovalerchuk, E. Vityaev. Data Mining in Finance: Advances in Relational and Hybrid methods. Kluwer, 2000.
- [6] S. Kuznetsov. On the intractability of computing the Duquenne–Guigues base. *J. UCS*, 10(8), pp. 927–933, 2004.
- [7] S. Kuznetsov, S. Obiedkov. Comparing performance of algorithms for generating concept lattices *J. Exp. Theor. Artif. Intell.*, 14(2-3), pp. 189–216, 2002.
- [8] D. Merwe, S. Obiedkov, D. Kourie. AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. *In Proc. ICFCA'2004*, pp. 372–385, 2004.
- [9] E. Vityaev. Knowledge Discovery. Computational Cognition. Models of Cognitive Processes. Novosibirsk State University, 293 p., 2006 (*in Russian*)
- [10] E. Vityaev. The logic of prediction. *In Proc. 9th Asian Logic Conference*, pp. 263–276, 2006.
- [11] E. Vityaev, B. Kovalerchuk. Empirical theories discovery based on the Measurement Theory. *Mind and Machine*, 14(4), pp. 551–573, 2005.
- [12] E. Vityaev, S. Smerdov. New definition of prediction without logical inference. *In Proc. CI 2009*, pp. 48–54, 2009.
- [13] Scientific Discovery website: <http://math.nsc.ru/AP/ScientificDiscovery/>

Versioning Based on Logical Difference for Lightweight Description Logic Terminologies

William Gatens and Boris Konev and Michel Ludwig and Frank Wolter

Department of Computer Science, University of Liverpool, UK

w.gatens@student.liverpool.ac.uk, {konev, michel.ludwig, wolter}@liverpool.ac.uk

Abstract

We study a logic-based approach to versioning of ontologies. Under this view, ontologies provide answers to queries about some vocabulary of interest. The difference between two versions of an ontology is given by the set of queries that receive different answers. We investigate this approach for terminologies given in the description logic $\mathcal{EL}$ extended with role inclusions and range restrictions for three distinct types of queries: subsumption, instance, and conjunctive queries. We present an implementation, CEX2, that computes finite representations of the respective difference between terminologies.

1 Introduction

Terminologies are lightweight ontologies that are used to provide a common vocabulary for a domain of interest together with descriptions of the meaning of terms built from the vocabulary and relationships between them. They are being used in areas such as medical informatics, bio-informatics, and the semantic web to capture domain semantics and promote interoperability. Terminologies are often large and complex. For example, the widely used medical terminology SNOMED CT (Systematized Nomenclature of Medicine Clinical Terms) contains more than 300 000 term definitions [(IHTSDO), 2008]. Another example is the National Cancer Institute ontology (NCI) consisting of more than 60 000 axioms [Golbeck *et al.*, 2003]. Engineering, maintaining, and using such terminologies is a complex and laborious task, which is practically unfeasible without appropriate tool support. In this contribution, we focus on a principled logic-based approach to support for *terminology versioning*.

Dealing with multiple versions of the same information unit is nothing new in computing, and version control is a well established computer technology. Although modern version control systems provide a range of operations including support for collaborative development, branching, merging, etc., these operations extend and rely on the basic operations of *detecting and representing the differences* between versions. In this paper, we focus on this basic problem of versioning.

The need for versioning support is recognised by the ontology research community and ontology users, and a large number of approaches and tools have been developed. In our

review of currently existing support for ontology versioning, we distinguish *three* approaches and describe them according to the difference between ontologies they compute:

1. versioning based on syntactic difference (syntactic diff);
2. versioning based on structural difference (structural diff);
3. versioning based on logical difference (logical diff).

The *syntactic diff* underlies most existing version control systems used in software development [Conradi and Westfechtel, 1998] (such as, for example, RCS, CVS, SCCS). It works with text files and represents the difference between versions as blocks of text present in one version but not another, ignoring any meta-information about the document. As observed already in [Noy and Musen, 2002], ontology versioning cannot rely on a purely syntactic diff operation since many syntactic differences (e.g., the order of ontology axioms) do not affect the semantics of ontologies. Therefore, ontology versioning based on syntactic difference is essentially limited to comparing rather informal change logs [Oliver *et al.*, 1999].

The *structural diff* extends the syntactic diff by taking into account some structural meta-information about the distinct versions of the ontologies compared. It has been suggested for dealing with structured and hierarchical documents such as UML diagrams, database schemas, or XML documents (see e.g. [Ohst *et al.*, 2003] and references within). For ontologies, the main characteristic of the structural diff is that it regards them as structured objects, such as an *is-a* taxonomy [Noy and Musen, 2002], a set of RDF triplets [Klein *et al.*, 2002], or a set of class defining axioms [Redmond *et al.*, 2008; Jimnez-Ruiz *et al.*, 2011]. Changes to ontologies are mostly described in terms of structural operations, for example, adding or deleting a class, extending a class, renaming slots, moving a class from one place in the hierarchy to another, adding or deleting an axiom, class renaming, etc.; sometimes basic logical properties of ontologies, e.g., the equivalence of different structural forms of concepts, are also taken into account [Jimnez-Ruiz *et al.*, 2011]. Ontology versioning based on structural diff of some form is available in most current ontology editors and ontology management systems either natively or through plugins [Noy and Musen, 2002; Klein *et al.*, 2002; Jimnez-Ruiz *et al.*, 2011].

Though very helpful, the structural diff still has the deficiency of having no unambiguous semantic foundation and

being syntax dependent. Moreover, it is tailored towards applications of ontologies which are based on the induced concept hierarchy (or some mild extension of it), but does not capture modern applications such as ontology based data access (OBDA) [Poggi *et al.*, 2008; Lutz *et al.*, 2009] in which ontologies are used to provide a user-oriented view of the data and make it accessible via queries formulated solely in the language of the ontology without any knowledge of the actual structure of the data.

The *logical diff* has only recently been introduced in [Konev *et al.*, 2008; Kontchakov *et al.*, 2010] and completely abstracts from the representation of the ontology. Here, an ontology is regarded as a set of axioms formulated in a logical language with a formal and unambiguous semantics. Under this view, ontologies provide answers to queries about some vocabulary of interest. Typical queries include *subsumption* between concepts and, if the ontology is used to access instance data, *instance checking* and *conjunctive query answering*. The logical diff is motivated by this view. If two versions of an ontology give the same answers to a class of queries relevant to an application domain, they may be deemed to have no difference regardless of their syntactic or structural form; and queries producing different answers from the versions may be considered as a characterisation of the difference itself. In this way one can, for example, define exactly the differences visible when querying instance data or exactly the differences expressed by subsumptions between concepts.

To make this approach work in practice, at least two problems have to be solved:

- For many ontology languages and classes of queries the computational complexity of even detecting if two ontology versions differ is at least one exponential harder than ontology classification and is sometimes undecidable [Lutz *et al.*, 2007; Lutz and Wolter, 2010; Grau *et al.*, 2008].
- If the set of queries producing different answers from the two versions is not empty, it is typically infinite and, therefore, cannot be presented to the user as such. Thus, techniques to succinctly characterise its elements and present them to the user are required.

The aim of this paper is to provide solutions to these problems for terminologies given in the description logic $\mathcal{ELH}^r$ that extends the description $\mathcal{EL}$ underlying the OWL2 EL profile with role inclusions and domain and range restrictions [Baader *et al.*, 2008]. This paper extends the results of [Konev *et al.*, 2008] for “pure” $\mathcal{EL}$ by firstly handling those language extensions, vital for versioning SNOMED CT and NCI terminologies, and secondly by developing algorithms computing the instance- and conjunctive query-based logical diff. In contrast to $\mathcal{EL}$ itself, for the latter there are complex differences that are not visible when only subsumption queries are considered. We also present an extension, CEX2, of the CEX system introduced in [Konev *et al.*, 2008] for computing the logical difference between $\mathcal{EL}$ terminologies and a plugin for Protégé. CEX2 also extends the functionality of the OwlDiff plugin [Křemen *et al.*,], which implements the algorithms from [Konev *et al.*, 2008] and does not cover $\mathcal{ELH}^r$ nor instance differences.

Concept	Translation	Inclusion	Translation
$\top$	$x = x$	$C \sqsubseteq D$	$\forall x (C^\sharp(x) \rightarrow D^\sharp(x))$
A	$A(x)$	$C \equiv D$	$\forall x (C^\sharp(x) \leftrightarrow D^\sharp(x))$
$C \sqcap D$	$C^\sharp(x) \wedge D^\sharp(x)$	$r \sqsubseteq s$	$\forall xy (r(x, y) \rightarrow s(x, y))$
$\exists r.C$	$\exists y (r(x, y) \wedge C^\sharp(y))$		
$\text{dom}(r)$	$\exists y (r(x, y))$		
$\text{ran}(r)$	$\exists y (r(y, x))$		

Figure 1: Standard Translation

2 Preliminaries

Let N_C , N_R , and N_I be countably infinite and mutually disjoint sets of concept names, role names, and individual names. $\mathcal{EL}$ -concepts C are built according to the rule

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

where $A \in N_C$, $r \in N_R$, and C, D range over $\mathcal{EL}$ -concepts. The set of $\mathcal{ELH}^r$ -inclusions consists of *concept inclusions* $C \sqsubseteq D$ and *concept equations* $C \equiv D$, *domain restrictions* $\text{dom}(r) \sqsubseteq C$, *range restrictions* $\text{ran}(r) \sqsubseteq C$ and *role inclusions* $r \sqsubseteq s$, where C, D are $\mathcal{EL}$ -concepts and $r, s \in N_R$. An $\mathcal{ELH}^r$ -TBox $\mathcal{T}$ is a finite set of $\mathcal{ELH}^r$ -inclusions. An $\mathcal{ELH}^r$ -TBox is called an $\mathcal{ELH}^r$ -terminology if all its concept inclusions and equations are of the form $A \sqsubseteq C$ and $A \equiv C$ and no concept name occurs more than once on the left hand side.

