Graph Based Knowledge Representation and Reasoning: Practical AI Applications

Madalina Croitoru

Reasoning About Knowledge as Graphs:
Practical Artificial Intelligence Applications

Madalina CROITORU

Univ. Montpellier 2
Equipe-projet INRIA GraphIK
Montpellier, France

Date Novembre 2014

JURY

Leila AMGOUD	Directeur de Recherche CNRS, IRIT, Toulouse	Rapporteur
Ollivier HAEMMERLE	Professeur, IRIT, Toulouse	Rapporteur
Jan TOP	Professeur, Vrije Universiteit, Netherlands	Rapporteur
Marie-Laure MUGNIER	Professeur, Université Montpellier II	Examinateur

Thanks

I thank the members of the jury for the enthusiasm they have accepted to review my thesis with. I thank my colleagues at LIRMM and IUT for their fruitful discussions and exchanges. I thank my many collaborators and co-authors that each taught me something new and valuable. I thank the GraphIK group for their long standing influence in my research, dating back to my first year of PhD. I thank my family for their education, support and love. I thank Cosmina for always being so happy and sweet. Thanks to Julien for his insights and humour and his family for their kindness.

Contents

Chapter 1

Introduction

1.1 Overview

My research interests are in the field of knowledge representation and reasoning (KRR). During and since my PhD (throughout my post-doc and lectureship) I have investigated graph-based KRR languages applied in different Artificial Intelligence (AI) fields. Syntactically, the objects I am interested in are graphs. Semantically these graphs represent different subsets of logics such that the logic operations are sound and complete with graph operations. This adds logical background to the well-known modelling capacity of graphs. In the next section, I put in context this research vision with respect to my work.

1.2 Context

During my PhD the logics subset I focused on is the existential, positive fragment of First Order Logic (FOL). Conceptual Graphs [57] are one labelled bipartite graph representation of such logic. Logical subsumption can be translated (in a sound and complete manner) into labelled graph homomorphism, called projection. My contribution looked at improving both the (1) reasoning efficiency of such graphs and (2) their expressivity. Regarding reasoning I have proposed a projection optimisation based on the reduction of the labelled homomorphism into a max clique problem. Regarding the Conceptual Graphs expressivity I looked into their employability in Natural Language Processing, Multi-Agents Systems and the Semantic Web.

In Natural Language Processing I have investigated the *Generation of Referring Expressions*, where one is interested in obtaining a unique description of an object in a scene. Representing this problem using Conceptual Graphs allowed a clear contribution in terms of logical expressivity of the problem [27]. I have then extended Conceptual Graphs to represent *multiple views* on the same piece

of knowledge according to different agents interested in that knowledge. For the new syntactic object (a labelled tri-partite graph) I have proposed appropriate semantics that allow obtaining either a consensual subset of knowledge or different degrees of shared knowledge between agents [26]. Finally, Conceptual Graphs were used in order to express a summarised view of different knowledge bases in order to optimise distributed query answering. This summary (called a *Knowledge Oriented Specification*) could be done at different levels of detail and was expressed using another extension of Conceptual Graphs for representing hierarchical knowledge: *Layered Conceptual Graphs*. Once again, the corresponding semantics and reasoning mechanisms (based on graph operations) were provided [19].

During my PostDoc I continued this research in graph-based KRR and I further focused on Multi-Agents Systems and Semantic Web. The two application domains were given by the two European Projects I was working on.

The first project, *HealthAgents: Agent-based Distributed Decision Support System for brain tumour diagnosis and prognosis*(http://www.healthagents. net), was looking at creating a unified repository of brain tumour data shareable amongst the different medical partners in the project. The data compatibility was ensured by the use of a domain ontology [39]. The data interoperability was then ensured by the use of the Multi-Agents System framework designed as an abstraction of the several exchanges in the system [32]. My work involved the Semantic Web aspects of the project, notably *building the domain ontology* and linking the common domain ontology to the different medical databases available in the system. The proposed ontology was expressed in OWL and built using the Protege ontology building tool for W3C standardisation issues. The second contribution was a tool that allowed a semi-automatic map from the databases to the proposed ontology [21]. This graph-based tool was of great help to the domain experts. Finally, the *security rights* for the systems were formalised into a separate ontology using Conceptual Graphs to allow for both reasoning and domain expert feedback [65].

The second project I was involved in was *Open Knowledge* (http://www. openk.org). The purpose of the project was to investigate a simple *Multi-Agents coordination language* for the sharing of multi-media resources. My involvement in the project looked at Semantic Web aspects and Multi-Agents Systems aspects. More precisely, I was involved into adapting different standard multi-media ontologies for our project. I was also working on indexing aspects for ontology sharing [38]. From a Multi-Agents viewpoint one of the challenges involved in the project was the *elicitation of agent preferences over resources*. The agents had to be able to define preferences over a set of well defined resources. When these preferences are expressed using numerical values this problem corresponds to the problem of designing bidding languages for Combinatorial Auctions. In Combinatorial Auctions an auctioneer has a number of resources (objects) for sell, and the bidders will give numerical values on subsets of those objects. The Winner Determination is based on finding the allocation of objects to bidders that maximises the auctioneer's revenue. It is

a NP-hard problem being equivalent to weighted set packing. In this context I have proposed a *graph-based bidding language* based (NET-bid) on an extension of network flows. This representation was also provided with according semantics where the max-flow on the aggregated NET-bids from all users corresponded to Winner Determination [20].

My PostDoc allowed me to continue my research interests in investigating graph-based KRR, either for FOL reasoning for the Semantic Web, or for reasoning in Multi-Agents Systems.

This line of work (graph-based KRR) was pursued further when arriving at LIRMM where the graph-based formalisms I have investigated can be viewed from two perspectives: a paradigm perspective and an application perspective. The two main paradigms (in the continuity of my PhD and PostDoc work) I have investigated are Multi-Agents Systems and ontological query answering. The application perspective was given by agronomy applications and bibliographic applications. The application domains allowed to merge the two paradigms of my research. I investigated the link between query answering under inconsistency and argumentation in the context of agronomy applications. I also looked at the problem of Multi-Agents knowledge allocation and information selling mechanism design. Currently I am investigating the link between inconsistency based reasoning and clustering semantics in the context of bibliographic knowledge bases.

In the remainder of the document I will detail the results obtained for each paradigm: conjunctive query answering and Multi-Agents systems, as well as results obtained at the intersection of these two AI domains. The conjunctive query answering work was in the continuity of my PhD work in conceptual graphs and extended towards three main directions to address current challenges in applications: *large knowledge bases*, *inconsistent knowledge bases* and *query answer allocation*. The Multi-Agents systems graph based knowledge representation and reasoning formalisms I investigated are centered around *resource allocation* (combinatorial auctions, coalition formation), *norm representation* and *argumentation*. The applications I have investigated are representation and reasoning in the *bibliographical domain* and representation and reasoning for *agri-food chains*.

It is also worthy to note that while at LIRMM I have co-initiated the *GKR@IJCAI International Workshop Series (2009, 2011, 2013)* reporting on graph-based KRR formalisms (that is, endowed with graph-based syntax and logical semantics). In 2010 I was also the *program chair* of the ICCS (International Conference on Conceptual Structures) conference, the main international conference focused on Conceptual Graphs and in 2014 I was the *general chair* of ICCS.

Let me conclude this section by noting that since 2010 I am part of the GraphIK INRIA group, a project team which gathers LIRMM members as well as INRA members. Since its creation I got involved with agronomy applications working closely with INRA colleagues on three projects: EcoBioCap, DUR DUR

and CEPIA AIC. The lessons learnt and the results obtainedS during this collaboration shaped my vision for my future research as explained in my research plan. Let us first enumerate the obtained results since 2008 onwards.

1.3   Thesis Organisation

I chose to present the research work I did in the past six years while at LIRMM, by the means of a collection of publications, each explained and put in context in the chapters at the beginning of the thesis.

Each chapter gives a quick and intuitive "story" of the research topics I am interested in. Then at the end of each chapter one or two significant papers are cited (and the reader directed towards the chapter that shows the papers in question). The papers are chosen given their potential to illustrate the point I was trying to bring across in the respective Section.

The chapters are grouped by: theory, application and future work. In Chapter 2, I present the main theoretical domain of research I have been interested in and namely graph based knowledge representation and reasoning. This Chapter is split in several Sections. In Section 2.2 I demonstrate labelled bipartite graphs, homomorphism and their application to query answering. In Section 2.3 I show network flows and their application to Multi-Agents resource allocation. In Section 2.4 I talk about argumentation graphs. Mixed domain results are then presented in Section 2.5 where the setting of Multi-Agents knowledge allocation is presented as well as in Section 2.6 where inconsistent query answering and argumentation equivalent semantics are revealed. All papers detailing the results of this Chapter are available in Chapter 5.

Chapter 3 presents the research projects I have participated in that involve non computing experts. This choice of presentation is due to the fact that such projects bring non negligible challenges in mixing applied and fundamental research that are not easy to overcome. In Section 3.2 I present an agronomy project aiming at improving packaging conception. In Section 3.3 and Section 3.4 I present two other agronomy projects, this times aimed at food challenges. Finally in Section3.5 I present a bibliographic data management project. All papers detailing the work presented in this Chapter are available in Chapter 6.

Chapter 4 concludes the thesis by presenting a research project aiming at developing a hybrid architecture for inconsistency handling in the agronomy domain.

Let me make a note by saying that this is *a posteriori* choice in presentation. Research is an online process and different interesting topics of research as well as application projects cannot be fully determined in advance. The presentation of work in this thesis and "how things happened" are not always identical. Going in more details of certain topics, while abandoning others that looked promising were not always research justified and depend greatly on the time, collaborators,

chance and funding opportunities. However, from my first year of PhD the main underlying topic of my work is the use of graph structures for knowledge representation and reasoning.

Chapter 2

Reasoning Using Graphs

2.1 Introduction

This chapter focuses on the core of my research so far and namely my interest in using graph-based formalisms for knowledge representation and reasoning. Knowledge representation and reasoning (KRR) has long been recognized as a central issue in Artificial Intelligence (AI). The development of effective techniques for KRR is a crucial aspect of successful intelligent systems. Different representation paradigms, as well as their use in dedicated reasoning systems, have been extensively studied in the past. The problem consists of how to encode human knowledge and reasoning by symbols that can be processed by a computer to obtain intelligent behavior.

New challenges, problems, and issues have emerged in the context of knowledge representation in AI, involving the logical manipulation of increasingly large information sets (see for example Semantic Web, BioInformatics and so on). Improvements in storage capacity and performance of computing infrastructure have also affected the nature of KRR systems, shifting their focus towards representational power and execution performance. Therefore, KRR research is faced with a challenge of developing knowledge representation structures optimized for large scale reasoning.

This new generation of KRR systems includes graph-based knowledge representation formalisms also appealing for their visual qualities. This is due to several factors:

- Firstly, graphs are simple mathematical objects (they only use elementary naive set theory notions such as elements, sets and relations) which have graphical representations (a set of points and lines connecting some pairs of points) and thus can be visualised.

- Secondly, there is a rich collection of efficient algorithms for processing graphs, thus graphs can be used as effective computational objects (they

are widely used, for instance, in Operational Research).

- Thirdly, graphs can be equipped with a logical semantics: the graph-based mechanisms they are provided with are sound and complete with respect to deduction in the assigned logic.

Furthermore, graph-based mechanisms can be explained to the user because they can be easily visualized on the graphs themselves. The results presented in this chapter will be put in context by the applications presented in Chapter 3. The results include the use of bipartite graphs and labelled graph homomorphisms for query answering (Section 2.2), network flow algorithms for multi agent resource allocation (Section 2.3) and argumentation graphs (Section 2.4). Results obtained at the intersection of the above fields are next presented. Section 2.5 shows how to use allocation techniques for query answering. Section 2.6 shows the link between inconsistent query answering techniques and argumentation.

Every section presents the general setting of the research question investigated. For more details, each section is illustrated by a representative paper in Chapter 5.

2.2 Query Answering and Graphs

One of the fundamental problems in KRR languages is entailment checking: is a given piece of knowledge entailed by other pieces of knowledge, for instance from a knowledge base (KB)? Another important problem is consistency checking: is a set of knowledge pieces (for instance the knowledge base itself) consistent, i.e., is it sure that nothing absurd can be entailed from it? The query answering problem asks for the set of answers to a query in the KB. In the special case of boolean queries (i.e., queries with a yes/no answer), it can be recast as entailment checking.

The ONTOLOGY-BASED DATA ACCESS (ODBA) problem [43] takes a set of facts, an ontology and a conjunctive query and aims to find if there is an answer / all the answers to the query in the facts (eventually enriched by the ontology). The large number of ontologies and data sources defined on top of ontologies on the Web brought-on further challenges due to the data integration inherent to this setting: identity problems and large knowledge bases.

We consider these problems one by one in the remainder of the section. Section 2.2.1 discusses the advantages of using graph-based formalisms in the query answering context. Section 2.2.2 presents a generic platform for reasoning in the heterogeneous data setting of OBDA.

2.2.1 Conceptual Graphs Query Answering

Conceptual Graphs were introduced by John Sowa (cf. [56, 57]) as a diagrammatic system of logic with the purpose "to express meaning in a form that is logically precise, humanly readable, and computationally tractable" (cf. [57]). Throughout the remainder of this thesis we use the term "Conceptual Graphs" to denote the family of formalisms rooted in Sowa's work and then enriched and further developed with a graph-based approach (cf. [16]).

In Conceptual Graphs all kinds of knowledge are encoded as graphs and thus can be visualised in a natural way:

- The vocabulary, which can be seen as a basic ontology (background knowledge) is composed of hierarchies of concepts and relations. These hierarchies can be visualized by their Hasse diagram, the usual way of drawing a partial order.

- All other kinds of knowledge are based on the representation of entities and their relationships. This representation is encoded by a labelled graph, with two kinds of nodes, respectively corresponding to entities and relations. Edges link entity nodes to relation nodes. These nodes are labelled by elements in the vocabulary.

These graphs have a semantics in first-order logic (FOL), i.e., a knowledge base can be translated into a set of first-order logical formulas. Reasoning tasks operate directly on the knowledge defined by the user and not on their translation into logical formulas. This is done by the means of labelled homomorphism between the knowledge base graph and the query graph. Stated in an other way, the logical semantics is only used to formally ground the graph model, i.e., representation and reasoning mechanisms. This makes it possible to explain reasoning to the end-user because it can be visualized in a natural way on the pieces of knowledge he/she is familiar with. Would a logical prover be used on the logical translation of these pieces of knowledge to compute reasoning, reasoning would become a black box for the user and could not be explained. In the GraphIK group the Cogui tool has been developed: a graph based knowledge representation and reasoning editor implementing the Conceptual Graphs model.

In the paper "Visual Reasoning with Graph-based Mechanisms: the Good, the Better and the Best" by Michel Chein, Marie-Laure Mugnier and Madalina Croitoru [17] (available in Chapter 5) we show the main constructs of this language and the labelled graph homomorphism operation that allows to reason with Conceptual Graphs. We illustrate these operations on an annotation application and use Cogui in order to depict the graphs and illustrate the practical added value of using Conceptual Graphs.

In knowledge bases (KB), the open world assumption and the ability to express variables may lead to an answer redundancy problem. This problem oc-

curs when the returned answers are comparable. In collaboration with Michel Leclere and Nicolas Moreau we defined a framework to distinguish amongst answers. Our method is based on adding contextual knowledge extracted from the KB that allows clarification of the notion of redundancy between answers. Our work was done in a Conceptual Graphs setting and we provided a definition for the set of answers to be computed from a query, which ensures both properties of non-redundancy and completeness. A special case of this problem (when the answer is a concept node) corresponds to the problem of Generation of Referring Expressions (GRE) studied in the Conceptual Graphs context in [27], extended in this query answering setting.

In the paper "Distinguishing Answers in Conceptual Graph Knowledge Bases" by Nicolas Moreau, Michel Leclere, and Madalina Croitoru [46] (available in Chapter5) we address the problem of query answers identification.

Let us mention that the RDF (Resource Description Framework) [1] query language SPARQL offers a way to describe answers (by the DESCRIBE primitive) but this is not done according to semantically sound syntactic criteria. More generally *RDF* is close to our Conceptual Graphs framework. RDF has a graph-based representation, it is provided with a formal semantics and an associated entailment notion ([36], [37]), but it does not come with an effective reasoning mechanism, even less with a graph-based mechanism that would operate on the graph representation. It is worth mentioning at this point that a translation from RDF to Conceptual Graphs has been investigated in [7] and implemented in Cogui. This work has been taken further and, since the logical subset represented by Conceptual Graphs and RDF(S) overlaps, a generic approach to KRR was investigated. This is presented in the next Section 2.2.2.

2.2.2 The ALASKA Platform

A problem particularly relevant in the context of multiple data/metadata sources is querying hybrid knowledge bases. In a hybrid knowledge base, each component may have its own formalism and its own reasoning mechanisms. Several languages have been proposed in the literature where the language expressiveness / tractability trade-off is justified by the needs of given applications. In description logics, the need to answer conjunctive queries has led to the definition and study of less expressive languages, such as the $\mathcal{EL}$ [6] and DL-Lite families [13]. Properties of these languages were used to define profiles of the Semantic Web OWL 2 language (`www.w3.org/TR/owl-overview`).

When the above languages are used by real world applications, they are encoded in different data structures (e.g. relational databases, Triple Stores, graph structures). Justification for data structure choices include (1) storage speed (important for enriching the facts with the ontology) and (2) query efficiency. Therefore, deciding on what data structure is best for one's application

---

[1]`http://www.w3.org/TR/REC-rdf-syntax/`

is a tedious task. While storing RDF(S) has been investigated from a database inspired structure [35], other logical languages did not have the same privilege. Even RDF(S), often seen as a *graph*, has not been thoroughly investigated from a *ODBA perspective* w.r.t. *graph structures* and emergence of *graph databases* in the NoSQL world.

ALASKA (acronym stands for **A**bstract and **L**ogic-based **A**rchitecture for **S**torage systems and **K**nowledge bases **A**nalysis) is a Java library dedicated to the storage and querying of large knowledge bases. It intends to be the foundation layer of our OBDA (Ontology Based Data Access) software developments. ***ALASKA has been built, first as part of a Masters thesis, and then part of the PhD of Bruno Paiva Lima da Silva (co-supervised with J.-F. Baget).***

The ALASKA core (data structures and functions) is written independently of any language used by storage systems it will access. The advantage of using a subset of First Order Logic to maintain this genericity is to be able to access and retrieve data stored in any system by the means of a logically sound common data structure. Local encodings will be transformed and translated into any other representation language at any time. Basically, the abstract layer of ALASKA is a logical layer. Storage systems are used to store data that can be seen as sets of logical atoms. Wrappers are used to encode this atom according to the storage system paradigm. Whatever this storage system, ALASKA only reads and writes atoms or sets of atoms.

This abstract layer is not only used to read and write in an uniform manner into various storage systems, but also to process queries. A conjunctive query can also be seen as a set of atoms. ALASKA is able to transform them into, for example, SQL or SPARQL queries, to benefit from the native querying mechanism of specific storage systems. Moreover, a generic backtracking algorithm has been designed, that allows to process these queries on any of these storage systems. This backtrack relies upon elementary queries, that check whether or not a grounded atom is stored in the system, or enumerate all atoms that specialize a given one. This backtrack does not incorporate powerful optimizations and pruning features, since it is designed to process simple queries.

In the paper "ALASKA for Ontology Based Data Access" by Jean-Francois Baget, Madalina Croitoru and Bruno Paiva Lima da Silva [8] (available in Chapter 5) we give a demonstration of the ALASKA architecture.

2.3   Resource Allocation and Graphs

Representing agent preferences is another way for eliciting an agent's knowledge. This can be further used in argumentation or normative systems (see Section 2.4). A preference is defined as a function from all the subsets of the set of resources / goals / pieces of knowledge over which the agent is expressing its preferences into a set of values. If the values are given numerically, the order re-

17

lation over preferences is total. If the values are not numeric, the (partial) order relation over preferences also has to be provided. The allocation of resources will then maximise the aggregated preferences of the agents. The Multi Agent Resource Allocation (MARA) setting has numerous applications in Multi-Agents Systems, notably for Combinatorial Auctions and Coalition Formation.

Combinatorial auctions (CAs) can be viewed as a method of allocating multiple heterogeneous goods among a number of individuals, each of whom can obtain not one but a bundle of goods. Bidding is the process of transmitting one's valuation function over the set of goods on offer to the auctioneer. The communication language is called a bidding language and it plays a key role in both central aspects of the allocation problem: preference elicitation and winner-determination (WD). Several bidding languages for CAs have previously been proposed, arguably the most compelling of which allow bidders to explicitly represent the logical structure of their valuation over goods via standard logical operators (e.g., [47]).

We proposed in [20] a visual paradigm for representing bidding languages for combinatorial auctions based on generalised network flows (used to represent the bids) and, the interpretation of the WD as a fair aggregation of individual preferences making use of max flow algorithms. Our work on combinatorial auctions led me to take an interest in a very closely related field: coalitional games.

Coalitional games are theoretical constructs that capture opportunities for cooperation by explicitly allowing agents to take joint actions as primitives [40]. One of the key issues in multi-agent coalition formation is the Coalition Structure Generation (CSG) problem, which involves dividing the set of agents into subsets (i.e., coalitions) so that the overall efficiency of the system is maximised. Such a division is referred to as a coalition structure. This problem, which is extremely challenging due to the exponential size of the search space, has recently attracted considerable attention in the Multi-Agents System literature [54, 52, 51]. It is typically assumed that the coalitions within every coalition structure are disjoint, and that they cover the entire set of agents. Moreover, the input to the problem is typically represented using a characteristic function, which assigns a value for every possible coalition, representing its performance.

The same network flow based representation was used as the backbone to a Generalised Network Flow representation for Coalition Games and gave the CF-NETs. Specifically, this representation is based on the observation that the coalition formation process can be viewed as the problem of directing the flow through a network where every edge has certain capacity constraints. We show that our new way of representing this process is intuitive, fully expressive, and allows for representing certain patterns in a significantly more concise manner compared to the conventional approach.

In the paper "Coalitional Games via Network Flows" by Talal Rahwan, Tri-Dung Nguyen, Tomasz P. Michalak, Maria Polukarov, Madalina Croitoru and Nicholas R. Jennings [53] (available in Chap-

ter 5) we show that CF-NETs have the flexibility to represent charac-teristic function games, coalitional games with overlapping coalitions, and coalitional games with identical agents.

2.4 Argumentation, Norms and Graphs

As explained in Section2.1, the Ontology Based Data Access (OBDA) setting can yield to data inconsistency. Argumentation theory is a well-known method for dealing with inconsistent knowledge. Logic-based argumentation considers constructing arguments from inconsistent knowledge bases, identifying attacks between them and selecting acceptable arguments and their conclusions. In order to know which arguments to accept, one applies a particular argumentation semantics.

When arguing about a course of actions to pursue, agents can introduce a multitude of arguments aimed at supporting their position and undermining other possible views. When represented in some underlying logic, formal argumentation theory provides the tools to reason about these arguments, allowing one to, for example, identify a consistent set of arguments and the course of actions they advocate. A core question that arises is what arguments an agent should utter in order to achieve these goals. This dialogue planning problem is, in most cases, computationally challenging, and work on argument strategy [3, 41, 48, 55] has identified heuristics which are used to guide an agent's utterances.

In [22] we sought to use a graph-based method to identify all arguments that an agent must advance at a specific point in time. ***The salient point of our work was that the argument structure was translated into a network flow structure, and then we could utilise graph operations (max flow) in order to calculate the appropriate set of arguments to advance. This paper, "Information Revelation Strategies in Abstract Argument Frameworks Using Graph Based Reasoning." by Madalina Croitoru and Nir Oren is available in Chapter 5.***

The argumentation work also led me to investigate *normative systems*. The reason for investigating norms stemmed from the fact that some of the agents' arguments could have a different impact rather than other agents. These agents could impose norms of behaviour on all agents that should be respected. In normative systems, the norms to be adhered by all agents are represented using tuples of First Order Logic formulae. These norms apply on all agents and the question is: at a given time and for a given state of the world what norms are violated. ***In the paper "Graphical norms via conceptual graphs" by Madalina Croitoru, Nir Oren, Simon Miles and Mike Luch [23] (available in Chapter5) we looked into representing the norm struc-ture using a graph-based KRR where normative reasoning corresponds to graph-based operations on these graphs. This work has been ex-tended for reasoning with different kind of norms (permissions, obli-***

gations).

2.5 Resource Allocation for Query Answering

The assumption behind semantic data integration and querying is that different agents accessing the integrated data repository will have equal interest in the querying results. In the setting of the query answering problem the multiplicity of knowledge requesters was simply regarded as a simple extension of the individual case. This assumption does not always hold in practical applications where the requesters are in direct competition for knowledge (newsagents, military applications). This is not always true in a data sensitive scenario where the knowledge provider might want to allocate the query answers to the agents based on their valuations. Furthermore, the agents might want some information exclusively (and thus offer a valuation that allows it) while others might want it shared. To this end, new mechanisms of allocation of query answers are needed.

We draw from the fields of query answering in information systems and multi agent resource allocation and propose the multi agent knowledge allocation setting (MAKA). We have proposed a graph based method, inspired by network flows, for solving it. These results were investigated jointly with Dr. Sebastian Rudolph from University of Dresden. *In the paper "Exclusivity-based allocation of knowledge" by Madalina Croitoru and Sebastian Rudolph [25] (available in Chapter 5) we define a bidding language based on exclusivity-annotated conjunctive queries and show a way to succinctly translate the allocation problem into a graph structure allowing to employ a wide range of constraint solving techniques to find the optimal allocation.*

I also investigated the mechanism design aspects of such valuations in collaboration with Dr. Iannis Vetsikas from the University of Athens. If we were to apply well known auction mechanisms to selling pieces of information that can be shared, hence are infinitely copied, no profit would be made; this would happen because competition is what drives the prices up and offering more items - here infinite - than buyers essentially removes any competition in this setting. And yet getting profit is usually the first goal of a seller. Moreover, the mechanisms for auctioning this kind of goods need to be incentive compatible, meaning that bidders would have an incentive to lie which would break down the mechanism.

In the paper "How much should you pay for information?" by Ioannis Vetsikas and Madalina Croitoru [64] (available in Chapter 5) we present and analyse several incentive compatible mechanisms for selling a single sharable good to bidders who are happy to share it, aiming at creating competition by restricting the number of winners.

2.6 Argumentation for Query Answering

As previously mentioned, argumentation theory is a well-known method for dealing with inconsistent knowledge [9, 2]. Logic-based argumentation [10] considers constructing arguments from inconsistent knowledge bases, identifying attacks between them and selecting acceptable arguments and their conclusions. Recently, this question was also considered in the OBDA case [42, 11] where maximal consistent subsets of the KB, called repairs, are then considered and different semantics (based on classical entailment on repairs) are proposed in order to compute the set of accepted formulae. Once the repairs calculated, there are different ways to calculate the set of facts that follow from an inconsistent knowledge base. For example, we may want to accept a query if it is entailed in all repairs (AR semantics). Another possibility is to check whether the query is entailed from the intersection of closed repairs (ICR semantics). Finally, another possibility is to consider the intersection of all repairs (IAR semantics).

Our work starts from the observation that both inconsistent ontological KB query answering and instantiated argumentation theory deal with the same issue, which is reasoning under inconsistent information. Furthermore, both communities have several mechanisms to select acceptable conclusions. Natural questions one could immediately ask are: Is there a link between the semantics used in inconsistent ontological KB query answering and those from argumentation theory? Is it possible to instantiate Dung's [31] abstract argumentation theory in a way to implement the existing semantics from ontological KB query answering? If so, which semantics from ontological KB query answering correspond to which semantics from argumentation theory? Does the proposed instantiation of Dung's abstract argumentation theory satisfy the rationality postulates [15]?

There are several benefits from answering those questions. First, it would allow to import some results from argumentation theory to ontological query answering and vice versa, and more generally open the way to the Argumentation Web [50]. Second, it might be possible to use these results in order to *explain* to users how repairs are constructed and why a particular conclusion holds in a given semantics by constructing and evaluating arguments in favour of different conclusions [30]. Also, on a more theoretical side, proving a link between argumentation theory and the results in the knowledge representation community would be a step forward in understanding the *expressibility* of Dung's abstract theory for logic-based argumentation [63].

This work has been done jointly with Dr. Srdjan Vesic from CRIL, Univ. Artois. *In the paper "What Can Argumentation Do for Inconsistent Ontology Query Answering?" by Madalina Croitoru and Srdjan Vesic [28] (available in Chapter 5) we show that it is possible to instantiate Dung's abstract argumentation theory in a way to deal with inconsistency in an ontological knowledge base (KB). Second, we formally prove the links between the semantics from ontological KB query an-*

*swering and those from argumentation theory: **ICR** semantics corresponds to sceptical acceptance under stable or preferred argumentation semantics, **AR** semantics corresponds to universal acceptance under stable / preferred argumentation semantics and **IAR** semantics corresponds to acceptance under grounded argumentation semantics. Third, we show that the instantiation we define satisfies the rationality postulates.*

The results allowed to use results from argumentation theory in order to explain reasoning under inconsistency in knowledge bases. This topic is at the center of the thesis of Abdallah Arioua (co-supervised with P. Buche and J. Fortin) and first results have already been showed. ***In the paper "On Conceptual Graphs and Explanation of Query Answering Under Inconsistency" by Abdallah Arioua, Nouredine Tamani and Madalina Croitoru [4] (available in Chapter 5) we present query answering explanation strategies inspired from the link between the OBDA inconsistent-tolerant semantics and argumentation acceptance semantics.***

Chapter 3

Inter-Disciplinary Projects

3.1   Introduction

In this chapter I detail the research projects I have been involved in since my arrival at LIRMM in 2008. I will pay special attention to the interdisciplinary projects, where at least one of the main participants is not a computing science center. I believe that applying research in practice is a major challenge of current Artificial Intelligence (I only focus on Artificial Intelligence since it is my primary field of expertise). The time and expertise needed to develop both aspects and the incertitude inherent to research and development make this a difficult task to fulfill.

In this chapter I will detail three projects within an agronomy scenario. One project looks at packaging aspects (Section 3.2) while the other two focus on food aspects (Section 3.4 and Section 3.3). While the project presented in Section 3.4 has only started in 2014, I think it is important to show the first results that were obtained. The fourth project I will detail is a project made in collaboration with bibliographic data experts presented in Section 3.5.

3.2   EU FP7 EcoBioCap

Conceiving food packagings that are taking into account all factors from different stakeholders in the production chain is an exciting, timely and important challenge to face. Changes such as (1) growing demands for ready to eat foods, (2) globalisation of the food business, (3) limiting the amount of waste generated by packagings are driving a search for innovative ways to package foods while maintaining quality, freshness, and safety. Packaging conception should take into account many aspects. Criteria such as visual appeal, labelling norm compliance, stability, safety, environmental impact and cost-effectiveness have to be considered before full exploitation. However, taking into account all cri-

teria can yield contradictory actions. Domain experts cannot express complete rules (along with the many exceptions of such rules) due to the complexity of the knowledge. Therefore, when trying to reason with the incompletely elicited knowledge available numerous inconsistencies occur and the important problem underlying the food packaging conception comes down on to how to reason under inconsistency and incompleteness with non monotonic knowledge.

Within the framework of the European project EcoBioCap (`www.ecobiocap.eu`) about the design of next generation packagings using advanced composite structures based on constituents derived from the food industry, we developed a Decision Support System (DSS) for packaging material selection. The DSS is made of two parts: (i) a flexible querying process which is based on a bipolar querying approach [29] dealing with imprecise data corresponding to the characteristics related to the food product to pack like the optimal permeance, the dimension of the packaging, its shape, etc., and (ii) an argumentation process which aims at aggregating several stakeholders requirements expressed as simple text arguments, to enrich the querying process by stakeholders' justified preferences. The former implements a database containing the respiration parameters of the packed food, and a second database storing the characteristics of packaging materials (O_2 and CO_2 permeance, biodegradability, transparency, etc.). The user can also specify some preferences such as the preferred storage temperature, dimensions of the packaging, etc, which could be mandatory or optional. This part of the software combines these inputs to compute the optimal permeance which guaranties the best shelf life for the packed food. Then, this optimal permeance are mixed with other user preference to form a bipolar query addressed to the packaging database. The returned list of packaging is ranked from the most to the least relevant one with regard to the expressed preferences.

For example, in order to pack cheese, researchers focus on the permeance properties in their following argument: *"a wheat gluten based material packaging is suitable for cheese because it offers a good atmosphere control"*, and cheese producers can retort with the following counter-argument *"a gluten layer cannot put in contact with cheese since the bacteria in the crust would degrade the packaging"*, considering here the interaction with the packed food aspect. As we see packagings have to be selected according to several aspects or criteria (permeance, interaction with the packed food, end of life, etc.), highlighted by the expressed stakeholders' arguments. To collect the user preferences, different surveys have been carried out among the stakeholders involved in the design of packaging: researchers, packaging industries, food producers, consumers, etc. The asked questions are about cost, end of life, biodegradability, the use of nanoparticles in the packaging, etc. For each criterion expressed through surveys, the stakeholders identified its importance, indicated the preferred values, and the reasons that justify their choice. The importance permits to give a priority to criteria over others, used in the bipolar approach considering mandatory and optional preferences. For instance, sanitary criteria ensuring a good preservation of the packed food are naturally more important than the color or the

transparency of the packaging. The values, in the other hand, can be easily used as predicates in the bipolar query.

Please note that the problem at hand does not simply consist in addressing a multi-criteria optimization problem [12]: the domain experts would need to be able to justify why a certain packaging (or set of possible packagings) are chosen.

Work in this project has been carried out jointly with members of IATE working on packaging conception. A joint publication (as seen below) with the experts was produced in which we detail the architecture and use of the module implemented. Part of the project, jointly with Patrice Buche, I have co-supervised Nouredine Tamani (a post-doctoral researcher) and Patricio Mosse (a Master student and then a programming assistant).

To be able to model this kind of reasoning, we need a structured format for arguments in which it is possible to express the involved concepts and rules modeling the semantics behind the text. We rely in this work on a subset of a logical structured argumentation system which corresponds to the smallest common subset of approaches presented in [1, 49, 45]. Such structured argumentation system (i) allows the expression of logical arguments as a combination of facts and inference rules, (ii) defines attacks and defeat relations between arguments based on a logical conflict notion.

The reasoning process underlying arguments is related to a domain which can be easily set-based interpreted. It is then possible to consider certain subsets of description logics to define concepts and rules forming arguments, for their well known trade off between a good level of expressivity and computational decidability. Here we make a choice to use the DLR-Lite description logic [18, 14] since it also allows a direct connection to a database schema and relational tables needed in the second step of querying. Concepts are defined by m-ary relations connected to sets of tuples from a database and rules can be expressed by natural subsumption between concepts. In this way, the interpretation of the arguments is closely related to the domain of definition of a database, and the result of the argumentation process can be directly harnessed in the querying phase. For each criterion expressed through surveys, the stakeholders identified the reasons that justify their choice which were modelled as arguments pros or cons some choices or values. They can justify why a packaging is better than another and can be then used to enrich the bipolar querying system. So, we detail in the next sections how arguments are logically modelled within a structured argumentation system and how the delivered justified conclusions can be used in the querying process. In Chapter 6 we show two papers:

- *The paper "Conflicting Viewpoint Relational Database Querying: An Argumentation Approach" by Nouredine Tamani, Madalina Croitoru and Patrice Buche [60] (available in Chapter 6) summarises the use of argumentation in EcoBioCap and provides related theoretical work pointers.*

25

- *In the paper "Eco-Efficient Packaging Material Selection for Fresh Produce: Industrial Session" by Nouredine Tamani, Patricio Mosse, Madalina Croitoru, Patrice Buche, Valerie Guillard, Carole Guillaume and Nathalie Gontard [61] (available in Chapter 6) we show the architecture of our DSS based on argumentation.*

3.3   INRA CEPIA AIC T80 - T65

The case of study considered in this project relates to the debate around the change of ash content in flour used for common French bread. Various actors of the agronomy sector are concerned, in particular the Ministry for Health through its recommendations within the framework of the PNNS ("National Program for Nutrition and Health"), the millers, the bakers, the nutritionists and the consumers.