In description logic, instance data are represented by *ABox assertions* of the form $A(a)$ and $r(a, b)$, where $a, b \in N_I$, $A \in N_C$, and $r \in N_R$. An *ABox* is a finite set of ABox-assertions. By $\text{obj}(\mathcal{A})$ we denote the set of individual names in $\mathcal{A}$. A *knowledge base* (KB) is a pair $(\mathcal{T}, \mathcal{A})$ consisting of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. Assertions of the form $C(a)$ and $r(a, b)$, where $a, b \in N_I$, C an $\mathcal{EL}$ -concept, and $r \in N_R$, are called instance assertions. To define the semantics of $\mathcal{ELH}^r$ we make use of the fact that $\mathcal{ELH}^r$ expressions can be regarded as formulas in FO, where FO denotes the set of first-order predicate logic formulas with equality using unary predicates in N_C , binary predicates in N_R , and constants from N_I ; see Figure 1. In what follows, we will not distinguish between $\mathcal{ELH}^r$ expressions and their translation into FO and regard TBoxes, ABoxes and KBs as finite subsets of FO. Thus, we use $\mathcal{T} \models \varphi$ to denote that φ follows from $\mathcal{T}$ in first-order logic even if φ is an $\mathcal{ELH}^r$ -inclusion and $\mathcal{T}$ a subset of FO. FO (and, therefore, $\mathcal{ELH}^r$) is interpreted in models $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where the *domain* $\Delta^\mathcal{I}$ is a non-empty set, and $\cdot^\mathcal{I}$ is a function mapping each concept name A to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$, each role name r to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and each individual name a to an element $a^\mathcal{I} \in \Delta^\mathcal{I}$.

The most important ways of querying $\mathcal{ELH}^r$ -TBoxes and KBs are

- *subsumption*: check whether $\mathcal{T} \models \alpha$, for an $\mathcal{ELH}^r$ -inclusion α and TBox $\mathcal{T}$,
- *instance checking*: check whether $(\mathcal{T}, \mathcal{A}) \models \alpha$ for an instance assertion α and KB $(\mathcal{T}, \mathcal{A})$, and
- *conjunctive query answering*.

To define latter, call a first-order formula $q(\vec{x})$ a *conjunctive query* if it is of the form $\exists \vec{y} \psi(\vec{x}, \vec{y})$, where ψ is a conjunction of expressions $A(t)$ and $r(t_1, t_2)$ with t, t_1, t_2 drawn from N_I and the sequences of variables $\vec{x}$ and $\vec{y}$. If $\vec{x}$ has length k ,

then a sequence $\vec{a}$ of elements of $\text{obj}(\mathcal{A})$ of length k is called a *certain answer* to $q(\vec{x})$ of a KB $(\mathcal{T}, \mathcal{A})$ if $(\mathcal{T}, \mathcal{A}) \models q(\vec{a})$.

3 Logical Difference

In this section, we introduce three notions of logical difference between TBoxes and the derived notion of Σ -inseparability. Intuitively, the logical difference between two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ should be the set of all ‘relevant formulas’ φ such that $\mathcal{T}_1 \models \varphi$ and $\mathcal{T}_2 \not\models \varphi$ or vice versa. Of course, which formulas φ are relevant depends on the application domain. In many applications only subsumptions between concepts are relevant, but if TBoxes are employed to access instance data, then answers to instance or even conjunctive queries can be relevant as well. In addition, in applications of large-scale terminologies such as SNOMED CT and NCI only a very small subset of the vocabulary of the terminology is relevant. Thus, a meaningful notion of logical difference should take into account only those formulas that are given in a certain *signature* of interest, where a signature Σ is a subset of $N_C \cup N_R$. Given a concept, role, concept inclusion, TBox, ABox, FO-sentence, set of FO-sentences E , we denote by $\text{sig}(E)$ the signature of E , that is, the set of concept and role names occurring in it. We use the term $\mathcal{ELH}_\Sigma^r$ -inclusion (Σ -ABox, Σ -query, etc.) to denote $\mathcal{ELH}^r$ -inclusions (ABoxes, queries, etc.) whose signature is contained in Σ .

The first notion of logical difference we introduce corresponds to applications in which only subsumptions are relevant.

Definition 1 (Σ -concept difference). *The Σ -concept difference between $\mathcal{ELH}^r$ TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ is the set $\text{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ of all $\mathcal{ELH}_\Sigma^r$ -inclusions α such that $\mathcal{T}_1 \models \alpha$ and $\mathcal{T}_2 \not\models \alpha$. We say that $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ -concept inseparable, in symbols $\mathcal{T}_1 \equiv_\Sigma^c \mathcal{T}_2$, if $\text{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \text{cDiff}_\Sigma(\mathcal{T}_2, \mathcal{T}_1) = \emptyset$.*

Σ -concept inseparability between $\mathcal{T}_1$ and $\mathcal{T}_2$ means that $\mathcal{T}_1$ can be replaced by $\mathcal{T}_2$ in any application that is only concerned with $\mathcal{ELH}_\Sigma^r$ -inclusions. As the following example shows, however, Σ -concept inseparable terminologies can give different answers for the same instance query and data.

Example 2. Let $\mathcal{T}_1 = \{\text{ran}(r) \sqsubseteq A_1, \text{ran}(s) \sqsubseteq A_2, B \equiv A_1 \sqcap A_2\}$, $\mathcal{T}_2 = \emptyset$, $\Sigma = \{r, s, B\}$. One can show that $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ -concept inseparable. However, for the Σ -ABox $\mathcal{A} = \{r(a_0, b), s(a_1, b)\}$ we have $(\mathcal{T}_1, \mathcal{A}) \models B(b)$ but $(\mathcal{T}_2, \mathcal{A}) \not\models B(b)$.

To take into account the differences between TBoxes that are relevant if TBoxes are used to access instance data, we consider the Σ -instance difference:

Definition 3 (Σ -instance difference). *The Σ -instance difference between TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ is the set $\text{iDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ of pairs of the form $(\mathcal{A}, \alpha)$, where $\mathcal{A}$ is a Σ -ABox and α is Σ -instance assertion such that $(\mathcal{T}_1, \mathcal{A}) \models \alpha$ and $(\mathcal{T}_2, \mathcal{A}) \not\models \alpha$. We say that $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ -instance inseparable, in symbols $\mathcal{T}_1 \equiv_\Sigma^i \mathcal{T}_2$, if $\text{iDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \text{iDiff}_\Sigma(\mathcal{T}_2, \mathcal{T}_1) = \emptyset$.*

In contrast to $\mathcal{ELH}^r$, it has been shown in [Lutz and Wolter, 2010] that for $\mathcal{EL}$ -TBoxes there is no difference between Σ -concept inseparability and Σ -instance inseparability:

Theorem 4 (Lutz and Wolter, 2010). *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be $\mathcal{EL}$ -TBoxes and Σ a signature. Then $\mathcal{T}_1 \equiv_\Sigma^c \mathcal{T}_2$ iff $\mathcal{T}_1 \equiv_\Sigma^i \mathcal{T}_2$.*

Sometimes, instance queries are not sufficiently expressive, and conjunctive queries are employed. In that case, the following notion of difference is appropriate.

Definition 5 (Σ -query-difference). *The Σ -query difference between TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ is the set $\text{qDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ of pairs of the form $(\mathcal{A}, q(\vec{a}))$, where $\mathcal{A}$ is a Σ -ABox, and $q(\vec{x})$ a Σ -conjunctive query, and $\vec{a}$ a tuple of individual names in $\mathcal{A}$ such that $(\mathcal{T}_1, \mathcal{A}) \models q(\vec{a})$ and $(\mathcal{T}_2, \mathcal{A}) \not\models q(\vec{a})$.*

$\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ -query inseparable, in symbols $\mathcal{T}_1 \equiv_\Sigma^q \mathcal{T}_2$, if $\text{qDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \text{qDiff}_\Sigma(\mathcal{T}_2, \mathcal{T}_1) = \emptyset$.

As observed in [Lutz and Wolter, 2010] already, even for $\mathcal{EL}$ Σ -instance inseparability does not imply Σ -query inseparability. The following is a simple example.

Example 6. Let $\mathcal{T}_1 = \{A \sqsubseteq \exists r.B\}$, $\mathcal{T}_2 = \emptyset$, $\Sigma = \{A, B\}$. Then $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ -instance inseparable, but they are not Σ -query inseparable. Consider the Σ -ABox $\mathcal{A} = \{A(a)\}$ and the Σ -query $q = \exists x B(x)$. Then $(\mathcal{T}_1, \mathcal{A}) \models q$ but $(\mathcal{T}_2, \mathcal{A}) \not\models q$.

It is shown in [Lutz and Wolter, 2010] that Example 6 is essentially the only situation in which there is a difference between Σ -instance inseparability and Σ -query inseparability in $\mathcal{EL}$: the two notions become equivalent for $\mathcal{EL}$ if the universal role is admitted in instance queries (e.g., the conjunctive query $\exists x B(x)$ corresponds to the instance query $\exists u.B(a)$ for the universal role u). In contrast, for $\mathcal{ELH}^r$ there are more subtle differences between the instance and the query case.

Example 7. Let $\mathcal{T}_1 = \{A \sqsubseteq \exists s.\top, s \sqsubseteq r_1, s \sqsubseteq r_2\}$, $\mathcal{T}_2 = \{A \sqsubseteq \exists r_1.\top \sqcap \exists r_2.\top\}$, $\Sigma = \{A, r_1, r_2\}$. Then $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ -concept and Σ -instance inseparable, but they are not Σ -query inseparable. To show the latter, let $\mathcal{A} = \{A(a)\}$ and let $q = \exists x(r_1(a, x) \wedge r_2(a, x))$. Then $(\mathcal{T}_1, \mathcal{A}) \models q$ but $(\mathcal{T}_2, \mathcal{A}) \not\models q$.

Note that one can easily show for all $\mathcal{ELH}^r$ terminologies $\mathcal{T}_1$ and $\mathcal{T}_2$ and all signatures Σ :

$$\mathcal{T}_1 \equiv_\Sigma^q \mathcal{T}_2 \Rightarrow \mathcal{T}_1 \equiv_\Sigma^i \mathcal{T}_2 \Rightarrow \mathcal{T}_1 \equiv_\Sigma^c \mathcal{T}_2.$$

Having introduced the three notions of logical difference we are interested in, we now face two problems: how to detect whether there is a difference between $\mathcal{ELH}^r$ terminologies and, if so, how to represent the differences.

4 Σ -Concept Difference

We start with the concept difference. The following lemma states that for any $\varphi \in \text{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ one can find a “primitive” representative $\varphi' \in \text{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ of φ .