The PNNS recommends to privilege the whole-grain cereal products and in particular to pass to a common bread of T80 type, i.e., made with flour containing an ash content (mineral matter rate) of 0.8%, instead of the type T65 (0.65% of mineral matter) currently used. Increasing the ash content comes down to using a more complete flour, since mineral matter is concentrated in the peripheral layers of the wheat grain, as well as a good amount of components of nutritional interest (vitamins, fibres). However, the peripheral layers of the grain are also exposed to the phytosanitary products, which does not make them advisable from a health point of view, unless one uses organic flour.

Other arguments (and of various nature) are in favour or discredit whole-grain bread. From an organoleptic point of view for example, the bread loses out in its "being crusty". From a nutritional point of view, the argument according to which the fibres are beneficial for health is discussed, some fibres could irritate the digestive system. From an economic point of view, the bakers fear selling less bread, because whole-grain bread increases satiety − which is beneficial from a nutritional point of view, for the regulation of the appetite and the fight against food imbalances and pathologies. However whole-grain bread requires also less flour and more water for its production, thus reducing the cost. The millers also fear a decrease in the quality of the technical methods used in the flour production.

Beyond the polemic on the choice between two alternatives (T65 or T80), one can take the debate further by distinguishing the various points of view concerned, identifying the desirable target characteristics, estimating the means of reaching that point. The contribution of our work is showing how using argumentation can help towards such practical goals.

Reverse engineering is known to be challenging from a methodological viewpoint due to the difficulty of defining the specifications for the expected finished product. The desired quality criteria are multiple, questionable, and not necessarily compatible. Furthermore, the impact of different steps of food processing

26

and their order is not completely known. Some steps are more studied than others, several successive steps can have opposite effects (or unknown effects), the target criteria may be outside of the characteristics of products.

We proposed a methodology combining the reverse engineering and logical based argumentation for selecting the actions to take towards the agronomy application at hand. We used the Conceptual Graphs knowledge representation and reasoning formalism (see Chapter 2 and the logical argumentation instantiation of [28].

The set of goals, viewpoints, as well as the knowledge associated with the goals /viewpoints is elicited either by the means of interviews with the domain experts or manually from different scientific papers. This step of the application has been the most time consuming but the most important. If the knowledge elicited is not complete, sound or precise the outcome of the system is compromised. Then, based on the knowledge elicited from the knowledge experts and their goals, we enrich the knowledge bases using reverse engineering (implemented using backwards chaining algorithms i.e., algorithms that work up from the query, using the rules, to enrich the set of facts in the knowledge base). Putting together the enriched knowledge bases obtained by backwards chaining from the different goals will lead to inconsistencies. The argumentation process is used at this step and the extensions yield by the applications computed. Based on the extensions and the associated viewpoints we can use voting functions to determine the application choice of viewpoints.

The evaluation of the system implemented was done via a series of interviews with domain experts. The above knowledge and reasoning procedures were implemented using the Cogui knowledge representation tool with an extension of 2000 lines of supplemental code. Three experts have validated our approach: two researchers in food science and cereal technologies of the French national institute of agronomic research, specialists respectively of the grain-to-flour transformation process and of the breadmaking process, and one industrial expert - the president of the French National Institute of Bread and Pastry.

Two interests of the approach were more particularly highlighted. They concern cognitive considerations. First, experts were conscious that the elicitation procedure was done according to their thought processes. The system was thus able to restitute the knowledge in a different manner than the experts usually do. Secondly, from a problem that could initially seem simple, the experts realised that it covered a huge complexity that a human mind could hardly address alone. The tool is currently available to them under restricted access.

In the paper "Decision support for agri-food chains: A reverse engineering argumentation-based approach" by Rallou Thomopoulos, Madalina Croitoru and Nouredine Tamani [62] (available in Chapter 6) we fully describe the technical details and knowledge represented within this application.

3.4 French ANR DUR-DUR

The ANR DURDUR project aims at re-organising the durum what agrifood chain in the aim of lowering pesticide use and increasing productivity and wheat disease resistance. The project only started at the beginning of 2014 thus the domain information is still scarce. This is the reason why the approach presented so far (and under implementation) is purely theoretical. But the approach addresses a key aspect of knowledge capitalisation in this setting: how to explain knowledge reasoning results to domain experts, not necessarily logicians. The task is even more daunting because the knowledge used as a backbone for reasoning is inconsistent and thus different inconsistency tolerant mechanisms need to be employed in order to perform deduction.

We address the problem of providing explanation facilities for query answering (query acceptance and query failure) in the ONTOLOGY-BASED DATA ACCESS (OBDA) setting under inconsistency-tolerant semantics (explained in Chapter 2). Query answering under these semantics may not be intuitively straightforward and can lead to loss of user's trust and satisfaction, affecting the system's usability [44]. Moreover, *explanation facilities* should not just account for user's "Why Q ?" question (why a query holds under a given inconsistency-tolerant semantics) but also for question like "Why not Q?" (why a query does not hold under a given semantics).

Given an inconsistent OBDA setting equipped with an inconsistency-tolerant semantics and given a boolean conjunctive query Q we consider two query answering problems and namely: $(\mathcal{RQ}_1)$ "Why does Q hold under such semantics?" and $(\mathcal{RQ}_2)$ "Why Q does not hold under such semantics?". Knowledge base explanation that takes the form of 'Justification' is more effective then other type of explanation (i.e. Line of Reasoning and Strategy). Justification according to the survey is considered as "the most effective type of explanation". This type of explanation aims at showing the reason why certain conclusion has drawn in particular circumstance. Following this result, we consider the logical instantiation of an argumentation framework for OBDA and exploit the equivalence between inconsistency tolerant semantics and argumentation semantics. The explanation takes the form of a dialogue between the *User* and the *Reasoner* with the purpose of explaining the query failure. At each level of the dialogue, we use language-based introduced primitives such as *clarification* and *deepening* to further refine the answer.

In the paper "Query Failure Explanation in Inconsistent Knowledge Bases Using Argumentation" by Abdallah Arioua, Nouredine Tamani, Madalina Croitoru and Patrice Buche [5] (available in Chapter 6) we provide the technical details of this promising approach.

3.5   French ANR Qualinca

The SUDOC (catalogue du Systeme Universitaire de Documentation) is a large bibliographical knowledge base managed by ABES (Agence Bibliographique de l'Enseignement Superieur). The SUDOC contains $\approx$ 10.000.000 document descriptions, and $\approx$ 2.000.000 person descriptions. A person description possesses some attributes (ppn[1], appellation set, date of birth...). A document description also possesses some attributes (title, ppn, language, publication date...) and a link to a person description. A link is labeled by a role (such as *author*, *illustrator* or *thesis advisor*) and means that the person has participated as the labelled role to the document.

One of the most important tasks for ABES experts is to reference a new book in SUDOC. To this end, the expert has to register the title, number of pages, types of publication domains, language, publication date, and so on. This new description represents the physical books in the librarian hands which he/she is registering. He/she also has to register people which participated to the book's creation. In order to do that, for each contributor, he/she selects every person description which has an appellation similar to the book contributor. Unfortunately, there is not that much information in person descriptions because the librarian politics is to give minimal information, solely in order to distinguish two person descriptions which have the same appellation, and nothing more (they reference books, not people). So the librarian has to look at document description which are linked in order to see whether the book in his/her hands seems to be a part of the bibliography of a particular candidate. If it is the case, he/she links the new document description to this candidate and looks at the next unlinked contributor. If there is no good candidate, he/she creates a new person description to represent the contributor.

This task is fastidious because it is possible to have a lot of candidates for a single contributor (as much as 27 for a contributor named Bernard Alain). This creates errors, which in turn can create new errors since linking is an incremental process. In order to help experts to repair erroneous links, we proposed two partitioning semantics in [33] which enable us to detect erroneous links in bibliographic knowledge bases. A partitioning semantics evaluates and compares partitions of elements couples (person description, document description).

In the paper "An analysis of the SUDOC bibliographic knowledge base from a link validity viewpoint" by Lea Guizol, Olivier Rousseaux, Madalina Croitoru, Yann Nicolas and Aline Le Provost [34] (available in Chapter 6) we practically evaluate the qualitative results of partitioning semantics on a real SUDOC sample. Let us mention that this work has been done within the PhD thesis of Lea Guizol supervised by Madalina Croitoru.

---

[1]A ppn identifies a description.

Chapter 4

Perspectives

4.1 Introduction

This chapter details the current and future research activities I currently work on and plan to undertake next. My long term research plan is to build an unified platform for knowledge representation and reasoning in presence of inconsistency. I plan to test and evaluate the practical need of this platform in the agronomy domain.

This chapter is structured as follows. I start by presenting the specificities of the agronomy domain and why reasoning in presence of inconsistency is a key aspect of knowledge representation and reasoning in this field (Section 4.2). A project for such a platform is then presented in Section 4.3. Of course this platform will evolve in the course of the next years as discussed in the final section of the chapter, Section 4.4.

4.2 Application Setting

A privileged domain application of my work in the past two years has been agronomy. The choice of the agronomy domain as a focus is motivated both by the local context of INRIA GraphIK project group (some members of GraphIK are also members of the UMR IATE) and by its adequation in terms of research challenges. The challenges in the agronomy field seem to be particularly well-adapted to artificial intelligence techniques. So far there are no mathematical models available to solve the problems related to the quality of agrifood chains. Their specificity is that such problems need to be stated at a conceptual level. Furthermore, solving these problems requires an integrated approach that takes into account expert knowledge, which is typically symbolic, as well as numeric data, vague or uncertain information and multi-granularity knowledge.

In agrifood chains, the products traditionally go through the intermediate

stages of processing, storage, transport, packaging and reach the consumer (the demand) from the producer (the supply). Due to an increase in quality constraints, several parties are involved in production process, such as consumers, industrials, health and sanitary authorities, etc. expressing their requirements on the final product as different point of views which could be conflicting. We accept that several complementary points of view - possibly contradictory - can be expressed (nutritional, environmental, taste, etc.). We then need to assess their compatibility (or incompatibility) and identify solutions satisfying a maximum set of viewpoints.

Eliciting (gathering knowledge from users) knowledge in such setting is a long process involving multiple and potentially conflicting viewpoints and actors. Knowledge representation needs to follow a language which is easily interoperable with other potentially useful knowledge repositories of the studied field. One of the main aspects of such representation effort is the capitalisation need. Choosing the knowledge representation language adapted for the application at hand in such way that the represented knowledge can benefit from similar efforts, and that it can be further re-used by other applications is an important point to consider in this setting.

Once the knowledge representation step over, reasoning capabilities should be made available. Reasoning in presence of inconsistency can be dealt with in two ways: fixing the inconsistency or living with the inconsistency.

The first alternative includes several methods from lifting to a more general knowledge representation language to deleting pieces of knowledge or adding pieces of knowledge. Please note that deleting pieces of knowledge, in certain cases is not ideal for our application since it means deleting some knowledge that the experts purposely represented in the system.

The second case presents many possibilities. So far, as methods of reasoning in presence of inconsistency, I have considered argumentation, belief revision and default logics. Each come with their own specific inconsistency handling use case: argumentation has been extensively investigated as an inconsistency handling framework due to its explanatory power, belief revision considers the scenario of adding / deleting information while default logics concern incomplete information (and thus this kind of method could be used in tight coupling with the first method).

We will show in the next section how we envisage to build a knowledge representation and reasoning platform that will make use of all these methods in a particular logically instantiated case equivalent to Conceptual Graphs. Furthermore, reasoning will be used in order to direct the knowledge elicitation process and, when possible, lift some inconsistencies.

4.3 Unified Inconsistency Reasoning

The primary and innovative aim of my project is to provide an unified methodology of elicitation and reasoning in presence of data inconsistency.

This will be done by two means:

- Eliciting new knowledge to lift, if possible, the inconsistency. By the means of automatically generated explanations for such inconsistencies – targeting the inconsistent subsets of knowledge (conflict sets)– new knowledge can be elicited from the end user. The new knowledge needs to be incorporated in the knowledge base in a semantic and principled manner.

- Answer queries in presence of such inconsistencies and explain the query answers.

Such approach is based on a given knowledge representation and reasoning logical language. I plan to use the logical subset equivalent to Conceptual Graphs given their graph-based representation and reasoning mechanisms (and the interaction with other Semantic Web languages via the Cogui platform). The reasoning in presence of inconsistency can use several methods such as argumentation, belief revision and non-monotonic logics such as default logics. Let us highlight at this point that the elicitation based on the inconsistency can only happen if the process is guided by a high level view point reasoning (which in turn can make use of argumentation techniques) and that will guide the belief revision process. A high level overview of the proposed project is given in Figure 4.1.

Before such tool can be built one must first face important foundational challenges:

- Knowledge Acquisition and Elicitation: Find a semi automatic method for extracting different knowledge (rules, facts, constraints) from controlled natural language text entered by domain experts.

- Given a knowledge base, devise algorithms to efficiently identify its minimal inconsistent subsets.

- Propose explanation strategies in knowledge bases and evaluate their practical appeal to domain experts.

- Use the explanation for the inconsistency as the basis to elicitate new knowledge.

- Incorporate new knowledge into the system and assess consistency.

- Reason (eventually in presence of inconsistency) and explain query answering results.
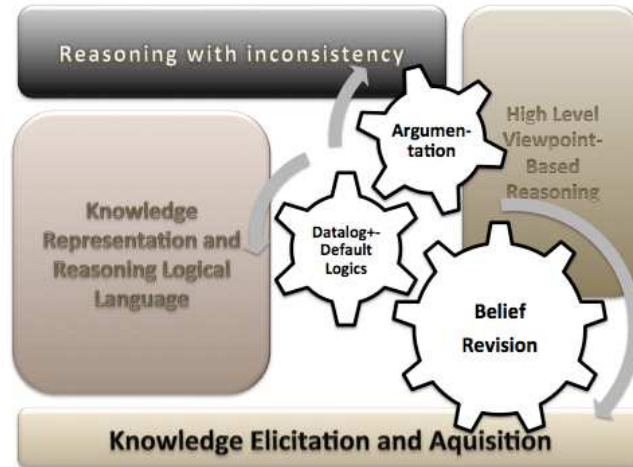
33

Figure 4.1: Hyena Project High Level Overview

We started this year to develop Hyena (HYbrid knowlEdge reasoNing and representAtion), a hybrid platform which allows to elicit, represent, reason with and explain inconsistency in Semantic Web knowledge bases.

We follow a rule-based representation of ontologies and base our work on the Alaska platform (see Section 2.2.2). Hyena relies on five modules (depicted in Figure4.2):

1. The *Knowledge Base Module*. The user imports a Knowledge Base using the ALASKA Manager, which stores it in an ALASKA project. Users import Knowledge Bases from different files: Facts are imported from RDF and N3 files, while Rules and Negative Constraints are imported from Datalog files.

2. The *Visualisation Module*. The *Visualisation Module*, which uses the popular Graphviz library, displays the knowledge base in a graphical manner on the screen.

3. The *Belief Revision Module*. The *Belief Revision Module* allows updating the (possibly inconsistent) knowledge base.

4. The *Reasoning Module*. The *Reasoning Module* allows for reasoning in presence of inconsistency.

5. The *Explanation Module*. The *Explanation Module* receives the query and one result, and analyses the (possibly inconsistent) Knowledge Base to understand how that result was reached.
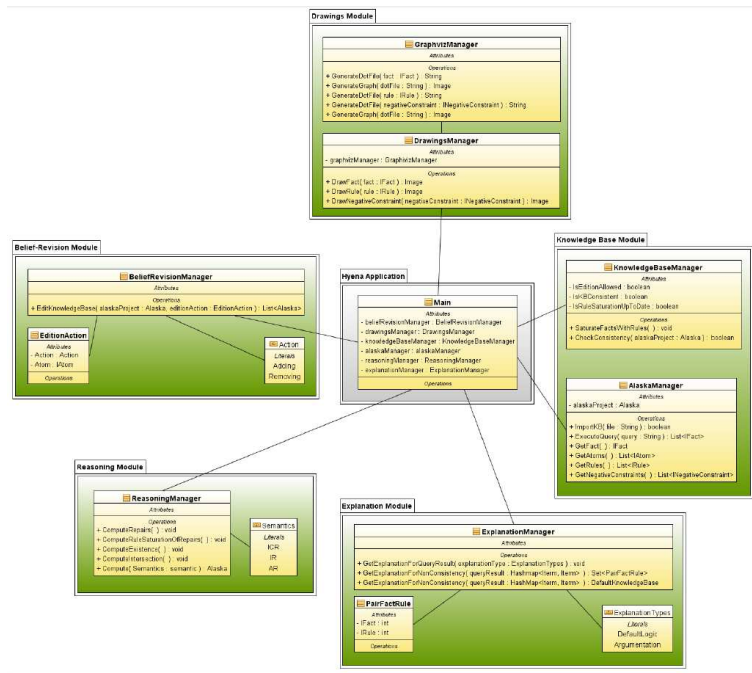
Figure 4.2: Hyena Class Diagram

In Figure 4.3 we depict the workflow of the Hyena architecture.

The user interacts with Hyena. In order to reason with knowledge a high level view point, filtering is needed (this is needed in complex domains where the multitude of viewpoints and the inherent conflicting information they bring makes the user querying task confusing). After this stage, reasoning in presence of inconsistency is performed and explained. This reasoning is based on alternative methods such as non-monotonic specially dedicated knowledge, argumentation and belief revision. The latter will help update the knowledge base with information the user might want to bring based on the provided explanations.

Two recent theoretical results should be reminded. First, together with Jerome Fortin and Abdallah Arioua from the GraphIK group we investigated an efficient (with respect to time and space complexity) mapping between inconsistent ontological knowledge bases (expressed in a general rule-based language) and a class of default theories (semi-monotonic, precisely *prerequisite-free closed normal default theories*). We formally proved the equivalence between the *inconsistency-tolerant semantics* in OBDA and inference in Default Logic as well the property of semi-monotonicity for inconsistency-tolerant semantics that can serve as a basis for an *anytime algorithm* for query answering. This means that default logic inspired algorithms for repair and inconsistency tolerant se-
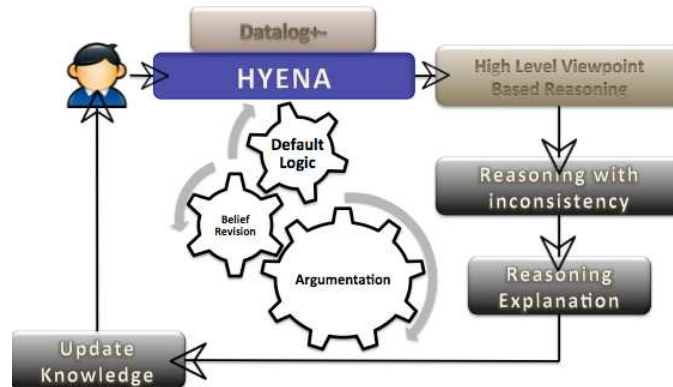
Figure 4.3: Hyena Workflow

mantics are possible. Second, together with Ricardo Rodriguez from University of Buenos Aires we have been studying the link between belief revision and inconsistency tolerant semantics in [24]. We showed how to get first results towards an axiomatic characterisation of certain semantics (see Chapter 2 for their definitions). The axiomatisation was done by covering the semantics using known consolidation operators from belief revision.

4.4 Future Directions

The immediate line of work envisaged for Hyena is developing and analysing properties of query answering explanation algorithms. Eventual optimisations of these algorithms could be envisaged, for example when considering preferences between sets of atoms. Such preferences could be interpreted as priorities over the knowledge base sets imported in Hyena, or by considering an ordering over viewpoints of experts. Then, if the preference relation satisfies certain properties one can show interesting properties of inconsistency tolerant semantics. This last line of research is investigated with Slawek Staworko from University of Lille.

Preferences could also be numerical and the use of fuzzy logics in this setting could bring interesting results. While the use of a fuzzy logic for low expressiveness description logics has already been investigated in EcoBioCap [58, 59] extending these results to the more expressive fragment of logics could bring promising research avenues. Furthermore, the tool developed in EcoBioCap could be extended to reason with expressive rules and thus be incorporated in the Hyena workflow.

Figure 4.4 shows an example of how Cogui, the explanation algorithms and the extension of EcoBioCap tool could function together for representation, reasoning and elicitation of knowledge.
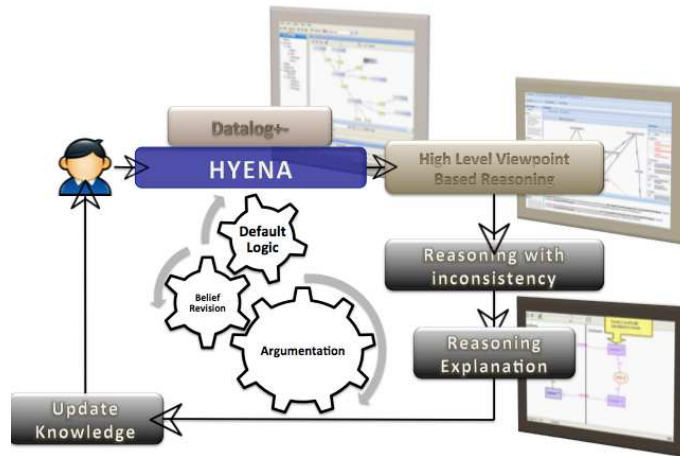
Figure 4.4: Hyena Workflow Enriched with Tools

As previously mentioned Hyena is a long term project which interacts with other projects developed in the GraphIK and Axe 5 at IATE groups. Hyena also behaves as complement to the capitalisation tools developed in Axe 5. Therefore in the future, special attention should be given to the knowledge representation and reasoning formalism and their interoperability properties. This should allow for interoperability with other Semantic Web tools as well as tools developed in the group.

Let me conclude by re-stating the observation made in the introduction of this thesis: research is an online process that, inevitably, contains a part of unknown and surprise. While the research project I presented is something that I am fully dedicated and committed to, in ten years changes in Artificial Intelligence tools and paradigms might change its methodology. However, I am confident in saying that reasoning in presence of inconsistency and elicitation are and will be at the core of knowledge representation and reasoning challenges in the years to come.

Chapter 5

Papers on Graph Reasoning

Visual Reasoning with Graph-based Mechanisms: the Good, the Better and the Best

Michel Chein, Marie-Laure Mugnier, Madalina Croitoru

LIRMM, 161 rue ADA, F34392 Montpellier Cedex 5, Montpellier, France
E-mail: {chein,mugnier,croitoru}@lirmm.fr

Abstract

This paper presents a graph-based knowledge representation and reasoning language. This language benefits from an important syntactic operation, which is called a graph homomorphism. This operation is sound and complete with respect to logical deduction. Hence, it is possible to do logical reasoning without using the language of logic but only graphical, thus visual, notions. This paper presents the main knowledge constructs of this language, elementary graph-based reasoning mechanisms, as well as the graph homomorphism, which encompasses all these elementary transformations in one global step. We put our work in context by presenting a concrete semantic annotation application example.

1 Introduction

Knowledge Representation and Reasoning. Knowledge representation and reasoning (KR) has long been recognized as a central issue in Artificial Intelligence (AI). Very generally speaking, the problem is how to encode human knowledge and reasoning by symbols that can be processed by a computer to obtain intelligent behavior. In general, Artificial Intelligence is concerned with qualitative, rather than quantitative, problem solving. As a basic building block of AI applications, a knowledge representation formalism should reflect this idea by supporting reasoning instead of calculation. This is done by organizing the knowledge into an easily processable form, classifying information by its properties, preprocessing the information, etc. The subfield of AI precisely called KR especially studies computational models, languages and systems able to represent knowledge in an explicit way and to do inferences, or reasoning, on this knowledge.

Even if there have been heated debates about KR in the past, in particular with respect to the role of logic in KR, there is nowadays an agreement on some important properties of a KR language, namely: to be logically founded, to allow for a structured representation of knowledge, to have good computational properties, and to allow users to have a maximal understanding and control over each step of the knowledge base cycle.

The first point concerns the fact that the language is translatable into a logic: the expressions should be translatable into sentences of a given logic, and inferences in that language should correspond to deduction in this logic. In other words, the inference mechanisms should be logically *sound* (i.e., every piece of knowledge inferred is deducible in the target logical fragment) and *complete* (any piece of knowledge that cannot be inferred cannot be deduced either in the target logical fragment). This allows to give a precise semantics to expressions and inferences, and to compare different languages from an expressiveness viewpoint.

Knowledge structuring means that semantically related pieces of knowledge should be grouped together, and that different kinds of knowledge (such as facts, rules, constraints, goals etc.) should

be represented by different knowledge constructs. This can be motivated by model adequacy (i.e., its conformity to the modeling of the application domain) and by computational efficiency concerns. A large part of KR research can be seen as the search of good tradeoffs between the expressivity of a representation formalism and the computational complexity of the associated reasoning.

Knowledge-based Systems. Knowledge-based systems (KBS) are systems, built upon models, able to represent knowledge in an explicit way and do reasoning with. These systems include a *knowledge base* (KB), composed of different kinds of knowledge, and a reasoning engine. The reasoning engine processes knowledge in the KB to answer a question or to reach a certain goal (for instance to check the consistency of the KB or to build a sequence of actions in order to achieve a goal). The cornerstone of the knowledge base is the *ontology*. From an epistemological viewpoint, an ontology answers the question "what kinds of things exist in the application domain?" We consider here computational ontologies, which provide a symbolic representation of classes of objects, called *concepts*, as well as the possible relationships between objects, called *relations* or *roles*. All other pieces of knowledge in the KB are expressed by structures built with the ontology terms (concepts and relations).

Classically, in the building of a knowledge-based system, the first phase consists of knowledge *elicitation*: obtaining a system specification expressed in a language understandable by human beings, so that it can be checked by domain experts. This *mediating representation* does not need to be precise, consistent, or complete. Its role is to allow the experts to freely explicit knowledge and to communicate with others. The second phase consists of translating (the largest possible part of) this informal representation into a formal representation, expressed in a formal language provided with reasoning mechanisms, so that it can be checked whether this representation is consistent and complete with respect to the task to be solved. A major difficulty of this way of doing is ensuring the faithfulness of the formal representation with respect to the mediating representation. Checking faithfulness needs a validation and a revision cycle which is usually long: while the expert understands the mediating representation, this representation cannot be checked, and he/she does not understand the formal representation provided with the reasoning tools. One of the solutions proposed in (Shaw and Gaines (1995); Bos et al. (1997)) is to create mediating representations that are formal: expressing the representation in a language which is both understandable by human beings and formal and in this way ensuring expert simulation at an early modeling stage. This is precisely the qualities claimed by our graph-based language detailed in Section 2.

However, for a KBS to be really used in practice, an essential point is that the user understands and controls the *whole process* whose main steps are not only building a knowledge base and running the KBS (as previously discussed) but also obtaining results. By user, we either mean an end-user, who may be a knowledge engineer, who builds the KB, or an expert of the application domain, who uses the implemented representation to check its conformity with her own representation of the domain, or a user for whom the system has been built and who wants to solve real problems. It should be easy for this user not only to enter different pieces of knowledge and to understand their meaning but also to understand the results of the system and how the system computed these results. The last point, namely the ability of understanding why the system gives a certain answer, is especially important since the user computing expertise may vary. Furthermore, for any domain and level of expertise, explaining to the user each step that makes up the logical inference, generally remains a difficult process.

2

A Graph-based Language. In this paper, by "graph" we understand the classical mathematical notion in *graph theory*, i.e., a structure that consists of a set of nodes (also called vertices) and a set of edges that establish relationships (connections) between nodes. Please note that, regrettably, "graph" in elementary mathematics also refers to a function graph, i.e., a plot.

In the proposed graph-based approach, where all pieces of knowledge are represented by labeled graphs, *the same language* is used at all levels and for all KBS functionalities. The benefits of using graphs for representing knowledge at all levels of the KBS stem from the following:

- Firstly, graphs are simple mathematical objects (they only use elementary naive set theory notions such as elements, sets and relations) which have graphical representations (a set of points and lines connecting some pairs of points) and thus can be visualized.
- Secondly, there is a rich collection of efficient algorithms for processing graphs, thus graphs can be used as effective computational objects (they are widely used, for instance, in Operational Research).
- Thirdly, graphs can be equipped with a logical semantics: the graph-based mechanisms they are provided with are sound and complete with respect to deduction in the assigned logic.

Furthermore, graph-based mechanisms can be explained to the user because they can be easily visualized on the graphs themselves, either as a sequence of very simple operations or as a "global" operation. This point will be further detailed in Section 3.

Semantic Networks. In general, KR languages rely on a purely textual representation with strict syntactic and semantic rules. Domain concepts, their properties, relations and restrictions, are all represented by words and sentences of the representation language. Textual communication is often supplemented with visual properties (such as character types, styles, structure, or layout), but a knowledge representation language can be regarded as visual only if it is based on a pictorial expression. The human short term memory is limited, but visual organizations enable brain sensory information storage and ability to break down complex structures into more easily manageable chunks. Due to their visual qualities, semantic networks, which were originally developed as cognitive models, have been used for knowledge representation since the early days of Artificial Intelligence, especially in natural language processing.

The term *semantic network* encompasses an entire family of graph-based visual representations. Since Nude (Richens (1956)) and the semantic network T (Masterman (1962)), which concern natural language processing, many semantic networks systems have been introduced (cf. Lehman (1992) for a collection of papers concerning various families of network-based structures). They all share the basic idea of representing domain knowledge using a graph, but there are differences concerning notation, as well as rules or inferences supported by the language. In semantic networks, the *diagrammatical reasoning* is mainly based on path construction in the network. We can distinguish two major families of languages born in the eighties. Let us start with the KL-ONE family. In addition to the large number of systems implementing KL-ONE variants (Woods and Schmolze (1992)), KL-ONE is considered as the ancestor of Description Logics (DLs) (Baader et al. (2003)), which are nowadays the most prominent KR languages dedicated to reasoning on ontologies. However, Description Logics have lost their graphical origins. Secondly, Conceptual Graphs. They were introduced by Sowa (cf. Sowa (1976, 1984)) as a diagrammatic system of logic with the purpose "to express meaning in a form that is logically precise, humanly readable, and computationally tractable" (cf. Sowa (1984)). Throughout the remainder of this paper we use the term "Conceptual Graphs" to denote the family of formalisms rooted in Sowa's work and then enriched and further developed with a graph-based approach (cf. Chein and Mugnier (2009)).

Paper Organization. The sequel of this paper is structured as follows. Section 2 presents the main syntactic constructs of the language. Section 3 presents the elementary graph-based reasoning mechanisms and explains how labeled graph homomorphism encompasses all elementary operations in one global operation. Please note that, in this paper, we do not enter into precise definitions and notations but rather rely on visual intuition (see Chein and Mugnier (2009) for more details and technical developments). An application scenario is presented in Section 4. We conclude the paper with Section 5 that presents related work in the domain and lays down future work directions. All figures depict graphs drawn using the conceptual graph editor Cogui [1]. Please note that Cogui is also fully integrated with the conceptual graph engine Cogitant [2] to perform reasoning on the above mentioned graphs.

2 Using graphs for representation: the good

In our approach, all kinds of knowledge are encoded as graphs and thus can be visualized in a natural way:

- The vocabulary, which can be seen as a basic ontology, is composed of hierarchies of concepts and relations. These hierarchies can be visualized by their Hasse diagram, the usual way of drawing a partial order (see Figures 1 and 2).
- All other kinds of knowledge are based on the representation of entities and their relationships. This representation is encoded by a labeled graph, with two kinds of nodes, respectively corresponding to entities and relations. Edges link entity nodes to relation nodes. These nodes are labeled by elements in the vocabulary (see Figure 3).

These graphs have a semantics in first-order logic (FOL), i.e., a knowledge base can be translated into a set of first-order logical formulas. Reasoning tasks operate directly on the knowledge defined by the user and not on their translation into logical formulas. Stated in an other way, the logical semantics is only used to formally ground the graph model, i.e., representation and reasoning mechanisms. This makes it possible to explain reasoning to the end-user because it can be visualized in a natural way on the pieces of knowledge he/she is familiar with. Would a logical prover be used on the logical translation of these pieces of knowledge to compute reasoning, reasoning would become a black box for the user and could not be explained.

2.1 Conceptual vocabulary

The vocabulary is composed of two partially ordered sets: a set of *concepts* and a set of *relations* of any arity (the arity is the number of arguments of the relation). The partial order represents a specialization relation: $t' \leq t$ is read as "t' is a specialization of t". If t and t' are concepts, $t' \leq t$ means that "every instance of the concept t' is also an instance of the concept t". If t and t' are relations, then these relations have the same arity, say k, and $t' \leq t$ means that "if t' holds between k entities, then t also holds between these k entities"). Figures 1 and 2 shows parts of these hierarchies visualized by their Hasse diagram ($t' \leq t$ if there is a path from t' up to t). For instance, the concept TeddyBear is a specialization of the concept Object, because of the path (TeddyBear, Toy, Object); the relation siblingOf is a specialization of the relation $link_2$ (which stands for any binary relation) because of the path (siblingOf, relativeOf, $link_2$). Note that a hierarchy is not necessarily a tree: for instance, there are two paths from Woman to Person, namely (Woman, Female, Person) and (Woman, Adult, Person).

Names of specific individuals can also be included in the vocabulary. The vocabulary can be further enriched by *signatures* for relations indicating the maximal concept that can be assigned to each of the relation arguments (e.g. the first argument of the relation motherOf is of maximal type Woman and its second argument is of maximal type Person). It can also contain statements of *disjointness* between concepts (e.g. the two types Object and Person are disjoint).
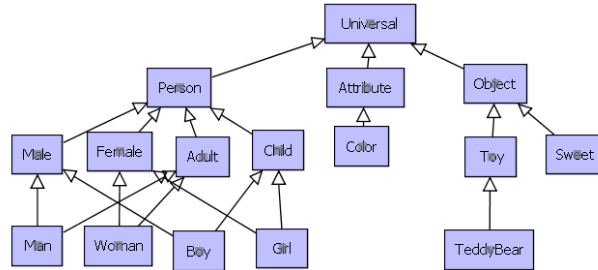
[1] http://www.lirmm.fr/cogui/
[2] http://cogitant.sourceforge.net/

Figure 1 Basic Ontology
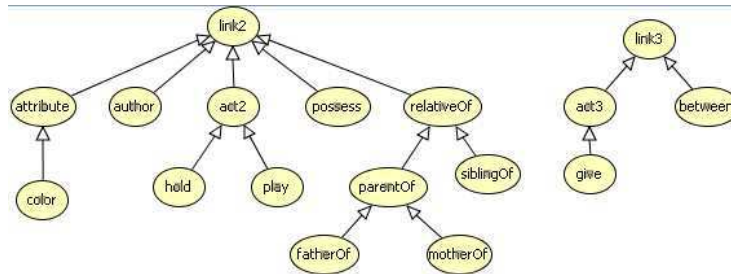*Concepts for the childhood domain*

Figure 2 Basic Ontology
Relations for the childhood domain

2.2 Basic graphs

A basic conceptual graph (BG) is a bipartite graph: one class of nodes, called *concept* nodes, represents entities and the other, called *relation* nodes, represents relationships between these entities or properties of them. E.g., the BG in Figure 3 graphically represents the following situation: "The boy John is a sibling of the girl Eva. John is giving a sweet to Eva, who is holding a red teddy bear belonging to John."

A concept node is labeled by a couple $t : m$ where t is a concept (and more generally, a list of concepts) called the type of the node, and m is called the marker of this node: this marker is either the generic marker, denoted by $*$, if the node refers to an unspecified entity, otherwise this marker is a specific individual name. E.g., in Figure 3 the node [Sweet:*] refers to "a" sweet, while the node [Boy:John] refers to "the" boy John. A relation node is labeled by a relation r called its type, and, if k is the arity of r, this node is incidental to k totally ordered edges. E.g., in Figure 3 the relation node of ternary type *give* has three incidental edges and the order on these edges allows to distinguish between the *agent* of the gift act [Boy:John], its *recipient* [Girl:Eva] and its *object* [Sweet:*]. Classically, concept nodes are drawn as rectangles and relation nodes as ovals and the order on edges incidental to a *k-ary* relation node are numbered from 1 to k.