Lemma 8. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be $\mathcal{ELH}^r$ -terminologies and Σ a signature. If $\varphi \in \text{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, then either there exist $\{r, s\} \subseteq \text{sig}(\varphi)$ with $r \sqsubseteq s \in \text{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ or one of*

(i) $C \sqsubseteq A$ or $A \sqsubseteq D$

(ii) $\exists r.C \sqsubseteq \exists r.A$

(iii) $\text{dom}(r) \sqsubseteq D$ or $\text{ran}(r) \sqsubseteq D$

is a member of $\text{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, where $r \in \text{sig}(\varphi)$, $A \in \text{sig}(\varphi)$ is a concept name and C, D are $\mathcal{EL}$ concepts occurring in φ ; the size of C and D is at most exponential in $\mathcal{T}_1$ and $\mathcal{T}_2$.

The bounds on the size of C and D are optimal: in Lemma 8, in the worst case one can only find concepts C and D of exponential size in $\mathcal{T}_1$ and $\mathcal{T}_2$. To obtain a finite, uniquely determined, and high-level representation of the Σ -concept difference, we call $x \in \Sigma$ a Σ -concept difference witness if either x is a concept name A satisfying condition (i) or condition (ii); or x is a role name r satisfying condition (iii). (Σ -differences of the form $r \sqsubseteq s$ can be easily computed and represented, and will not be further analysed in this paper.)

Theorem 9. *The set of all Σ -concept difference witnesses can be computed in polynomial time.*

Σ -concept difference witnesses can be regarded as representatives of primitive $\text{cDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$ elements. In particular, if there are no difference witnesses and no role inclusions in $\text{cDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$, then $\text{cDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$ and we obtain:

Corollary 10. *For $\mathcal{ELH}^r$, Σ -concept inseparability is decidable in polynomial time.*

Our proofs of these results extends the Gentzen-style sequent calculus of [Hofmann, 2005] for $\mathcal{EL}$ to $\mathcal{ELH}^r$ and then employs the extension to analyse derivations in $\mathcal{ELH}^r$.

5 Data-Observable Differences

As we have seen above, there are differences between TBoxes that are visible using ABoxes but not visible using $\mathcal{ELH}^r$ -inclusions. The following result shows how one can represent the additional members of the Σ -instance difference.

Lemma 11. *If $(\mathcal{A}, \alpha) \in \text{iDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$, then*

- $\varphi \in \text{cDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$, for some $\mathcal{ELH}^r$ -inclusion φ such that $\text{sig}(\varphi) \subseteq \text{sig}(\mathcal{A}) \cup \text{sig}(\alpha)$ or
- $(A', A(b)) \in \text{iDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$, for some concept name $A \in \text{sig}(\alpha)$, $A' \subseteq \mathcal{A}$ of polynomial size in $\mathcal{T}_1$ and $\mathcal{T}_2$ and some $b \in \text{obj}(A')$.

Similarly to the Σ -concept difference case, call a concept name $A \in \Sigma$ satisfying Point 2 of Lemma 11 for some ABox $\mathcal{A}'$ a Σ -instance difference witness.

Theorem 12. *The set of all Σ -instance difference witnesses can be computed in polynomial time. In particular, for $\mathcal{ELH}^r$, Σ -instance inseparability is decidable in polynomial time.*

Note that, in addition, for any Σ -instance difference witness A , one can compute in polynomial time some $\mathcal{A}'$ with $(\mathcal{A}', A(b)) \in \text{iDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$, for some b .

The following lemma shows how one can represent members of the Σ -query difference that are not members of the Σ -instance difference.

Lemma 13. *If $(\mathcal{A}, q(\vec{a})) \in \text{qDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$, then*

- $(\mathcal{A}, \alpha) \in \text{iDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$, for some instance assertion α with $\text{sig}(\alpha) \subseteq \text{sig}(q)$ or
- $(\mathcal{A}', q'(b)) \in \text{qDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$, for some ABox $\mathcal{A}'$ of cardinality at most two and $\text{sig}(\mathcal{A}') \subseteq \text{sig}(\mathcal{A})$, conjunctive query $q'(x)$ with $\text{sig}(q') \subseteq \text{sig}(q)$ and $b \in \text{obj}(\mathcal{A}')$.

Call an ABox $\mathcal{A}'$ satisfying Point 2 of Lemma 13 for some conjunctive query $q'(b)$ a Σ -query difference witness.

Theorem 14. *The set of all Σ -query difference witnesses can be computed in polynomial time. In particular, for $\mathcal{ELH}^r$, Σ -query inseparability is decidable in polynomial time.*

The proofs of Lemmas 11 and 13 employ model-theoretic methods and the sequent calculus developed for $\mathcal{ELH}^r$.

We close this section with the observation that tractability of deciding Σ -inseparability is extremely non-robust under extensions of the language. For example, one can show that the addition of disjointness assertions $A \sqcap B \sqsubseteq \perp$, A, B concept names, to $\mathcal{EL}$ -terminologies makes the three inseparability notions discussed in this section NP-hard. Inseparability becomes undecidable if general role inclusions $r_1 \circ \dots \circ r_n \sqsubseteq r$ are added to $\mathcal{EL}$ terminologies. The complexity of inseparability for simpler conditions on roles (such as transitivity) is unknown.

6 Systems and Experiments

We have implemented polynomial time algorithms computing difference witnesses for $\text{cDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$ and $\text{iDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$ for $\mathcal{ELH}^r$ -terminologies in the CEX2 tool¹. CEX2 also computes representatives, in the sense of Lemmas 8 and 11, of $\text{cDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$ and $\text{iDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$, respectively, which are invaluable for understanding the difference witnesses. Note that CEX2 currently cannot analyse the Σ -query difference $\text{qDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$ between $\mathcal{ELH}^r$ -terminologies. CEX2 extends CEX [Konev *et al.*, 2008], which can only enumerate Σ -concept difference witnesses for $\mathcal{EL}$ terminologies. The performance of CEX2 is similar to that of CEX, for an evaluation of which we refer the reader to [Konev *et al.*, 2008]. We applied CEX2 to compare the January 2009 (SM09a) and July 2009 (SM09b) versions of SNOMED CT. SM09a contains 310013 concept names and 62 role names, whereas SM09b contains 307693 concept names and the same role names. For our experiments we used 159 signatures ranging over so called SNOMED CT subsets used in the UK for the deployment of SNOMED CT in specific areas.

We illustrate the results on a typical subset called “Specimen Material Type”². Its signature contains 6510 concept names shared with SM09b, which we extended with all 62 SNOMED CT role names. The output of CEX2 shows that

- (i) $\text{cDiff}_{\Sigma}(\text{SM09b}, \text{SM09a}) = \text{iDiff}_{\Sigma}(\text{SM09b}, \text{SM09a})$;
- (ii) there are no differences regarding role inclusions;
- (iii) there are 10 concept names A such that there are concept inclusions of the form $C \sqsubseteq A$ contained in the set $\text{cDiff}_{\Sigma}(\text{SM09b}, \text{SM09a})$;
- (iv) there are 46 concept names A such that there are concept inclusions of the form $A \sqsubseteq D$ in $\text{cDiff}_{\Sigma}(\text{SM09b}, \text{SM09a})$.

In Point (iii) and (iv), the longest representatives C, D computed by CEX2 had twelve concept and role name occurrences (thus were far smaller than the exponential worst case suggests). The relatively small number of Σ -concept difference witnesses allows one to analyse the inclusions from

¹Available under an open-source license at <http://www.csc.liv.ac.uk/~michel/software/cex2/>

²Additional experimental results can be found at <http://www.csc.liv.ac.uk/~michel/arcoe11/>

- (1) $\text{LaboratoryTest} \sqsubseteq \text{LaboratoryProcedure} \sqcap \text{EvaluationProcedure}$
- (2) $\text{BloodTest} \equiv \text{LaboratoryTest} \sqcap \exists \text{roleGroup} . \exists \text{hasSpecimen} . \text{BloodSpecimen}$
- (3) $\text{VenipunctureForBloodTest} \equiv (\exists \text{roleGroup} . \exists \text{hasFocus} . \text{BloodTest}) \sqcap \text{Venipuncture}$
 $\sqcap (\exists \text{roleGroup} . ((\exists \text{procedureSiteDirect} . \text{VenousStructure}) \sqcap (\exists \text{method} . \text{PunctureAction})))$

Figure 2: Minimal Axiom Set

$\text{cDiff}_{\Sigma}(\text{SM09b}, \text{SM09a})$ computed by CEX2 further. For example, to explain and understand why

- (*) $\text{VenipunctureForBloodTest}$
 $\sqsubseteq \exists \text{roleGroup} . \exists \text{hasFocus} . \text{EvaluationProcedure}$

is a member of the Σ -difference, we have computed, using axiom pinpointing [Baader *et al.*, 2007], a minimal set of axioms from SM09b which entails the concept inclusion above; the set is shown in Fig. 2. Axioms 2 and 3 are in both terminologies, but SM09a contains

$$\text{LaboratoryTest} \sqsubseteq \text{LaboratoryProcedure}$$

instead of Axiom 1. Concept and role names from Σ are shaded in gray. It can be seen that the interaction between Σ -concepts heavily depends on inclusions that are built up mainly from non- Σ -concepts; actually none of inclusions required to derive (*) is a Σ -inclusion.

CEX2 is a text-based tool. In order to make it more accessible to ontology users, a Protégé plugin, LogDiffViz³, was created, which calls CEX2 and visualises both ontology versions and the differences as a hierarchical structure. LogDiffViz also provides basic axiom pinpointing. The plugin is distributed as a self-contained Java archive file (JAR) in which CEX2 is bundled.

7 Conclusion

We have presented a theoretical foundation for ontology versioning based on the logical diff for $\mathcal{ELH}^r$ terminologies. The algorithms have been implemented in the CEX2 tool and applied to SNOMED CT. We facilitate the use of the CEX2 tool through a Protégé plugin. It would be of interest to extend the theory and implementation further so as to cover more expressive role inclusions (e.g., transitive roles).

References

- [Baader *et al.*, 2007] F. Baader, R. Peñaloza, and B. Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In *Proc. KI*, pages 52–67, 2007.
- [Baader *et al.*, 2008] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope further. In K. Clark and P. F. Patel-Schneider, editors, In *Proc. OWLED*, 2008.
- [Conradi and Westfechtel, 1998] R. Conradi and B. Westfechtel. Version models for software configuration management. *ACM Comput. Surv.*, 30(2):232–282, 1998.
- [Golbeck *et al.*, 2003] J. Golbeck, G. Fragaso, F. Hartel, J. Hendler, B. Parsia, and J. Oberhaler. The national cancer institute’s thesaurus and ontology. *J. of Web Semantics*, pages 75–80, 2003.
- [Grau *et al.*, 2008] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research*, 31:273–318, 2008.
- [Hofmann, 2005] M. Hofmann. Proof-theoretic approach to description-logic. In *Proc. LICS*, pages 229–237, 2005.
- [IHTSDO], 2008] International Health Terminology Standards Development Organisation (IHTSDO). *SNOMED CT User Guide*. 2008. Available from <http://www.ihtsdo.org/snomed-ct/snomed-ct-publications/>.
- [Jimnez-Ruiz *et al.*, 2011] E. Jimnez-Ruiz, B. Cuenca Grau, I. Horrocks, and R. Berlanga Llavori. Supporting concurrent ontology development: Framework, algorithms and tool. *Data Knowl. Eng.*, 70(1):146–164, 2011.
- [Klein *et al.*, 2002] M. C. A. Klein, D. Fensel, A. Kiryakov, and D. Ognyanov. Ontology versioning and change detection on the web. In *Proc. EKAW, LNCS*, pages 197–212, 2002.
- [Konev *et al.*, 2008] B. Konev, D. Walther, and F. Wolter. The logical difference problem for description logic terminologies. In *Proc. IJCAR, LNAI*, pages 259–274, 2008.
- [Kontchakov *et al.*, 2010] R. Kontchakov, F. Wolter, and M. Zhakharyashev. Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artif. Intell.*, 174(15):1093–1141, 2010.
- [Křemen *et al.*,] P. Křemen, M. Schwarz, Z. Kouba, M. Šmíd, J. Abrahamčík, J. Pufler, and P. Butka. <http://krizik.felk.cvut.cz/km/owlldiff/>.
- [Lutz and Wolter, 2010] C. Lutz and F. Wolter. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *J. Symb. Comput.*, 45(2):194–228, 2010.
- [Lutz *et al.*, 2007] C. Lutz, D. Walther, and F. Wolter. Conservative extensions in expressive description logics. In *Proc. IJCAI*, pages 453–458, 2007.
- [Lutz *et al.*, 2009] C. Lutz, D. Toman, and F. Wolter. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *Proc. IJCAI*, pages 2070–2075, 2009.
- [Noy and Musen, 2002] N. F. Noy and M. A. Musen. PromptDiff: A fixed-point algorithm for comparing ontology versions. In *Proc. AAAI/IAAI*, pages 744–750, 2002.
- [Ohst *et al.*, 2003] D. Ohst, M. Welle, and U. Kelter. Differences between versions of UML diagrams. In *Proc. ESEC / SIGSOFT FSE*, pages 227–236, 2003.
- [Oliver *et al.*, 1999] D. E. Oliver, Y. Shahar, E. H. Shortliffe, and M. A. Musen. Representation of change in controlled medical terminologies. *Artif. Intell. in Medicine*, 15(1):53–76, 1999.
- [Poggi *et al.*, 2008] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. Data Semantics*, 10:133–173, 2008.
- [Redmond *et al.*, 2008] T. Redmond, M. Smith, N. Drummond, and T. Tudorache. Managing change: An ontology version control system. In *Proc. OWLED*, 2008.

³Available from <http://www.csc.liv.ac.uk/~cs8wg/LogDiffViz/>

SemCaDo: a serendipitous causal discovery algorithm for ontology evolution

Montassar Ben Messaoud

LARODEC, Institut Supérieur de Gestion,
LINA, Ecole Polytechnique
de l'Université de Nantes.
benmessaoud.montassar@hotmail.fr

Philippe Leray

LINA, UMR 6241, France,
Ecole Polytechnique de l'Université de Nantes.
philippe.leray@univ-nantes.fr

Nahla Ben Amor

LARODEC, 41, Avenue de la liberté,
2000 Le Bardo, Tunisie.
nahla.benamor@gmx.fr

Abstract

With the rising need to reuse the existing knowledge when learning Causal Bayesian Networks (CBNs), the ontologies can supply valuable semantic information to make further interesting discoveries with the minimum expected cost and effort. In this paper, we propose a cyclic approach in which we make use of the ontology in an interchangeable way. The first direction involves the integration of semantic knowledge to anticipate the optimal choice of experimentations via a serendipitous causal discovery strategy. The second complementary direction concerns an enrichment process by which it will be possible to reuse these causal discoveries, support the evolving character of the semantic background and make an ontology evolution.

1 Introduction

Bayesian networks (BNs), first introduced by Pearl [Pearl, 1988], are compact graphical probabilistic models that are capable of modeling domains comprising uncertainty. Due to the Markov equivalence property, when learning the Completed Partially Directed Acyclic Graph (CPDAG) from observational data and randomly choosing one possible complete instantiation in the equivalence space, we are not so sure if that graph reflects the true causal structure. For this reason, an extension of traditional BNs is introduced, where each edge is interpreted as a direct causal influence between a parent node and a child node, relative to the other nodes in the network [Pearl, 2000].

This paper provides a substantially extended version of our previous work [Ben Messaoud *et al.*, 2009] in which we introduce the preliminary findings for integrating a semantic distance calculus to choose the appropriate interventions. Further developments along this direction have been made in order to deploy more efficient strategies to integrate the semantic prior knowledge, improve the causal discovery process and reuse the new discovered information to support the ontology evolution.

The remainder of this paper is arranged as follows: Section 2 gives the necessary background for both CBNs and ontologies. Section 3 sets out how to use the ontological knowledge

to enhance the causal discovery and vice versa. In Section 4, we show simulation results to evaluate the performances of the proposed algorithm. Concluding remarks and future works are given in Section 5.

2 Basic concepts & background

2.1 Ontologies

There are different definitions in the literature of what should be an ontology. The most notorious was given by Tom Gruber [Gruber, 1995], stipulating that an ontology is an *explicit* specification of a *conceptualization*. The "conceptualization", here, refers to an abstract model of some phenomenon having real by identifying its relevant concepts. The word "explicit" means that all concepts used and the constraints on their use are explicitly defined.

In the simplest case, an ontology describes a hierarchy of concepts (i.e. classes) related by taxonomic relationships (is-a, part-of). In more sophisticated cases, an ontology describes domain classes, properties (or attributes) for each class, class instances (or individuals) and also the relationships that hold between class instances. It is also possible to add some logical axioms to constrain concept interpretation and express complex relationships between concepts.

Hence, more formally, an ontology can be defined as a set of labeled classes $C = \{C_1, \dots, C_n\}$, hierarchically ordered by the subclass relations (i.e. is-a, part-of relations). For each concept C_i we identify k meaningful properties p_j , where $j \in [1, k]$. We use H_i to denote the finite domain of instance candidates with each concept C_i and c_i to denote any instance of C_i . We also use R to represent the set of semantical (i.e non-hierarchical) relations between concepts and R_c to represent the subset of causal ones. Finally, formal axioms or structural assertions $\langle c_i, c_j, s \rangle$ can be included, where $s \in S$ is a constraint-relationship like "must, must not, should, should not, etc".

Practically speaking, the ontologies are often a very large and complex structure, requiring a great deal of effort and expertise to maintain and upgrade the existing knowledge. Such proposals can take several different forms such as a change in the domain, the diffusion of new discoveries or just an information received by some external source [Flouris *et al.*, 2008].

There are many ways to change the ontology in response

to the fast-changing environment. One possible direction is the ontology evolution which consists in taking the ontology from one consistent state to another by updating (adding or modifying) the concepts, their properties and the associated relations [Khattak *et al.*, 2009].

The ontology evolution can be of two types [Khattak *et al.*, 2009]:

- **Ontology population:** When new concept instances are added, the ontology is said to be populated.
- **Ontology enrichment:** Which consists in updating (adding or modifying) concepts, properties and relations in a given ontology.

In order to establish the context in which the ontology evolution takes place, the principle of ontology continuity should be fulfilled [Xuan *et al.*, 2006]. It supposes that the ontology evolution should not make false an axiom that was previously true. When changes do not fulfill the requirement of ontological continuity, it is not any more an evolution, it is rather an ontology revolution.

2.2 Causal Bayesian Networks

A Causal Bayesian Network (CBN) is a Directed Acyclic Graph (DAG) where the set of nodes V represents discrete random variables $X = \{X_1, X_2, \dots, X_n\}$ and the set of edges E represents causal direct dependencies over V . We use D_i to denote the finite domain associated with each variable X_i and x_i to denote any instance of X_i . We denote by $Pa(X_i)$ the set of parents nodes for X_i and $Nei(X_i)$ the set of its neighboring variables.

To learn CBNs, the observational data don't contain enough information to discover the true structure of the graph, which will be restricted to the Completed Partially Directed Acyclic Graph (CPDAG). Thus we have to collect further information on causality via interventions (i.e. actions tentatively adopted without being sure of the outcome). Here, we should note that intervening on a system may be very expensive, time-consuming or even impossible to perform. For this reason, the choice of variables to experiment on can be vital when the number of interventions is restricted.

3 SemCaDo: a serendipitous causal discovery algorithm for ontology evolution

Generally, in the research area, scientific discoveries represent a payoff for years of well-planned works with clear objectives.

This affirmation did not exclude the case of other important discoveries that are made while researchers were conducting their works in totally unrelated fields and the examples are abundant from Nobel's flash of inspiration while testing the effect of dynamite to Pasteur brainstorm when he accidentally discovered the role of attenuated microbes in immunization.

In this way, we propose a new causal discovery algorithm which stimulates serendipitous discoveries when performing the experimentations using the following CBN-Ontology correspondences.

3.1 CBNs vs Ontologies

One of the main motivations when realizing this work is the similarities between CBNs and ontologies. This is particularly true when comparing the structure of the two models as shown in the following correspondences:

1. Nodes (V_i) $\leftrightarrow$ Concepts (C_i): The ontology concepts, which are relevant to the considered domain are represented by the nodes of the CBN.
2. Random variables (X_i) $\leftrightarrow$ Concept attributes ($C_i.p_j$): All random variables in the CBN are represented as specific concept attributes in the ontology.
3. Causal dependencies (E) $\leftrightarrow$ Semantic causal relations (R_c): The correspondence between the two models in term of causality will be as follows:
 - A causal relation between two concepts in the ontology will be represented by a directed link between the corresponding CBN nodes. It is read as $c_Y.p_j$ is the direct consequence of $c_X.p_j$, where p_j is the concept attribute used to make the correspondence.
 - A causal dependency represented by a directed link in the CBN will be represented by a specific causal relation between the appropriate concepts in the ontology.
4. Observational or experimental data ($D_{obs,int}$) $\leftrightarrow$ Concept-attribute instances ($c_i.p_j$): We make a correspondence between the observational (resp. interventional) data at our disposal and the instances of the domain ontology. Each observation (resp. intervention) can be viewed, in the ontological context, as a state instantiation of a given concept attribute.