BGs are used to represent assertions called *facts*. They are also building blocks for more complex kinds of knowledge, as outlined in the next section.
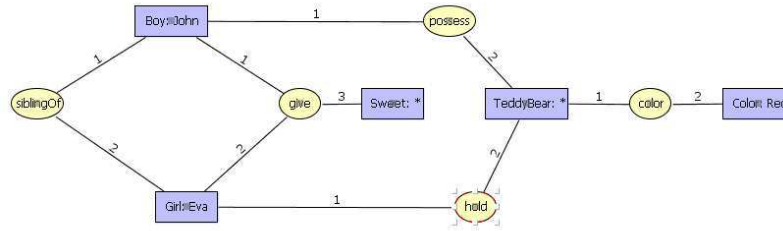
Figure 3 A situation described by a basic graph (H)

2.3 More complex graphs

In the following we present two examples of basic conceptual graph extensions: *nested graphs*, which allow to structure facts by level of detail; and *inference rules*, which enrich the basic ontology with general knowledge about a domain.

Nested graphs. In a nested graph, a concept node may itself contain a (nested) graph, whose role is to further describe the entity represented by the node. This allows to distinguish between internal and external information about an object, to represent zooming into an object or to contextualize the description of an object. For instance, let us consider the nested graph in Figure 4. At the outermost level, this graph says that "Eva is giving a picture that she did to John" (note however that time is not represented here) and the graph nested in the node referring to the picture further describes this picture: "this picture shows a boy holding a teddy bear". The dotted line is a *coreference link*: it links two nodes that refer to the same entity (in this example the boy called John). The information that the picture has been done by Eva can be seen as an *external* information about the picture, while the detail of what is in the picture can be seen as an *internal* information, that can be obtained by zooming into the picture. It can also be said that the piece of information nested in a node is relevant within the context represented by this node (here, John is holding a teddy bear in the context of the picture, but it may not be true in the outermost context).
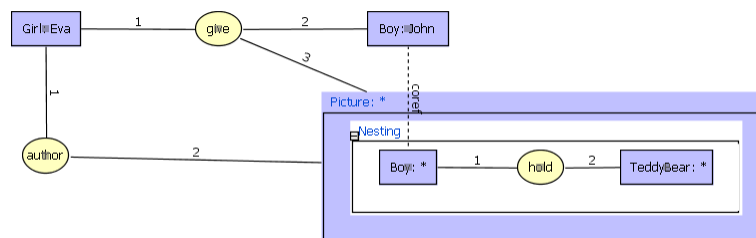

Figure 4 A nested graph

6

Rules. A rule expresses implicit knowledge of the form "if *hypothesis* then *conclusion*", where hypothesis and conclusion are both basic graphs. This knowledge can be made explicit by applying the rule to a specific fact: intuitively, when the hypothesis graph is found in a fact, then the conclusion graph can be added to this fact (see Sect. 3.4 for more details). There is a one to one correspondence between some concept nodes in the hypothesis with concept nodes in the conclusion. Two nodes in correspondence refer to the same entity. These nodes are said to be *connection nodes*. For instance, Figures 5 and 6 present two rules, with the hypothesis on the left hand side and the conclusion on the right, separated by a vertical line. In Figure 5, rule R_1 says that "if a person is a sibling of a person, then the inverse relation holds between these persons". More formally: "for all persons x and y, if x is a sibling of y then y is a sibling of x"(all concept nodes are connection nodes). In Figure 6, rule R_2 says that "for all persons x and y, if x is a sibling of y, then they have a common parent, i.e., there is a person who is a parent of x and of y" (the node representing this person is not a connection node, since it is not in correspondence with a node in the hypothesis). Let us add that rules can also be defined as pairs of nested graphs instead of basic graphs.

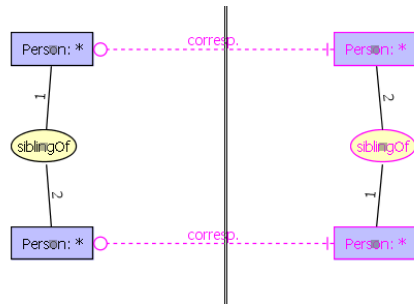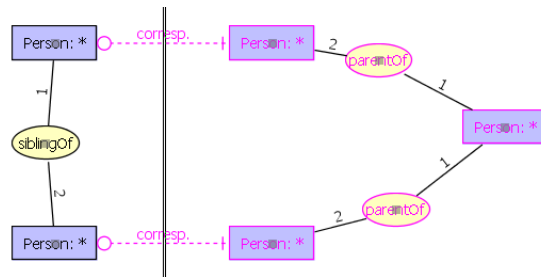**Figure 5** A rule (R_1)

Figure 6 Another rule (R_2)

All these graphical objects, i.e., the vocabulary as well as basic graphs, nested graphs and rules, are provided with a semantics in first-order logic. This semantics specifies the meaning of knowledge constructs and allows to show the correctness of the associated graph mechanisms with respect to logical deduction (see Sect. 3.3).

3   Using graphs for reasoning: the better

Different kinds of reasoning concerning BGs can be graphically defined, e.g., applying inference rules or contextual reasoning. These reasonings are based on a *subsumption* relation between conceptual graphs. This section is devoted to the presentation of this fundamental reasoning notion in the simplest case, i.e., BGs.

Let G and H be two BGs over the same vocabulary. Intuitively, G subsumes H if the fact—or the information—represented by H entails the fact represented by G, or in other words, if all information contained in G is also contained in H.

A query-answering mechanism using BGs and subsumption can be defined as follows. Let us consider a knowledge base B composed of a set of BGs, representing some assertions about a modeled world, e.g., the fact in Figure 3. Elements in B answering a query Q are intuitively defined as the elements that entail Q, or equivalently, elements that are specializations of Q, or also, elements that are subsumed by Q. Let us consider for instance the query in Figure 7. This query is easily visualized as a graph, but is more complex to express textually: it asks for a situation where a boy and a girl, who is one of the boy's relatives, are each in relation with a red toy. We will see hereafter why and how the fact in Figure 3 answers this query.
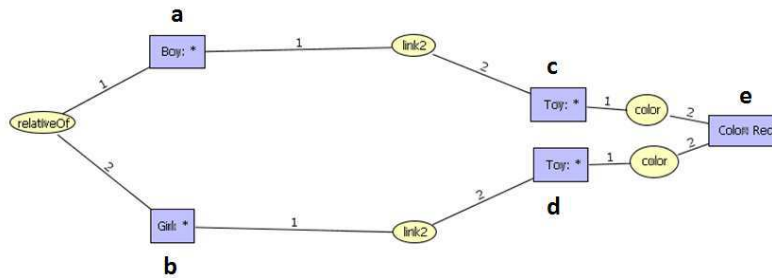
Figure 7 A query described by a basic graph (G)

Relationships with logics is mentioned at the end of this section and we will see that the subsumption relation exactly corresponds to deduction in a fragment of first-order logic. The subsumption relation can be defined either by a sequence of elementary operations or by the classical homomorphism notion applied to BGs. Both are very easily visualizable and they are defined below by means of drawings.

3.1   Generalization and Specialization Operations

There are five elementary generalization operations and five inverse operations, called elementary specialization operations. The terms *generalization* and *specialization* are used here with the following intuitive meaning: let I be a piece of information; whenever a piece of information is added to I the obtained information is a specialization of I (it contains more specific knowledge) and, conversely, deleting a piece of information from I yields a final piece of information that is more general than I (it contains less precise knowledge).

3.1.1   Elementary Generalization Operations for BGs

Any generalization operation is a "unary" operation, i.e., it transforms a BG into another BG. It can be pictured by a drawing representing the transition from the input BG to the output BG. Generalization has to be taken in a broad sense, i.e., the BG obtained may contain a strictly more general information than the initial BG or the same information.

The elementary generalization operations can be graphically defined as follows.

- **Substract** (Figure 8). The *Substract* operation consists of deleting some connected components of a BG. The BG obtained is clearly more general than (or equivalent to) the original BG, since a piece of information is deleted.
- **Detach** (Figure 9). The *Detach* operation consists of splitting a concept node c into two concept nodes c_1 and c_2, the edges incident to c being shared between c_1 and c_2. For instance, let us consider the detachment of the generic concept node c in Figure 9. In the initial BG it is said that "there is a boy who possesses something and is giving something to a person", and in the resulting BG it is said that "there is a boy who possesses something and there is a boy giving something to a person". In the resulting BG, the two boys may be different, while in the initial BG it is necessarily the same boy. Therefore, the final situation is more general than the initial situation.
- **Increase** (Figure 10). The *Increase* operation consists of increasing the label of a concept or relation node. In the case of a concept node, it means that one can increase its type, e.g., replacing "the girl Eva" by "the person Eva" and/or replace an individual marker by the generic marker, e.g., transforming "the person Eva" into "a person". Formally, the generic marker is considered as greater than all individual markers, which are pairwise non comparable. Thus, increasing a concept node label consists of increasing its type and/or its marker. Clearly, "a person" is more general (in an intuitive sense) than "the girl Eva". In the same way, replacing a relation type by a greater type is also clearly a generalization operation, e.g., "John is a relative of Eva" is more general than "John is a sibling of Eva." Let us remark that checking if a type t is greater than another type t' is the most frequent operation in hierarchies. It is a basic operation (t is greater than t' if and only if there is a path from t' to t) for which efficient algorithms have been developed (cf. Chein and Mugnier (2009)). In Figure 11, the bold paths contain all types greater than the type Boy.
- **Relation duplicate** (Figure 12). The *Relation duplicate* operation consists of duplicating a relation node, i.e., adding a new relation node r' having the same type and the same list of arguments as a relation node r. r and r' are said to be *twin* relation nodes.
- **Copy** (Figure 13). The *Copy* operation consists of duplicating a whole BG, i.e., adding to it an isomorphic and disjoint copy of it.
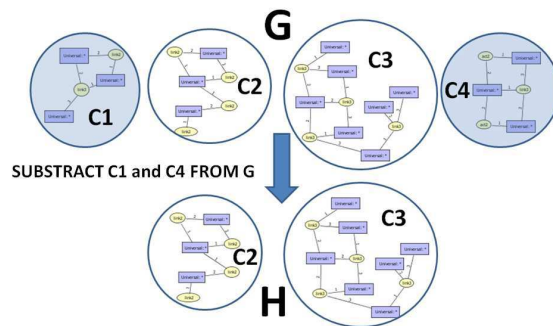
Figure 8 Substract
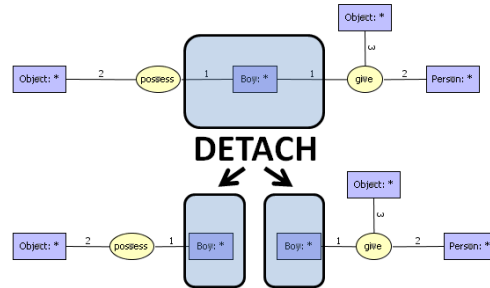
Figure 9 Detach
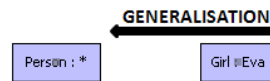
Figure 10 Increase
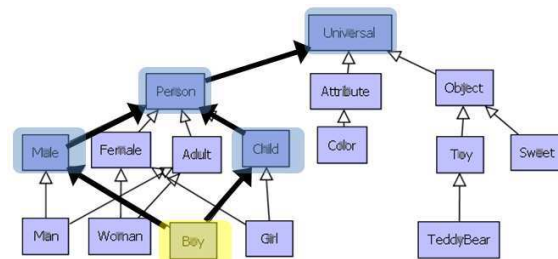
Figure 11 Types greater than or equal to the type Boy

In the last two operations the structure of the obtained BG contains the structure of the initial BG (one adds something), thus, at first glance, one can think that they are specialization operations. Nevertheless, as these operations duplicate *already existing information*, the BG obtained is semantically equivalent to the initial one, thus they are also generalization operations (in a broad sense).

We can now precisely define what *"G is a generalization of H"* means: G is a generalization of H if there is a sequence of elementary generalization operations leading from H to G, i.e., there is a sequence of BGs $H_0 = (H), H_1, ..., H_n = (G)$, such that for all $i = 1, \ldots, n$, H_i is obtained from H_{i-1} by an elementary generalization operation.

Figures 14, 15, 16, 17, 18, present some of the graphs occurring in a generalization sequence from H, the BG in Figure 3 to G, the BG in Figure 7. H_1 is obtained from H by splitting both nodes [Girl:Eva] and [Boy: John] (two *Detach* operations). H_2 is obtained from H_1 by deleting the connected component containing the *give* relation node (*Substract* operation). H_3 is obtained from H_2 by duplicating the relation node *color* (*Relation duplicate* operation). H_4 is obtained from H_3 by a *Detach* operation on the node [TeddyBear:*]. G is obtained from H_4 by a sequence of seven *Increase* operations: the relation $siblingOf$ is replaced by $relativeOf$, the
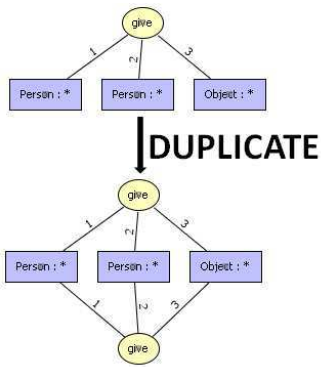
Figure 12 Duplicate
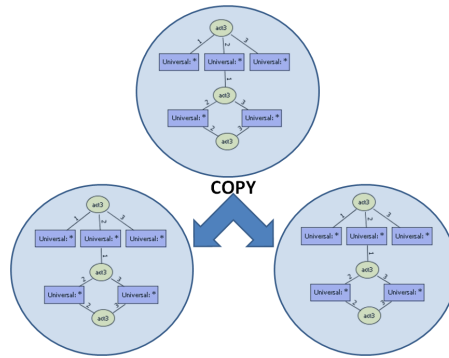
Figure 13 Copy

relations *possess* and *hold* are replaced by *link*2; the individual concept nodes [Boy:John] and [Girl:Eva] are made generic, and the teddy bears become toys.
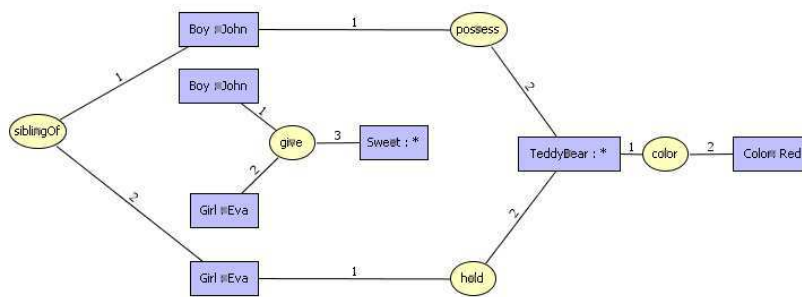
Figure 14 Generalization from H to G - step I (H_1)
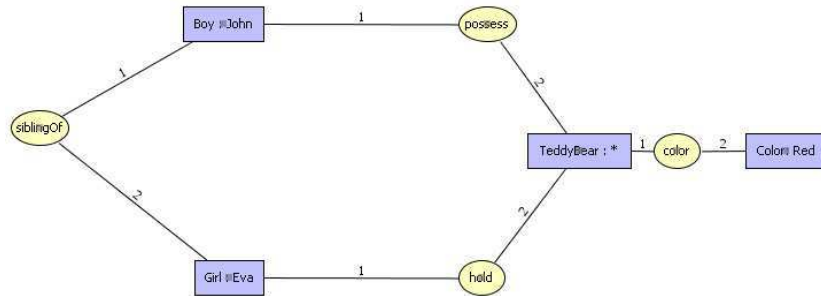
Figure 15 Generalization from H to G - step II (H_2)
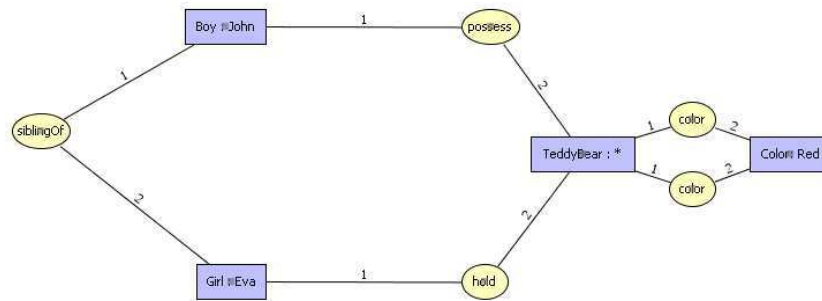
Figure 16 Generalization from H to G - step III (H_3)
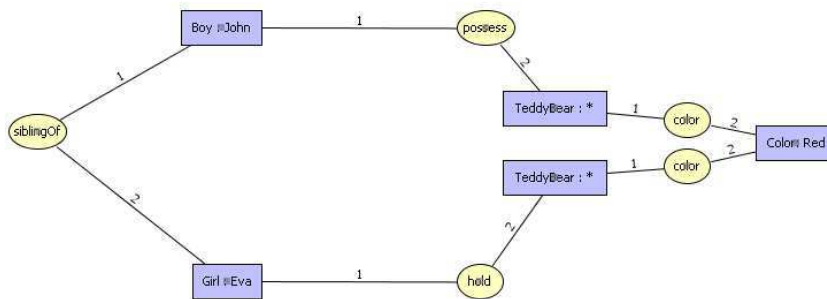
Figure 17 Generalization from H to G - step IV (H_4)
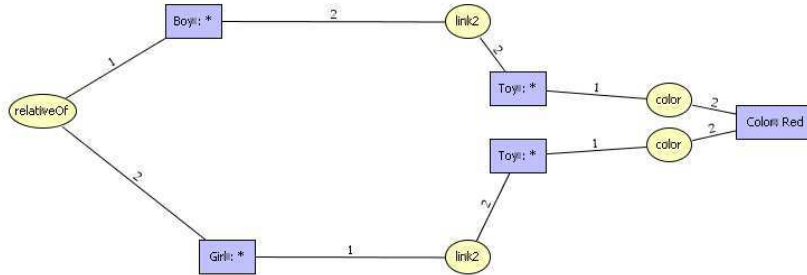
Figure 18 Generalization from H to G - step V (G)

3.1.2 Elementary Specialization Operations

We have seen in the previous section that the elementary generalization operations are indeed generalization operations in an intuitive manner, they are easy to draw and to understand, and they allow for a precise definition of the subsumption relation between BGs. Nevertheless, let us consider again the query-answering problem previously stated using generalization. Let Q be a BG query and B a set of BG facts. Answering Q consists of looking for subgraphs of BGs in B that are subsumed by Q, i.e., such that Q generalizes them. It seems more intuitive and it is more efficient to start from the query Q and to look for specializations of Q in B. This can be done by defining elementary specialization operations which are the inverse of the elementary generalization operations defined previously.

- **Disjoint sum** (Figure 19). Given two disjoint BGs, their *Disjoint sum* is the BG obtained by juxtaposing two copies of these BGs. This operation is the inverse of the *Substract* operation.
- **Join** (Figure 20). Given a BG, joining two concept nodes with the same label in this BG consists of merging them. This operation is the inverse of the *Detach* operation.
- **Restrict** (Figure 21). *Restrict* consists of decreasing the label of a concept or relation node. This operation is the inverse of the *Increase* operation.
- **Relation simplify**(Figure 22). This operation consists of deleting a twin relation node. This operation is the inverse of the *Relation duplicate* operation.
- **Copy**. This operation has already been defined as a generalization operation.
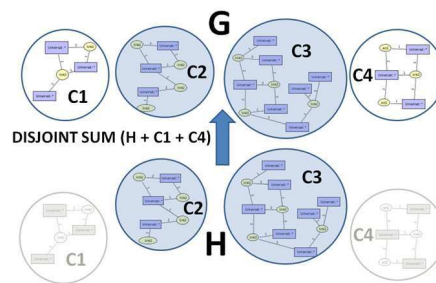
Figure 19 Disjoint sum
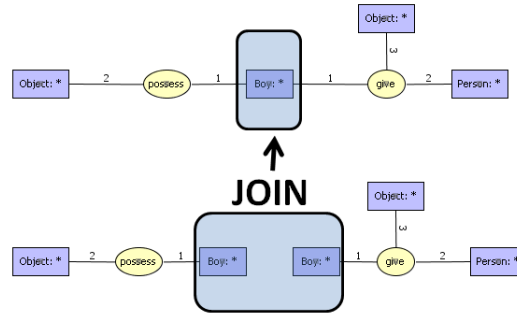
Figure 20 Join

Figure 21 Restrict
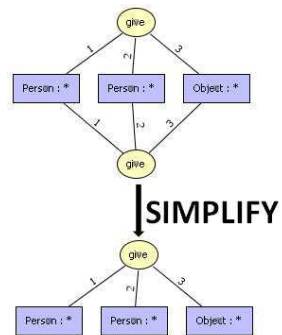
Figure 22 Simplify

H is a *specialization* of G if H can be obtained from G by a sequence of elementary specialization operations. In reading Figures 14, 15, 16, 17, 18 backwards, one obtains a specialization sequence from G to H. This sequence begins with a set of *Restrict* operations (leading from G to H_4). Then, H_3 is obtained from H_4 by a *Join* operation. H_2 is obtained from H_3 by *Relation Simplify*. H_1 is obtained from H_2 by a *Disjoint sum* operation, and H is obtained from H_1 by two *Join* operations.

The following property is straightforward to check:

G is a generalization of H if and only if H is a specialization of G.

14

3.2   Homomorphism

We have seen specialization operations, which are more convenient than generalization operations when considering the query-answering problem. Now, the problem is how to *find* a sequence of specialization operations from a BG to another BG.

In this section, we introduce the homomorphism notion between BGs. A homomorphism from G to H is a mapping from the node set of G to the node set of H, which preserves the adjacency between nodes of G and can decrease the node labels. If there is a homomorphism (say π) from G to H, we say that G *maps* to H (by π).

Let us consider again the query G in Figure 7 and the fact H in Figure 3. There is a homomorphism from G to H which is pictured by the dashed lines in Figure 23. The concept node $a = $ [Boy:*] in G is mapped to [Boy : John] in H, the concept node $b = $ [Girl:*] in G is mapped to [Girl : Eva] in H, both concept nodes $c = $ [Toy:*] and $d = $ [Toy:*] in G are mapped to the same node [TeddyBear:*] in H, the concept node $e = $ [Color:Red] in G is mapped to [Color:Red] in H, the relation node ($relativeOf$) is mapped to the relation node ($siblingOf$), the relation node ($link2$) from a to c is mapped to the relation node ($possess$) and the other relation node ($link2$) from b to d is mapped to the relation node ($hold$); finally, both relation nodes ($color$) are mapped to the same node ($color$).
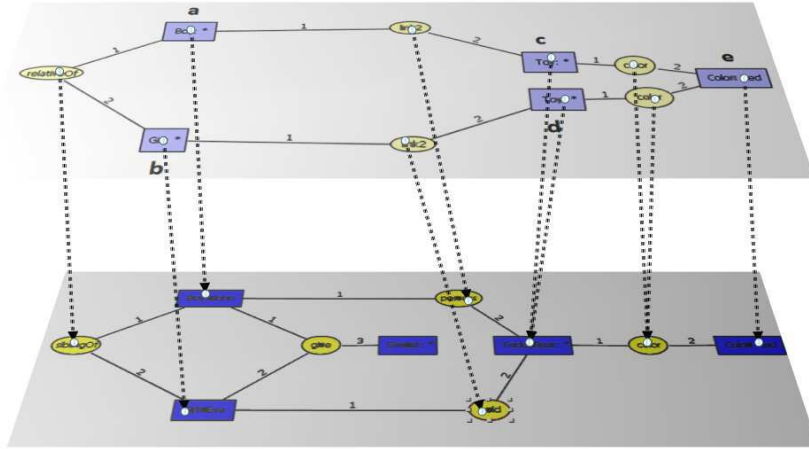
Figure 23   A homomorphism from G to H

A BG homomorphism can be more precisely defined as follows. A *homomorphism* π from G to H is a mapping from the concept node set of G to the concept node set of H and from the relation node set of G to the relation node set of H, which preserves edges and may decrease concept and relation labels, that is:

- for any edge labeled i between nodes c and r in G, there is an edge labeled i between nodes $\pi(c)$ and $\pi(r)$ in H; in other words, if a relation r has neighbors $c_1 \ldots c_k$ (in this order) then its image $\pi(r)$ has neighbors $\pi(c_1) \ldots \pi(c_k)$ (in this order).
- for any (concept or relation) node x in G, the label of its image $\pi(x)$ in H is less than or equal to the label of x.

Let us consider again the homomorphism in Figure 23. Figure 24 highlights the subgraph of H induced by the nodes which are images of nodes in G. This subgraph is called the "image graph" of G by this homomorphism.
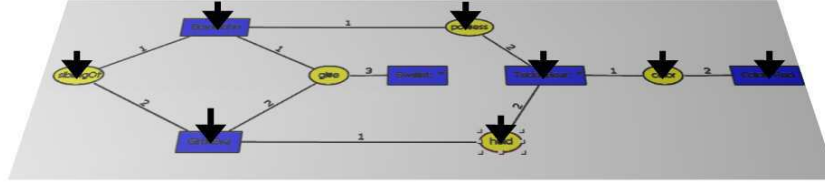
Figure 24 Image of G by the homomorphism in Figure 23

The following theorem holds. Given two BGs G and H, the three following propositions are equivalent:

1. G is a generalization of H
2. H is a specialization of G
3. There is a homomorphism from G to H

Even though it is easy to visualize a homomorphism and to check if a mapping between two BGs is a homomorphism (cf. Figure 23), this global graph matching can be replaced, if needed, for more detailed explanation purposes, by a sequence of elementary operations (cf. Figures 14 to 18). Assume for instance that the answer to the query G is visualized as in Figure 24 and the user wants to understand why the *image graph* of G is indeed a specialization of G. The homomorphism can be decomposed into a sequence of elementary specialization operations, starting from G, as follows:

- Firstly, a sequence of Restrict showing how the label of each node of G is specialized (cf. the transformation from G to H_4);
- Secondly, a sequence of Join showing which concept nodes are merged into the same node (these are the nodes in G with the same image by the homomorphism, cf. the transformation from H_4 to H_3);
- Thirdly, a sequence of Relation simplify removing relation nodes that have become redundant (cf. the transformation from H_3 to H_2).

After these three steps, the image graph of G is obtained, which is sufficient to show that the fact contains a specialization of G. To build the fact graph itself from the image graph of G, one would need a disjoint sum (cf. the transformation from H_2 to H_1) and some joins (cf. the transformation from H_1 to H).

3.3   Logical correctness

Until now, we have introduced the reasoning operations (generalization, specialization and subsumption) using simple graphical operations. It remains to relate these notions to usual reasoning notions, i.e., essentially, to relate BG subsumption to logical deduction. This can be done using the logical semantics of the knowledge constructs. This semantics is defined by a mapping from graphical objects to logical formulas. It is classically denoted by Φ in conceptual graphs (Sowa (1984)). The fundamental theorem states that given two BGs G and H built on a vocabulary $\mathcal{V}$, there is a homomorphism from G to H if and only if $\Phi(G)$ is a semantic consequence of $\Phi(H)$ and $\Phi(\mathcal{V})$ (this is a soundness and completeness theorem of BG homomorphism w.r.t. FOL entailment, cf. Chein and Mugnier (1992)). Let us point out that BGs are in fact equivalent to the positive, conjunctive and existential fragment of FOL.

Once again, logic is used to give a semantics to BGs, but not to reason with them. As detailed before, one motivation for preferring graph-based reasoning is its visual aspects permitting to understand (logically based) reasoning without doing logic. Another motivation is that BGs have

16

good computational properties. Efficient graph-based deduction algorithms have been built, which are not the translations of logical procedures (see f.i. Chein and Mugnier (2009)).

3.4   Overview of reasoning on more complex pieces of knowledge

Previous generalization and specialization elementary operations, as well as the corresponding homomorphism notion, can be extended to nested graphs, while preserving soundness and completeness with respect to deduction in the associated fragment of FOL.

Let us now consider the graph rules presented in Section 2. A rule R can be applied to a BG H if there is a homomorphism from its hypothesis to H. Applying R to H according to such a homomorphism π consists of "attaching" to H the conclusion of R by merging each connection node in the conclusion with the image by π of the corresponding connection node in the hypothesis. See for instance the graph H in Figure 3 and the rules R_1 and R_2 in Figures 5 and 6. There is a homomorphism from the hypothesis of R_1 to H, thus R_1 can be applied to H, which yields the graph K (Figure 25). Similarly, R_2 can be applied to H, or to K. Let us apply R_2 to K: we obtain L (Figure 25). R_2 can be applied another time with a new homomorphism to L: however, this application would be redundant, since the part to be added is already present in L.
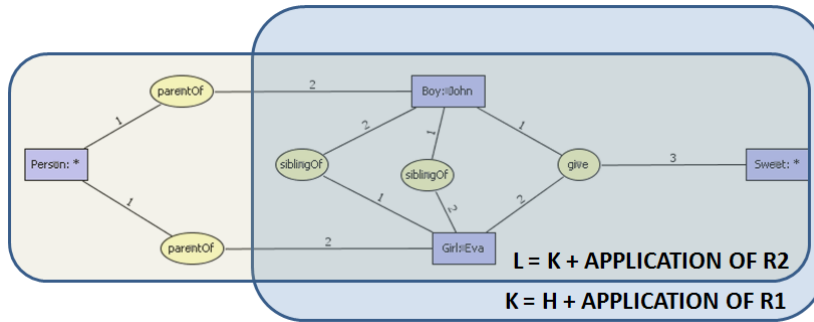
Figure 25   Applying rules

When a knowledge base contains a set of facts and a set of rules, the query mechanism has to take implicit knowledge coded in rules into account. The knowledge base answers a query Q if a BG F' can be derived from the facts using the rules, such that Q maps to F'. Let us consider again the fact H in Figure 3 and the rules R_1 and R_2 in Figures 5, and 6 and let Q be the query in Figure 26 asking if there is someone who is parent of a boy and a girl: there is no homomorphism from Q to H; however, the knowledge base containing H and the rules R_1 and R_2 answers Q: indeed Q maps to L (Figure 25), which is derived from H by applying the rules. This reasoning can be explained by visualizing a homomorphism from Q to L, as well as a sequence of rule applications allowing to add the knowledge involved in this homomorphism (for instance, the application of R_1 is not needed in our example).

Finally, let us point out that this rule application mechanism is sound and complete, i.e., given a KB $\mathcal{K}$ and a BG Q, Q maps to a graph derived from $\mathcal{K}$ if and only if $\Phi(\mathcal{K}) \models \Phi(Q)$, where $\Phi(\mathcal{K})$ is the set of formulas assigned to the vocabulary, the set of facts and the set of rules composing $\mathcal{K}$.
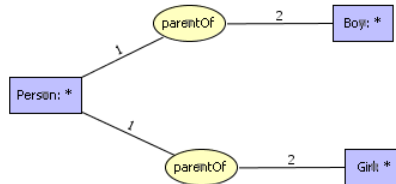
Figure 26 Another query example

4 Using graphs for applications: the best

In this section we present a concrete Artificial Intelligence application using the above described language for image annotation. The choice of the application is motivated twofold. Firstly, this application clearly demonstrate the visual appeal of Conceptual Graphs from a representation depiction faithfulness viewpoint. Secondly, the visual reasoning capabilities allow for all levels of expertise when building, querying the knowledge base and understanding why certain results will be returned.

Within the image annotation process we will distinguish between *resources* (in this case electronic image files) and *metadata*. A metadata is a computational object always associated with a resource. Each resource is identified by an identifier, which is used in the metadata base for referencing the resource.

Metadata can be roughly categorized into two classes: *objective* metadata and *subjective* metadata. Examples of objective metadata include: resource address, authors name and the document size. Subjective metadata aims at representing information generally depending on the author of the metadata. Examples of subjective metadata include: the representation of the content of a resource (indexation of the resource), the representation of a comment, note, or remark etc. In this case an annotation is simply a piece of knowledge associated with a resource. In the following, we will present a Conceptual Graph approach for building and querying a knowledge base aimed at annotating family photos. The knowledge base used to illustrate notions throughout this section has been edited with the tool Cogui.

Such annotation is built from an ontology fundamentally composed of a hierarchy of concepts (or terms) and a hierarchy of relations between concepts. The ontology can also contain representations of other knowledge. Relation signatures indicate the types of relation arguments. Rules represent knowledge of the form "if a piece of information is present (within an annotation) then further information can be added (to this annotation)". Thus, rules can be used to automatically complete an annotation by implicit and general knowledge. Another kind of knowledge with the same syntactic form as rules but a different semantics are *constraints*. Constraints represent knowledge of the form "if a piece of information is present (within an annotation) then other information cannot be present (in the annotation)". Signatures and constraints are used to avoid the construction of absurd annotations. All these kinds of knowledge are represented by labeled graphs.

For clarity purposes the example given in this paper is very simple. This framework have been successfully employed for annotation in the large scale context of the LOGOS Framework 6 European Project (cf. for instance, Lalande et al. (2009)). as well as in other French projects (Moreau et al. (2007), Genest and Chein (2005)).

Let us consider the photograph in Figure 27. A semantic annotation of this image is depicted in Figure 28 where a fact represents a girl, the relative of a child, playing with the same red train that the child is playing with. As explained above, all of the concepts and the relations used in

the facts need to be described and organized in the vocabulary. Figure 29 shows the concept and relation hierarchies purposely built for annotating family images.

Figure 27 Example of photo that needs to be semantically annotated
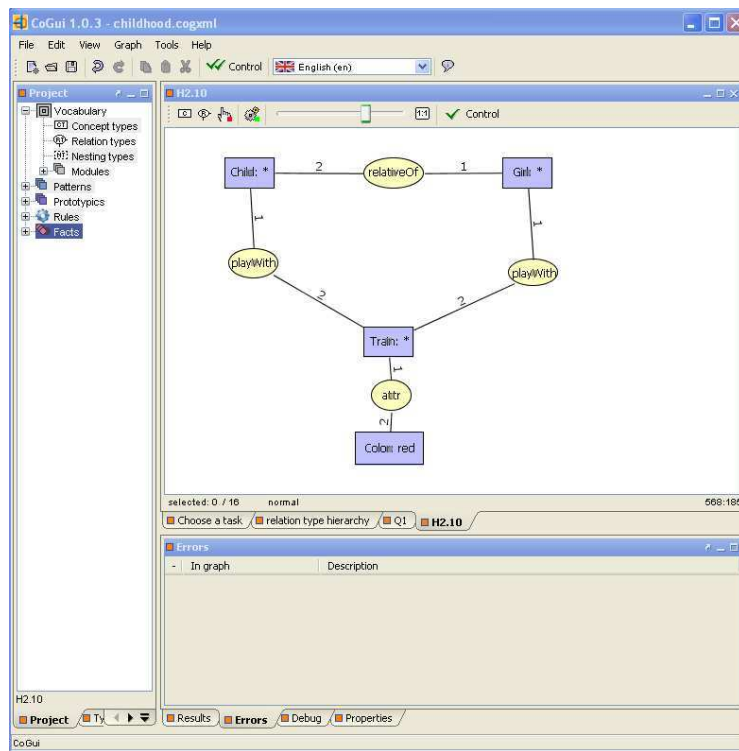
Figure 28 Example of a semantic annotation
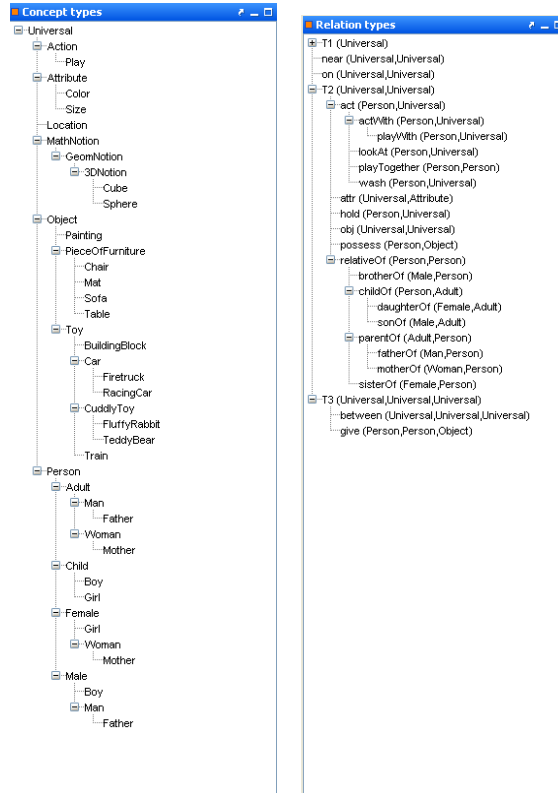
Figure 29 Concept / Relation hierarchy for photo annotation

Please note that the constructs introduced before (such as rules and nested graphs) are also used for annotation. Since certain chunks of knowledge appear often, other kinds of constructs have been introduced specifically for speeding up the annotation process, such as *prototype graphs*. For example, for a child we could always like to have annotated the fact that it is playing with a certain object. Notice this is not a rule, e.g., it may apply in certain situation and not in others. Figure 30 represents how to insert the prototypical graph for a given concept.