3.2 SemCaDo Sketch

Our approach relies on extending the MyCaDo algorithm [Meganck *et al.*, 2006] in order to incorporate available knowledge from domain ontologies. The original character of the SemCaDo (Semantic Causal Discovery) algorithm is essentially its ability to make impressive discoveries and reuse the capitalized knowledge in CBNs.

The correspondences between CBNs and ontologies in SemCaDo must respect the following constraints:

- Only a single ontology should be specified for each causal discovery task.
- Each causal graph node must be modeled by a corresponding concept in the domain ontology. The concepts which are candidates to be a member of such correspondence have to share the same studied attribute p_j .
- The causal discoveries concern concepts sharing the same semantic type (e.g. direct transcriptional regulation between genes). This means that all concepts C_i modeled in the CBN must belong to the same super-concept SC and the causal relationship under study R_c should be defined for any element of SC to any other one.
- The ontology evolution should be realized without introducing inconsistencies or admitting axiom violations.

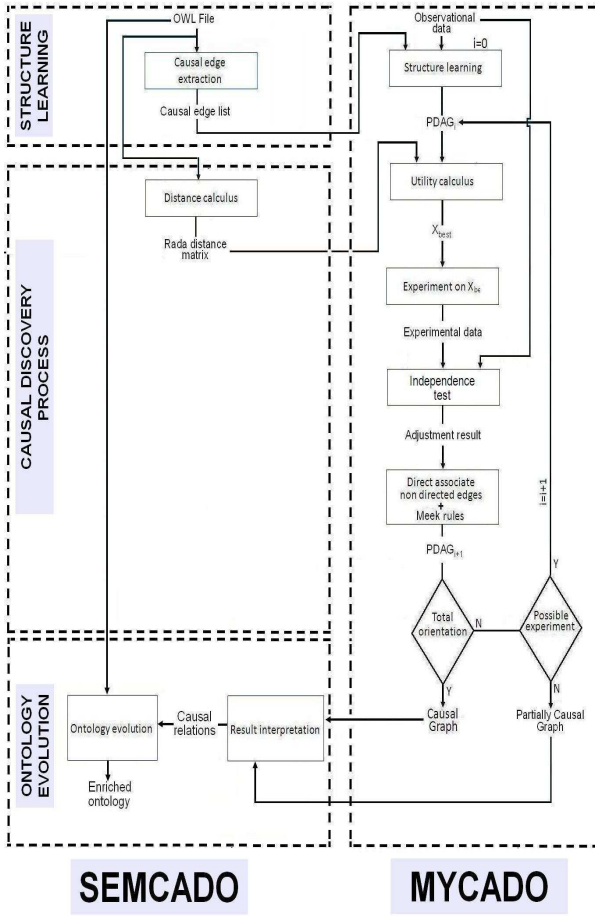


Figure 1: The SemCaDo Algorithm

The general overview of the SemCaDo algorithm is given in Figure 1. As inputs, SemCaDo needs an observational dataset and a corresponding domain ontology. Then it will proceed through three main phases:

1) Learning the initial structure using causal prior knowledge:

The ontology in input may contain some causal relations in addition to hierarchical and semantic relations. Those causal relations should be integrated from the beginning in the structure learning process in order to reduce the task complexity and better the final output. Therefore, such direct cause to effect relations will be incorporated as constraints when using structure learning algorithms. Our main objective is to narrow the corresponding search space by introducing some restrictions that all elements in this space must satisfy.

In our context, the only constraint that will be defined is edge existence. But we could also imagine in future work that some axioms in the ontology also give us some information about forbidden edges. All these edge constraints can easily be incorporated in usual BN structure learning algorithm [de Campos and Castellano, 2007]. Under some condition of

consistency, these existence restrictions shall be fulfilled, in the sense that they are assumed to be true for the CBN representing the domain knowledge, and therefore all potential Partially Directed Acyclic Graph (PDAG) must necessarily satisfy them.

Definition 1 Given a domain ontology $\mathcal{O}$, let $G=(C, R_c)$ be the DAG where $R_c: C_i \times C_j$ represents the subset of semantic causal relations extracted from $\mathcal{O}$. This subset included both direct and logically derivable semantic causal relations. Let $H=(X, E_h)$ be a PDAG, where X is the set of the corresponding random variables and E_h corresponds to the causal dependencies between them. H is consistent with the existence restrictions in G if and only if:

$$\forall C_i, C_j \in C, \text{ if } C_i \rightarrow C_j \in R_c \text{ then } X_i \rightarrow X_j \in E_h.$$

When we are specifying the set of existence restrictions to be used, it is necessary to make sure that these restrictions can indeed be satisfied. In fact, such causal integration may lead to possible conflicts between the two models. When this occurs, we have to maintain the initial causal information in the PDAG since we are supposed to use perfect observational data. On the other hand, we should ensure the consistency of the existence restrictions in such a way that no directed cycles are created in G .

2) Causal discovery process:

Before delving into the details of our approach, we first review the principal idea of the causal discovery process. When performing an experimentation on X_i , we have to measure all neighboring variables and accordingly to the result direct all edges connecting X_i and $\text{Nei}(X_i)$. This edge orientation represents one instantiation among all possible instantiations. It is then possible to continue the edge orientation by using the Meek rules [Meek, 1995] to infer new causal relations.

The aim of this phase is to decide which experiment will be performed and hence also which variables will be altered and measured. For this purpose, the strategy we propose in our approach makes use of a semantic distance calculus (e.g. Rada distance [Rada *et al.*, 1989]) provided by the ontology structure. So, for each node in the graph, SemCaDo calculates its semantic inertia, denoted by $\text{SemIn}(X_i)$ and expressed as follows:

$$\text{SemIn}(X_i) = \frac{\sum_{X_j \in \text{Nei}(X_i) \cup X_i} \text{dist}_{\text{Rada}}(\text{mscs}(\text{Nei}(X_i) \cup X_i), X_j)}{\text{Card}(\text{Nei}(X_i) \cup X_i)} \quad (1)$$

where:

- $\text{mscs}(c_i, c_j)$: the most specific common subsumer of the two concepts c_i and c_j , where $i \neq j$.
- $\text{dist}_{\text{Rada}}(c_i, c_j)$: the shortest path between c_i and c_j , where $i \neq j$,
- $\text{Card}(M)$: the cardinality of any set.

Moreover, the semantic inertia presents the following properties:

- When the experimented variable and all its neighbors lie at the same level in the concept hierarchy, the semantic inertia will be equal to the number of hierarchical levels needed to reach the mscs.

- It essentially depends on semantic distance between the studied concepts. This means that the more this distance is important, the more the SemIn will be maximized.

Further to these, we also integrate a semantic cumulus relative to inferred edges [Meek, 1995] denoted by *Inferred_Gain* in our utility function. For this purpose, we use $I(X_i)$ to denote the set of nodes attached by inferred edges after performing an experimentation on X_i . So, the *Inferred_Gain* formula is expressed as follows:

$$Inferred_Gain(X_i) = \frac{\sum_{X_j \in I(X_i)} dist_{Rada}(mscs(I(X_i)), X_j)}{Card(I(X_i))} \quad (2)$$

Note that we don't use here all the information provided by the ontology. We should also consider the axioms to check if any new relation could be inferred from the semantic point of view. Better interacting with the axioms is one of our perspectives for future work.

When using the two proposed terms weighted by the cost of experiment (i.e. $cost(A_{X_i})$) and measurement (i.e. $cost(M_{X_i})$), our utility function will be as follows:

$$U(X_i) = \frac{SemIn(X_i) + Inferred_Gain(X_i)}{\alpha cost(A_{X_i}) + \beta cost(M_{X_i})} \quad (3)$$

where measures of importance $\alpha, \beta \in [0,1]$.

This utility function will be of great importance to highlight the serendipitous character of SemCaDo algorithm by guiding the causal discovery process to investigate unexplored areas and conduct more informative experiments.

3) Edge orientation & ontology evolution:

Once the specified intervention performed, the meek rules [Meek, 1995] will be applied to infer new arcs until no more edges can be oriented. We note that these orientation rules are proven to be correct and complete subject to any additional background knowledge.

Since certain experimentation can not be performed, either because of ethical reasons or simply because it is impossible to do it, the final causal graph can be either a CBN or a partially causal graph. In both cases, the causal knowledge will be extracted and interpreted for an eventual ontology evolution.

In this way, the causal relations will be translated as semantic causal relations between the corresponding ontology concepts. For this purpose, SemCaDo algorithm uses a six-phases evolution process [Stojanovic *et al.*, 2002]:

- Change capturing: The aim of this initial step in the ontology evolution process is to capture the new discovered causal relations on the current causal graph which are not actually modeled. It starts after finishing the structure learning step in order to treat all changes in a consistent and unified manner.
- Change representation: In order to be correctly implemented, we have to represent these causal changes formally, explicitly and in a suitable format. In the context of SemCaDo algorithm, we only handle elementary changes (i.e. restricted to adding semantic causal relations) that cannot be decomposed into simpler ones.

- Semantics of change: The semantics of change is the phase that enables the resolution of ontology changes in a systematic manner by ensuring the consistency of the ontology. In our case, conflicting knowledge is highly possible to occur when deducing causal conclusions from the ontology axioms. Such inconsistencies should be handled by automated reasoning. This step also prevents the creation of new cycles in the ontology when integrating the causal discoveries. This consistency rule is maintained since the causal discovery step in SemCaDo avoid the creation of cycles during the structure learning.
- Implementation: In order to avoid performing unwanted changes, a list of all consequences in the ontology and dependent artifacts should be generated and presented to the ontology engineer, who should then be able to accept the change or reject it. If the implementer agrees to add the new causal relationships, all actions to apply the change have to be performed.
- Propagation: Pursuing and adopting the new causal discoveries can generate additional changes in the other parts of the ontology. These changes are called derived changes. That is why, during this step, it is necessary to determine the direct and indirect types of changes to be applied. In case of ambiguity, the ontology expert decide on the action to occur. A human intervention at this level is essential to remove the ambiguity and to make the final decision.
- Validation: Change validation enables justification of performed changes and undoing them at user's request. If the output of SemCaDo causal discovery step is a partially directed graph, it is possible to restart the cycle when there sufficient budget to make further discoveries.

4 Experimental study

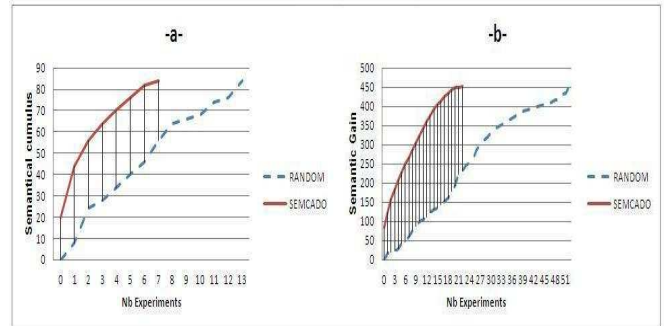


Figure 2: The semantic gain given the number of experiments using SemCaDo and random approach on relatively small graphs (a) and bigger ones (b).

In the experimental evaluation, we will compare our approach to a random causal discovery strategy.

For this purpose, we randomly create a set of syntectic 50 and 200 node graphs and simulate the result of a structure

learning algorithm working with a perfect infinite dataset. For each simulated graph, we automatically generate a corresponding concept hierarchy in which we integrate a varying percentage (10% to 40%) of the initial causal relations. As we do not dispose of a real system to intervene upon, we decide to simulate the experimentations directly in the previously generated CBNs.

Another point to consider in our experimental study concerns the calculation of the semantic gain. In fact, after each SemCaDo (resp. Random) iteration, we measure the sum of semantic distances ([Rada *et al.*, 1989] in these experiments) relative to the new directed edges in the graph and update a semantic cumulus. In both strategies, the two corresponding curves are increasing in, meaning that the higher is the number of experimented variables, the higher is the value of the semantic gain. Nevertheless the more the curve is increasing faster, the more the approach is converging to the best and most impressive experiments.

Figure 2 shows that, during the experimentation process, SemCaDo comfortably outperforms the random approach in term of semantic gain. This is essentially due to the initial causal knowledge integration and the causal discovery strategy when performing the experimentations. But if the two curves reach the same maxima when obtaining a fully directed graph, where is the evolutionary contribution of SemCaDo? Let us remember that we are approaching a decision problem which is subject to the experimentation costs and the budget allocation. Taking into account this constraint, the semantic domination of SemCaDo will be extremely beneficial when the number of experiments is limited.

All these experimental results show how the SemCaDo algorithm can adopt a serendipitous attitude with the minimum expected cost and effort. This is indeed a new avenue of causal investigation, moving far away from traditional techniques.

5 Conclusions and Future Works

In this paper, we outlined our serendipitous and cyclic approach which aims to: i) Integrate the causal prior knowledge contained in the domain ontology when learning the structure of the partially directed graph from observational data. ii) Use the semantic distance calculus to guide the iterative causal discovery process and investigate unexpected causal relations. iii) Capture the required causal discoveries to be applied to ontology evolution.

The SemCaDo algorithm is an initial attempt towards a more ambitious framework exploiting the power of CBNs and ontologies. Future works will detail how to revolutionize the ontologies by adding new axioms and ignoring others when incorporating causal discoveries.

References

- [Ben Messaoud *et al.*, 2009] Montassar Ben Messaoud, Philippe Leray, and Nahla Ben Amor. Integrating ontological knowledge for iterative causal discovery and visualization. In *ECSQARU*, pages 168–179, 2009.
- [de Campos and Castellano, 2007] Luis M. de Campos and Javier G. Castellano. Bayesian network learning algorithms using structural restrictions. *International Journal of Approximate Reasoning*, pages 233–254, 2007.
- [Flouris *et al.*, 2008] Giorgos Flouris, Dimitris Manakanatas, Haridimos Kondylakis, Dimitris Plexousakis, and Grigoris Antoniou. Ontology change: classification and survey. In *Knowledge Engineering Review*, volume 23, pages 117–152, 2008.
- [Gruber, 1995] T. R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies Vol. 43, Issues 5-6*, pages 907–928, 1995.
- [Khattak *et al.*, 2009] Asad M. Khattak, Khalid Latif, Songyoung Lee, and Young-Koo Lee. Ontology Evolution: A Survey and Future Challenges. In Dominik Ślzak, Tai-hoon Kim, Jianhua Ma, Wai-Chi Fang, Frode E. Sandnes, Byeong-Ho Kang, and Bonggen Gu, editors, *U-and E-Service, Science and Technology*, volume 62, pages 68–75. Springer Berlin Heidelberg, 2009.
- [Meek, 1995] Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Proceedings of the Eleventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pages 403–41, San Francisco, CA, 1995. Morgan Kaufmann.
- [Meganck *et al.*, 2006] Stijn Meganck, Philippe Leray, and Bernard Manderick. Learning causal bayesian networks from observations and experiments: A decision theoretic approach. In Vicen Torra, Yasuo Narukawa, Ada Valls, and Josep Domingo-Ferrer, editors, *MDAI*, volume 3885 of *Lecture Notes in Computer Science*, pages 58–69. Springer, 2006.
- [Pearl, 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.
- [Pearl, 2000] Judea Pearl. *Causality: models, reasoning, and inference*. 2000.
- [Rada *et al.*, 1989] Roy Rada, Hamed Mili, Ellen Bicknell, and Maria Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1):17–30, 1989.
- [Stojanovic *et al.*, 2002] Ljiljana Stojanovic, Alexander Maedche, Boris Motik, and Nenad Stojanovic. User-driven ontology evolution management. In *EKAU '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management*, pages 285–300, London, UK, 2002.
- [Xuan *et al.*, 2006] Dung Nguyen Xuan, Ladjel Bellatreche, and Guy Pierra. A versioning management model for ontology-based data warehouses. In A. Min Tjoa and Juan Trujillo, editors, *DaWaK*, volume 4081 of *Lecture Notes in Computer Science*, pages 195–206. Springer, 2006.

GALILEO: A System for Automating Ontology Evolution

Michael Chan Jos Lehmann Alan Bundy
 School of Informatics, University of Edinburgh
 Edinburgh, U.K.
 {mchan, jlehmann, bundy}@inf.ed.ac.uk

Abstract

We describe the GALILEO system, which is designed for automating the evolution of higher-order logic ontologies by incorporating user interaction for diagnosing and repairing faults. In particular, we present our approach to ontological conflict diagnosis, which circumvents problems posed by HOL's undecidability by means of: formalising modular ontologies as Isabelle *locales* and preparing the system by user interaction; applying *ontology repair plans* and automatically identifying logically valid terms that are responsible for the conflict; and, automatically eliminating physically meaningless terms.

1 Introduction

The process of revising an ontology in the face of new information is key to many areas of Computer Science. The literature on the subject calls such processes *ontology evolution* and concentrates on how Description Logic (DL) ontologies for Semantic Web applications need to evolve in order to maintain their coherence and consistency (ontology debugging) or relative to other ontologies (ontology alignment). We decided to investigate ontology evolution by *formalizing and automating in higher-order logic* (HOL) the mechanisms to repair *locally consistent* but possibly *globally inconsistent* ontologies, i.e. ontologies that are individually consistent but may give rise to an inconsistency when merged. Working with locally consistent ontologies enables the reasoning about the shape of the cause of the global conflict, allowing for specific meaningful repairs. The notion of ontologies we adopted is a general one, i.e. a specification of a conceptualisation [Gruber, 1993], which is compatible with our treatment of HOL theories as ontologies. Unlike the less expressive logics, including DL, our approach naturally allows for the formalisation of ontology evolution both as belief revision and syntactic manipulation (e.g., splitting a function, changing its arity, etc.). HOL has proven advantageous in at least three other ways. Firstly, HOL's polymorphism of variables (and other symbols such as $\leq$, $\geq$, $+$, $-$, etc.) permits the generality of the proposed repair plans for evolution and their applicability over diverse cases. Secondly, HOL-based theorem provers, such as Isabelle, enable HO-reasoning for ontology evolution

and reasoning over locally consistent but globally inconsistent ontologies that share variables. Finally, many complex concepts are better represented as HOL objects, e.g., the orbit of a star – this is relevant because the examples used for developing and testing GALILEO are based on the evolution of physics, which involves concepts best represented as functions. This, therefore, relates our work to scientific discovery [Langley, 1981].

In this paper, we provide an overview of the GALILEO system, which is the first system to attempt to automate ontology evolution in HOL. In particular, we present our approach to ontological conflict diagnosis, which circumvents problems posed by HOL's undecidability by means of: a) formalising modular ontologies as *locales* [Ballarin, 2006] and preparing the system by user interaction; b) automatically identifying logically valid terms that are responsible for the conflict; and, c) automatically eliminating physically meaningless terms. We have designed mechanisms called *ontology repair plans* (ORPs) [Bundy and Chan, 2008], each of which defines a trigger pattern for detecting that certain faults exist in the given ontologies and a set of transformation rules for resolving the detected conflict. The identification of the term that gives rise to the inconsistency, hereafter designated by “conflict term” and the variable *stuff*, underpins the diagnosis component and is a relatively complex reasoning task – whereas the repair component typically involves relatively simple syntactic manipulation of the ontologies based on the result of the diagnosis. Thus, in this paper, we present only GALILEO's diagnosis component and briefly outline the repair.

2 Case Study and Ontology Repair Plans

We describe below the running example for the rest of the paper: the contradiction between the predicted and the observed orbital velocity of galaxies. We then describe two ORPs, which can be applied to resolve the conflict that underlies the case study and produce two different but historically proposed explanations.

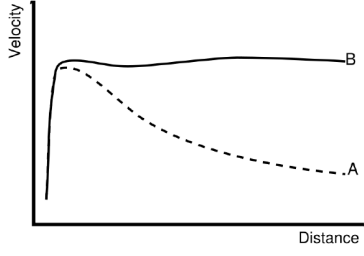


Figure 1: The x -axis is the distance of the stars in the galaxy $Glxy1$ from the galactic centre and the y -axis is their orbital velocities. The dotted and solid lines are the predicted and actual curve that is observed.^a

^aTaken from http://en.wikipedia.org/wiki/Galaxy_rotation_problem.

2.1 Case Study: Predicted vs. Observed Orbital Velocity of Galaxies

Newtonian dynamics predicts that orbital velocities¹ of stars in a spiral galaxy decrease inversely with the square root of the distance from the galactic centre, or the *radius*. However, the velocity observed was almost constant out to large radii (Figure 1), as first observed by Rubin [Rubin *et al.*, 1980]. One explanation is about the theoretical existence of *dark matter*, which is matter that is inferred to exist in the universe but not give off light, and is based on various sources of evidence, including the fact that the observed orbital velocities of stars in spiral galaxies exceeds the predicted values. Rubin concluded that some invisible matter is exerting a gravitational force on these stars, causing the unexpectedly high velocities.