As shown before, the user can then look for certain photos that contain a child and its relative, a girl, both acting with a toy. Such query is visually represented in Figure 31. Based on homomorphism (as previously explained) the query can be "mapped" into the fact represented in Figure 28 and this will return the photo attached to it.

Note that we have only presented here the kernel of an information retrieval system, in which the search process is based on graph homomorphism. In order to take the intrinsic vagueness of information retrieval into account, i.e., to search documents imprecisely and incompletely represented in order to answer a vague query, the exact search realized by graph homomorphism is not sufficient. Additional techniques based on graph transformations for doing approximate search and for ranking the answers to a query have been developed (cf. for instance Genest and Chein (2005)).
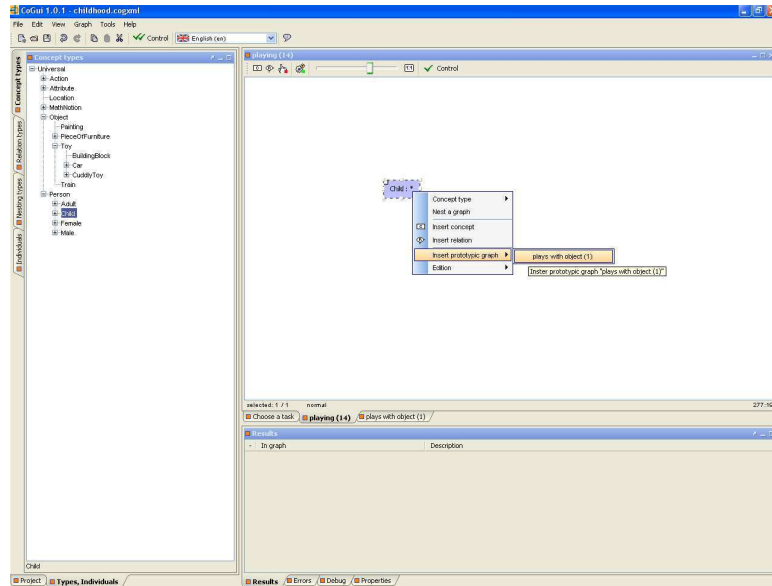
Figure 30 Prototypical graph insert for photo annotation
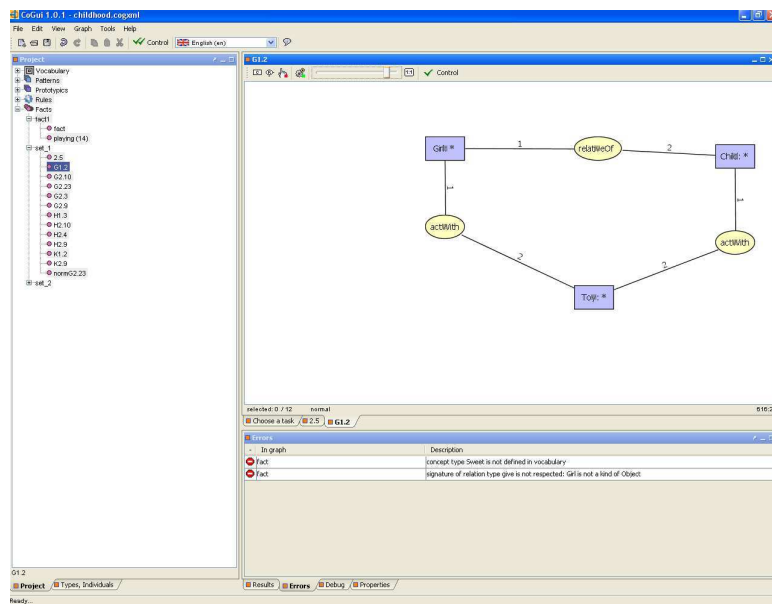
Figure 31 Query graph for photo annotation

5   Related work and conclusion

In this section, we place our language in the landscape of graphical knowledge representation languages in order to enhance its original features. We will only detail each language in the light of its respective distinction with Conceptual Graphs. This choice is motivated by our aim of demonstrating the originality of our proposal in the context of graph-based languages for both knowledge representation and reasoning as opposed of doing a general synthesis of existing languages for KR.

An important criterium distinguishing Conceptual Graphs from other graph-based languages for knowledge modeling is the *reasoning* aspect. Indeed, numerous graphical languages have been proposed for data and knowledge modeling. Prominently amongst them, *UML* (Unified Modeling Language) unifies several previous languages and allows to model a system or a program by diagrams. However, this language does not have a denotational semantics and is not provided with inference mechanisms.

Furthermore, let us focus solely on graphical languages fundamentally dedicated to representing knowledge in the form of relationships between concepts and/or concept instances. The name "map" is often used to describe mediating, thus informal, representations. A *concept map* is a diagram representing relationships between concepts, or more vaguely, ideas, images, words, etc. the aim being to organize informal knowledge (Novak and Canas (2006)). A *cognitive map* is a graphical representation of an influence network between notions. A *topic map* is a diagram representing associations between topics, and is more specifically dedicated to the description of resources for the Semantic Web. If all these languages provide graphical views of knowledge, none of them possesses a formally based reasoning mechanism.

Closer to our proposal, let us cite *RDF* (Resource Description Framework) [3], which is the basic Semantic Web language. RDF has a graph-based representation, it is provided with a formal semantics and an associated entailment notion (Hayes (2004), Horst (2004)), but it does not come with an effective reasoning mechanism, even less with a graph-based mechanism that would operate on the graph representation.

Let us point out that for several of these languages, some form of formally founded visual reasoning based on graph homomorphism has been proposed: Raimbault et al. (2005); Chauvin et al. (2008); Aissaoui et al. (2003); Carloni et al. (2006); Baget (2005).

We have already mentioned in Section 1 the semantic network family, and its successors, description logics, which are logically founded knowledge representation and reasoning languages, which have lost their diagrammatical aspects. In contrast, the Semantic Network Processing System (SNePS) (see Shapiro (1979, 2000)), specially dedicated to the implementation of cognitive agents, remains graph-based. It is provided with three kinds of inference mechanisms: a sound (but not complete) logic-based inference, as well as a path-based inference and a frame-based inference. All three kinds of inferences can be integrated, but there is no formal semantics for this combination.

Conceptual Graphs finally appears to be the only knowledge representation language both logic-based and graph-based, with logically sound and complete graph inference mechanisms (at least for the fragment developed here), thus allowing for visual reasoning. Another important feature of conceptual graphs is that relations can be of any arity (i.e., they can have any number of arguments), which allows for a more natural representation in many cases (for instance, when relations are extracted from data tables, their arity is the number of columns in the table). This latter feature is shared with topic maps, but none of the other languages mentioned above.

The sound and complete graph-based mechanisms for reasoning presented in this paper have been fully implemented and are available as part of the Cogui Editor and the CG reasoning engine Cogitant. While homomorphism proves to be an intuitive mechanism for query answering, more advanced graph-based tools could be envisaged to make this notion even more intuitive (see for

---

[3] http://www.w3.org/TR/REC-rdf-syntax/

example the 3D manipulation of the images in Figure 23). This is not the only possible envisaged manipulation: different layouts, smooth zooming capabilities or colors could also be employed to increase the presentive qualities of our language.

References

G. Aissaoui, D. Genest, and S. Loiseau. Cognitive map of conceptual graphs: A graphical model to help for decision. In A. de Moor, W. Lex, and B. Ganter, editors, *ICCS*, volume 2746 of *Lecture Notes in Computer Science*, pages 337–350. Springer, 2003. ISBN 3-540-40576-3.

F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.

J. F. Baget. RDF Entailment as a Graph Homomorphism. In *Proc. of ISWC'05*, 2005.

C. Bos, B. Botella, and P. Vanheeghe. Modeling and Simulating Human Behaviors with Conceptual Graphs. In *Proc. of ICCS'97*, volume 1257 of *LNAI*, pages 275–289. Springer, 1997.

O. Carloni, M. Leclère, and M.-L. Mugnier. Introducing graph-based reasoning into a knowledge management tool: An industrial case study. In *IEA/AIE*, pages 590–599, 2006.

L. Chauvin, D. Genest, and S. Loiseau. Contextual cognitive map. In P. W. Eklund and O. Haemmerlé, editors, *ICCS*, volume 5113 of *Lecture Notes in Computer Science*, pages 231–241. Springer, 2008. ISBN 978-3-540-70595-6.

M. Chein and M. Mugnier. *Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs*. Springer, 2009.

M. Chein and M.-L. Mugnier. Conceptual Graphs: Fundamental Notions. *Revue d'Intelligence Artificielle*, 6(4):365–406, 1992.

D. Genest and M. Chein. A content-search information retrieval process based on conceptual graphs. *Knowledge and Information Systems (KAIS)*, 8:292–309, 2005.

P. Hayes. RDF Semantics. W3C Recommendation. http://www.w3.org/TR/2004/REC-rdf-mt-20040210/, 2004.

H. Horst. Extending the RDFS Entailment Lemma. In *Proc. of the Third International Semantic Web ConferenceISWC'04*, volume 3298 of *LNCS*, page 7791. Springer, 2004.

S. Lalande, K. Staykova, M. Chein, A. Gutierrez, V. Saraydarova, and D. Dochev. Using Domain Knowledge to Speed up the Annotation of Digital Content with Conceptual Graphs. *Cybernetics and Information Technology*, 9(3):22–38, 2009.

F. Lehman. *Semantics Networks in Artificial Intelligence*. Pergamon Press, 1992.

M. Masterman. Semantic message detection for machine translation, using an interlingua. In *International Conference on Machine Translation of Languages and Applied Language Analysis*, 1962.

N. Moreau, M. Leclère, M. Chein, and A. Gutierrez. Formal and Graphical Annotations for Digital Objects. In *Proc. of International Workshop on Semantically Aware Document Processing and Indexing (SADPI'07)*, volume 259 of *ACM Digital Library, ACM International Conference Proceeding Series*, pages 69–78. ACM, 2007.

J. Novak and A. Canas. The origins of the concept mapping tool and the continuing evolution of the tool. *Information Visualization Journal*, 5:175–184, 2006.

T. Raimbault, D. Genest, and S. Loiseau. A new method to interrogate and check uml class diagrams. In F. Dau, M.-L. Mugnier, and G. Stumme, editors, *ICCS*, volume 3596 of *Lecture Notes in Computer Science*, pages 353–366. Springer, 2005. ISBN 3-540-27783-8.

R. Richens. Programming for mechanical translation. *Mechanical Translation*, 3:Discontinued Journal, 1956.

S. Shapiro. Sneps: A logic for natural language understanding and commonsense reasoning. In *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, 2000.

S. Shapiro. The sneps semantic network processing system. In *The representation and use of knowledge by computers*, 1979.

M. Shaw and B. Gaines. Knowledge and requirements engineering. In *Proceedings of the 9th Banff Knowledge Accquisition For Knowledge-Based Systems Workshop*, 1995.

J. F. Sowa. Conceptual Graphs. *IBM Journal of Research and Development*, 20(4):336–375, 1976.

J. F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.

W. Woods and J. Schmolze. The kl-one family. *Computers Math. Applic.*, 23:133–177, 1992.

Distinguishing Answers in Conceptual Graph Knowledge Bases

Nicolas Moreau, Michel Leclère, and Madalina Croitoru

LIRMM, Univ. Montpellier 2, CNRS
161, rue Ada
34392 Montpellier, France
{moreau,leclere,croitoru}@lirmm.fr

Abstract. In knowledge bases (KB), the open world assumption and the ability to express variables may lead to an answer redundancy problem. This problem occurs when the returned answers are comparable. In this paper, we define a framework to distinguish amongst answers. Our method is based on adding contextual knowledge extracted from the KB. The construction of such descriptions allows clarification of the notion of redundancy between answers, based not only on the images of the requested pattern but also on the whole KB. We propose a definition for the set of answers to be computed from a query, which ensures both properties of non-redundancy and completeness. While all answers of this set can be distinguished from others with a description, an open question remains concerning what is a good description to return to an end-user. We introduce the notion of smart answer and give an algorithm that computes a set of smart answers based on a vertex neighborhood distance.

1 Motivation

In the semantic web age, a large number of applications strongly rely on the building and processing of knowledge bases (KB) for different domains (multimedia, information management, semantic portals, e-learning, etc.). The formal languages used for representation will encounter obvious scaling problems and therefore rely on implicit or explicit graph based representations (see for example Topic Maps, RDF, Conceptual Graphs, etc.) As a direct consequence, querying such systems will have to be done through graph based mechanisms and, accordingly, optimization techniques implemented [1].

In ICCS'08 [2], we identified the *semantic database context* in which a set of answers has to be computed. For this case, there are two kinds of answer graphs: answers as subgraphs of the knowledge base graph that are used for browsing the KB or for applying rules; and answers as graphs, independent of the KB, corresponding to the classical use of a querying system[1]. With this latter kind

---

[1] For example, in SPARQL, as blank node labels can be renamed in results, see [1] ¶ 2.4.

of answer, an important problem concerns detecting redundant answers. Unlike classical databases, redundancies are not limited to duplicate tuples. Indeed, the presence of unspecified entities (generic markers in CG or blank nodes in RDF) and also a type hierarchy leads us to consider that an answer which is more general than another as redundant. In [2], we studied this problem and proposed to return irredundant forms of the most specific answers to a query.

An important problem arising in this context is ultimately related to the nature of a KB vs. a database. With a classical database, one can assume that a query designer knows the database schema and is thus able to build a query corresponding to its needs. With a KB, the schema is extended to an ontology and the different assertions are not supposed to instantiate a specific frame (the knowledge is semi-structured). Consequently, it is difficult for a query designer to specify the content of the searched knowledge: he/she wants to obtain some information about a specific pattern of knowledge. The suppression of redundant answers only by comparing answer graphs independently of the KB results in the problem not being considered. A better way to address the redundancy problem consists of completing the answer graphs with their neighborhood to obtain more detailed answers in order to return some relevant knowledge to the end-user in order to get insight (restitution, reflet) into the diversity of the knowledge in the KB. Moreover, users seem to prefer answers in their context (e.g. paragraph) rather than the exact answer[3].

In this paper we focus on answers given in a graph based form. Our motivation stems from the homogeneity of preserving the same format between the KB, query and answer. Moreover, this will allow the reuse of answers as a KB for different future answers (see for example nested queries). Note that while this paper focuses solely on the problem of distinguishing graph based answers, the same research problem will arise and results will be obtained when of answers are represented as a tuple.

2   Contribution and related work

Figure 1 shows a query, a conceptual graph formalized KB (see section 3.1) and all five answers to the query in the KB (from A_1 to A_5, in gray in the picture).

The problem that arises in this scenario is how to define relevant answers when they are independent of the KB (*i.e.* a set of answer graphs and not a set of answer subgraphs of the KB). For instance, answers A_1 and A_5 are equivalent (*"there is a human who owns an animal"*), and knowledge expressed by these answers is expressed by all of the others. The open world assumption makes it impossible to state that all humans or animals represent distinct elements of the described world. Therefore it seems preferable to only return answers that bring more knowledge, i.e. in our example that *"Mary owns a cat"* and *"a human owns a dog"*. But another relevant answer could be that *"there is a human knowing Mary who owns a cat"*.

The contribution of the paper is to refine our preceding notion of redundancy between answers, to take the knowledge of the KB into account, through the
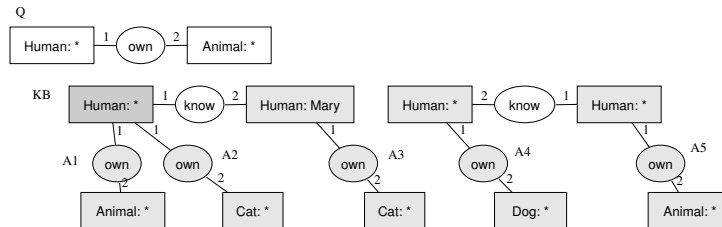
Fig. 1. Query, KB and highlighted answers.

notion of a *fair extension* of an answer, a graph that specifically describes an answer. Thus, an answer is irredundant if there is such a "fair" extension. A special case of this problem (when the answer is a concept node) corresponds to the problem of Generation of Referring Expressions (GRE) studied in the conceptual graph context in [4], that we extend for our purposes. Based on the extended notion of redundancy, we define two answer subset properties of *non-redundancy* (there is no redundant answer in the subset) and *completeness* (each answer is redundant to an answer of the subset). We then explain how to compute all the non-redundant and complete subsets of answers, based on a *redundancy graph*. We then discuss what is a good set of referring graphs to be returned to the user, and give an algorithm that fulfills these good properties.

As previously mentioned, a special case of answer identification was studied in [4]. The RDF query language SPARQL offers a way to describe answers (by the DESCRIBE primitive) but it does not address the specific problem of distinguishing one answer from another according to semantically sound syntactic criteria. The problem of answer redundancy in the Semantic Web context has been studied in [5], but the redundancy stated in this paper concerns the union of all answers, and corresponds, in the CG field, to the classical notion of irredundancy (see section 3.1), as opposed to our redundancy between answers. In an article about OWL-QL [6] the notion of server terseness is defined, which is the ability of a server to always produce a response collection that contains no redundant answers (i.e. there is no answer that subsumes another one). This corresponds to our previous notion of redundancy defined in [2].

In the next section, preliminary notions about conceptual graphs and our query framework are given. Section 4 deals with the extended notion of redundancy between answers and its application to a subset of answers. In section 5 we discuss the relevancy of the set of referring graphs returned to the user. We conclude our work in the last section.

3 Preliminary notions

3.1 Simple Graphs

The conceptual graph formalism we use in this paper has been developed at LIRMM over the last 15 years [7]. The main difference with respect to the initial

general model of Sowa [8] is that only representation primitives allowing graph-based reasoning are accepted.

Simple graphs (SGs) are built upon a *support*, which is a structure $S = (T_C, T_R, I, \sigma)$, where T_C is the set of concept types, T_R is the set of relations with any arity (arity is the number of arguments of the relation). T_C and T_R are partially ordered sets. The partial order represents a specialization relation ($t' \leq t$ is read as "t' is a specialization of t"). I is a set of individual markers. The mapping σ assigns a signature to each relation specifying its arity and the maximal type for each of its arguments.

SGs are labeled bipartite graphs denoted $G = (C_G, R_G, E_G, l_G)$ where C_G and R_G are the concept and relation node sets respectively, E_G is the set of edges and l_G is the mapping that labels nodes and edges. Concept nodes are labeled by a couple $(t : m)$, where t is a concept type and m is a marker. If the node represents an unspecified entity, its marker is the generic marker, denoted $*$, and the node is called a *generic* node, otherwise its marker is an element of I, and the node is called an *individual* node. Relation nodes are labeled by a relation r and, if n is the arity of r, it is incidental to n totally ordered edges.

A specialization/generalization relation corresponding to a deduction notion is defined over SGs and can be easily characterized by a graph homomorphism called *projection*. When there is a projection π from G to H, H is considered to be more specialized than G, denoted $H \leq G$. More specifically, a projection π from G to H is a mapping from C_G to C_H and from R_G to R_H, which preserves edges (if there is an edge numbered i between r and c in G then there is an edge numbered i between $\pi(r)$ and $\pi(c)$ in H) and may specialize labels (by observing type orders and allowing substitution of a generic marker by an individual one).

In the following, we use the notion of *bicolored SG* that was first introduced in [9]. A bicolored SG is an SG $H = \langle H_0, H_1 \rangle$ in which a color on $\{0, 1\}$ is assigned to each node of H, in such a way that the subgraph generated by 0-colored nodes, denoted H_0 and called the core of H, is a subSG. H_1, which is the sub-SG defined by 1-colored vertices and 0-colored concepts that are in relation with at least one 1-colored concept, is called the description.

3.2   Query framework

The chosen context is a base composed of assertions of entity existences and relations over these entities, called *facts*, and stored in a single graph (not necessarily connected) named the *knowledge base*. This graph is assumed to be *normalized* if it does not contain two individual nodes with the same marker i. A normal form is easily computed by merging duplicate individual nodes of the graph. On the other hand, we do not require the KB graph to be in *irredundant form*: a graph G is in irredundant form iff there is no a projection from G in one of these strict subgraphs; otherwise this graph is said to be in *redundant form*. Indeed, computation of the irredundant form of a graph is expensive as the base can be large [10] and, moreover, there is not any local criterion for computation of the irredundant form and thus no incremental method (started at each updating of the base) can be expected. Such a KB graph B is simply queried by specifying a

SG Q called the *query*. There is no constraint on the query (normalization or irredundancy). The answers to the query are found by computing the set $\Pi(Q, B)$ of projections from Q to B. The primary notion of answer consists of returning the set of subgraphs of B image of Q by a projection in $\Pi(Q, B)$.

Definition 1 (Answer set). *The set of answers of a query Q in a base B, denoted $Q(B)$, is $\{\pi(Q) \mid \pi \in \Pi(Q, B)\}$*[2].

The first research question we address in this paper is whether this set of answers contains redundant answers? In fact, three kinds of redundancies can arise:

1. **Duplication:** two answers are identical. There is a duplication when two projections define the same subgraph. This problem can be solved easily by only keeping one of the duplicate subgraphs (this is done in the answer set $Q(B)$). An example of duplication is given in fig. 3(b) taken as the KB and fig. 3(c) taken as the query: there are two projections from the query whose images are the whole KB.
2. **Inclusion:** An answer is contained in another one. There is an inclusion when an answer is in a redundant form. Then its subgraph, in irredundant form, is also an answer. In the previous example (fig. 3(b) and 3(c)), there are two projections that define included answers.
3. **Redundancy:** An answer is more general or equivalent than another one. There is a redundancy when two answers are comparable (thus the knowledge expressed by one is also expressed by another); as an example answers A_1 and A_2 of figure 1.

Inclusion will be studied in section 4.4. The true redundancy problem becomes crucial since the answers are no longer KB subgraphs. Indeed, two answers can appear redundant when they are not really redundant. In [2], we define the notion of redundancy based only on the comparability of the answer graphs. This approach was motivated by the following argument: as the returned set of answers is independent of the KB, the subset of the *more specific irredundant answers*, denoted R_{min}, is sufficient to bring the entire range of answers. Moreover, R_{min} is minimal in terms of vertex number. In the following section, we characterize the true redundancy.

4   Dealing with true redundancy

In [2], the *completeness* criterion (the knowledge expressed by each initial answer is expressed by one of the answers contained in the returned subset of answers) ensures that no knowledge is lost when a redundant answer is deleted. But this redundancy is based only on the comparability of answer subgraphs (that are <u>semantically </u>close as they are specializations of the query).

[2] "Answer set", denoted $Q(B)$, and "answer" notions correspond respectively in our previous work [2] to notions of "answers by image subgraphs", denoted $R_{IP}(Q, B)$, and "images of proof". Names have been changed for simplification.

4.1 The true redundancy

The true redundancy has to take the knowledge brought by the neighbor vertices of the answer into account. With this aim, we introduced the notion of extended answer, which is an answer supplemented with some knowledge extracted from its neighborhood.

Definition 2 (Extension of an answer). *Let B be a KB graph and Q a query graph. An extension of an answer A from $Q(B)$ is a bicolored graph $E = \langle E_0, E_1 \rangle$, where E_0 is isomorphic to A and such that there is a projection π from E to B with $\pi(E_0) = A$.*
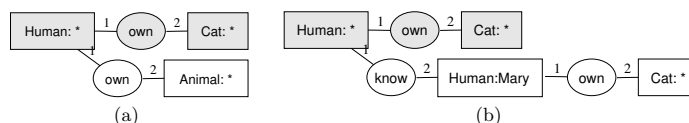
Fig. 2. Several extensions of answer A_2 of fig. 1.

Fig. 2 represents several extensions of answer A_2 of fig. 1. Note that the extensions are not necessarily isomorphic copies of subgraphs of the KB.

With the previous definition of redundancy (section 3.2), if two answers have two incomparable extensions, they are two distinct answers and thus have to be considered as non-redundant answers, the one w.r.t. the other one. However, these two answers must not be distinguished in an "artificial" way: i.e. if the answers are distinguished by selectively adding knowledge to the extension of only certain answers but not to all of those which possess this knowledge in their neighborhood.

Definition 3 (Fairness property). *An extension E of an answer A from a set $Q(B)$ is fair iff there is no extension E' of another answer A' from $Q(B)$ such that there is a projection π from E to E' with $\pi(E_0) = E_0'$ and $\pi(E_1) = E_1'$.*

The extension of figure 2(b) is a fair extension of the answer A_2 of figure 1, contrary to the extension of figure 2(a). One can now give a definition of the true redundancy:

Definition 4 (True redundancy). *An answer A from a set of answers $Q(B)$ is truly redundant if there is no fair extension of this answer.*

In the example of fig. 1, answer A_1 is redundant. The search for a fair extension may seem to be a difficult task, considering that the number of extensions that can be generated for an answer is infinite. However, these extensions are semantically bounded by a more specific one that corresponds to an isomorphic copy of the base, $\langle A, B \setminus A \rangle$[3]. Naturally, the more one adds knowledge to the

---

[3] For clarity, we sometimes denote subgraphs of KB as cores or descriptions of bicolored graph instead of their isomorphic copies.

extension (the more it is specific), the more one potentially distinguishes this answer. So A is irredundant only if $\langle A, B \setminus A \rangle$ is a fair extension.

Theorem 1. *There is a fair extension E of an answer A iff $\langle A, B \setminus A \rangle$ is a fair extension.*

Proof. By contraposition. Suppose that there is a fair extension $E = \langle E_0, E_1 \rangle$ of A and that $\langle A, B \setminus A \rangle$ is not a fair extension of A. $\langle A, B \setminus A \rangle$ is an extension of A and is isomorphic (without considering the colors) to B. As E is an extension of A, there is a projection π from E to B such that $\pi(E_0) = A$. As $\langle A, B \setminus A \rangle$ is not a fair extension of A, there is a projection π' from $\langle A, B \setminus A \rangle$ to B such that $\pi(R_0) \neq A$. So there is a projection $\pi'' = \pi \circ \pi'$ from E to B such that $\pi''(E_0) \neq A$. Thus E is not a fair extension of A. $\qquad\square$

Corollary 1. *An answer A of $Q(B)$ is truly irredundant iff $\langle A, B \setminus A \rangle$ is a fair extension.*

4.2   The GRE problem

The problem of finding a fair extension of an answer is strongly related to the problem of generation of referring expressions (GRE) known in the natural language processing field, which aims to describe an object in a scene such that the description only refers to this object. The GRE problem was formalized in the CG framework in [4], where the scene is an SG and the object to identify is a concept of the SG. A *referring graph* of a concept v is a subgraph of the KB containing this concept, and a *distinguishing graph* is a referring graph whose v is a fixed point for all projections of the graph in the KB. Our problem of answer redundancy can be seen as an extension of this formalization.

Definition 5 (Referring graph). *A referring graph of a subgraph G' of a graph G is a subgraph R of G that contains G' as a subgraph. To distinguish G' from the rest of the referring graph, we denote it as a bicolored graph $R = \langle R_0, R_1 \rangle$ where $R_0 = G'$ and $R_1 = R \setminus G'$.*

Definition 6 (Distinguishing graph). *A referring graph R of a subgraph G' of a graph G is distinguishing if, for each projection π from R to G, $\pi(R_0) = G'$.*

The referring graph of figure 2(b) is a distinguishing graph of the answer A_2 of figure 1, contrary of the referring graph of figure 2(a). We can now link the true redundancy notion with the existence of a distinguishing graph for a given answer. The next properties strengthen the notion of true redundancy by showing its independence in the query and thus in the other answers.

Property 1. Let A be an answer of $Q(B)$. There is a fair extension of A iff there is a distinguishing graph of A w.r.t. B.

Proof. Let $E = \langle E_0, E_1 \rangle$ be a fair extension of A. Then $\pi(E)$ is a distinguishing graph of A w.r.t. B. On the other hand, a distinguishing graph of A is also a fair extension of A. $\qquad\square$

Corollary 2. *An answer A of $Q(B)$ is irredundant iff $R = \langle A, B \setminus A \rangle$ is a distinguishing graph of A w.r.t. to B.*

Thus, determining whether an answer is irredundant can be done by testing the distinguishness of the referring graph built from KB. However, the cost of such a test is exponential in the size of the KB.

4.3 Redundancy and subset of answers

Since the redundancy of an answer has been defined, it could be considered that for building a set of answers without redundancies one could simply remove redundant answers. However, the redundancy defined in the preceding section hides the fact that they are two types of redundancy. This is shown in fig. 3 (a query and three different KBs) :

The first case will arise when an answer is completely redundant with respect to another answer (fig. 3(b)). This means that A_2 is redundant w.r.t. A_1 but the reverse does not hold. In this case, we say that A_2 is strongly redundant w.r.t. A_1. Thus, we only return the answer A_1.

The second case arises when there are a set of answers which are redundant amongst themselves (fig. 3(c) and 3(d)). In these two examples, an answer is redundant with respect to the others and vice-versa. In this case, we say that the answers are mutually redundant and we have to choose an answer in this set.

Note that the answer redundancy problem still holds in KBs in irredundant forms, as in the example of fig. 3(d).
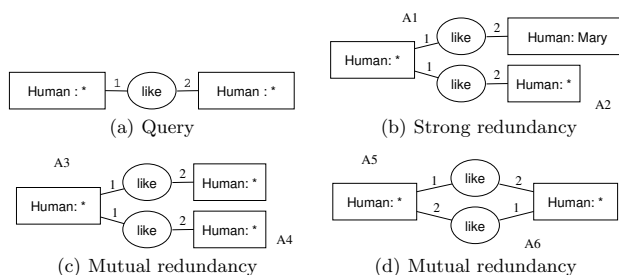
Fig. 3. Different cases of redundant answers

Strong and mutual redundancies are based on the redundancy of an answer *w.r.t. another one*:

Definition 7 (Redundancy relation). *An answer A is redundant to an answer A' iff for each referring graph R^A of A, there is a projection π from R^A to a referring graph $R^{A'}$ of A' with $\pi(R_0^A) = \pi(R_0^{A'}) = A'$.*

One can test the redundancy relation with the KB referring graph:

Property 2. An answer A is redundant to an answer A' iff there is a projection π from $\langle A, B \setminus A \rangle$ to a referring graph $R^{A'}$ of A' with $\pi(A) = A'$.

Therefore an answer A is redundant to an answer A', the redundancy relation between A' and A defines the two previously seen redundancy cases:

Definition 8 (Redundancies between answers). *Given A and $A' \in Q(B)$, such that A is redundant to A':*

- *A is strongly redundant to A' if A' is not redundant to A ;*
- *A and A' are mutually redundant if A' is redundant to A ;*

Based on the notion of true redundancy, we can refine our notions of *non-redundancy* and *completeness* of subsets of answers that we defined in [2] :

Definition 9 (Non-redundant subset of answers). *A subset of answers $\mathcal{A}$ of $Q(B)$ is non-redundant if there are no two answers A and $A' \in \mathcal{A}$ such that $A \neq A'$ and A is redundant to A'.*

Definition 10 (Completeness). *A subset of answers $\mathcal{A}$ of $Q(B)$ is complete if for each answer A of $Q(B)$ there is an answer A' of $\mathcal{A}$ such that A is redundant to A'.*

We define a *graph of redundancies* based on the notions of strong and mutual redundancies. All types of redundancies of definition 8 can be viewed as relations on $Q(B)^2$, for example (A, A') belongs to the strong redundancy relation if A is strongly redundant to A'. Thus we can characterize properties of these relations. Particularly, mutual redundancy defines equivalent classes over the set of answers, and strong redundancy links all answers of an equivalent class to all answers of another equivalent class. We construct the redundancy graph such that each vertex is an equivalent class, and is linked by the strong redundancy:

Definition 11 (Graph of redundancy). *The graph of redundancy $G = (V, E)$ of $Q(B)$ is a directed graph, where vertices represent equivalent classes of the mutual redundancy relation and where there is an edge between v_1 and v_2 if all answers of the class represented by v_1 are strongly redundant to all answers of the class represented by v_2.*

Fig. 4(a) represents the redundancy graph of query and KB of fig. 1, whereas fig. 4(b) represents the redundancy graph of query of fig. 3(a) and KB defined by the union of KBs of fig. 3(b), 3(c) and 3(d).

(a)   (b)

Fig. 4. Redundancy graph of the previous examples.

To construct a complete and non-redundant set of answers, one has to only choose a single answer per equivalent class (to avoid mutual redundancy), only from equivalent classes that are sinks (to avoid strong redundancy), and for all of them (to be complete). This is why there is more than one non-redundant complete subset in the second example (fig. 4(b)): there is an equivalent class that is a sink and contains more than one element ($\{A_5, A_6\}$).

Theorem 2. *A subset of answers $\mathcal{A}$ of $Q(B)$ is complete and non-redundant iff for each sink in the redundancy graph there is a single answer of $\mathcal{A}$ that is represented by this sink.*

Proof. Given a non-redundant and complete subset of answers $\mathcal{A}$. Non-redundancy means that there is a single answer for each equivalent class represented in $\mathcal{A}$ and that there are no two answers A_i and A_j such that there is a path from A_i to A_j. Completeness ensures that for all answers A_i of $Q(B)$ (particularly answers belonging to a sink) there is an answer A_j such that A_i is redundant to A_j. So $\mathcal{A}$ has to contain an answer of each sink. When combining completeness and non-redundancy, only one answer of each sink is taken.
• Given a subset of answers $\mathcal{A}$ that is composed of an element of each sink of the redundancy graph. Given two answers of $\mathcal{A}$. These answers are not mutually redundant because there are no two answers of the same equivalent class. These answers are not strongly redundant because each answer comes from a sink. Thus $\mathcal{A}$ is non-redundant. For each answer A_i of $Q(B)$, either there is an answer A_j of the same equivalent class in $\mathcal{A}$ (thus A_j is redundant to A_i), or there is an answer A_k in $\mathcal{A}$ that comes from a sink such that there is a path from the equivalent class of A_i to the sink, and thus A_i is redundant to A_k. $\mathcal{A}$ is complete.

The construction of a redundancy graph combines the problem of finding all projections of a graph into another graph (to compute the set of answers) and the problem of computation of the irredundant form of a graph. Indeed, as stated by property 2, computation of all redundancy links of all answers used to construct the graph is based on all projections of the most specific referring graph of each answer (i.e. a bicolored isomorphic copy of the whole KB). Therefore, a way to compute all redundancy links is to compute all projections from the KB into itself, and check images of a each answer by all projections.

4.4 Inclusion of answers

As mentioned in section 3.2, the inclusion of answers is one of the problems that can occur. If treated as a kind of duplication, included answers are just deleted from the answer set before computation of a non-redundant complete subset. But this approach is not the best one. We think that the inclusion problem should be treated after the redundancy problem.

In the example of fig. 5, there are seven answers: three that contain only one relation (A_1, A_2, A_3), three that contain two relations (A_{12}[4], A_{13}, A_{23}), and

---

[4] Answer A_{ij} represents the answer that is the union of answers A_i and A_j.

one with three relations (A_{123}). Considering all answers, only answers A_2, A_3 and A_{23} are irredundant (see fig. 5(c)), but it is not suitable to keep A_2 and A_3. Otherwise, if only non-included answers are kept, this leads to keeping answers that were redundant to deleted answers (here A_{123}, redundant to A_{23}), which is not good.

(a) Query   (b) KB
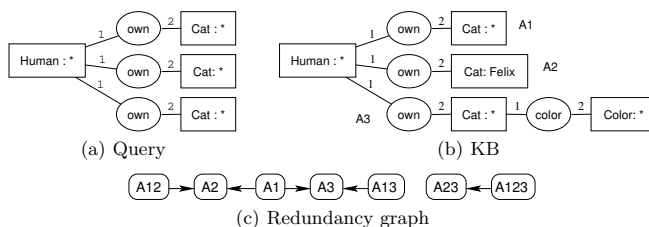
(c) Redundancy graph

Fig. 5. A query and a KB that produce included answers, and their redundancy graph.