Suppose the contradiction arises from observing the galaxy rotation curves Ga and Gb of type $Gly \mapsto Real \mapsto Real$, which are the curves based on prediction and observation, respectively, where Gly is the type of galaxy. Let the ontology of the predictive theory Thy contain definitions, including that of orbital velocity, and assertions about the value of various properties of different stars; let the sensory ontology, $Sens$, contain observations made on the *rotational velocity* of a galaxy, as reported in [Rubin *et al.*, 1980]. Figure 2 highlights some of the axioms available in the two ontologies, where Evt , Gly , Dst , and Str denote types for representing events, galaxies, distances, and stars, respectively; G denotes the universal gravitational constant; OV , the orbital velocity; g_r , the set of stars in the galaxy g up to the radius r ; M , the mass of a body; Rad , the radius of a star from its galactic centre; RtV , the rotational velocity; RdV , the radial velocity; SyV , the velocity of the galactic system relative to the observer; Inc , the inclination of the galaxy; $\lambda Shift$, the shift in wave length; $Glxy1$, the galaxy being observed; $Star1$ and $Star2$, stars in the observed galaxy; and, $Obs6$, the observation event.

Note that the graph object in (1) is defined using equality on functions, whereas that in (4) is defined using the equality

¹We assume that observed orbital velocity and rotational velocity for stars are the same.

$$Ax(Thy) := \{ \forall e:Evt. Ga(Glxy1) := OV(e, Glxy1) \} \quad (1)$$

$$\forall e:Evt, g:Gly, r:Dst. OV(e, g) := \quad (2)$$

$$\sqrt{\frac{G \times \sum_{s \in g_r} M(e, s)}{r}}, \quad G := 6.67 \times 10^{-11},$$

$$M(Obs6, Star1) := 5, \quad M(Obs6, Star2) := 9 \dots \} \quad (3)$$

$$Ax(Sens) := \{ \forall s:Str. \quad (4)$$

$$Gb(Glxy1, Rad(s, Glxy1)) :=$$

$$RtV(Obs6, s, Glxy1),$$

$$\forall e:Evt, s:Str, g:Gly. RtV(e, s, g) := \quad (5)$$

$$\frac{RdV(e, s, g) - SyV(e, g)}{\sin(Inc(e, g))},$$

$$\lambda Shift(Obs6, Star1, Glxy1, \dots) = 300, \dots \} \quad (6)$$

Figure 2: Example axioms for the representation of the case study in §2.1.

on reals; this illustrates a benefit of the polymorphism in HOL over less expressive logics. Moreover, we assume both Thy and $Sens$ contain some knowledge about shapes of curves, e.g., the curve that has a positive gradient between zero and some point and a zero gradient thereafter. A modular setup permits the extension and sharing of such ontological knowledge.

Some symbols in $Sens$, such as RtV and RdV , are not in the language of Thy . To link together the seemingly disparate terms, an example axiom of the bridging meta-ontology, Ob_b , is:

$$\forall v, r:Real. (Sens \vdash Rad(s, g) = r \wedge \quad (7)$$

$$Sens \vdash RtV(e, s, g) = v) \longleftrightarrow Thy \vdash OV(e, g, r) = v$$

which relates RtV in $Sens$ to the OV in Thy by Rad . Rubin's proposed resolution is the hypothetical existence of dark matter, which introduces a new form of matter. In contrast, the *Modified Newtonian Dynamics* (MOND) [Milgrom, 1983] argues that the definition of G is flawed and that it is not a constant, but is a function depending on the acceleration of the star. By MOND, the value of G is greater for stars with very low accelerations.

2.2 Ontology Repair Plans

We introduce here two ORPs, *Where's My Stuff* and *Inconstancy*, and focus mostly on the diagnostic aspect, i.e. their trigger formulae.

The Where's My Stuff Ontology Repair Plan

In physics, a common type of conflict is caused by a difference between the predicted value of some property and the value according to some corresponding sensory information arising from an experiment. This conflict is typically caused by the use of a theoretical definition that is based on an incorrect definition of the property measured or that of a dependency of the property measured. The error is generally a consequence of a misconceptualisation with a component of

the property neglected from the definition; for instance, when speaking about the matter distribution inside a galaxy, only the visible matter is included in the theoretical definition and dark matter is neglected.

Suppose an ontology O_1 , representing the current state of a predictive theory, disagrees over the value of $f(stuff)$ with another ontology O_2 , representing some sensory information. $f(stuff)$ might, for instance, be the graph plotting the orbital velocities of stars – a HO-object.

Trigger: If $f(stuff)$ has different values, the following formula will be triggered:

$$\exists O_1, O_2:Onto, \tau, \tau':Types, f:\tau \mapsto \tau', \quad (8)$$

$$stuff:\tau, v:\tau'.$$

$$(O_1 \vdash f(stuff) >_{\tau'} v \wedge O_2 \vdash f(stuff) \leq_{\tau'} v) \quad (9)$$

$$\vee$$

$$(O_1 \vdash f(stuff) \geq_{\tau'} v \wedge O_2 \vdash f(stuff) <_{\tau'} v) \quad (10)$$

where $O \vdash \phi$ means that ϕ is a theorem of ontology O ; $t:Types$ means t is a type; $o:Onto$ means o is an ontology; v is a value of type τ' ; and, $>_{\tau}$ is a partial order for τ .

WMS is triggered if the return value of $f(stuff)$ deduced from O_1 is different from that deduced from O_2 (9, 10). The two variables in $f(stuff)$, $stuff$ and f , respectively represent the part of the term that is responsible and not responsible for the conflict. WMS resolves the detected conflict by splitting $stuff$ into three parts: visible $stuff$, invisible $stuff$, and total $stuff$, and defining invisible $stuff$ in terms of total and visible $stuff$ s in the repaired O_1 , $\nu(O_1)$. The new O_2 , $\nu(O_2)$, is the same as O_2 except for the renaming of old $stuff$ to visible $stuff$. If Thy and $Sens \cup O_b$ in §2.1 were instantiations of O_1 and O_2 , respectively, and $stuff$ was instantiated to $Glxy1$, where $\cup$ merges ontologies, then WMS emulates Rubin's solution by introducing a third component to the galaxy: dark matter.

2.3 The Inconstancy Ontology Repair Plan

In order to better define a repair operation, so that, e.g., the number of possible values a new variable created by the repair process can take is dramatically reduced, extra sensory information can help resolve such ambiguity. We have identified from historical records of physics development that the inclusion of additional sensory information collected under different conditions can indeed help identify new dependencies for a property.

Suppose an ontology O_1 , representing the current state of a physical theory, is in conflict with a set of ontologies $O_2(\vec{r}_i)$, representing sensory information arising from experiments under different circumstances, represented by $\vec{r}_i$, such that unexpected variations in the value of $f(stuff)$ arise under these circumstances.

Trigger: If $f(stuff)$ is measured to take different values in different circumstances, then the following trigger for-

mulae will be matched:

$$\exists O_1, O_2:Onto, \tau, \tau':Types, \vec{r}_1, \vec{r}_2:Cir,$$

$$f:\tau \mapsto \tau', stuff:\tau, \vec{s}_1, \vec{s}_2:\tau, c_1, c_2:\tau'.$$

$$O_1 \vdash stuff =_{\tau'} c \wedge \quad (11)$$

$$O_2(\vec{r}_1) \vdash f(stuff) =_{\tau'} c_1 \wedge \dots \quad (12)$$

$$O_2(\vec{r}_n) \vdash f(stuff) =_{\tau'} c_n \wedge$$

$$\exists i, j \leq n. O_1 \vdash c_i \neq_{\tau'} c_j \vee c \neq_{\tau'} c_i \quad (13)$$

where $O_2(\vec{r})$ is the sensory ontology containing observations made under the condition $\vec{r}$; Cir is the type of conditions describing circumstances; $\vec{s}_1$ and $\vec{s}_2$ are arguments of $stuff$; and, c_1, c_2 are values of type τ' .

Inconstancy is triggered if $f(stuff)$ is predicted to be independent of $\vec{r}$, but the return values of $f(stuff)$ unexpectedly vary when $\vec{r}$ varies (11, 12, 13). The resolution of the detected conflict is to retain all O_1 -axioms in the repaired O_1 , $\nu(O_1)$, and retain all $O_2(\vec{r}_i)$ -axioms in the repaired $O_2(\vec{r}_i)$, $\nu(O_2(\vec{r}_i))$, except for the replacement of old $stuff$ with $\nu(stuff)$ and the replacement of the definition of $stuff$ by a new definition of $\nu(stuff)$ in $\nu(O_1)$ – the new definition establishes a relationship between $\vec{r}_i$ and $stuff$. If G in §2.1 is the instantiation of $stuff$, then G is given a new dependency, which is the solution proposed by MOND.

3 The GALILEO System

Both WMS and Inconstancy have been implemented in the GALILEO system. The reasoning capabilities of GALILEO are provided by the Isabelle proof assistant [Paulson, 1994], so soundness of the reasoning is guaranteed. Beside reasoning, Isabelle provides a contextualisation facility called *locales* [Ballarin, 2006], which can be used for formalising modular ontologies. The GALILEO system handles flexible ontological configurations, including ontological extensions and those with heterogeneous signatures. For heterogeneous configurations, *bridging ontologies* are used to align signatures across different ontologies. GALILEO analyses ontologies expressed in an Isabelle theory file, in which each ontology is represented as a locale.

There are three major phases in the diagnosis procedure: a) *preparation*, in which the system acquires the instantiation of some of the existential variables in (9, 10) or (11, 12, 13) in order to detect a WMS type or an Inconstancy type of fault; b) *conflict term search*, in which all logically correct instantiations of $stuff$ are discovered; and, c) *search space control*, in which only physically meaningful instantiations are retained. Below, we present our approach in more depth by applying it to an example record of ontology evolution in physics.