So the best way is to deal with redundancy first (by computing a subset that is non-redundant and complete) and to delete included answers after that. In the previous example, this strategy led to the subset $\{A_{23}\}$.

4.5 Redundancy at a considered distance

In the query framework section, it was stated that computation of the irredundant form of the KB is a difficult problem. But we also see that computation of the redundancy graph also requires finding all projections of the KB into itself. Thus, we propose to restrict referring graphs of an answer to a portion of the base that is "near" this answer. This is also due to the fact that a referring graph that contains too much knowledge, even if it distinguishes an answer, does not help the user much. So we propose to bound referring graphs by a distance k, and to only consider vertices that are in the distance field of one of the vertex of an answer, which forms the *k-neighborhood graph* of the answer:

Definition 12 (k-neighborhood graph). *Given a KB B, a subgraph S of B, and a step k ($k \geq 0$), the k-neighborhood graph, denoted $N^k(S)$, is defined recursively by:*

- $N^0(S) = S$
- $N^{n+1}(S)$ *is composed of $N^n(S)$ expanded by every relation r not in $N^n(S)$ and which is linked to a concept of $N^n(S)$, and by all concepts linked to r.*

Recursion stops when $n = k$ or $N^n(S) = N^{n+1}(S)$ and returns $N^n(S)$.

It seems obvious that all definitions put forward previously should now take this constraint into account. Bounding referring graphs can be seen as a restriction of the KB, which is now considered as the union of the k-neighborhood graph of each answer, for a given distance.

74

Definition 13 (Truncated KB). *Given a KB B, a query Q, and a distance $k \geq 0$, the truncated KB at distance k is $B_k = \bigcup_{A \in Q(B)} N^k(A)$*

Now we can apply all previous definitions to the truncated KB. For example, an answer A of $Q(B)$ is irredundant considering the distance k iff $R = \langle A, B_k \setminus A \rangle$ is a distinguishing graph of A w.r.t. B_k (see corollary 2).

5 Building smart answers

In the previous section, redundancy and completeness were only studied from a theoretical standpoint. A more user-aspect oriented standpoint was introduced in the section 4.5. Even if the user gets a subset of answers that is non-redundant and complete, he/she has no way to distinguish answers, i.e. the only assumption that can be made is that there is, for each answer, a way to distinguish them from the others. Otherwise, property 1 states that the most specific referring graph is one of them, but it usually contains too much knowledge.

So what are the properties of a really good set of referring graphs returned to the user? To keep the notions of non-redundancy and completeness, only answers of such a subset of answers should have a referring graph. As all of these answers can be distinguished, all referring graphs should be a distinguishing graph. Finally, to not introduce unnecessary redundancies in the set of referring graphs, there should be only one distinguishing graph by referred answer. A set of referring graphs that fulfills these properties is called a *set of smart answers*:

Definition 14 (Set of smart answers). *A set of smart answers $\mathcal{S}$ of a query Q on a KB B is a set of distinguishing graphs of all answers of a non-redundant complete set $\mathcal{A}$ of $Q(B)$ such as $|\mathcal{S}| = |\mathcal{A}|$.*

We propose the Bounded Smart Answers (BSA) algorithm (see algo. 1) that takes the answers and a distance as parameters, and returns a set of smart answers of truncated KB at distance k such that each extension is the minimal k-neighborhood graph that distinguishes the answer[5].

Theorem 3. *Algorithm $BSA(Q(B), k)$ produces a set of smart answers of Q on the truncated KB B_k.*

Proof. In DAK, all referring graphs are constructed with the same distance. Therefore, thanks to the distance conservation property of the projection, that if there is a bicolored projection from $N^k(A_i)$ to $N^k(A_j)$, there is a projection from $N^k(A_i)$ to B_k such that image of the core of $N^k(A_i)$ is equal to A_j (i.e. $N^k(A_i)$ is a referring graph of A_j in B_k). For each answer A that belongs to an equivalent class that is not a sink, A will not be returned by DAK because of the second "foreach" of DAK. The third foreach of DAK ensures that for each equivalent class that is a sink, the algorithm will only add one (the first taken) answer that belongs to this class once the good distance is reached. Thus

---

[5] Note that comparisons in DAK (algo. 2) are between bicolored graphs, that is $B \leq B'$ iff there is a π from B' to B such that $\pi(B'_0) = B_0$.

Data: Answers $Q(B)$, a distance k
Result: A set of smart answers of truncated KB B_k
begin
   $i \leftarrow 0$
   $D \leftarrow \emptyset$ `// distinguished answers`
   $S \leftarrow \emptyset$ `// smart answers`
   while $i \leq k$ **or** $D \neq Q(B)$ **do**
      $D' \leftarrow \mathtt{DISTING\_AT\_K}(Q(B), D, i)$
      $D \leftarrow D \cup D'$
      foreach $A \in D'$ **do**
         $S \leftarrow S \cup \{\langle A, N^i(A) \setminus A \rangle\}$
   return S
end

Algorithm 1: BOUNDED SMART ANSWERS (BSA)

the union of all answers returned by DAK is a non-redundant complete subset of answers of B_k. As BSA returns only one graph per answer of the computed non-redundant complete subset, BSA returns a set of smart answers of B_k. $\quad\square$

6 Conclusion

We extended our previously defined notion of answer redundancy. Our new framework now considers answers but also descriptions (called referring graphs) that can distinguish an answer amongst others. These descriptions could be adapted to query languages of the semantic web, e.g. by giving a formal definition of the SPARQL DESCRIBE query form. Therefore this new redundancy is now linked to the whole KB. Based on this new redundancy, we refined two other previous definitions concerning subsets of answers: *non-redundancy* property (there is no redundant answer in the subset) and *completeness* (each answer is redundant to an answer of the subset). We proposed a way to construct all non-redundant and complete subsets of answers using a *redundancy graph*. We introduced the notion of smart answer and gave an algorithm that computes a set of smart answers based on a vertex neighborhood distance.

Answer redundancy arises because of open world assumption and undefined objects (generic concepts). A deeper redundancy still exists, that is not related to the formalized world (the KB), but rather to the "real world" (described by the KB). For example, it is possible that non-redundant answers "a big cat" and "a white cat" refer to a single cat of the "real world", which is big and white. The study of this new kind of redundancy can provide a foundation for using aggregation operators (e.g. number of results to a query) in graph based KB query languages.

References

1. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. Technical report, W3C (2008)

Data: Answers $Q(B)$, distinguished answers D, a distance k
Result: A set D' of distinguished answers at distance k
begin
    $D' \leftarrow \emptyset$
    foreach $A \in Q(B) \setminus D$ **do**
        $disting \leftarrow$ **true**
        foreach $A_2 \in Q(B) \setminus \{A\}$ **do**
            if $N^k(A) \geq N^k(A_2)$ **and** $N^k(A_2) \not\geq N^k(A)$ **then**
                $disting \leftarrow$ **false**

        foreach $A_3 \in D'$ **do**
            if $N^k(A) \geq N^k(A_3)$ **then**
                $disting \leftarrow$ **false**

        if $disting =$ **true then**
            $D' \leftarrow D' \cup \{A\}$
    return D'
end

Algorithm 2: DISTING AT K (DAK)

2. Leclère, M., Moreau, N.: Query-answering cg knowledge bases. In: ICCS '08: Proceedings of the 16th international conference on Conceptual Structures, Berlin, Heidelberg, Springer-Verlag (2008) 147–160
3. Lin, J.J., Quan, D., Sinha, V., Bakshi, K., Huynh, D., Katz, B., Karger, D.R.: What makes a good answer? the role of context in question answering. In: INTERACT. (2003)
4. Croitoru, M., van Deemter, K.: A Conceptual Graph approach to the Generation of Referring Expressions. In: Proceedings of the 20 th International Joint Conference on Artificial Intelligenc (IJCAI-2007). (2007)
5. Gutierrez, C., Hurtado, C., Mendelzon, A.O.: Foundations of semantic web databases. In: PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, New York, NY, USA, ACM Press (2004) 95–106
6. Fikes, R., Hayes, P., Horrocks, I.: OWL-QL, a language for deductive query answering on the Semantic Web. Web Semantics: Science, Services and Agents on the World Wide Web **2**(1) (2004) 19–29
7. Chein, M., Mugnier, M.L.: Conceptual Graphs: Fundamental Notions. Revue d'Intelligence Artificielle **6**(4) (1992) 365–406
8. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley (1984)
9. Baget, J.F., Mugnier, M.L.: The SG family: Extensions of simple conceptual graphs. In: IJCAI. (2001) 205–212
10. Mugnier, M.: On generalization/specialization for conceptual graphs. Journal of Experimental & Theoretical Artificial Intelligence **7**(3) (1995) 325–344

ALASKA for Ontology Based Data Access

Jean-François Baget, Madalina Croitoru, Bruno Paiva Lima da Silva

LIRMM (University of Montpellier II & CNRS), INRIA Sophia-Antipolis, France

Abstract. Choosing the tools for the management of large and semi-structured knowledge bases has always been considered as a quite crafty task. This is due to the emergence of different solutions in a short period of time, and also to the lack of benchmarking available solutions. In this paper, we use ALASKA, a logical framework, that enables the comparison of different storage solutions at the same logical level. ALASKA translates different data representation languages such as relational databases, graph structures or RDF triples into logics. We use the platform to load semi-structured knowledge bases, store, and perform conjunctive queries over relational and non-relational storage systems.

1 Motivation and Impact

The ONTOLOGY-BASED DATA ACCESS (ODBA) problem [4] takes a set of facts, an ontology and a conjunctive query and aims to find if there is an answer / all the answers to the query in the facts (eventually enriched by the ontology). Several languages have been proposed in the literature where the language expressiveness / tractability trade-off is justified by the needs of given applications. In description logics, the need to answer conjunctive queries has led to the definition and study of less expressive languages, such as the $\mathcal{EL}$ ([1]) and DL-Lite families [2]. Properties of these languages were used to define profiles of the Semantic Web OWL 2 language (`www.w3.org/TR/owl-overview`).

When the above languages are used by real world application, they are encoded in different data structures (e.g. relational databases, Triple Stores, graph structures). Justification for data structure choice include (1) storage speed (important for enriching the facts with the ontology) and (2) query efficiency. Therefore, deciding on what data structure is best for one's application is a tedious task. While storing RDF(S) has been investigated from a database inspired structure [3], other logical languages did not have the same privilege. Even RDF(S), often seen as a *graph*, has not been thoroughly investigated from a *ODBA perspective* wrt *graph structures* and emergence of *graph databases* in the NoSQL world.

This demo will allow to answer the following research question: "*How to design an unifying logic-based architecture for ontology-based data access?*".

2 ALASKA

We thus demonstrate the ALASKA (acronym stands for **A**bstract and **L**ogic-based **A**rchitecture for **S**torage systems and **K**nowledge bases **A**nalysis) platform. ALASKA's goal is to enable and perform ODBA in a logical, generic manner, over existing, heterogenous storage systems. The platform architecture is multi-layered.
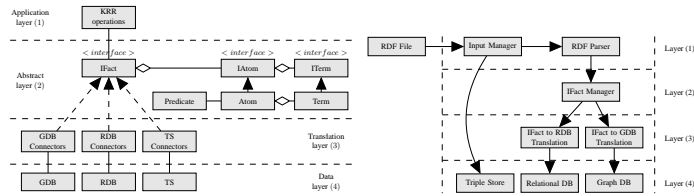
Fig. 1. ALASKA class diagram, and workflow of storage protocol.

The first layer is (1) the **application** layer. Programs in this layer use data structures and call methods defined in the (2) **abstract** layer. Under the abstract layer, the (3) **translation** layer contains functions by which logical expressions are translated into the languages of several storage systems. Those systems, when connected to the rest of the architecture, compose the (4) **data** layer. Performing higher level reasoning operations within this architecture consists of writing programs and functions that use exclusively the formalism defined in the abstract layer. Once this is done, every program becomes compatible to any storage system connected to architecture.

To have a functional architecture, representative storage systems were selected. The systems already connected to ALASKA are listed below (please note that this list is not final and subject to constant updates):

→ **Relational databases:** Sqlite 3.3.6[1], MySQL 5.0.77[2]
→ **Graph databases:** Neo4J 1.8.1[3], DEX 4.7[4], OrientDB 1.0rc6[5], HyperGraphDB 1.1[6]
→ **Triples Stores:** Jena TDB 0.9.4[7]

Figure 1 displays, on the left-hand side, the class diagram of the architecture. On the right-hand side the workflow of knowledge base storing is illustrated. Let us analyse the workflow. We consider a RDF file as input. The RDF file is passed to the Input Manager (layer 1). According to the storage system needs the Input Manager directs it accordingly. If the RDF file will be stored in a Triple Store than the file is directly passed to the Triple Store of choice (layer 4). If the RDF file needs to be stored in a graph database the file is first transformed in an IFact object (layer 2). It is then translated (layer 3) to the language of the system of choice (graph database in this case) before being stored onto disk (layer 4).

Querying in ALASKA follows a similar workflow as the storage. In Figure 2, on the left hand side we show the storing workflow for storing a fact F in either a relational database or a graph database (for simplification reasons). On the right hand side of

---

[1] http://www.sqlite.org/
[2] http://www.mysql.com/
[3] http://www.neo4j.org/
[4] http://www.sparsity-technologies.com/dex
[5] http://www.orientechnologies.com/orient-db.htm
[6] http://www.hypergraphdb.org/
[7] http://jena.sourceforge.net/

the figure the querying workflow is depicted for graph and relational databases. Let us consider a fact F both stored in a relational database and in a graph database. Let us also consider a query Q. This query can either be expressed in SQL (or in a graph language of choice) and be sent directly to the respective storage system (e.g. the SQL Q query to the F in the relational database). Alternatively, the query can be translated in the abstract logic language and a generic backtrack algorithm used for answering Q in F. This generic backtrack algorithm will solely use the native language "elementary" operations for accessing data.

Fig. 2. ALASKA storage and querying workflow.

3 ALASKA Demo Procedure

In a nutshell, the demo procedure of ALASKA goes as follows. Given a knowledge base (user provided or selected amongst benchmarks provided by ALASKA) a set of storage systems of interest are selected by the user. The knowledge base is then transformed in the Abstract Architecture and consequently stored in the selected systems. The storage time per system is then showed to the user (excluding the time needed for translation into Abstract Layer). Once the storage step is finished, users are able to perform conjunctive queries over the knowledge bases and, once again, compare the time of each system for query answering.

Let us consider an example. The knowledge base used here has been introduced by the SP2B project [5]. The SP2B project supplies a generator that creates knowledge bases with a certain parametrised quantity of triples maintaining a similar structure to the original DBLP knowledge base. The generator was used to create 5 knowledge bases of increasing sizes (5 million triples, 20, 40, 75 and respectively 100). Each of the knowledge bases has been stored in Jena, DEX, SQLite and Neo4J. In Figure 3 we show the time for storing the knowledge bases and their respective sizes on disk.

The user can see that the behavior of Jena is worse than the other storage systems. This is due to the Jena RDF parser uses central memory for buffering purposes when loading a file. For comparison, the other systems use the custom made RDF parser of ALASKA. Let us also note that DEX behaves much better than Neo4J and this is due to the fact that ACID transactions are not required for DEX (while being respected by Neo4J). Second, the size of storage is also available to the user. One can see, for instance, that the size of the knowledge base stored in DEX and Neo4J is well under

the size of initial RDF file. However, the size of the file stored in Jena is bigger than the one stored in SQLite and bigger than the initial size of the RDF file.
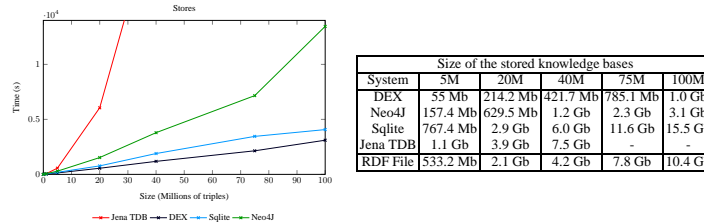

Size of the stored knowledge bases					
System	5M	20M	40M	75M	100M
DEX	55 Mb	214.2 Mb	421.7 Mb	785.1 Mb	1.0 Gb
Neo4J	157.4 Mb	629.5 Mb	1.2 Gb	2.3 Gb	3.1 Gb
Sqlite	767.4 Mb	2.9 Gb	6.0 Gb	11.6 Gb	15.5 Gb
Jena TDB	1.1 Gb	3.9 Gb	7.5 Gb	-	-
RDF File	533.2 Mb	2.1 Gb	4.2 Gb	7.8 Gb	10.4 Gb

Fig. 3. Storage time and KB sizes in different systems

Once the storage step is finished, users are able to perform conjunctive queries. As already explained, querying the newly-stored knowledge base using the native interrogation engine (SQL for relational databases, SPARQL for 3Stores, etc.) is still possible with ALASKA. However, ALASKA also allows the possibility to perform conjunctive queries that access any storage system included in the platform using the same backtrack algorithm. The queries we have used here are:

1. `type(X,Article)`
   Returns all the elements which are of type article.
2. `creator(X,PaulErdoes) . creator(X,Y)`
   Returns the persons and the papers that were written with Paul Erdoes.
3. `type(X,Article) . journal(X,Journal1-1940) . creator(X,Y)`
   Returns the creators of all the elements that are articles and were published in Journal 1 (1940).
4. `type(X,Article) . creator(X,PaulErdoes)`
   Returns all the articles created by Paul Erdoes.

In the graphs in Figure 4 and 5 we show the combination storage and querying algorithm. For instance Jena(BT) stands for using Jena for elemntary access operations and the generic backtrack for querying. SQLite(SQL) uses directly the SQL querying engine over the data stored in SQLite. In the graph corresponding to Q1 we also study the behavior of SQLite using the genric backtrack. For other queries we did not show it because the behavior is much worse that the other systems. We can also observe that for Q1, Q3 and Q4 queries SQLite and Jena behave faster than the graph bases. However, for Q2 this is no longer the case. In this case the fastest system for the generic backtrack is Jena followed by Neo4J and DEX, while SQLite explodes. The intuition behind this behavior is due to the phase transition phenomenon in relational databases but these aspects are out of the scope of this demonstration.

4 Discussion

An abstract platform (ALASKA) was created in order to perform storage operations independently of the data location. In order to enable the comparison between different
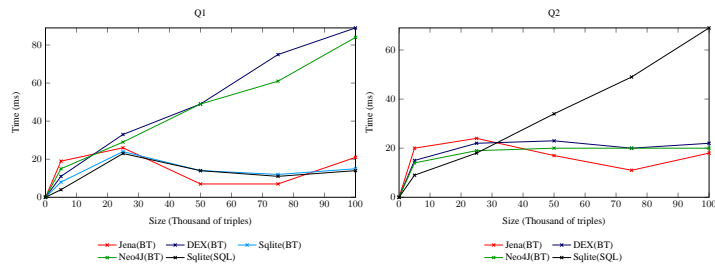
Fig. 4. Querying performance using ALASKA for large knowledge bases: Q1 and Q2
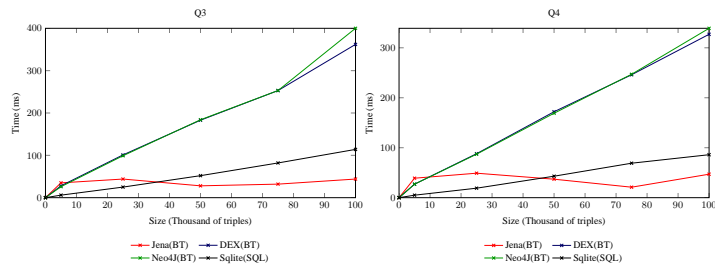
Fig. 5. Querying performance using ALASKA for large knowledge bases: Q3 and Q4

storage paradigms, ALASKA has to translate a knowledge base from a common language (i.e. First Order Logic) into different other representation language. Comparing different storage and querying paradigms becomes then possible. A knowledge base stored in a relational database can be also stored in a graph based database as well as a Triple Store and queried with an in built SPARQL engine etc.

References

1. F. Baader, S. Brandt, and C. Lutz. Pushing the el envelope. In *Proc. of IJCAI 2005*, 2005.
2. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The dl-lite family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
3. B. Haslhofer, E. M. Roochi, B. Schandl, and S. Zander. Europeana RDF store report. Technical report, University of Vienna, Vienna, Mar. 2011.
4. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS 2002*, 2002.
5. M. Schmidt, T. Hornung, G. Lausen, and C. Pinkel. Sp2bench: A sparql performance benchmark. *CoRR*, abs/0806.4627, 2008.

Coalitional Games via Network Flows

Talal Rahwan[1], **Tri–Dung Nguyen**[1], **Tomasz P. Michalak**[2,3],
Maria Polukarov[1], **Madalina Croitoru**[4] **and Nicholas R. Jennings**[1]
[1] University of Southampton, UK   [2] University of Oxford, UK
[3] University of Warsaw, Poland   [4] University of Montpelier, France

Abstract

We introduce a new representation scheme for coalitional games, called *coalition-flow networks* (CF-NETs), where the formation of effective coalitions *in a task-based setting* is reduced to the problem of directing flow through a network. We show that our representation is intuitive, fully expressive, and captures certain patterns in a significantly more concise manner compared to the conventional approach. Furthermore, our representation has the flexibility to express various classes of games, such as characteristic function games, coalitional games with overlapping coalitions, and coalitional games with agent types. As such, to the best of our knowledge, CF-NETs is the first representation that allows for switching conveniently and efficiently between overlapping/non-overlapping coalitions, with/without agent types. We demonstrate the efficiency of our scheme on the *coalition structure generation* problem, where near-optimal solutions for large instances can be found in a matter of seconds.

1 Introduction

Cooperation is a central concept in artificial intelligence and at the heart of multi-agent systems design. Coalitional game theory is the standard framework to model cooperation; it provides theoretical constructs that allow agents to take joint actions as primitives [Ieong and Shoham, 2006]. One of the most important research questions in this domain is the *coalition structure generation problem* (CSG), i.e., how can the set of agents be partitioned into groups (coalitions) such that social welfare is maximized? This problem has attracted considerable attention in the recent AI literature [Bachrach and Rosenschein, 2008; Michalak *et al.*, 2010; Rahwan *et al.*, 2011], as it has important applications and inspires new theoretical and computational challenges. Existing work on coalition structure generation focuses on *characteristic function games* (CFGs), i.e., games in which the outcome obtained by a coalition is not influenced by agents outside the coalition.[1]

---

[1]Recently, a coalition structure generation algorithm has been proposed in games with *"externalities"* (i.e., possible influences be-

The prevailing assumption in CSG algorithms is that coalition structures are disjoint. However, there exist many realistic scenarios in which this assumption does not hold [Chalkiadakis *et al.*, 2010]. For example, in multi-sensor systems, where sensors coordinate their movements to monitor particular targets of interest, the performance can be improved by allowing the sensors to form overlapping coalitions [Dang *et al.*, 2006].

A second prevailing assumption in existing CSG algorithms is that every agent is unique, i.e., such works do not consider situations where some agents are identical. Again, the ability to differentiate between identical and non-identical agents can often improve performance. For example, in disaster response scenarios, one should distinguish between a fire brigade and an ambulance (as they are aimed for different activities), but may be indifferent between two ambulances with equal capabilities. The theory of coalition structure generation has been extended only recently to such domains [Aziz and de Keijzer, 2011; Ueda *et al.*, 2011].

The standard CFG representation explicitly lists the values of all the possible coalitions and requires space that is exponential in the number of agents. However, in many classes of games, there exists additional structure (such as identical agents) that can be exploited, and well-crafted representation schemes can significantly improve performance [Deng and Papadimitriou, 1994; Ieong and Shoham, 2006; Elkind *et al.*, 2009; Aadithya *et al.*, 2011]. However, the majority of works in this line of research focused on computing solution concepts, rather than solving the coalition structure generation problem.

To date, with very few exceptions (e.g., [Shehory and Kraus, 1998; Dang and Jennings, 2006]) the incorporation of tasks is absent from the research on coalition structure generation. The advantage of incorporating tasks is that it allows for capturing a much wider range of applications. In particular, many real-world applications involve a set of tasks to be achieved, where the agents need to form coalitions since (some of) those tasks cannot be performed by an individual agent, or can be achieved more efficiently by a group of agents (compared to an individual). However, the disadvantage of incorporating tasks is that it makes the optimization problems (such as coalition structure generation) significantly harder to solve. This is because the value of a coalition does

---

tween co-existing coalitions) [Rahwan *et al.*, 2012].

324

not depend solely on the identities of its members; it depends on the tasks being achieved by the coalition. As such, a coalition does not necessarily have a single value. Instead, it can have a different value for every task.

Our Contribution

We formulate a unified approach for modelling different types of coalitional games *in task-based settings*. In particular, we formulate the coalition structure generation problem as a network flow problem—a problem widely studied in the combinatorial optimization literature, with many important applications in the real world [Schrijver, 2003]. More specifically:

- We propose a new representation of coalitional games, namely *coalition-flow networks* (CF-NETs), where the coalition formation process is represented as the process of directing the flow through a network with edge-capacity constraints. We show that CF-NETs are an appropriate representation for several important classes of coalitional games, such as conventional (non-overlapping) coalitional games, overlapping coalitional games, and coalitional games with agent types. Importantly, to the best of our knowledge, this is the first representation with which one can easily and efficiently switch between cases with overlapping/non-overlapping coalitions, with/without agent types. In addition, we show how CF-NETs allow for the succinct storage of certain, potentially useful patterns.

- We show that under the CF-NET representation, the coalition structure generation problem in a task-based setting can be reformulated as a mathematical program related to the well-known production-transportation problem [Tuy *et al.*, 1996; Holmberg and Tuy, 1999; Hochbaum and Hong, 1996].

- We provide an anytime approximation technique for the CSG problem, which provides worst-case guarantees on the quality of the solutions produced. Using numerical simulations, we show that our algorithm scales to thousands of agents. For example, for $n = 5000$ agents, it finds solutions within 75-85% of the optimum in a matter of seconds.

Related Work

Researchers have attempted to circumvent the intractability of the characteristic function form by developing alternative representations, which are more compact and allow efficient computations. These representations can be divided into two main categories:

1. The representation is guaranteed to be succinct, but is not fully expressive (i.e., it cannot represent any arbitrary characteristic function game) [Deng and Papadimitriou, 1994; Wooldridge and Dunne, 2006].

2. The representation is fully expressive, but is only succinct for some problem instances [Ieong and Shoham, 2006; Elkind *et al.*, 2009; Conitzer and Sandholm, 2004b]). Our CF-NET representation scheme falls in this category.

Most previous work focuses on the computation of solution concepts, such as the Shapley value [Deng and Papadimitriou, 1994; Ieong and Shoham, 2006; Elkind *et al.*, 2009; Conitzer and Sandholm, 2004b], the core [Conitzer and Sandholm, 2004a; 2004b], the bargaining set, and the kernel [Greco *et al.*, 2009]. On the other hand, the coalition structure generation problem has recently attracted attention, with most of the studies focusing on characteristic function games [Rahwan *et al.*, 2009].

Among the very few works that consider overlapping coalitions, we mention the works by Shehory and Kraus [1996] and Dang *et al.* [2006]. The former only proposes a greedy algorithm for a subclass of these games, whereas the latter makes heavy use of several strong assumptions placed on the characteristic function. Recently, Chalkiadakis *et al.* [2010] introduced a formal model of overlapping coalition formation, in which an agent can participate in multiple coalitions simultaneously, by contributing a portion of its resources to each coalition containing it.
Our CF-NET representation is designed for the purpose of solving the coalition structure generation problem efficiently. The only other representation designed with this problem in mind is due to Ohta *et al.* [2009]. However, unlike our representation, theirs does not incorporate tasks, and does not consider games with agent types or overlapping coalitions.

The basic idea of applying network flows to model coalitional games has been examined before, for example by Kalai and Zemel [1982] and by Bachrach and Rosenschein [2009]. However, the fundamental difference is that, in those papers, the units of flow can be thought of as utility units (so the solution to a network flow problem influences the value of a coalition). In our representation, what flows in the network is "agents", so solving a flow problem returns a coalition structure. This is precisely what gives flexibility to our framework, allowing for overlapping coalitions, or identical agents, to be considered whenever needed.

2 Preliminaries

In this section, we provide the standard definitions for the classes of coalitional games studied in this work.

We define a task-based *characteristic function game*, TCFG, as a coalitional game given by a tuple $\langle A, K, v \rangle$, where

- $A = \{a_1, \ldots, a_n\}$ is a set of agents;
- $K = \{k_1, \ldots, k_q\}$ is a set of tasks;
- $v : 2^A \times K \to \mathbb{R}$ is a function that assigns a value to every pair (C, k_i), where C is a coalition and k_i is a task.

In this paper, we assume that every performed task is performed by exactly one coalition. The possibility of having a group of agents that does not perform any tasks can easily be incorporated by adding a single dummy task. On the other hand, the possibility of having a single coalition perform multiple tasks can be incorporated by allowing overlapping coalitions (in which case every performed task is performed by a single, not necessarily unique, coalition).

An implicit assumption is that every agent can participate in exactly one coalition. When an agent is allowed to belong to multiple coalitions simultaneously, the tuple $\langle A, v \rangle$

defines a task-based *coalitional game with overlapping coalitions*, denoted by TCFG^o. In such a game, each agent can be a member of up to 2^{n-1} coalitions. In reality, this number is reasonable only if coordination costs are low and the number of available resources is sufficiently large.

Thus, the definition can naturally be extended to a task-based *resource-constrained coalitional game with overlapping coalitions* (TCFG^{rco}). In such a game, each agent $a_i \in A$ has an upper bound on the number of coalitions it can join, which can be interpreted as a a limit on the resources that can be employed by an agent to perform tasks. In this setting, a game is formally defined by a tuple $\langle A, r, v \rangle$, where A and v are defined as above, and the function $r : A \rightarrow \{1, \ldots, 2^{n-1}\}$ assigns to every agent its resource constraints.[2] Note that the TCFG^o and TCFG representations are special cases of TCFG^{rco}, where the number of coalitions that an agent can join is maximal (2^{n-1}) and minimal (1), respectively.

Next, we formalize task-based coalitional games with agent types. Let A and v be defined as above and $T = \{t_1, \ldots, t_m : m \leq n\}$ a set of types, such that each agent in A is associated with one type in T. Let $t : A \rightarrow T$ be a function that returns the type of any given agent. An important situation to model in such games is that of two agents with the same type bringing the same marginal contribution to any coalition they belong to. That is, given a type $t_k \in T$, for any two agents a_i, a_j of type t_k and any coalition $C \subseteq A \backslash \{a_i, a_j\}$, the following holds: $v(C \cup \{a_i\}) - v(C) = v(C \cup \{a_j\}) - v(C)$.

Recall that the conventional TCFG representation can be used to represent (albeit not concisely) the agent-type setting. Clearly, however, the game can be represented more concisely by considering the values of coalitions as a function of the types (rather than the identities of the agents). To this end, we define coalitional games with agent types.

Definition 1 *A task-based* coalitional game with agent types (TCFG^{at}) *is a tuple* $\langle A, T, t, \mathcal{T}, v^{at} \rangle$, *where:*

- $A = \{a_1, \ldots, a_n\}$ *is a set of agents;*
- $T = \{t_1, \ldots, t_{m \leq n}\}$ *is a set of agent types;*
- $t : A \rightarrow T$ *is a function that returns the type of any agent, $a_i \in A$;*
- $\mathcal{T} = \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_m$ *is a multi-set of agent types, such that* $\mathcal{T}_k = \{t_k, \cdots, t_k\}$, *where* $|\mathcal{T}_k| = |\{a_i \in A : t(a_i) = t_k\}|, \forall k \in \{1, \cdots, m\};$
- $K = \{k_1, \ldots, k_q\}$ *is a set of tasks;*
- $v^{at} : 2^{\mathcal{T}} \times 2^K \rightarrow \mathbb{R}$ *is a characteristic function of agent types.*

3 The CF-NET Representation

Next, we introduce the framework of *coalition-flow networks* (CF-NETs) to represent the classes of coalitional games discussed above.

Definition 2 *A* coalition-flow network *(CF-NET) is a tuple* $\langle N, E, \mathcal{X}, \mathcal{Y}, \mathcal{Z} \rangle$, *where (1)* (N, E) *is an acyclic digraph with a set of nodes N and a set of directed edges E, and (2) $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ are sets of constraints. In particular:*

1. **The set of nodes**, N, *is the union of the following disjoint sets:*
   - $\{S, G\}$ *contains the* source *node S from which the "flow" is pushed into the network and the* sink *node G towards which the flow needs to be directed.*
   - N^a *is the set of* agent *nodes: each agent (or type) is represented by exactly one such node. As such,* $|N^a| = n$.
   - N^k *is the set of* task *nodes: each task is represented by exactly one such node. Thus, $|N^k| = q$.*

2. **The set of edges** *is $E = (\{S\} \times N^a) \cup E' \cup (N^k \times \{G\})$, where $E' \subseteq \{(n_i, n_j) : n_i \in N^a, n_j \in N^k\}$.*

3. **The set of constraints** *is $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$, where $\mathcal{X} = \{X_i : i = 1 \ldots n\}$, $\mathcal{Y} = \{Y_{ij} : i = 1 \ldots n, j = 1 \ldots q\}$, and $\mathcal{Z} = \{Z_j : j = 1 \ldots q\}$, which restrict the possible values of the flow through the edges in $\{S\} \times N^a$, E' and $N^k \times \{G\}$, respectively. In particular:*
   - X_i *represents the permitted multiplicities of agent a_i (or the permitted numbers of agents whose type is t_i) in the game.*
   - Y_{ij} *represents the permitted multiplicities of agent a_i (or permitted numbers of agents of type t_i) that will perform task k_j.*
   - Z_j *represents the sizes of the coalition that will perform task k_j.*

An illustration of the CF-NET representation can be found Figure 1). Now, based on the network structure (N, E) defined above, and the constraints $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ imposed on the flow through the edges, we introduce next the notion of *CF-flow*. Intuitively, the CF-flow depicts the coalition formation process. Analogously to the flow in the network, the process of directing the flow from the source node to the sink node (through the agent/type nodes and the task nodes) can be interpreted as the process of determining which coalition should perform each task.

Definition 3 *A* coalition formation flow *(CF-flow) in a CF-NET* $\langle N, E, \mathcal{X}, \mathcal{Y}, \mathcal{Z} \rangle$ *is a function $f : E \rightarrow \mathbb{R}$ with the following properties:*

1. $f(S, n_i) \in \mathcal{X}, \forall n_i \in N^a$.
2. $f(n_i, n_j) \in \mathcal{Y}, \forall (n_i, n_j) \in E'$.
3. $f(n_j, G) \in \mathcal{Z}, \forall n_j \in N^k$.
4. $\displaystyle\sum_{(n_i, n_j) \in E} f(n_i, n_j) = \sum_{(n_j, n_k) \in E} f(n_j, n_k),$
   $\forall n_j \in N^a \cup N^k$.

Let (x, y, z) be an *instantiation* of a CF-flow f in a CF-NET $\langle N, E, \mathcal{X}, \mathcal{Y}, \mathcal{Z} \rangle$, such that $x_i = f(S, n_i)$ for $n_i \in N^a$, $y_{ij} = f(n_i, n_j)$ for $(n_i, n_j) \in E'$, and $z_j = f(n_j, G)$ for $n_j \in N^k$. Then, from Definition 3, we have:

$$x_i = \sum_{j=1}^m y_{ij}, \; \forall \, i = 1, \ldots, n; \;\; z_j = \sum_{i=1}^n y_{ij}, \; \forall \, j = 1, \ldots, m; \quad (1)$$

$$x_i \in \mathcal{X}_i, \; y_{ij} \in Y_{ij}, \; z_j \in Z_j, \; \forall \, i = 1, \ldots, n, \forall \, j = 1, \ldots, m.$$
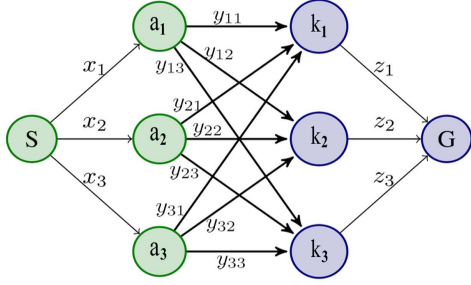
326

Figure 1: Sample CF-NET representation

Given a CF-flow, we define next the values of the coalitions constructed by it. To this end, we equip a CF-NET with three valuation functions as follows:

Definition 4 *Given a CF-NET* $\langle N, E, \mathcal{X}, \mathcal{Y}, \mathcal{Z} \rangle$*, let* c, d, g *be the* valuation functions *defined, respectively, on* $\{S\} \times N^a$*,* E' *and* $N^k \times \{G\}$*, such that:*

- *For all* $(S, n_i) \in \{S\} \times N^a$*,* $c_i = c(S, n_i)$ *represents the* value *of a* singleton *coalition of agent* a_i *(or an agent of type* t_i*).*
- *For all* $(n_i, n_j) \in E'$*,* $d_{ij} = d(n_i, n_j)$ *is the* contribution *of agent* a_i *(or an agent of type* t_i*) to task* k_j*.*
- *For all* $(n_j, G) \in N^k \times \{G\}$*,* $g_j(z_j) = g(n_j, G)$ *is a* synergy *function, where* $z_j \in Z_j$ *is the size of the coalition performing task* k_j*.*[3]

That is, the value *of a coalition* C *performing task* k_j *is given as:*

$$v(C, k_j) = \sum_{a_i \in C} d_{ij} y_{ij} + g_j(|C|). \tag{2}$$

The first advantage of CF-NETs is their ability to represent any of the four classes of games discussed above, (i.e. TCFG, TCFG^o, TCFG^{rco}, and TCFG^{at}). This can be done simply by setting the appropriate $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$. In more detail,

- In TCFG, $\forall a_i \in A, X_i = \{1\}$, $\forall a_i \in A, \forall k_j \in K, Y_{ij} = \{0, 1\}$, and $\forall k_j \in K, Z_j = \{0, \ldots, n\}$;
- In TCFG^o, $\forall a_i \in A, X_i = \{1, \ldots, 2^{n-1}\}$, $\forall a_i \in A, \forall k_j \in K, Y_{ij} = \{0, 1\}$, and $\forall k_j \in K, Z_j = \{0, \ldots, n\}$;
- In TCFG^{rco}, $\forall a_i \in A, X_i = \{1, \ldots, r(a_i)\}$, $\forall a_i \in A, \forall k_j \in K, Y_{ij} = \{0, 1\}$, and $\forall k_j \in K, Z_j = \{\{0, \ldots, n\}\}$;
- In TCFG^{at}, $\forall a_i \in A, X_i = \{|\mathcal{T}_i|\}$, $\forall a_i \in A, \forall k_j \in K, Y_{ij} = \{0, \ldots, |\mathcal{T}_i|\}$, and $\forall k_j \in K, Z_j = \{0, \ldots, m\}$.

Next, we discuss expressiveness.

Proposition 1 *Every coalitional game that can be modelled as a TCFG, TCFG^o, TCFG^{rco} and/or TCFG^{at}, can also be represented as a CF-NET.*

---
[3]Throughout the paper this is assumed that $g_j(0) = 0$.

Proof : We demonstrate that, for any arbitrary coalitional game under consideration, there exists a CF-NET representation that uniquely defines this game. Specifically:

1. TCFG: Our aim is to construct CF-NET representing an arbitrary game $\langle A, K, v \rangle$. First, we create a one-to-one function $I : 2^A \times K \to \{1, \ldots, 2^n \times q\}$. Now, for every coalition $C \in 2^A$ and task $k_j \in K$, we create a hypothetical task $w_{I(C,k_j)}$ which is connected to all the agents in C and none of the agents in $A \backslash C$. Furthermore, for every $a_i \in C$, we set $X_i = \{1\}$, $Y_{i,I(C,k_j)} = \{0, 1\}$, and $d_{i,I(C,k_j)} = 0$. We also set $Z_{I(C,k_j)} = \{0, |C|\}$, and set $g_{I(C,k_j)}(|C|) = v(C, k_j)$ and $g_{I(C,k_j)}(0) = 0$.

2. TCFG^o/TCFG^{rco}: This case is similar to the previous one. The difference is *only* in the definition of X_i, which is now given by $X_i = \{1, \ldots, 2^{n-1}\}$ in the case of TCFG^o, or by $X_i = \{1, \ldots, r(a_i)\}$ in the case of TCFG^{rco}.

3. TCFG^{at}: Here, agent nodes depict available types of agents. That is, every node $n_i \in N^a$ represents a type t_i. This case is similar to TCFG case, except that we now set $X_i = \{|\mathcal{T}_i|\}$ and $Y_{i,I(C,k_j)} = \{0, |C \cap \mathcal{T}_i|\}$.

$\square$

The constructs in the proof of Proposition 1 imply that CF-NETs are no less concise than the corresponding TCFG, TCFG^o/TCFG^{rco} and TCFG^{at} representations. Indeed, the edges do not need to be explicitly represented; for every hypothetical task w_x it is possible to identify to it simply by using the inverse of function I (which in turn can be concisely represented).

Observe that, for certain patterns often encountered in coalitional games, CF-NETs provide much more concise representations. For instance, suppose there exist additional requirements for coalitions to be formed, such as certain agents being incompatible with each other, or constraints on the coalition sizes. In these cases, one can simply exclude "infeasible" coalitions from the set of coalition nodes. This only decreases the size of the representation, which for certain game classes makes CF-NETs exponentially more concise than the corresponding characteristic function representations.

Next we prove our main technical result, demonstrating the computational power of the CF-NET representation in the coalition structure generation problem.

4 Coalition Structure Generation in CF-NETs

In this section, we formally define the coalition structure generation (CSG) problem and propose an approximation method for solving it on CF-NETs. Our technique utilizes the advantages of the CF-NET representation to produce anytime solutions and estimate their quality.

The CSG Problem

First, we make explicit the notion of a coalition structure for each of the classes TCFG, TCFG^o, TCFG^{rco} and TCFG^{at}.

1. TCFG: A coalition structure, π, is a *partition* of the agents:

$$\pi = \left\{ C : C \subseteq A, \ \bigcup_{C \in \pi} C = A, \ \forall C, C' \in \pi : C \cap C' = \emptyset \right\}$$

327

2. TCFG°: Similar to TCFG, except that coalitions can now overlap. A coalition structure, π^o, is defined as:

$$\pi^o = \left\{ C : C \subseteq A, \bigcup_{C \in \pi} C = A \right\}$$

3. TCFG^{rco}: In this case, each agent $a_i \in A$ can belong to at most $r(a_i)$ coalitions simultaneously:

$$\pi^{rco} = \left\{ C : C \subseteq A, \bigcup_{C \in \pi} C = A, \forall a_i \in A : |\{C \in \pi : a_i \in C\}| \leq r(a_i) \right\}$$

4. TCFG^{at}: Coalitions are multi-sets of types, rather than sets of agents as in previous cases. Let $m_y(x)$ denote the multiplicity of element x in multi-set y; then:

$$\pi^{at} = \left\{ C : C \subseteq \mathcal{T}, \bigcup_{C \in \pi} C = \mathcal{T}, \forall t_i \in \mathcal{T} : \sum_{C \in \pi} m_C(t_i) = m_{\mathcal{T}}(t_i) \right\}$$

Let Π^x denote the sets of all possible coalition structures in TCFG^x, where index x is either empty or stands for "o", "rco" or "at". The CSG problem in TCFG^x is to find an optimal coalition structure, π^x_{OPT}, that maximizes the sum of coalition values:

$$\pi^x_{OPT} \in \arg \max_{\pi=\{C_1,\ldots,C_{|\pi|}\}\in\Pi^x, K^\pi=\{k_1,\ldots,k_{|\pi|}\}\subseteq K} \sum_{C_i \in \pi} v^x(C_i, k_i),$$

where $v^x = v^{at}$ if TCFG^{at}, and $v^i = v$, otherwise.

Solving the CSG Problem in CF-NETs

We now formalize the CSG problem in terms of CF-NETs. Here, since the value of a coalition performing a task is defined by (2), the value of a coalition structure is:

$$\underbrace{\sum_{i=1}^n \sum_{j=1}^q d_{ij} y_{ij}}_{\text{individual marginal contributions}} + \underbrace{\sum_{j=1}^q g_j(z_j)}_{\text{synergy values}} .$$

For ease of exposition, we focus next on the non-overlapping model. However, we emphasize the fact that our method can be easily extended to games with overlapping coalitions or with agent types (by defining X_i, Y_{ij} and Z_j accordingly).

For all i, j, we have $X_i = \{1\}$, $Y_{ij} = \{0, 1\}$ and $Z_j = \{0, \ldots, n\}$. The coalition structure generation problem can then be formulated as a mathematical program:

$$
\begin{aligned}
CSG \quad := \quad & \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \sum_{i=1}^n \sum_{j=1}^q d_{ij} y_{ij} + \sum_{j=1}^q g_j(z_j) \\
s.t. \quad & z_j = \sum_{i=1}^n y_{ij}, \qquad \forall j = 1, \ldots, q, \quad (3) \\
& x_i = \sum_{j=1}^q y_{ij}, \qquad \forall i = 1, \ldots, n, \quad (4) \\
& \boldsymbol{x} = \{1\}^n, \boldsymbol{y} \in \{0,1\}^{n \times q}, \boldsymbol{z} \in \{0, \ldots, n\}^q. \quad (5)
\end{aligned}
$$

Note that if the synergy functions g_j are linear, the problem is equivalent to the classical maximum network flow problem and can be solved efficiently [Cormen *et al.*, 2001]. However, for non-linear synergies, CSG becomes a non-linear integer

programming problem, which is generally NP-hard [Sandholm *et al.*, 1999]. In this paper, we solve/approximate CSG for general synergy functions.

First, observe that our problem is related to the *production-transportation problem* from Operations Research [Tuy *et al.*, 1996; Holmberg and Tuy, 1999], which also has a network flow interpretation. Here, the non-linear terms $g_j(z_j)$ can be viewed as the production part that the decision makers have to decide upon. Once the production $\boldsymbol{z}$ has been fixed, the problem becomes a standard transportation problem – of finding $(\boldsymbol{x}, \boldsymbol{y})$ – and can be solved efficiently. However, solving for optimal $(\boldsymbol{x}, \boldsymbol{y})$ and $\boldsymbol{z}$ simultaneously is nontrivial [Hochbaum and Hong, 1996].

Therefore, in this work we develop an approximation technique for CSG. Our method has two advantages: it gives an anytime algorithm and it provides upper and lower bounds on the optimal value; as a result, one can quantify how far the solution provided is from the optimal solution. Specifically, we use constraint relaxation and duality to modify the mathematical program as follows.

Relax the constraint (3) $z_i = \sum_{i=1}^n y_{ij}$ in CSG and consider the corresponding dual problem:

$$
\begin{aligned}
\min_{\boldsymbol{\lambda}} \quad & \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \sum_{i=1}^n \sum_{j=1}^q d_{ij} y_{ij} + \sum_{j=1}^q g_j(z_j) \\
& - \sum_{j=1}^q \lambda_j(z_j - \sum_{i=1}^n y_{ij}) \\
s.t. \quad & (4), (5)
\end{aligned}
$$

Now, for each fixed $\boldsymbol{\lambda}$, consider the inner problem:

$$
\begin{aligned}
h(\boldsymbol{\lambda}) \quad = \quad & \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \sum_{i=1}^n \sum_{j=1}^q (d_{ij} + \lambda_j) y_{ij} \\
& + \sum_{j=1}^q (g_j(z_j) - \lambda_j z_j) \\
s.t. \quad & (4), (5)
\end{aligned}
$$

By replacing every x_i with $\sum_{j=1}^q y_{ij}$ and noticing that $h(\boldsymbol{\lambda})$ is now separable in $\boldsymbol{y}$ and $\boldsymbol{z}$, we can show that $h(\boldsymbol{\lambda}) = \sum_{i=1}^n h_{1i}(\boldsymbol{\lambda}) + \sum_{j=1}^q h_{2j}(\lambda_j)$, where:

$$
\begin{aligned}
h_{1i}(\boldsymbol{\lambda}) \quad = \quad & \max_{\boldsymbol{y}_{i.}} \sum_{i=1}^n \sum_{j=1}^q (d_{ij} + \lambda_j) y_{ij} \\
s.t. \quad & \sum_{j=1}^q y_{ij} \leq 1, \\
& \boldsymbol{y}_{i.} \in \{0,1\}^q, \quad \text{and} \\
h_{2j}(\lambda_j) \quad = \quad & \max_{z_j \in \{0,\ldots,n\}} g_j(z_j) - \lambda_j z_j.
\end{aligned}
$$

Finally, by simplifying $h_{1i}(\boldsymbol{\lambda})$ to $h_{1i}(\boldsymbol{\lambda}) = \max\{0, \max_j(d_{ij} + \lambda_j)\}$, we get a *one variable* integer programming problem, which can be easily solved.

Thus, for each fixed $\boldsymbol{\lambda}$, we can compute $h(\boldsymbol{\lambda})$ very efficiently. Notice that $\min_{\boldsymbol{\lambda}} h(\boldsymbol{\lambda})$ provides an upper bound on CSG (duality theory) and hence any choice of $\boldsymbol{\lambda}$ gives

an upper bound (but we are interested in the smallest one, $\min_{\boldsymbol{\lambda}} h(\boldsymbol{\lambda})$). Notice also that $h(\boldsymbol{\lambda})$ is a piece-wise linear function on $\boldsymbol{\lambda}$ with possible jumps. We then can solve the problem $\min_{\boldsymbol{\lambda}} h(\boldsymbol{\lambda})$ using a sub-gradient based method for updating $\boldsymbol{\lambda}$, i.e., we reduce λ_j if $(z_j - \sum_{i=1}^{n} y_{ij}) < 0$ and increase λ_j otherwise. Although we might not be able to solve $\min_{\boldsymbol{\lambda}} h(\boldsymbol{\lambda})$ to optimality, a sub-optimal solution still provides us with an upper bound on CSG.

Finally, lower bounds are obtained. First, for each $\boldsymbol{\lambda}$, the inner problem can be solved to find its optimal solution on $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. This solution is a feasible solution to CSG and hence its objective value provides a lower bound. Another method for producing a lower bound is to fix the optimal $\boldsymbol{z}$ found in the inner problem and then solve the transportation problem to find the corresponding optimal $(\boldsymbol{x}, \boldsymbol{y})$. This will give another feasible solution and hence, a new lower bound.[4]

5 Performance Evaluation

We perform numerical tests on the algorithm for various settings with the number of agents n varying between 100 and 5000 and the number of tasks q varying between 50 and 200. For each combination of (n, q), we generate 100 random samples using random seeds between 1 and 100. In total, we have tested the algorithm with 2000 random instances. On each instance, the parameters $\boldsymbol{c}$ and $\boldsymbol{d}$ are generated uniformly, i.e., $d_{ij} \sim U[0, 1]$. The synergy function $g_j(z_j)$ is also a random discrete function of the following form:

$$g_j(k) = \epsilon_{j1} + \epsilon_{j2} + \ldots + \epsilon_{jk}, \; \forall k = 1 \ldots n, \; \forall j = 1 \ldots q,$$

where $\epsilon_{ji} \sim U[0, 1]$ are uniform random variables. This means the synergy function g_j is the sum of uniform random variables and the coalition value increase by ϵ_{js} when the coalition size increases from $s - 1$ to s. By creating 100 random instances, we can test the robustness of the algorithm when input data varies[5].

Figure 2 shows the performance of the algorithm when the number of agents varies between 100 and 5000, while the number of tasks is fixed at 100. Sub-figure (A) shows the total computational time, sub-figure (B) shows the number of iteration, while sub-figure (C) shows the optimality bound between the feasible coalition structure found and the worst upper bound. We can see a linear trend in the computational time from sub-figure (A) with less than three minutes[6] to solve the largest and the worst instance (among 600 random instances for this case). The linear trend in the computational time can be explained by the fact that the number of arithmetic operations in each iteration grows linearly, while the number of iterations (shown in sub-figure (B)) does not change much.

Sub-figure (C) shows the guaranteed bound between the feasible coalition structure found and the optimality. Notice that these bounds are guaranteed despite the fact that we don't know the optimal coalition structures thanks to the availability of the upper bounds derived by the algorithm. This also means that the actual optimality bounds could be higher than the average optimality bounds between 75-81% that appear in sub-figure (C).

Figure 3 shows the performance of the algorithm when the number of agents varies is fixed at $n = 2000$, while the number of tasks varies between 50 and 200. We can see a very similar linear trend in the computational time in sub-figure (A) and the optimality bounds between 75-81% in sub-figure (C). The total computational time for the largest instance is less than 35 seconds.

Figures 4, 5 show the same set of statistics as in Figures 2, 3 except that the CFG$^{\mathrm{rco}}$ games now allow each players to join up to 5 coalitions. The guaranteed optimality bounds vary between 76-85% on these instances. We can also see a linear trend in the computational time as the number of players and the number of tasks increase and the algorithm takes less than 90 seconds for the worst instance.

6 Conclusions

We introduced CF-NETs, a representation scheme for coalitional games in task-based settings, which is inspired by network flows. We examined its qualities with respect to conventional coalitional games with non-overlapping coalitions, (resource-constrained) overlapping coalitional games, and coalitional games with agent types. We utilized the advantages of this representation to develop an approximation technique for coalition structure generation, which applies to all these game classes and allows to effectively solve large instances of the problem.

Our work can be extended in several ways. It would be interesting to extend the CF-NET framework with components that would allow to capture game patterns other than those considered in this paper. Furthermore, we are keen on testing the properties of the CF-NET representation with respect to computing different solution concepts such as the Shapley value and the core.

---

[4]With the obtained bounds, it is possible to extend the algorithm in the future by incorporating branch-and-bound techniques to improve solution quality.

[5]For each pair of (n, q), we will present box plots that show the statistics among 100 random instances generated with the middle red horizontal lines showing the medians, the boxes showing the 25 and 75 percentiles, and the red crosses showing the outliers.

[6]All the numerical tests appear in this manuscript are performed on a personal computer, Intel® Xeon® CPU W3520 @2.67GHz with 12GB RAM and under Windows 7 operation system. The code was written and tested on Matlab R2012a.

Figure 2: Finding near-optimal coalition structures in CFG games, given different numbers of *agents*.

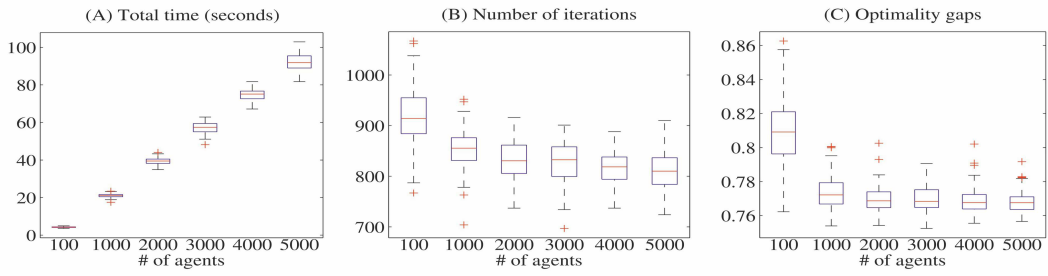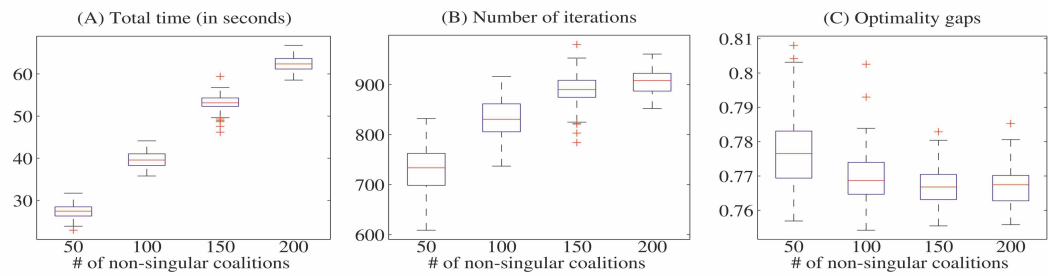Figure 3: Finding near-optimal coalition structures in CFG games, given different numbers of *tasks*.
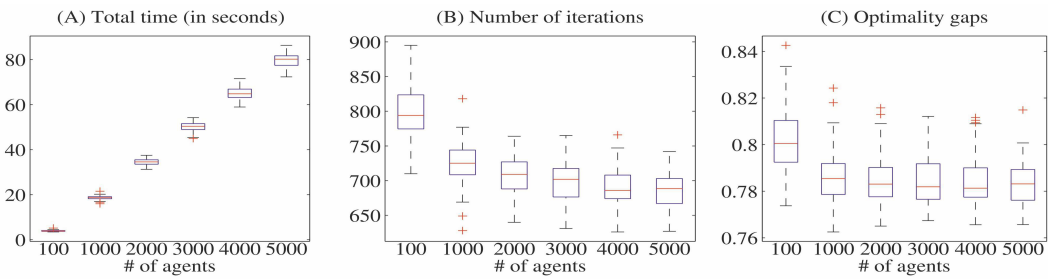

Figure 4: Finding near-optimal coalition structures in CFG^{rco} games, given different numbers of *agents*.
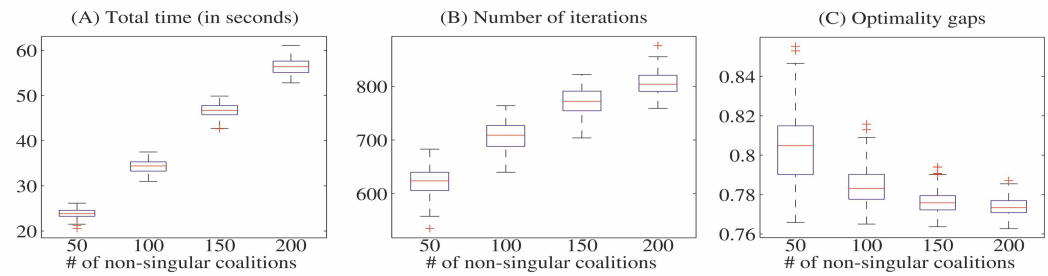

Figure 5: Finding near-optimal coalition structures in CFG^{rco} games, given different numbers of *tasks*.

330

89

References

[Aadithya *et al.*, 2011] K. V. Aadithya, T. P. Michalak, and N. R. Jennings. Representation of coalitional games with algebraic decision diagrams. In *Proceedings of AAMAS*, pages 1121–1122, 2011.

[Aziz and de Keijzer, 2011] Haris Aziz and Bart de Keijzer. Complexity of coalition structure generation. In *Proceedings of AAMAS*, pages 191–198, 2011.

[Bachrach and Rosenschein, 2008] Y. Bachrach and J. S. Rosenschein. Coalitional skill games. In *Proceedings of AAMAS*, pages 1023–1030, 2008.

[Bachrach and Rosenschein, 2009] Y. Bachrach and J.S. Rosenschein. Power in threshold network flow games. *AAMAS*, 18(1):106–132, 2009.

[Chalkiadakis *et al.*, 2010] G. Chalkiadakis, E. Elkind, E. Markakis, M. Polukarov, and N. R. Jennings. Cooperative games with overlapping coalitions. *J. of Artif. Int. Res. (JAIR)*, 39(1):179–216, 2010.

[Conitzer and Sandholm, 2004a] V. Conitzer and T. Sandholm. Complexity of Determining Nonemptiness in The Core. In *Proceedings of IJCAI*, pages 219–225, 2004.

[Conitzer and Sandholm, 2004b] V. Conitzer and T. Sandholm. Computing shapley values, manipulating value division schemes and checking core membership in multi-issue domains. In *AAAI*, pages 42–47, 2004.

[Cormen *et al.*, 2001] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms: 26. Maximum Flows*. McGraw-Hill Higher Education, 2nd edition, 2001.

[Dang and Jennings, 2006] Viet Dung Dang and Nicholas R. Jennings. Coalition structure generation in task-based settings. In *ECAI*, pages 210–214, 2006.

[Dang *et al.*, 2006] V. D. Dang, R. K. Dash, A. Rogers, and N. R. Jennings. Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *Proceedings of AAAI*, pages 635–640, 2006.

[Deng and Papadimitriou, 1994] X. Deng and C. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematical OR*, (19):257–266, 1994.

[Elkind *et al.*, 2009] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. A tractable and expressive class of marginal contribution nets and its applications. *Mathematical Logic Quarterly*, 55(4):362 – 376, 2009.

[Greco *et al.*, 2009] G. Greco, E. Malizia, L. Palopoli, and F. Scarcello. On the complexity of compact coalitional games. In *Proceedings of IJCAI*, pages 147–152, 2009.

[Hochbaum and Hong, 1996] D.S. Hochbaum and S.P. Hong. On the complexity of the production-transportation problem. *SIAM J. on Optimization*, 6(1):250–264, 1996.

[Holmberg and Tuy, 1999] K. Holmberg and H. Tuy. A production-transportation problem with stochastic demand and concave production costs. *Mathematical programming*, 85(1):157–179, 1999.

[Ieong and Shoham, 2006] S. Ieong and Y. Shoham. Marginal contribution nets: A complact representation scheme for coalitional games. In *Proceedings of ACM EC*, pages 170–179, 2006.

[Kalai and ZemelL, 1982] E. Kalai and E. ZemelL. On Totally Balanced Games and Games of Flow. *In Mathematics of Operations Research*, 7(3):476–478, 1982.

[Michalak *et al.*, 2010] Tomasz Michalak, Jacek Sroka, Talal Rahwan, Michael Wooldridge, Peter McBurney, and Nicholas R. Jennings. A Distributed Algorithm for Anytime Coalition Structure Generation. In *Proceedings of AAMAS*, pages 1007–1014, 2010.

[Ohta *et al.*, 2009] N. Ohta, V. Conitzer, R. Ichimura, Y. Sakurai, A. Iwasaki, and M. Yokoo. Coalition structure generation utilizing compact characteristic function representations. In *Proceedings of CP (to appear)*, 2009.

[Rahwan *et al.*, 2009] T. Rahwan, S. D. Ramchurn, A. Giovannucci, and N. R. Jennings. An anytime algorithm for optimal coalition structure generation. *Journal of Artificial Intelligence Research (JAIR)*, 34:521–567, 2009.

[Rahwan *et al.*, 2011] Talal Rahwan, Tomasz P. Michalak, and Nicholas R. Jennings. Minimum search to establish worst-case guarantees in coalition structure generation. In *Proceedings of IJCAI*, pages 338–343, 2011.

[Rahwan *et al.*, 2012] Talal Rahwan, Tomasz P. Michalak, Michael Wooldridge, and Nicholas R. Jennings. Anytime coalition structure generation in multi-agent systems with positive or negative externalities. *Artificial Intelligence (AIJ)*, 186:95–122, 2012.

[Sandholm *et al.*, 1999] Tuomas Sandholm, Kate Larson, Martin Andersson, Onn Shehory, and Fernando Tohmé. Coalition structure generation with worst case guarantees. *Artif. Intell.*, 111(1-2):209–238, July 1999.

[Schrijver, 2003] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.

[Shehory and Kraus, 1996] O. Shehory and S. Kraus. Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. In *Proceedings of ICMAS*, pages 330–337, 1996.

[Shehory and Kraus, 1998] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence (AIJ)*, 1(101):165–200, 1998.

[Tuy *et al.*, 1996] H. Tuy, S. Ghannadan, A. Migdalas, and P. Värbrand. A strongly polynomial algorithm for a concave production-transportation problem with a fixed number of nonlinear variables. *Mathematical Programming*, 72(3):229–258, 1996.

[Ueda *et al.*, 2011] Suguru Ueda, Makoto Kitaki, Atsushi Iwasaki, and Makoto Yokoo. Concise characteristic function representations in coalitional games based on agent types. In *Proceedings of IJCAI*, pages 393–399, 2011.

[Wooldridge and Dunne, 2006] M. Wooldridge and P. Dunne. On the computational complexity of coalitional resource games. *Artificial Intelligence (AIJ)*, (170):835–871, 2006.

Information Revelation Strategies in Abstract Argument Frameworks using Graph based Reasoning

Madalina Croitoru[1], Nir Oren[2]

[1] University of Montpellier 2, France, `croitoru@lirmm.fr`

[2] University of Aberdeen, UK, `n.oren@abdn.ac.uk`

Abstract. The exchange of arguments between agents can enable the achievement of otherwise impossible goals, for example through persuading others to act in a certain way. In such a situation, the persuading argument can be seen to have a positive utility. However, arguments can also have a negative utility — uttering the argument could reveal sensitive information, or prevent the information from being used as a bargaining chip in the future. Previous work on arguing with confidential information suggested that a simple tree based search be used to identify which arguments an agent should utter in order to maximise their utility. In this paper, we analyse the problem of which arguments an agent should reveal in more detail. Our framework is constructed on top of a bipolar argument structure, from which we instantiate *bonds* — subsets of arguments that lead to some specific conclusions. While the general problem of identifying the maximal utility arguments is NP-complete, we give a polynomial time algorithm for identifying the maximum utility bond in situations where bond utilities are additive.

1 Introduction

When participating in dialogue, agents exchange arguments in order to achieve some goals (such as convincing others of some fact, obtaining a good price in negotiation, or the like). A core question that arises is what arguments an agent should utter in order to achieve these goals. This dialogue planning problem is, in most cases, computationally challenging, and work on argument strategy [2, 6, 7, 9] has identified heuristics which are used to guide an agent's utterances.

In this paper we consider a scenario where an agent must select some set of arguments to advance while taking into account the cost, or benefit, associated with revealing the arguments. [7] deals with a similar situation, and give the example of a government attempting to convince the public that weapons of mass distraction exist in some country. They assume that doing so will result in a positive utility gain. In order to back up their claims, the government must give some further evidence, and have a choice of arguments they can advance in doing so, ranging from citing claims made by intelligence resources on the ground, to showing spy satellite photographs, to withdrawing the claims. Each of these arguments has an associated utility cost, and the government must therefore

identify the set of arguments which will maximise its utility. In such a situation, it is clear that advancing all arguments is not always utility maximising for an agent, and [7] utilise a one-step lookahead heuristic to maximise utility while limiting computational overhead.

Unlike [7], in this paper we assume that an agent can advance several arguments simultaneously within a dialogue, and must justify these arguments when advancing them. We therefore seek to identify all arguments that an agent must advance at a specific point in time. To do so, we utilise a bipolar argument framework [3] to allow us to deal with both attacks between arguments and argument support.

We solve our problem through a translation of the argument structure into a graph structure, and then utilise graph operations in order to calculate the appropriate set of arguments to advance. Such a translation also allows us to derive an interesting result with regards to the complexity of bonds calculation.

In the next section we introduce the concept of a *bond* — a set of arguments that should be introduced together by an agent. We then examine the problem of computing a maximum utility bond. The paper concludes with a discussion of possible extensions.

2   Bonds

An argument can have several possible justifications. In the context of a dialogue, it is clearly desirable to advance the *maximal utility* justification. Importantly, this justification often does not coincide with the maximal justification in the set theoretic sense, as the utility of the entire set of arguments might be smaller than the utility of a subset of these arguments. A *bond* is then precisely the maximal utility justification for an argument. This is illustrated by the following informal example.

Example 1. A student is asked to justify why they did not hand in their homework on time, and can respond in several ways. First, they could claim they had done the homework, but that their new puppy ate it. Second, they could explain that they were ill. Furthermore, they could blame this illness on either a nasty virus they had picked up, or due to a hangover caused by over exuberance during the weekend. Clearly, providing all these reasons will not engender as much sympathy as simply blaming the virus. The latter therefore forms a maximal utility justification aimed at obtaining the teacher's sympathy, and forms a bond.

Bonds originate through the possibility that multiple lines of argument yield the same result, and that some of these have different utility costs and benefits when compared to others. A bond is made up of the subset of paths that maximise the agent's utility.

We situate bonds within Bipolar argumentation frameworks [1]. Unlike standard Dung argumentation frameworks, bipolar frameworks explicitly consider both support and attack between arguments. We begin by formalising Bipolar frameworks, following which we introduce the notion of a *coalition* [3]. Such
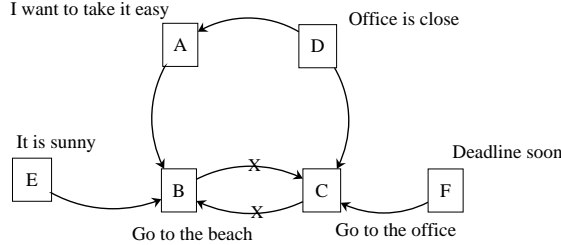
Fig. 1. Bipolar Argumentation System with argument valuations

coalitions can be thought of as the set of all justifications for an argument. Bonds are then a subset of justifications from within a coalition, representing a single line of justifications to the conclusion.

Definition 1. *(Bipolar Argument Framework)* *An abstract bipolar argument framework is a tuple $BAF = (\mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup})$ where $\mathcal{A}$ is a set of arguments; $\mathcal{R}_{def}$ is a binary relation $\subseteq \mathcal{A} \times \mathcal{A}$ called the defeat relation; $\mathcal{R}_{sup}$ is a binary relation $\subseteq \mathcal{A} \times \mathcal{A}$ called the support relation. A bipolar argument framework obeys the constraint that $\mathcal{R}_{def} \cap \mathcal{R}_{sup} = \emptyset$.*

Definition 2. *(Coalitions)* *Given a bipolar argument framework $BAF = (\mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup})$, a coalition is a set of arguments $\mathcal{C} \subseteq \mathcal{A}$ such that all of the following conditions hold.*

1. *The subgraph $(\mathcal{C}, \mathcal{R}_{sup} \cap \mathcal{C} \times \mathcal{C})$ is connected.*
2. *$\mathcal{C}$ is conflict free.*
3. *$\mathcal{C}$ is maximal with respect to set inclusion.*

Definition 3. *(Bonds)* *Given a bipolar argument framework $BAF = (\mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup})$, and $\mathcal{C} \subseteq \mathcal{A}$, a coalition within BAF, a subset $\mathcal{B} \subseteq \mathcal{C}$ is a bond if and only if there is no $a \in \mathcal{C}$ such that for some $b \in \mathcal{B}$, $(b, a) \in \mathcal{R}_{sup}$.*

Example 2. To illustrate these concepts, consider the bipolar argument framework illustrated in Figure 1. Here, arrows with crosses indicate attacks between arguments, while undecorated arrows represent support. Let us also associate utilities with each argument in the system as follows: $u(A) = 11$, $u(B) = 5$, $u(C) = -2$, $u(D) = -8$, $u(E) = 4$ and $u(F) = -10$.

Figure 2 depicts the two coalitions found in this framework, namely $\mathcal{C}_1 = \{D, A, B, E\}$ and $\mathcal{C}_2 = \{A, D, C, F\}$. The utility associated with the latter is 9, and with the former, 12. Now consider the following bond (which is a subset of $\mathcal{C}_1$): $\{A, E, B\}$. Its utility is 20, and in a dialogue, these are the arguments that an agent should advance.

With the definition of bonds in hand, we now turn our attention to how the maximum utility bond — $\{A, E, B\}$ in the previous example — can be computed.
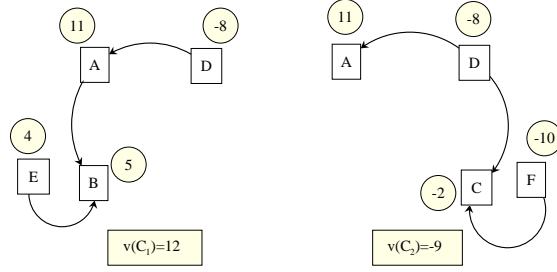
Fig. 2. Coalitions in Argumentation System in Figure 1

3   Identifying Bonds

A naïve approach to computing maximal utility begins with a coalition $\mathcal{C}$ and enumerating its bonds, beginning with arguments which do not support other arguments (these are bonds of cardinality 1), then considering bonds with a single support (i.e. bonds of cardinality 2), and so on. Once all bonds are computed, the maximal utility ones are identified and returned. Clearly, this approach is, in the worst case, exponential in the number of arguments in the domain.

We can construct a polynomial time solution by treating the problem as a maximum flow problem on an appropriate network. The complexity of this type of algorithm is $O(|\mathcal{C}|^3)$, where $|\mathcal{C}|$ is the number of nodes in $\mathcal{C}$ if we apply a push-relabel algorithm [5]. We begin by considering a *induced support graph* by the coalition over the original graph, defined next. The graphs of Figure 2 are examples of such induced support graphs.