3.1 Preparation

In order to detect a WMS type of fault, the system requires an indication of the ontologies that may contain the fault, which represents the instantiations of O_1 and O_2 . The system also requires a formula that expresses the conflict, *cearp*, which is a sentence that is derived to be true in one ontology but false in another. For example, in Rubin's graph of observed orbital velocities at various radii in a galaxy *glx*, the observed velocities near the edge of *glx* are greater than those predicted

[Rubin *et al.*, 1980]. Again, let *Sens* be the ontology containing sensory data and $Gb(glx)$ be the observed graph for glx . The diagnosis procedure begins with the substitution:

$$\{O_1/Thy, O_2/Sens, cexp/Ga(glx) = Gb(glx)\}$$

since $Thy \vdash Ga(glx) = Gb(glx)$ and $Sens \vdash Ga(glx) < Gb(glx)$ as Ga and Gb are expected to be identical but Gb turns out to be greater than Ga . To ensure that the two graphs indeed disagree, the user must discharge a proof obligation in the shape of the trigger formula but with the existentials instantiated, e.g.

$$(Thy \longrightarrow Ga(glx) \geq Gb(glx) \wedge \\ Sens \longrightarrow Ga(glx) < Gb(glx)) \vee \dots$$

in which each ontology is treated as a predicate expressed as a conjunction of its axioms. Although the system requires values of O_1 , O_2 and $cexp$ as user inputs, the instantiations of each of the remaining variables, i.e. f , $stuff$, and v , are automatically identified, so the rest of the procedure is completely automatic.

For Inconstancy to be triggered, pattern (11, 12, 13) needs to be instantiated. Similar to WMS, GALILEO requires from the user the ontologies that may give rise to a fault – in this case, one theoretical ontology and two sensory ontologies. In addition, the system requires the terms representing the unexpected variation in each of the two input sensory ontologies, $cexp_1$ and $cexp_2$, respectively. For instance, suppose in two sensory ontologies Obs_1 and Obs_2 , the observed orbital velocities of two stars, p_1 and p_2 , unexpectedly vary. So, $Obs_1 \vdash OV(p_1, glx) = v_1$ and $Obs_2 \vdash OV(p_2, glx) = v_2$, and in the theory, $Thy \vdash OV(p_1, glx) - v_1 \neq OV(p_2, glx) - v_2$. Thus, we have the substitution:

$$\{O_1/Thy, O_1(\vec{r}_1)/Obs_1, O_2(\vec{r}_2)/Obs_2, \\ cexp_1/OV(p_1, glx), cexp_2/OV(p_2, glx)\}.$$

Like WMS, the fault is verified by discharging a proof obligation in the shape of the trigger formula, and the system then identifies various instantiations of $stuff$.

3.2 Search for Conflict Terms

Having indicated the potentially faulty ontologies and $cexp$, GALILEO identifies instantiations of $stuff$ to completely instantiate the relevant trigger pattern. Isabelle’s polymorphism is partial as its type unification does not specialise schematic type variables to function types. Such a constraint, which also holds in most other theorem provers, renders type unification tractable, but it is an impediment to instantiating $stuff$ to different arity functions in different case studies. Letting $stuff$ be a variadic function allows for high generality of the trigger. GALILEO, therefore, explicitly generates a set of types for $stuff$, given a maximum function order and arity.

With WMS, instantiations of f , $stuff$, and v can be discovered by applying Isabelle’s higher-order matching algorithm to match $cexp$ against the pattern $f(stuff) = v$. For example, there exists the substitution $\{glx/stuff\}$ by matching with $Ga(glx)$, which suggests that the conceptualisation of the galaxy glx should be adjusted, e.g., by introducing dark matter. Unlike the HOL unification problem, HOL matching is decidable.

Inconstancy also adopts a general approach to the identification of the conflict term by matching $cexp_1$ and $cexp_2$ against the patterns $f(stuff_1)$ and $f(stuff_2)$. The system iterates through each substitution returned and generates a proof obligation for each instantiation of $stuff$. With the substitution for Inconstancy described in §3.1, instantiations of $stuff_i$ could be simply $OV(p_i, glx)$, then the proof obligation $\exists c_1, c_2. Obs_1 \vdash OV(p_1, glx) = c_1 \wedge Obs_2 \vdash OV(p_2, glx) = c_2 \wedge Thy \vdash c_1 \neq c_2$ will need to be discharged.

Often, the term expressed in $cexp_i$ may not represent the preferred concept for repair, e.g., $cexp_i$ represents measurements on orbital velocity but the preferred concept for repair is another, such as the gravitational constant G . To search for other potential concepts for repair, GALILEO automatically identifies other relevant instantiations of $stuff$. For example, Thy contains a definition of OV in terms of G and M (2); if $stuff$ is instantiated to OV , then $stuff$ can be instantiated to G , which is the fix proposed by MOND.

3.3 Search Space Control

For both ORPs, the search spaces of possible substitutions are typically unmanageable because f and $stuff$ are both polymorphic – there are more than 20,000 substitutions in both cases, but only a small fraction of these are physically meaningful. A solution is needed to avoid an explosion of substitutions.

Isabelle’s implementation of higher-order matching, like all others, returns logically correct substitutions without assessing the physical meaningfulness or relevance of the substitutions. Physical meaningfulness is crucial for controlling the search for suitable substitutions in order to perform a meaningful diagnosis, which must be compatible with real-world semantics. For instance, matcher 1 in Table 1 assigns $stuff$ to $Glx1$, which is the diagnosis required. However, we argue that some other example matches have little physical content or meaningfulness, e.g., matches 6 and 7 assign $stuff$ to a function containing the identity function and a function containing a free variable ($?s$) generated by the matching algorithm, respectively.

1.	$Glx1$	5.	$\lambda x.x(Ga(Glx1))$
2.	Ga	6.	$\lambda x.x(Ga(x(\lambda x.x)))$
3.	$Ga(Glx1)$	7.	$\lambda x.x(Ga(x(?s(x))))$
4.	$\lambda x.x(Glx1)$		

Table 1: Example instantiations of $stuff$ from pattern matching $f(stuff)$ against $Ga(Glx1)$.

To reduce the number of substitutions returned, several heuristics (Table 2) are designed and applied.

4 Evaluation

We evaluated our technique and implementation on a diverse range of records of ontology evolution in physics. Although only one example and two different fixes have been described in this paper, we base the evaluation also on examples already reported elsewhere, e.g., the *bouncing-ball paradox* [Chan *et al.*, 2010] and the *discovery of latent heat* [Bundy *et al.*,

#	Description
H1	<i>stuff</i> must not contain the $\lambda x.x$ term.
H2	f and <i>stuff</i> must not contain free variables.
H3	<i>stuff</i> must contain one or more constants.
H4	Two functions of the same function symbol but with permuted argument types are the same.

Table 2: Example heuristics for pruning the solution space: H1 ignores instantiations containing the identity function, which has no physical characteristic; H2 ignores instantiations that imply less ontological commitment than that we intended; H3 requires that the term to be repaired must contain an element of the signature, i.e. a concrete concept; and, H4 regards, e.g., $stuff:\alpha \rightarrow \beta \rightarrow \gamma$ and $stuff:\beta \rightarrow \alpha \rightarrow \gamma$ to be the same.

2010]; the goal is to demonstrate the high level of generality of our approach, not just being able to identify various types of *stuff*, but also to handle examples across distant subfields of physics. The rest of this section covers two key areas for evaluating our work: effectiveness of search space control and meaningfulness of the generated solutions.

4.1 Reduction in Candidate Matches

Our experiment has shown that our approach to automating conflict term discovery is intractable without a reduction in the search space containing candidate matches. Even with a low unification bound, e.g., 3, the procedure fails to return a solution on all case studies. The raw search spaces are, therefore, far too vast in practice. The use of the few example heuristics in Table 2, however, prunes away from the solution scope more than 20,000 matches for the case study in §2.1 and 57,000 for both the bouncing-ball and the latent heat case studies, and enables termination. The resulting sets of matches are significantly reduced to substantially more manageable amounts of 5, 21, and 27 for each respective example. Thus, the results suggest that the heuristics used are effective in controlling the search.

4.2 Meaningfulness of Solutions

Even though the solution spaces could be considerably reduced, we will examine the degree of physical meaningfulness of the instantiations of *stuff* in the remaining matches. For dark matter, only matches 1 to 5 in Table 1 remain in the pruned space. Every candidate match returned is in fact physically interpretable – for instance, instantiating *stuff* to *Glxy1* indicates that the galaxy should have an extra component; to *Ga* leads to a redefinition of all predicted curves; and, to $Ga(Glxy1)$ limits the modification to only the definition of the predicted curve for *Glxy1*.

5 Conclusion

The paper provided an overview of the GALILEO system and of two of its ORPs: WMS and Inconstancy. The diagnostic components were presented by successfully applying them to discover two fault types underlying the same set of conflicting observations in astronomy. This showed the generality of GALILEO and its ability to return only physically meaningful ontology repairs.

References

- [Ballarin, 2006] Clemens Ballarin. Interpretation of locales in Isabelle: Theories and proof contexts. In *Mathematical Knowledge Management*, volume LNCS 4108, pages 31–43. Springer Berlin/Heidelberg, 2006.
- [Bundy and Chan, 2008] Alan Bundy and Michael Chan. Towards ontology evolution in physics. In Wilfrid Hodges and Ruy de Queiroz, editors, *Logic, Language, Information and Computation*, volume 5110 of *Lecture Notes in Computer Science*, pages 98–110. Springer Berlin / Heidelberg, July 2008.
- [Bundy et al., 2010] A. Bundy, J. Lehmann, G. Qi, and I. Varzinczak, editors. *Notes of the ECAI-10 Workshop on Automated Reasoning about Context and Ontology Evolution*. ECAI-10, 2010.
- [Chan et al., 2010] M. Chan, J. Lehmann, and A. Bundy. Higher-order representation and reasoning for automated ontology evolution. In *Proceedings of the 2010 International Conference on Knowledge Engineering and Ontology Development*, pages 84–93. SciTePress, 2010.
- [Gruber, 1993] Thomas R. Gruber. A translation approach to portable ontology specification. *Knowledge Acquisition*, 5(2), 1993.
- [Langley, 1981] P. Langley. Data-driven discovery of physical laws. *Cognitive Science*, 5(1):31–54, 1981.
- [Milgrom, 1983] M. Milgrom. A modification of the Newtonian dynamics as a possible alternative to the hidden mass hypothesis. *The Astrophysical Journal*, 270:365–370, 1983.
- [Paulson, 1994] L. C. Paulson. *Isabelle: A generic theorem prover*. Springer-Verlag, 1994.
- [Rubin et al., 1980] V. C. Rubin, N. Thonnard, and Jr. Ford, W. K. Rotational properties of 21 SC galaxies with a large range of luminosities and radii, from NGC 4605 ($R = 4\text{kpc}$) to UGC 2885 ($R = 122\text{kpc}$). *Astrophysical Journal*, 238:471, 1980.



ARCOE-11 Workshop Notes