Definition 4. *(Induced Support Graph) Let $BAF = (\mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup})$ be an abstract bipolar argumentation framework and $\mathcal{C} \subseteq \mathcal{A}$ a coalition in BAF. We define the graph $G_{\mathcal{C}}^{BAF}$ (the induced support graph by $\mathcal{C}$) as the graph $G_{\mathcal{C}}^{BAF} = (N_{\mathcal{C}}, E_{sup}|_{\mathcal{C}})$ where:*

- *Each node in $N_{\mathcal{C}}$ corresponds to an argument in the coalition $\mathcal{C}$ and*
- *The edges are only the support edges restricted to the nodes in $\mathcal{C}$ (denoted by $E_{sup}|_{\mathcal{C}}$).*

Within such an induced support graph, a bond is a set $N_{\mathcal{B}} \subseteq N_{\mathcal{C}}$, where for each $n \in N_{\mathcal{B}}$, and for each edge $(n,m) \in E_{sup}|_{\mathcal{C}}$, it is also the case that $m \in N_{\mathcal{B}}$. Since we always compute the induced support graph with respect to some underlying bipolar argument framework, we will denote $G_{BAF}^{\mathcal{C}}$ as $G^{\mathcal{C}}$.

Additionally, we denote the utility of an argument corresponding to a node n in the graph as $u(n)$. The utility of a set $\mathcal{B}$ of arguments is defined as $u(N_{\mathcal{B}}) = \sum_{n \in N_{\mathcal{B}}} u(n)$. For convenience, we denote those nodes associated with a positive utility by $N_{\mathcal{C}}^+$, and those with a negative utility by $N_{\mathcal{C}}^-$.
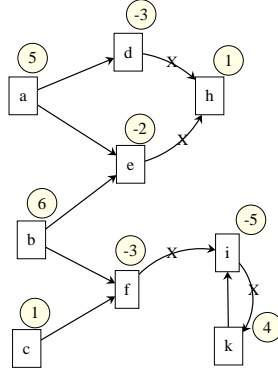
Fig. 3. Bipolar Argumentation System and its valuation

We now show how the problem of finding the maximum utility bond of a coalition can be solved by reducing it to a minimum-cut computation on an extended network $G^{\mathcal{C}}_{extended}$. The idea is to construct this new network such that a minimum cut will correspond to a maximum utility bond. This idea follows an approach used, for example, in the Project Selection Problem [10].

In order to construct $G^{\mathcal{C}}_{extended}$ we add a new source s and a new sink t to the graph $G^{\mathcal{C}}$. For each node $n \in N^+_{\mathcal{C}}$ we add an edge (s, n) with capacity $u(n)$. For each node $m \in N^-_{\mathcal{C}}$ we add an edge (m, t) with capacity $-u(m)$ (thus a positive utility). The rest of capacities (the capacities of edges corresponding to those in $G^{\mathcal{C}}$ are set to ∞.

Example 3. Consider the bipolar argument framework whose graph is shown in Figure 3. The corresponding $G^{\mathcal{C}}_{extended}$ for the coalition $\{a, b, c, d, e, f\}$ is shown in Figure 4. For readability, we have omitted the "∞" label on edges $\{(a, d), (a, e), (b, e), (b, f), (c, f)\}$.

Theorem 1. *If (A', B') is a minimum cut in $G^{\mathcal{C}}_{extended}$ then the set $A = A' - \{s\}$ is a maximum utility bond.*

Proof. The capacity of the cut $(\{s\}, \mathcal{C} \cup \{t\})$ is $C = \sum_{n \in N^+_{\mathcal{C}}} u(n)$. So, the maximum flow value in this network is at most C.

We want to ensure that if (A', B') is a minimum cut in the graph $G^{\mathcal{C}}_{extended}$, then $A = A' - \{s\}$ satisfies the bond property (that is, it contains all of the supported elements). Therefore, if the node $i \in A$ has an edge (i, j) in the graph then we must have $j \in A$. Since the capacities of all the edges coming from the graph $G^{\mathcal{C}}$ have capacity ∞ this means that we cannot cut along such edge (the flow would be ∞).
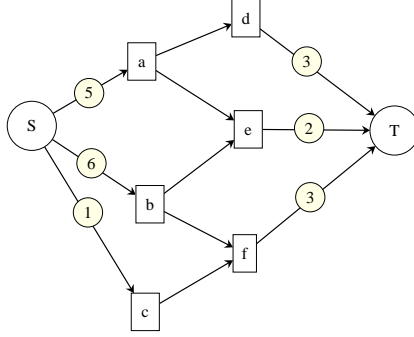
Fig. 4. Extended network of a coalition in a Bipolar Argumentation Framework

Therefore, if we compute a minimum cut (A', B') in $G_{extended}^{\mathcal{C}}$ we have that $A' - \{s\}$ is a bond. We now prove that it is of maximum utility.

Let us consider any bond $\mathcal{B}$. Let $A' = \mathcal{B} \cup \{s\}$ and $B' = (\mathcal{C} - \mathcal{B}) \cup \{t\}$ and consider the cut (A', B').

Since $\mathcal{B}$ is a bond, no edge (i, j) crosses this cut (if not there will be a supported argument not in $\mathcal{B}$). The capacity of the cut (A', B') satisfying the bond support constraints as defined from $\mathcal{C}$ is $c(A', B') = C - \sum_{a \in \mathcal{C}} u(a)$, where $C = \sum_{a \in \mathcal{C}+} u(a)$. We can now prove that the minimum cut in G' determines the bond of maximum utility. The cuts (A', B') of capacity at most C are in a one-to-one correspondence with bonds $A = A' - \{s\}$. The capacity of such a cut is:

$$c(A', B') = C - u(A)$$

The capacity value is a constant, independent of the cut, so the cut with the minimum capacity corresponds to maximum utility bonds.

We have thus proved a polynomial time algorithm for the maximum utility bond decision. While this seems like a strong result, it should be noted that we made use of the fact that our input was a coalition rather than the full argument system. Typically, agents must consider the entire set of arguments and must therefore identify the coalitions themselves. Doing so is an NP-complete problem [3].

To conclude, we discuss the complexity of the two decision problems (bond finding and coalition finding) in an abstract setting, where the input is a bipolar argument framework (as opposed to a compiled knowledge base such as ASPIC+ permits, though this results in an exponential blow-up of the size of the domain).

P1
Input: A bipolar argumentation framework, an utility function and $k \in \mathbf{Z}$.
Question: Is there a coalition with the utility $\geq$ k.

P2
Input: A bipolar argumentation framework, an utility function and $k \in \mathbf{Z}$.
Question: Is there a bond with the utility $\geq$ k.

Both problems are NP-complete.

Proof. (*Sketch*) Clearly, the two problems belongs to NP. To prove the NP-completeness, let us consider an undirected graph G and a utility function defined on its nodes. If the edges of this graph are considered directed (thus obtaining a digraph that will correspond to the attack digraph), and each non-edge in G is replaced by a pair of opposed directed support edges, the coalitions in the Bipolar Argumentation Framework obtained are exactly the maximal stables in the initial graph. Indeed, these sets are conflict-free in the Bipolar Argumentation Framework and clearly connected in the support digraph. Thus, deciding if a given undirected graph with weights on its vertices has a maximal stable set of total weight greater or equal than some threshold can be reduced to **P1** (or **P2**). Since this problem is NP-complete [4] we have that **P1** and **P2** are NP-complete.

4   Discussion and Conclusions

In this paper we have introduced the notion of a maximum utility bond. These bonds are related to the justification of arguments when considered in the context of a bipolar argument framework. Since there can be many argument justifications, one needs to identify a heuristic for computing the "best" justification when arguing. We considered a utility based heuristic in which we have assumed that each argument has associated either a positive or a negative numerical utility. Such utility could correspond to the information revealing cost of the argument, or the degree of confidence the agent has in the argument, etc. Furthermore we have assumed that the utility function is additive. We have then described a polynomial time algorithm for computing maximum utility bonds, assuming that coalitions have already been identified.

In the future, we intend to investigate two significant extensions to the current work. First, we have assumed that utilities are additive. However, this simplification is not — in practice — realistic. The presence, or absence, of combinations of arguments could result in very different utilities, such as in the case where revealing some secret information makes other secret information unimportant to keep hidden. To address this case, we can transform our problem into a multi-agent resource allocation (MARA) problem, where arguments are transformed into resources. We must potentially consider an exponentially large input domain (i.e. all possible combinations of arguments), and in [8] such an exponential input was dealt with in the context of coalition formation. We therefore intend to

apply their techniques, noting that additional constraints must be introduced on the technique's outputs to capture the nature of our domain.

Another potential avenue of future work arises by noting that we have implicitly combined the cost of revealing an argument, and attacks due to the argument on a bond into a single number (the negative utility). In reality, these two values are different; by separating them out, we could perform a minimisation that reflects the different potential preferences of a reasoner.

References

1. L. Amgoud, C. Cayrol, M. C. Lagasquie-Schiex, and P. Livet. On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems*, 23(10):1062–1093, 2008.
2. L. Amgoud and H. Prade. Reaching agreement through argumentation: a possiblistic approach. In *Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning*, pages 175–182, 2004.
3. C. Cayrol and M.-C. Lagasquie-Schiex. Coalitions of arguments: A tool for handling bipolar argumentation frameworks. *International Journal of Intelligent Systems*, 25(1):83–109, 2010.
4. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman and Co, 1979.
5. V. Goldberg and R. Tarjan. A new approach to the maximum flow problem. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 136–146, 1986.
6. A. C. Kakas, N. Maudet, and P. Moraitis. Layered strategies and protocols for argumentation-based agent interaction. In I. Rahwan, P. Moraitis, and C. Reed, editors, *Proceedings of the First International Workshop on Argumentation in Multiagent Systems*, volume 3366 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 66–79, New York, 2004. Springer-Verlag.
7. N. Oren, T. J. Norman, and A. D. Preece. Arguing with confidential information. In *Proc of ECAI 2006*, pages 280–284, 2006.
8. T. Rahwan, T. P. Michalak, M. Croitoru, J. Sroka, and N. R. Jennings. A network flow approach to coalitional games. In *ECAI*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 1017–1018. IOS Press, 2010.
9. T. Rienstra, M. Thimm, and N. Oren. Opponent models with uncertainty for strategic argumentation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, page to appear, 2013.
10. E. Tardos and J. Kleinberg. *Algorithm Design*. Addison-Wesley, 2005.
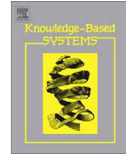
Graphical norms via conceptual graphs

Madalina Croitoru [a], Nir Oren [b,*], Simon Miles [c], Michael Luck [c]

[a] LIRMM, University Montpellier II, France
[b] Department of Computing Science, University of Aberdeen, UK
[c] Department of Informatics, King's College London, UK

ABSTRACT

The specification of acceptable behaviour can be achieved via the use of obligations, permissions and prohibitions, collectively known as *norms*, which identify the states of affairs that should, may, or should not hold. Norms provide the ability to constrain behaviour while preserving individual agent autonomy. While much work has focused on the semantics of norms, the *design* of normative systems, and in particular understanding the impact of norms on a system, has received little attention. Since norms often interact with each other (for example, a permission may temporarily derogate an obligation, or a prohibition and obligation may conflict), understanding the effects of norms and their interactions becomes increasingly difficult as the number of norms increases. Yet this understanding can be critical in facilitating the design and development of effective or efficient systems. In response, this paper addresses the problem of norm explanation for Naïve users by providing of a graphical norm representation that can explicate why a norm is applicable, violated or complied with, and identify the interactions between permissions and other types of norms. We adopt a *conceptual graph* based semantics to provide this graphical representation while maintaining a formal semantics.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Solutions to problems arising in the domain of multi-agent systems have often been inspired by approaches from human societies. Nowhere is this more evident than in addressing the problem of controlling the behaviour of agents within open systems. Here, interactions between agents can cause unexpected system behaviour, and traditional procedural approaches fail due to the unpredictability and complexity of these interactions, as well as the inherent autonomy of the agents involved. In human societies, behavioural control is achieved in a declarative manner, by specifying expectations regarding the behaviour of others, such as with laws or rules. These specifications, or *norms*, identify obligations, permissions and prohibitions that individuals are expected to comply with in particular situations. Drawing on this, there has been much work concerning the application of norms to artificial systems, in which agents are able to make use of concepts such as obligations, permissions, and prohibitions, to represent and reason about socially imposed goals and their execution. Such *norm aware* agents are able to decide whether to act in a manner consistent with norms, or whether to ignore them. In this context, norms are generally imposed on a set of agents in order to increase the overall utility of a system (often at the cost of individual utility) [18], or to reduce computational or communication overhead [4].

While the design and architecture of norm aware agents is critically important, this is not the only problem that must be addressed when utilising norms. Perhaps more interesting (and more challenging) is the problem of design time identification of which norms are needed in order to achieve some desired behaviour. Norms can interact with each other in unpredictable ways, and determining the effects of a norm on a system can thus be difficult. To identify these problematic norm interactions requires us to be able to *explain* the effects of a norm, and why, in some specific situation, it is applicable, violated, complied with, or in some other state, yet this has not been investigated to any real depth. Moreover, the ability to provide such explanations can enable designers to better understand the interactions between different norms, thereby allowing them to avoid introducing redundant norms [3], and to specify norms more precisely. Norm explanations can thus provide vital support for the design a normative system. In addition, from the perspective of users, norm explanation can facilitate a more intuitive appreciation of a system by providing a stronger understanding of the *reasons* why particular norms may have been brought to certain states in response to system events. Such a facility can increase and enhance the trust of a user in relation to operation of the system, providing confidence that it is in fact operating correctly.

Since much of the research into the formal properties of norms has taken place within the area of philosophy and deontic logic

[13,23], norms are typically specified within a knowledge-based system (KBS) using a logic which, for non-technical users, is often difficult to understand. However, in order for a KBS to be usable by such users, it is essential that they can understand and control not only the knowledge base construction process, but also how results are obtained from the running system. It should be easy for users not only to enter different pieces of knowledge and to understand their *meaning* but also to understand the *results* of the system, and *how* the system computed these results. This latter aspect, namely the ability to understand *why* the system gives a certain answer, is especially important since the expertise of different users may vary, and explaining each step of the logical inference process poses a difficult problem.

However, due to the core properties of norms, providing such explanations is not trivial. First, norms can be applicable only in specific circumstances, rather than over a system's entire lifetime. Thus, examining norms in isolation from a running system may not provide any useful explanation regarding an individual agent's behaviour. Second, multiple norms can interact with each other, collectively placing complex expectations on the various agents involved. Thus, while it may appear that an agent is violating some obligation, it may actually be the case that the agent is either currently exempt from this obligation due to it not being applicable in the current situation, or due to there being some *permission* that applies in the current circumstances, overriding the obligation. Given this, it should be clear that it is extremely difficult for non-technical users (indeed, also for technical experts) to interpret a large set of textually (logically) specified norms and identify their effects, and that an alternative solution to norm understanding is required.

In response, our aim in this paper is to provide a sound graphical representation of norms, by adopting a graph-based semantics and applying the semantics to normative systems. To do so, we adopt the normative framework of Oren et al. [17], a generic framework that enables updating and monitoring of the changing *status* of norms, and supports the normative reasoning process. Now, in order to provide such a graphical representation, we must be able to provide a sound and complete translation between the operations of the normative framework and the operations on the graph-based representation. Not only can this help in *understanding* the results of an update to the status of a norm, but it also allows for structural *optimisations* of norms that might not be obvious from the textual (logical) representation of the norm. Each of these is a significant challenge; in this paper, we focus on the former aspect of the graphical representation, leaving the latter for future work.

Oren et al.'s framework represents norms by means of sets of first order logic tuples, which are manipulated using a set of rules that can be reduced to first order logic subsumption on the individual tuple elements. The contribution of this paper is to map norms onto *conceptual graphs* [19,20], the only graph based formalism to have a sound and complete semantics corresponding to deduction (via subsumption) in first order logic. This formal semantics enables us to easily link Oren et al.'s norms, with their textual representation, to the conceptual graph's graphical representation, thereby providing a graphical explanation regarding the system's normative state to non-technical users. This aspect of our work was first discussed in [6], in which it was shown how individual obligations can be represented graphically. Representing permissions, and their interaction with obligations, introduces further complications, but we can extend the basic model to address this, as originally outlined in [16].

The remainder of this paper is structured as follows. In the next section, we provide the necessary formal background to the paper by briefly reviewing the normative framework and introducing the conceptual graph formalism. In Section 3, we show how the status of norms can be computed graphically. Section 4 then considers

the graphical representation of interactions between permissions and other norm types. In Section 5, the paper provides a discussion in two parts: first it offers an evaluation of the effectiveness of our approach, together with an assessment of what is needed for more substantial user studies; second, it reviews some important related work. Finally, Section 6 concludes the paper by considering possible extensions to our work.

2. Background

In order to provide the requisite context for the contributions of the paper, and the basis on which we are able to develop norm explanations, we begin in this section by reviewing the formal model of norms. The model focuses on the problem of monitoring in that it facilitates identification of the *status* of norms as the environment changes over time. We then introduce the graphical formalism used in the remainder of this paper, conceptual graphs (CGs), which we map to the normative model in Section 3. This mapping allows us to address the problem of *explanation*, identifying why a norm has a particular status at some point in time.

2.1. The normative model

We introduce the normative model in a somewhat informal manner, motivating it in the context of a small example and examining how the model can be applied. Consider a situation in which an agent takes their car to a repair shop in order to be repaired. This repair shop provides a guarantee to its customers that their cars will be repaired within seven days, and thus has an *obligation* upon it, whenever a car arrives, to repair it within seven days. Clearly, once this obligation is fulfilled, it is lifted, and the repair shop no longer needs to repair the car. However, the obligation remains on the repair shop *as long as* the car is not repaired (even after seven days have passed). Finally, circumstances beyond the repair shop's control (for example, a power failure), will give the repair shop permission to repair the car seven days later than otherwise required.

The requirement on the repair shop to mend a car within seven days only obliges the repair shop to take action once a car actually arrives. Until then, the norm is an *abstract norm*. When a customer brings in a car, the norm is instantiated, thereby obtaining normative force over the repair shop and obliging it to repair the car within seven days. A single abstract norm can result in multiple *instantiated norms*; if two cars arrive at the repair shop, two instantiations of the abstract norm will occur.

Given this example, we observe that a norm may be defined in terms of five components. First, a norm has a *type*, such as an obligation, or a permission. Second, a norm has an *activation condition*, identifying the situations in which the norm affects some agents. Third, a norm imposes some *normative condition* on the affected agents; if this normative condition does not hold, then the norm is not being complied with (or made use of in the case of a permission). Fourth, norms have an *expiration condition*, identifying the situations after which the norm no longer affects the agent. Finally, the norm must identify the agents to which it is directed (i.e. those it affects), referred to as the *norm targets*.

More formally, we assume that the permissions and obligations represented by the norm refer to states and events in some environment, represented by some logical predicate language $\mathcal{L}$, such as first order logic. A norm is then a tuple of the form:

⟨*NormType*,
NormActivation,
NormCondition,
NormExpiration,
NormTarget⟩,

where

1. *NormType* ∈ {*obligation*,*permission*}; and
2. *NormActivation*,*NormCondition*,*NormExpiration*,*NormTarget*   are all well formed formulae (*wff*) in $\mathcal{L}$.

Thus, for example, the following abstract norm represents the idea that a repair shop must repair a car within seven days of its arrival at the shop[1]:

⟨*obligation*,
arrivesAtRepairShop(X, Car, T_1),
repaired$(Car) \lor (currentTime(CurrentTime) \land$
before$(CurrentTime, T_1 + 7days))$,
repaired(Car),
repairShop(X)⟩.

For ease of presentation, we have taken a relaxed approach to the notation, and mixed events and states within this norm; a more complex underlying language, such as the Event Calculus [11], would allow disambiguation of these concepts (as well as providing a richer typology of temporal notions).

If, at any point, an abstract norm's *NormActivation* condition holds, an instantiated version of the norm is created (subject to the additional constraint that the norm is not already instantiated for the same reason, as discussed in detail in [17]). The instantiation of a norm involves creating a copy of the abstract norm in which the norm's variables are bound to the values that caused the *NormActivation* condition to evaluate to true. When instantiated, the individuals included in *NormTarget* are identified. These individuals are then either obliged or permitted to bring about the normative goal specifed by *NormCondition*, until such a time as the conditions specified by *NormExpiration* hold. In this way, instantiated norms *persist* until they expire. (Note that a more complete logical semantics for the instantiation and processing of norms in this way is provided in [17].)

Now, if a car, car_1, arrives at Bob's repair shop at time 12, we can instantiate the abstract norm above and obtain the following instantiated norm:

⟨*obligation*,
arrivesAtRepairShop$(bob, car_1, 12)$,
repaired$(car_1) \lor (currentTime(CurrentTime) \land$
before$(CurrentTime, 19))$,
repaired(car_1),
repairShop(bob)⟩.

It should be noted that there can be variables within an instantiated norm (as is the case for *CurrentTime* in the above norm), and that the norm target refers to both the agent (*bob*), and the role (*repairShop*) undertaken by the agent in the context of the norm. Given a *NormTarget*, one simple way of computing the set of agents affected by the norm, is as follows: {*X*|*NormTarget*⊢ *role*(*X*)} for any predicate *role*.

A key aspect of the normative framework is that it enables the identification of the changing *status* of norms over time. This status can include the fact that it is instantiated or abstract, whether it is being complied with or violated, and whether it has expired. This is critical in understanding the impact of norms on behaviour and determining what actions to take as a result; the work in [17] introduces several distinct predicates that capture these different

possibilities for status. For example, *violation* of a norm may require some remedial action, and is thus a relevant status value, with an associated predicate. Importantly, the status can also be referred to by other norms. For example, a norm stating that "*if a car has not been repaired after seven days, the repair must be free*", can be represented as follows (assuming that the norm above is labelled $n1$):

⟨*obligation*,
violated$(n1)$,
repairCost$(Car, 0)$,
false,
repairShop(X)⟩.

Here, the *violated*$(n1)$ predicate refers to the norm's status, and evaluates to true if and only if $n1$ is an instantiated obligation whose normative condition evaluates to false, and for which there is no permission that allows the *negation* of the normative condition. This, and other such predicates are formally defined in [17].

As seen in this example, norms can *explicitly* refer to other norms and the variables found within them (such as *Car* in the example above). In addition, as we will see later there may also be *implicit* references to other norms (most notably in the case of permissions). Determining the status of any particular norm thus requires an examination of the interactions between multiple norms; when a system contains many norms connected to each other by such implicit and explicit references, it can be extremely difficult to identify precisely why some norm has a particular associated status. In order to address this difficulty of understanding and identification, we seek an alternative means of examining the status of norms in such systems. In particular, since humans find it much easier to assimilate large amounts of graphical information, as opposed to information in other forms, it is appropriate to make use of a graphical model to represent and visualise norms. In doing so, we are able to make explicit the links between the norms described above in a way that is amenable to human inspection and understanding. Of course, since such a representation must also to be able to be processed by machine, the best choice of representation to use for this purpose is one that is well understood and has a formal semantics. In consequence, therefore, we adopt *conceptual graphs* as the foundation for our graphical representation mechanism. In the next subsection, we introduce and describe this conceptual graph formalism.

2.2. Conceptual graphs

Due to their visual qualities, *semantic networks*, which were originally developed as cognitive models, have been used for knowledge representation since the early days of artificial intelligence, especially in natural language processing. Different kinds of semantic networks all share the basic idea of representing domain knowledge using a graph, but there are differences concerning notation, as well as rules or inferences supported by the language. In semantic networks, *diagrammatical reasoning* is mainly based on path construction in the network.

In this context, we can distinguish two major families of languages resulting from work on *semantic networks*: KL-ONE and *conceptual graphs*. KL-ONE [22] is considered to be the ancestor of *description logics* (DLs) ([1]), which form the most prominent family of knowledge representation languages dedicated to reasoning about ontologies. However, description logics have now lost their graphical origins. In contrast, *conceptual graphs* were introduced by Sowa (cf. [19,20]) as a diagrammatic system of logic intended "to express meaning in a form that is logically precise, humanly readable, and computationally tractable" (cf. [20]). Throughout the remainder of this paper we use the term

---

[1] Unless otherwise stated, we make use of Prolog notation within our logical formulae. More specifically, variables are written with an initial capital letter, while constants begin with a lowercase letter.

"conceptual graphs" to denote the family of formalisms rooted in Sowa's work and then enriched and further developed with a graph-based approach (cf. [5]).

In the conceptual graph (CG) approach, all kinds of knowledge can be encoded as graphs and can thus be naturally visualised. More specifically, a CG partitions knowledge into two types, the first of which identifies the CG's *vocabulary*, and can be seen as a basic ontology, and the second of which, (referred to as the *basic graph*) stores facts encoded using the CG's vocabulary. The vocabulary, referred to as the CG's *support*, is composed of two distinct parts, namely a partial order of concepts, and a partial order of relations (of any arity). Since both parts of the support are partial orders, they can be visualised by their Hasse diagram, where the partial order represents a specialisation relation, $t' \leqslant t$, indicates that t' is a specialisation of t. More specifically, if t and t' are concepts, $t' \leqslant t$ indicates that every instance of the concept t' is also an instance of the concept t. Fig. 1 provides an example of a concept hierarchy constructed in this way, and which is used in the illustrative example throughout this paper. Similarly, if t and t' are relations, and these relations have the same arity, say k, then $t' \leqslant t$ means that if t' holds between k entities, then t also holds between these k entities. Fig. 2 shows the relation hierarchy that is used in the example throughout this paper. These relations are organised by arity — unary, binary and ternary and so on — with a separate graph for each.

Now, a CG's basic graph encodes knowledge based on the representation of entities and their relationships. This encoding takes the form of a bipartite graph, consisting of *concept nodes* that represent entities, and *relation nodes* that represent relationships between these entities or their properties. A concept node is labelled by a pair, $t{:}m$, where t is a concept drawn from the concept hierarchy, and m is called the *marker* of the node. Markers consist of either a specific individual name, or a marker, denoted $*$, which acts as a generic marker, and is used if the concept node refers to an unspecified entity. A relation node is labelled by a relation r taken from the relation hierarchy and, if r has an arity of k, the relation node must be incidental to k totally ordered edges. Classically, concept nodes are drawn as rectangles and relation nodes as ovals. The order on edges incidental to a *k-ary* relation node is then represented by labelling the edges with numbers from 1 to k. Fig. 3 provides an example of a basic graph that expresses the fact that a *vehicle arrived* at the *RepairShop* at a certain *Time*. Finally, since the notion of *instantiated norm* is central to our framework, we occasionally refer to an *instantiated basic graph*, which is simply a basic graph with no generic markers, as shown in Fig. 4.
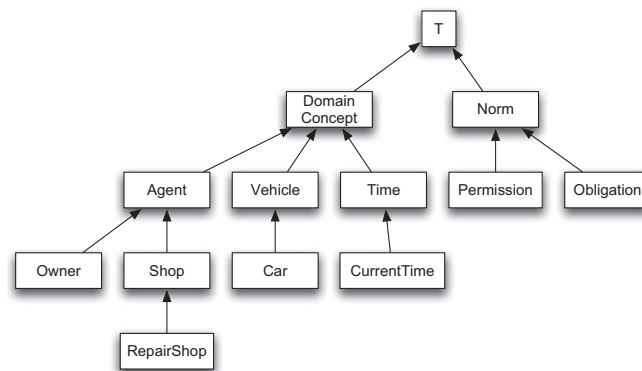
Given these basic notions of conceptual graphs, a mapping between a CG and first order logic can be used to provide the CG with a semantics. This mapping, denoted by Φ in the conceptual graphs literature, utilises a first order language corresponding to the elements of the conceptual graph's vocabulary (i.e. its relation and concept hierarchies). Elements from the concept hierarchy are translated into unary predicates, and elements from the relation hierarchy with an arity of k are mapped into k-ary predicates. Individual names are then constants in the logic. Formulae are added to the logic based on the partial orders of concepts and relations: if t and t' are concepts, and $t' < t$, then the formula $\forall x((t'(x) \rightarrow t(x))$ is obtained. Similarly, if r and r' are k-ary relations, with $r' < r$, then the formula $\forall x_1 \dots x_k (r'(x_1,\dots,x_k) \rightarrow r(x_1,\dots,x_k))$ is obtained. A fact G obtained from the basic graph can then be translated into a positive, conjunctive and existentially closed formula (via the mapping $\Phi(G)$), with each concept node being translated into a variable or a constant. If the concept node is a generic node (as in the concept nodes on the left hand side of Fig. 3), then $\Phi(G)$ results in a variable, otherwise $\Phi(G)$ returns the concept node's individual marker.

While we have shown how CGs can be mapped to a first order logic formula, mapping between CGs and first order logic in order to perform reasoning is cumbersome, and a large part of the power of the CG approach is obtained from the ability to perform reasoning over the graphs themselves. The fundamental operation used to perform such reasoning is *projection*. In order to describe this concept, we must first define the notion of a *homomorphism*.

Let G and H be two basic graphs (BGs). A *homomorphism* π, from G to H, is a mapping, from the concept node set of G to the concept node set of H, and from the relation node set of G to the relation node set of H, that preserves edges and may decrease concept and relation labels. That is:

- for any edge labelled i between the concept node c and relation node r in G, there is an edge labelled i between the nodes $\pi(c)$ and $\pi(r)$ in H; and
- for any (concept or relation) node x in G, the label of its *image* $\pi(x)$ in H is a specialisation of the label of x; that is, $\pi(x) \leqslant x$.

Homomorphisms are used to form *projections* between two BGs, as illustrated in Fig. 5. Here, the BG on the left hand side of the figure models the situations when some *Car* arrives at bob's repair shop at *Time* 12, while the right hand side of the figure models the case when some *Vehicle* arrives at some *RepairShop* at some *Time*. The homomorphism (indicated using dashed lines) indicates that node *RepairShop* on the right can be mapped onto the
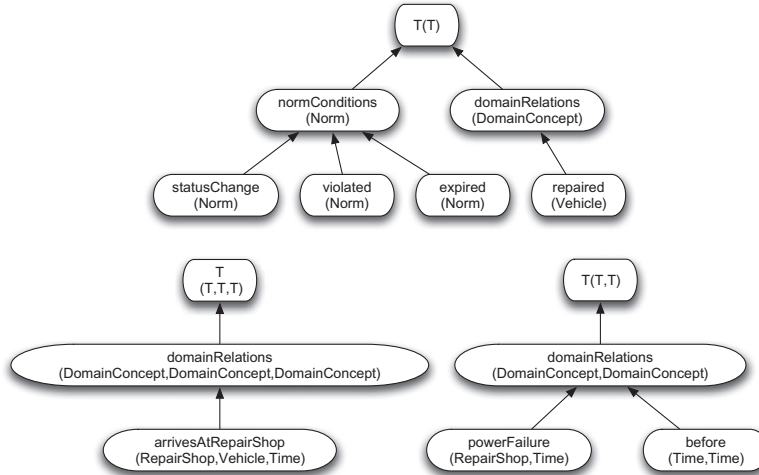

Fig. 1. Conceptual graph support: the *concept hierarchy*.

Fig. 2. Conceptual graph support: the *relation hierarchy*.
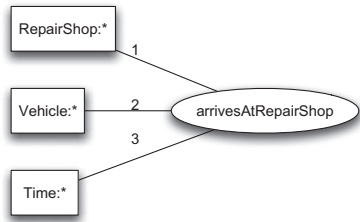

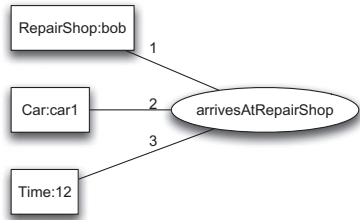**Fig. 3.** A generic *basic conceptual graph fact*.

Fig. 4. A ground, or instantiated, *basic conceptual graph fact*.

RepairShop node on the left with marker *bob*, that the *Vehicle* node on the right can be mapped to *Car* on the left, and that the generic *Time* node on the right can be mapped to a specific *Time* node. Finally, the *arrivesAtRepairShop* relation maps between the two BGs.

Now, an important theorem in the CG literature (referred to as the *fundamental theorem*) also allows us to map back from first order logic to conceptual graphs. This theorem states that, given two BGs, *G* and *H*, there is a homomorphism from *G* to *H* if and only if $\Phi(G)$ is a semantic consequence of $\Phi(H)$ and the logical translation of the vocabulary: $\Phi(\mathcal{V}), \Phi(\mathcal{H}) \models \Phi(\mathcal{G})$. This is a soundness and completeness theorem of BG homomorphism with respect to first order logic entailment, the consequence of which is that a

homomorphism between two graphs is, in effect, an explanation as to why logical subsumption takes place. Since such homomorphisms can be represented graphically, this allows for visual representations of logical subsumption, the explanation of which is a unique feature of CGs. Any alternative logic-based graphical representation language would have to include an additional separate explanation layer as well as the representation layer itself.

Given the fundamental building blocks we have now introduced, of the normative model and conceptual graphs, we can proceed to detail how a norm can be represented within a CG-based framework.

3. Graphically computing the status of norms

3.1. Modelling norms with CGs

By encoding structured knowledge graphically, CGs can provide a way to represent, illustrate and interpret the states through which norms proceed; that is, whether they have been activated, violated, fulfilled, or expired. Then, by connecting such representations (or depictions) of permissions and obligations, it is possible to interpret whether an obligation has truly been violated, or whether a permission derogates this obligation under particular circumstances.

One commonly encountered problem is that norms can sometimes be fulfilled by multiple different actions, events or states. Intuitively, if these conditions are separated by disjunctions, they can be evaluated in a tree-like structure by the norm reasoner. We make this explicit by representing norms in such a structure, with every level of the tree corresponding to one type of condition in the norm. Moreover, at every level, we break the condition into a disjunction of positive first order logic conjunctions. This representation ensures that normative reasoning is sound and complete with respect to a particular kind of path-finding in the norm tree (finding at least one satisfied level node). Now, when instantiated, a norm's activation condition becomes fixed, and its normative and expiration conditions are used to determine its status. This suggests that the tree structure is indeed suitable for use in representing a norm. In what follows, we proceed to define this tree structure, which we refer to as a *norm tree*.
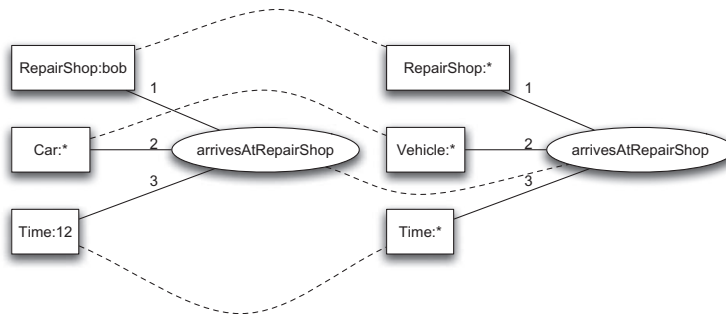
Fig. 5. A projection between two basic conceptual graph fact.

A norm tree represents both abstract and instantiated norms. Its root is associated with the entire norm (more specifically, its *type* and *target*), while the remaining levels represent different parts of the norm. (For this purpose, we also assume that a norm's *target* is a conjunctive formula, and can thus be represented as a conceptual graph). Nodes in the second level are associated with the activation condition, nodes in the third level are associated with the normative condition, and nodes in the fourth level with the expiration condition. Each of the nodes within the tree has an associated CG representation of its content, as illustrated in Fig. 6.

Given this basic structure, different branches of the norm tree can be used to represent disjunctive conditions within a specific norm attribute. Thus, for example, a norm with a normative condition of the form $a \vee b$ would have two branches at the norm tree's third level. As indicated above, we assume that the norm target parameter consists of a conjunctive combination of predicates (in other words, a norm is associated with a specific group of individuals rather than applying to some subgroup *or* another), and that all other parameters (except for norm type), may contain disjunctions. In this way, in order to represent the norm as a norm tree, we transform all of its attributes into disjunctive normal form, to get a norm represented as follows:

$$\left\langle Type, \bigvee_{i=1,a} AC_i, \bigvee_{j=1,c} NC_j, \bigvee_{k=1,e} EC_k, NT \right\rangle, \tag{1}$$

where AC_i, NC_j, EC_k and NT are all conjunctive first order formulae so that, for example, $AC = \bigvee_{i=1,a} AC_i$. Furthermore, by assuming negation as failure, we can ensure that all of these formulae are positive

(by introducing an explicit predicate for negation), and can therefore represent each as a conceptual graph, defined on some given support (i.e. the domain ontology).

Given a norm N in disjunctive normal form as in Eq. (1) above, we define its norm tree as a tree for which each node contains a norm and is labelled by a CG as follows.

1. The root node of the tree contains norm N and is labelled by a CG identifying the norm's type and targets (i.e. *Type* and *NT*).
2. The root node has a child nodes (i.e. nodes at level one) where, for $i = 1 \ldots, a$, child node i is labelled with the CG representing AC_i and contains a norm N^i of the form:

$$\left\langle Type, AC_i, \bigvee_{j=1,c} NC_j, \bigvee_{k=1,e} EC_k, NT \right\rangle.$$

3. Each node at level two, which is a child of N^i, and is labelled with a CG representing NC_j, contains a norm N^{ij} for $j = 1, \ldots, c$ of the form:

$$\left\langle Type, AC_i, NC_j, \bigvee_{k=1,e} EC_k, NT \right\rangle.$$

4. Each node at level three, which is a child of N^{ij}, and is labelled with a CG representing EC_k, contains a norm N^{ijk} for $k = 1, \ldots, e$ of the form:

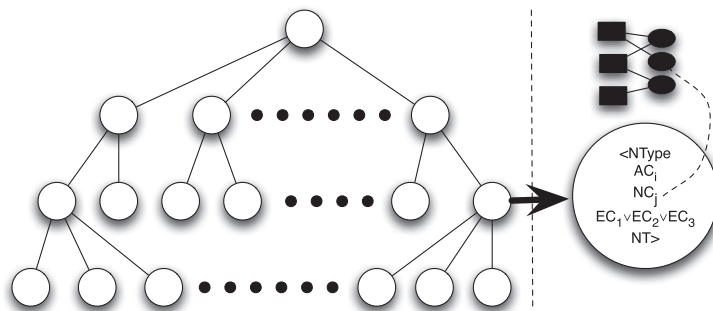$$\left\langle Type, AC_i, NC_j, EC_k, NT \right\rangle.$$

Fig. 6. A conceptual representation of a norm tree.

3.2. Modelling norms in the repair domain

Consider the norm of our car repair example, which obliges a repair shop to repair a car within seven days of its arrival. The left hand side of Fig. 7 illustrates the norm tree associated with this norm. For simplicity, we have ignored the norm target parameter, assuming that it is present in the root node. The dotted line between the nodes and CGs identifies which nodes are labelled with which CGs. It should be noted that the *function* relation, found in the right hand normative condition node, is used to compute whether the current time is greater than seven days from the time the car arrived for repair. This is used to simplify the CG shown in the figure; within a complete system, this CG would make use of an arithmetic function to add seven days to the car's arrival time, and then make use of an additional function or predicate to compare the current time to the deadline to determine whether the car has been repaired in time. Now again consider the nodes at the third level of the norm tree. These correspond to the norm condition and, when translated to first order logic, yield a formula of the form $repaired(Car) \lor function(CurrentTime, Time + 7\ days)$.

Note that there is a separation between the semantics of the normative model and its norms, and the semantics of the knowledge-based system. For a parameter (such as the normative condition) in the norm to evaluate to true, any of the disjunctions from which it is composed must evaluate to true (e.g. *repaired(Car)* in the above example). This aspect of a norm is captured by the normative model's semantics, and is thus represented by the norm tree structure. However, reasoning within the knowledge-based system is kept separate from the norm model semantics by means of conceptual graph annotations of the nodes in the normative tree. Thus, the knowledge-based system identifies *which* of the normative condition's disjunctions actually evaluated to true in the case where the normative condition is true. A user of the system could then be presented with the explanation of why the norm condition is valid: in the context of the repair shop norm, at least one node is satisfied (or both). While the figures in this paper are monochrome, colour can be added to a running system in order to identify the validity of a node (for example, red could mean invalid, while green could mean valid).

Finally, the right hand side of Fig. 7 illustrates the norm tree for the permission (to repair a car later than 7 days if there is a power failure) found in our example. Since no disjunctions exist within the activation, expiration and normative conditions, the norm tree has no branches.

This conceptual graph representation provides us with two advantages over a textual representation of the norm. First, the conceptual graph representation makes the types of concepts linked by predicates visually explicit (for example, *RepairShop:** as opposed to *X*). While this problem is easily addressed by manually changing the variable names of the textual logic representation (using *meaningful* literals), the heuristic employed could be confusing. Second, and more importantly, for *elaborated* pieces of knowledge (namely conjunctions with common variables) the translation between natural language and logical formulae becomes very difficult. For example, suppose that we are trying to represent the fact that a car arrives at a *Volvo* repair shop, that the repair shop accepts only cars of the same make, that the time at which the car arrives at the repair shop must be later than 9, and that this is the opening time of the repair shop. While the conceptual graph depiction is intuitive given its visual nature, the logic-based (textual) approach can be difficult to follow.

3.3. Instantiating norms

Now, consider the abstract norm illustrated on the left hand side of Fig. 7, and suppose that a new fact—that some car, c_1 arrived at the repair shop belonging to *bob* at time 12—is added to the knowledge base. In predicate form, we write *arrivesAtRepair-Shop(bob, c_1, 12)*. This piece of knowledge is *projected* to all the norm conditions in the system in the following way. Using projection, the fact is mapped onto the abstract norm of Fig. 7, and a new norm tree, with the appropriate CG nodes now labelled by constants, is created. This CG is shown in Fig. 8 in which, for clarity, nodes in the norm tree belonging to an abstract norm are depicted in white (c.f. Fig. 7).

It should be noted that there can be multiple instantiated versions of the same abstract norm simultaneously. However, each of these will have a different set of variable bindings, and thus a different CG associated with the norm tree.

3.4. Computing the status of norms

So far, we have shown how abstract and instantiated norms may be represented as norm trees, but we have not yet considered how to determine the status of a norm using our norm tree structure. Consider the left hand side of Fig. 8, which represents the instantiated norm from the repair shop example. The right hand side of the figure shows the CG representation of the environment,
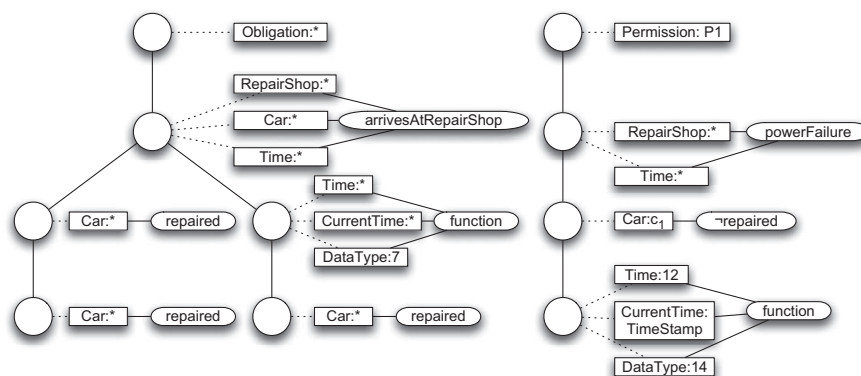

Fig. 7. The norm tree for the abstract obligation norm (left) and abstract permission norm (right) found in the repair shop example.
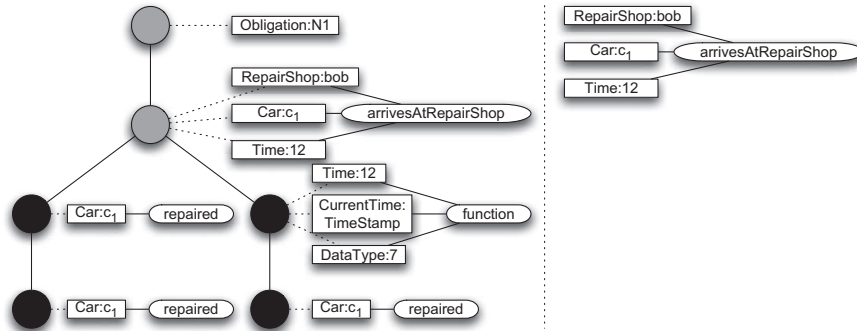
Fig. 8. An instantiated norm for the repair shop example.

as stored within the knowledge base. This CG represents the fact that a car c_1 arrived at the repair shop at time 12. To distinguish between abstract and instantiated norms, we colour the nodes of an instantiated norm using different colours (which are always non-white), as opposed to white abstract norms.

Now, as new facts appear and disappear within the knowledge base, the status of norms also changes. Determining this status may be achieved by checking for the existence of projections between the facts in the environment and the conceptual graph annotations of the norm tree. Fig. 9 illustrates the situation when an additional fact—namely that the current time is before time 19—is added to the environment (represented by the two CGs on the right of the figure). The norm tree on the left of Fig. 9 now contains a mixture of black and grey nodes. A grey node corresponds to the fact that the node is satisfied; that is, there is a projection between the environment and the corresponding CG annotation. The remaining nodes are black: they are not satisfied. Thus, in Fig. 9, illustrating the car repair example, there is no projection between the CG node representing the expiration condition, which states that the car is repaired, and the CG on the right of Fig. 9. Similarly, there is a projection (and thus the node is grey) between the CG on the right, and the CG linked to the node at the normative condition level stating that the current time is before 19 (the condition in this latter node is represented by the function taking in the datatype, time and current time). If, at some later point, the car is repaired, the black nodes within the norm tree will turn grey.

During its lifecycle, an abstract norm becomes instantiated. While instantiated, its normative condition may evaluate to true or false at different times. Eventually, the norm's expiration condition evaluates to true, after which the instantiated norm is deleted. We have already seen how one may determine whether a norm may be instantiated using a norm tree. A norm's normative condition is satisfied (that is, it evaluates to true), if any of the nodes at the norm condition level are grey. Similarly, a norm expires if any of the nodes at the expiration condition level are grey.

A norm's status includes whether it is activated or expiring, and whether it is being satisfied, and it is trivial to determine this from the norm tree. It is also possible to determine more sophisticated aspects providing a richer notion of the status of a norm from a norm tree. As an example, in the next section, we discuss how to determine whether an obligation has been *violated*. All aspects of the status of a norm can be computed by posing queries to the knowledge base, and thus, it is possible to *visually* determine the status of a norm.

4. Computing violation with permissions

One critical aspect of normative state that cannot be computed directly form a norm tree is whether the norm is violated. This is because of the way in which we treat permissions. In [2], Boella and van der Torre point out that permissions can be viewed as exceptions to obligations and prohibitions, and this is how
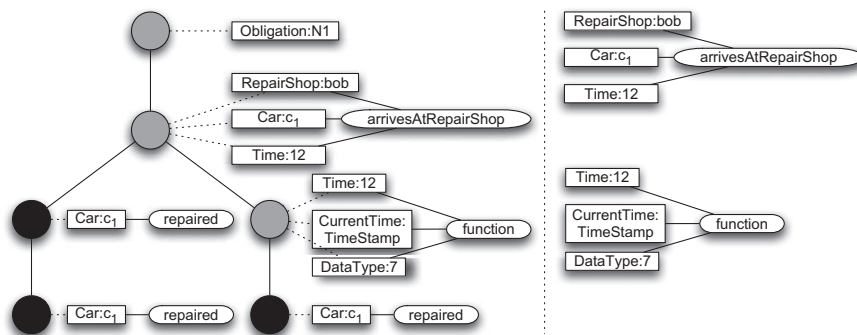

Fig. 9. A norm tree evaluated according to the knowledge base shown on the right.

permissions are handled by our model. Thus, for example, given an obligation on the repair shop to repair a car within 7 days, a permission to instead repair the car within 14 days *derogates* the obligation. While the obligation may not be complied with (because the car may not be repaired within 7 days), the repair shop will not be in violation of the obligation unless 14 days have expired.

Permissions thus do not exist in isolation, but instead act as exceptions to other types of norms. This means that in evaluating whether an obligation or prohibition is violated, one must consider not only the possibly violated norm itself, but also the permissions present. However, given a large normative system, identifying the appropriate permission that may prevent a violation from occurring can be challenging. Our visual approach can help overcome the cognitive load imposed by this problem by highlighting any relevant permissions that prevent a norm from being violated.

To illustrate, we return to our car repair example. If a power failure occurred at time 14, then the (instantiated) permission allowing Bob to repair car c_1 within 14 days (i.e. by day 28) is as follows:

⟨*permission*,
powerFailure(*bob*, 14),
¬*repaired*(c_1),
currentTime(*CurrentTime*) ∧ *before*(*CurrentTime*, 28*days*),
repairShop(*bob*)⟩.

Conceptually, in order to determine whether an instantiated and un-expired permission derogates an obligation or prohibition, we must check whether the permission's norm condition is *consistent* with the obligation. If it is not consistent, in the sense that the permission allows the negation of the obligation, then derogation takes place, otherwise the permission does not affect the obligation. In our example, ¬*repaired*(c_1) is inconsistent when evaluated against *repaired*(c_1), and the permission thus derogates the obligation. This check for consistency thus lies at the heart of our work.

Clearly, consistency checking requires the ability to represent and reason about the negation of a relation. However, the standard CG formalism is unable to represent such negated relations, and we make use of an extension to CGs first proposed by Mugnier and Leclère [15] to show how the consistency check can be performed from within the CG formalism. Mugnier and Leclère introduce the idea of a *negative relation node* which, when present as a node in a CG, identifies the fact that the named relation does not exist

between the concepts incident on the node. Now, the approach we adopt here makes use of the closed world assumption, and extends a CG to include its negative relation nodes. More specifically, we add all possible negative relation nodes that do not make the graph contradictory to the CG's basic graph. Thus, for example, if we do not know that a car has been repaired, we now explicitly state that it has not been repaired; if a node *repaired*(car_1) is not present in some CG, the *completed* form of the CG must include the node ¬ *repaired*(car_1).

Given this completed CG, if the permission's normative condition cannot be projected into the CG (because the car has in fact been repaired, for example), the permission derogates the obligation (or rather, that node in the norm tree for which the CG projection is unsuccessful, which will not be coloured black). The permission, and the relevant concepts and relations that derogate the permission, can then be displayed to the user to explain why the norm is not in violation. If, on the other hand, the permission is not relevant to the obligation, then a violation occurs, and the violated norm can again be highlighted in order to show the user its status. Thus, given a norm tree for an (instantiated, unexpired) obligation *N*, the norm it represents is violated if and only if all of its nodes at the normative condition level are coloured black.

Fig. 10 illustrates the derogation of an obligation by a permission. Dashed lines indicate links between the concepts and relations found in the two nodes, and the normative condition node marked with a grey node with a black centre in the obligation indicates that the node, while evaluating to false, is derogated by a permission. From the figure, it is clear that the obligation is not violated. Note that the permission's activation condition node is black. We assume that while a power failure occurred in the past (instantiating the permission), there is currently no power failure.

4.1. Case study

To illustrate the overall framework, we consider an additional scenario in which rapid response medical units must perform some duties when an emergency situation occurs. These units have the following obligation:

"If a state of emergency has been declared, a rescue unit is obliged to travel to a casualty, and then collect them, or provide them with medicine until they have no more space and are out of medicines".
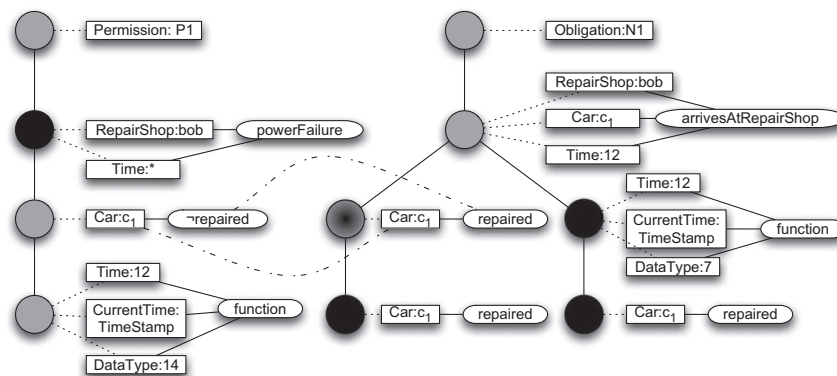
Fig. 10. A norm tree for a permission (left), and obligation (right) evaluated according to some knowledge, showing how the permission derogates the obligation.

Formally, this obligation is represented as follows:

⟨*obligation*,
stateOfEmergency() ∧ *casualty*(*C*),
travel(*U*, *C*) ∧ (*collect*(*U*, *C*) ∨ *medicate*(*U*, *C*)),
noSpace(*U*) ∧ *noMedicine*(*U*),
rescueUnit(*U*)⟩.

The disjunctive normal form of the obligation's normative condition is:

(*travel*(*U*, *C*) ∧ *collect*(*U*, *C*)) ∨ (*travel*(*U*, *C*) ∧ *medicate*(*U*, *C*)).

Given this, we assume a very simple permission representing casualty triage: "If the casualty is dead, there is no need to medicate them". Formally, this is as follows:

⟨*permission*, *dead*(*C*), ¬*medicate*(*U*, *C*), *false*, *rescueUnit*(*U*)⟩.

In order to construct the norm tree, we begin by identifying the concepts and relations found in this scenario, where the concepts include *StateOfEmergency*, *Casualty*, *RescueUnit* and *Dead*, and the relations include *travel*, *collect*, *medicate*, *noSpace* and *noMedicine*. These concepts and relations yield the support displayed in Fig. 11, and the abstract norms illustrated in Fig. 12.

Now, suppose that a state of emergency exists, and that a dead casualty c_1 has been detected by a rescue unit r_1. Furthermore, r_1 has space and medicine available. Given that the rescue unit has not travelled to the casualty, collected it, or provided medicine, is it in violation of its obligation? In order to determine this, we must compute the completed CG of the instantiated obligation's normative condition. Fig. 13 shows the completed form of the graph for both the left and right hand branches of the instantiated *obligation* norm tree's normative condition nodes. The dotted lines within Fig. 13 illustrate that the *permission*'s normative condition projects into the obligation's right hand branch normative condition.
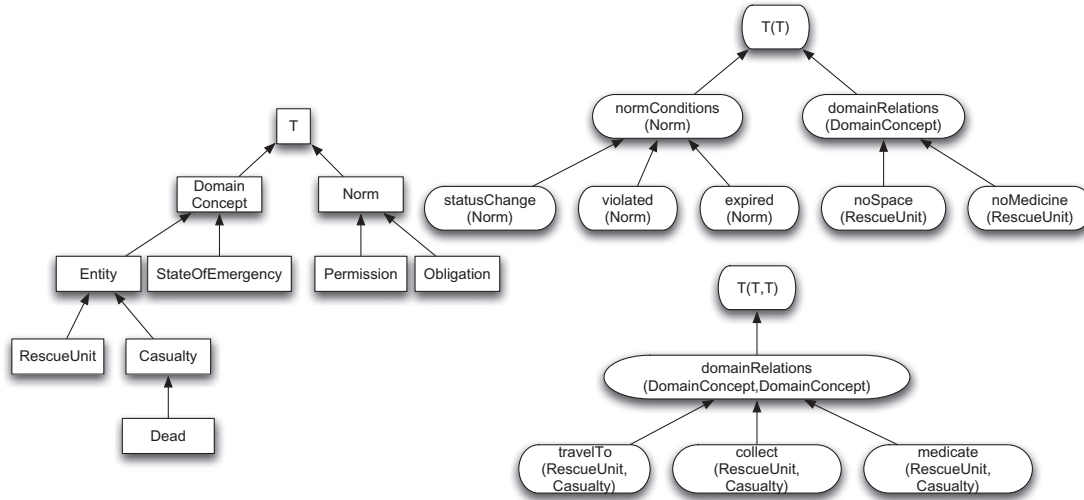
Fig. 11. The CG support composed of the concept hierarchy (left) and relation hierarchies (right).
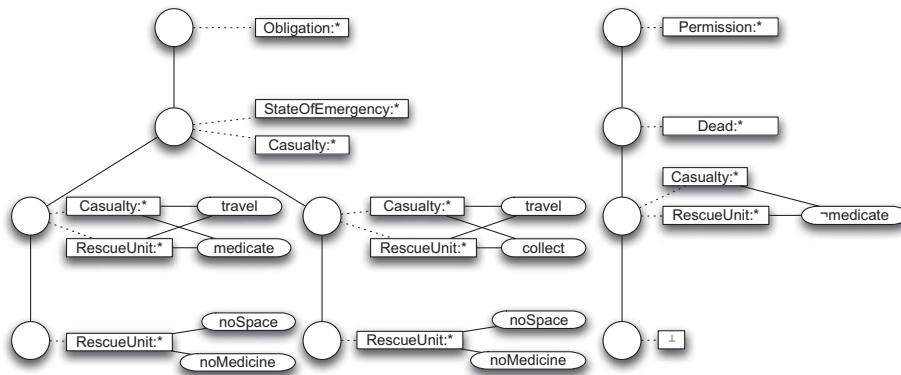
Fig. 12. The abstract norm trees.
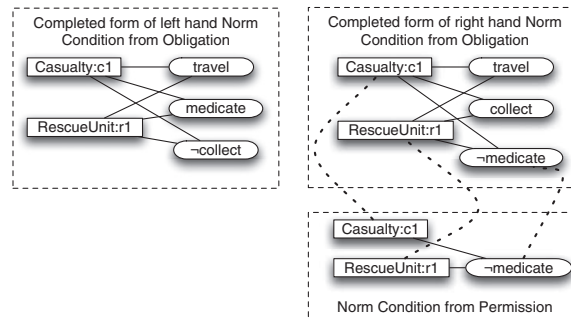
Fig. 13. The completed form of the obligation's normative condition (top left and top right), with the projection of the permission's normative condition.
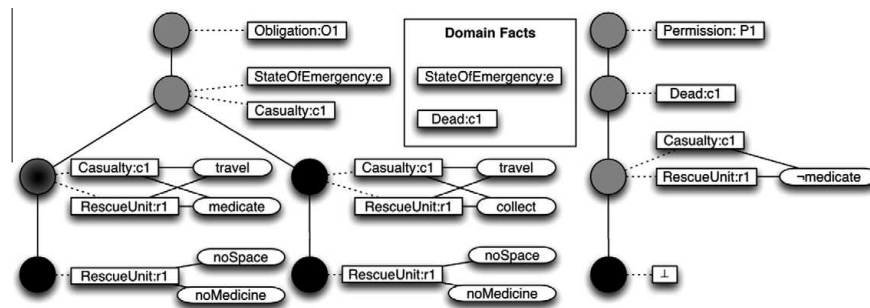
Fig. 14. Norm instantiation according to the domain facts.

However, no such projection is possible into the left hand branch. Therefore, the permission derogates the left hand branch of the obligation's norm condition, and the norm is not violated. This is shown in Fig. 14. In summary, our CG approach to norm explanation makes clear exactly how the permission enables a user to understand how the permission interacts with the obligation.

5. Discussion

5.1. Evaluation

Norms provide a means of regulating system behaviour, yet their structure and operation can often obscure the understanding that is possible, especially by end-users. In particular, it is important to understand not just the structure of norms, but also their status at different points in time, and the ways in which they interact. This latter aspects is critical, for the interaction between norms can affect their status. In order to provide an effective means for supporting user understanding, we have developed a visual model to explain the structure and status of a norm. This ability to provide explanations of a norm's status is especially useful; for example, complex contract disputes may require that some rewards or penalties be assigned by a human mediator, but in order to perform this assignment, the mediator must first understand which norms were violated, and which were complied with. Norm explanation is also important at the system design stage, where an understanding of norm status in different situations is needed to ensure correct system behaviour.

Previous work such as [5] has demonstrated that graphical systems excel in cases where non-technical users must be catered for,

and this is exactly the approach we adopt. More specifically, we can identify the following benefits of our graphical normative representation. First, the graphical system can be used to identify which elements of the environment impact on a norm, even when making use of specialisation or generalisation of concepts or relations (as illustrated in Fig. 5), when it is not clear to a user how different concepts may relate to each other. In this way, users can directly track the effects of changes in the environment on a norm. Similarly, through the association of CGs with norms, it is possible to support navigation between norms sharing identical, specialised or generalised relations or nodes, or sharing markers. The set of norms affected by changes to the environment can thus be easily tracked.

Importantly, a graphical system is able to provide the user with an easily understandable snapshot regarding the status of the system. More specifically, by adopting an approach in which the colours associated with the nodes of an instantiated norm's tree indicate their status, we provide a means for users to quickly identify which norms have what status, and why (as illustrated in Fig. 9).

Finally, and as indicated above, the interactions between different parts of a system can be made explicit. Since individual norms can combine in complex ways to give sophisticated structures by virtue of the links between permissions and obligations, for example, providing a visual representation can be argued to be vital to ensure clarity of presentation and understanding. Indeed, identifying obligations that are derogated due to permissions, and in turn identifying these permissions is not trivial, yet as we have shown in Section 4 (and Fig. 10), this becomes relatively straightforward with an appropriate representation.

All these aspects can be seen directly from the work presented in this paper. Clearly however, while this evaluation of our contribution is justified in its own terms, and demonstrates the validity of our approach in providing explanation, the claim of aiding users requires a more substantial (and more challenging) evaluation. In particular, since one of the core advantages of the graphical approach lies in enhancing user *understanding*, the next step in evaluating our framework must be to undertake user studies, comparing the graphical approach and standard, text-based techniques for representing norms, and their impact on and value to users. Current work is concerned with implementation of a software tool for exactly this purpose, providing clear visualisations of the status of norms as described earlier by displaying the norm trees found in a running system, colouring tree nodes as appropriate, displaying the node CG graphs (and enabling further analysis to identify and display projections, graph support and the like). More interesting and valuable functionalities are also anticipated, for example, if an obligation is derogated, selecting an appropriate node will allow a user to visualise the associated derogating permission, and *vice versa*.

Although anecdotal evidence from early trials with users already suggests that our approach has significant merit, this more substantial user evaluation will require a methodical effort with multiple users across different user categories (for example, expert users with an understanding of logic, non-expert users with less formal modelling experience, and the like). Clearly, this is a major undertaking that is beyond the scope of the current work, yet this will be important before an enhanced appreciation of the value of the graphical norm explanation approach can be established.

5.2. Related work

Much of the existing work on norms and normative reasoning originated from the philosophical domain. While recognising the conditional nature of norms, such work emphasised problems such as identifying what state of affairs should hold, or how to resolve normative conflict. However, apart from the work of Governatori et al. [9], few have considered how a normative system evolves when norms are fulfilled. Governatori et al. adopt a defeasible logic based approach to norm representation, with norms expiring when a *defeater* to them is introduced. Within a long lived system, this approach is cumbersome; reinstantiating a norm requires the introduction of a defeater to the defeater. In contrast, the framework presented in this paper is intended to capture the evolution of a norm over time, allowing for its instantiation and expiration, as well as recording the time periods during which a norm was complied with or violated. Since the internal structure of such a norm is somewhat complex, some technique for explaining why a norm is in a certain state is required, and we proposed a visual model for explaining this status of a norm. This ability to provide explanations of a norm's status in such domains is particularly useful; for example, complex contract disputes may require that some rewards or penalties be assigned by a human mediator, but in order to perform this assignment, the mediator must first understand which norms were violated, and which were complied with. Norm explanation is also important at the system design stage, where an understanding of norm status in different situations is needed to ensure correct system behaviour.

As described in Section 3, instantiated norms are created by copying abstract norms and modifying the labels within the norm's basic graph. Recent work on CGs [21] has examined the possibility of adding a special *evolves into* relation to capture the notion of transformation over time, and it is tempting to utilise this relation to formally represent the instantiation of a norm. However, this relation is currently only useful when the objects being represented will transform into the evolved object in a predictable manner, and can therefore not be directly applied to our work. Nevertheless, identifying a more formal approach to creating instantiated norms from abstract norms is worth pursuing, as this would allow us to answer questions about possible norm instantiations.

Our graphical representation highlights the link between permissions and obligations, and borrows some ideas from [7], wherein CGs were used to express and manage the interdependencies between security policy rules. Since norms can be used to express such rules [12], many issues identified there (such as the detection of redundant policies) map directly to the domain of norms.

More generally, however, we are aware of very little work dealing with the explanation of norms to users. This may be due to an implicit assumption that normative systems are fully automated, and that explanation is thus not necessary, or perhaps due to an assumption regarding the technical expertise of a system's users. However, even if a user is able to understand a norm representation, graphical explanations may still be advantageous when reasoning about complex interactions between large groups of norms. One exception to this is the recent work of Miles et al. [14], which touches on the concept of norm explanation. Here, a causal graph is used to analyse and explain norm violation, and then to identify whether there were mitigating circumstances for the violation.

6. Conclusions and future work

Norms have a complex lifecycle, becoming instantiated, and thus placing an expectation on an agent's behaviour at certain points in time, following which they may expire and cease to influence an agent. Within a long lived system, norms may be instantiated and expire multiple times; at any point in time, only a certain subset of norms may be relevant to identifying what behaviour should take place. Furthermore, examining a single norm in isolation does not provide enough information to determine whether an agent is acting in compliance with the norm. For example, as shown in Section 4, permissions may derogate norms, and multiple norms must be considered when reasoning about their effects. Critically, while existing norm representations are sufficient for automated reasoning, their form is not ideal for explaining the behaviour of the system to end-users. In order to provide a user with an effective understanding of a normative system, all of these issues must be taken into consideration.

The goal of our approach is to provide an effective tool for system understanding to end-users. Our underlying norm formalism is able to model the norm's lifecycle, while our conceptual graph based representation enables a user to consider the interactions between obligations and permissions, and understand them in an intuitive manner. Many avenues remain open for future investigation. Other studies have shown that graphical representations are more easily understood than logic-based ones [5] by non-experts, and though we have proceed on this legitimate assumption, we have yet to undertake the user studies that will confirm this empirically, but aim to do so in the short term. We also intend to leverage the formal power of our model, by investigating the use of graph theoretical operations to identify redundant norms [2]. Similarly, we believe that graph-based operations can be used to detect, and help resolve, normative conflict. Both of these applications effectively validate the structure of the norms, and we thus aim to apply existing work on CG validation [8] to aid us in this task. Furthermore, projection can also act as a *similarity* measure, and can thus be applied to determining the trustworthiness of contracts (as encoded by groups of norms) along the lines suggested by Groth et al. [10]. Finally, we have focused on the status of norms

at a single point in time; we plan to investigate how our approach can aid in explaining interactions not only between simultaneously active norms, but also how they can be used to identify and explain temporally distributed normative interactions.

References

[1] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider (Eds.), The Description Logic Handbook, Cambridge University Press, 2003.

[2] G. Boella, L. van der Torre, Permissions and obligations in hierarchical normative systems, in: Proceedings of the Nineth International Conference on Artificial Intelligence and Law (ICAIL-03), ACM, New York, NY, USA, 2003, pp. 109–118.

[3] G. Boella, L. van der Torre, Institutions with a hierarchy of authorities in distributed dynamic environments, Artificial Intelligence and Law 16 (2008) 53–71.

[4] W. Briggs, D. Cook, Flexible social laws, in: C. Mellish (Ed.), Proceedings of the 14th International Joint Conference on Artificial Intelligence, Morgan Kaufman, San Francisco, 1995, pp. 688–693.

[5] M. Chein, M. Mugnier, Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs, Springer, 2009.

[6] M. Croitoru, N. Oren, S. Miles, M. Luck, Graph-based norm explanation, in: M. Bramer, M. Petridis, A. Hopgood (Eds.), Research and Development in Intelligent Systems XXVII, Proceedings of AI-2010: The Thirtieth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, pp. 35–48.

[7] M. Croitoru, L. Xiao, D. Dupplaw, P. Lewis, Expressive security policy rules using layered conceptual graphs, Knowledge Based Systems 21 (2008) 209–216.

[8] J. Dibie-Barthélemy, O. Haemmerlé, E. Salvat, A semantic validation of conceptual graphs, Knowledge-Based Systems 19 (2006) 498–510.

[9] G. Governatori, J. Hulstijn, R. Riveret, A. Rotolo, Characterising deadlines in temporal modal defeasible logic, in: Proceedings of the 28th International Conference on Artificial Intelligence (AI-2007), Lecture Notes in Artificial Intelligence, vol. 4830, pp. 486–496.

[10] P. Groth, S. Miles, S. Modgil, N. Oren, M. Luck, Y. Gil, Determining the trustworthiness of new electronic contracts, in: Proceedings of the 10th Annual International Workshop on Engineering Societies in the Agents' World (ESAW 2009), Springer, 2009, pp. 132–147.

[11] R.A. Kowalski, M.J. Sergot, A logic-based calculus of events, New Generation Computing 4 (1986) 67–95.

[12] C. Krogh, The rights of agents, in: M. Wooldridge, J.P. Müller, M. Tambe (Eds.), Proceedings of the IJCAI Workshop on Intelligent Agents II: Agent Theories, Architectures, and Languages, Lecture Notes in Computer Science, vol. 1037, Springer-Verlag: Heidelberg, Germany, 1996, pp. 1–16.

[13] P. McNamara, Deontic logic, in: E.N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, 2010, Fall 2010 edition.

[14] S. Miles, P. Groth, M. Luck, Handling mitigating circumstances for electronic contracts, in: Proceedings of the AISB 2008 Symposium on Behaviour Regulation in Multi-agent Systems, pp. 37–42.

[15] M.L. Mugnier, M. Leclère, On querying simple conceptual graphs with negation, Data Knowledge Engineering 60 (2007) 468–493.

[16] N. Oren, M. Croitoru, S. Miles, M. Luck, Understanding permissions through graphical norms, in: J. Leite, P. Torroni, T. Agotnes, G. Boella, L. van der Torre (Eds.), Declarative Agent Languages and Technologies VIII, 8th International Workshop, DALT 2010, Toronto, Canada, May 10, 2010, Revised, Selected and Invited Papers, Lecture Notes in Computer Science, vol. 6814, Springer, 2011, pp. 167–184.

[17] N. Oren, S. Panagiotidi, J. Vazquez-Salceda, S. Modgil, M. Luck, S. Miles, Towards a formalisation of electronic contracting environments, in: J.F. Hubner, E.T. Matson, O. Boissier, V. Dignum (Eds.), Coordination, Organizations, Institutions and Norms in Agent Systems IV, COIN@AAMAS 2008/COIN@AAAI 2008, Lecture Notes in Artificial Intelligence, 5428, Springer, 2008, pp. 156–171.

[18] Y. Shoham, M. Tennenholtz, On social laws for artificial agent societies: Offline design, Artificial Intelligence 73 (1995) 231–252.

[19] J.F. Sowa, Conceptual graphs, IBM Journal of Research and Development 20 (1976) 336–375.

[20] J.F. Sowa, Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, 1984.

[21] R. Thomopoulos, J.R. Bourguet, B. Cuq, A. Ndiaye, Short communication: answering queries that may have results in the future: a case study in food science, Knowledge-Based Systems 23 (2010) 491–495.

[22] W. Woods, J. Schmolze, The KL-ONE family, Computers and Mathematics with Applications 23 (1992) 133–177.

[23] G.H. von Wright, Deontic logic, Mind 60 (1951) 1–15.

Exclusivity-based Allocation of Knowledge

(Extended Abstract)

Madalina Croitoru
University of Montpellier 2
France
croitoru@lirmm.fr

Sebastian Rudolph
Karlsruhe Institute of Technology
Germany
rudolph@kit.edu

ABSTRACT

The classical setting of query answering either assumes the existence of just one knowledge requester, or the knowledge requests from different parties are treated independently from each other. This assumption does not always hold in practical applicaters where requesters often are in direct competition for knowledge. We propose a formal model for this type of scenario by introducing the Multi-Agent Knowledge Allocation (MAKA) setting which combines the fields of query answering in information systems and multi-agent resource allocation.

Categories and Subject Descriptors

K.6.0 [**Management of Computing and Information Systems**]: General—Economics

General Terms

Economics, Theory

Keywords

Auction and mechanism design

1. INTRODUCTION

Conjunctive query answering (between a knowledge requester and a knowledge provider) constitutes the de-facto standard of interacting with resources of structured information: databases or ontological information systems. The classical setting in query answering is focused on the case where just one knowledge requester is present. In case multiple requesters are present, the queries posed by different parties are processed and answered as independent from each other, thus making the multi-requester scenario a straightforward extension of the individual case.

While the above practice is natural in some cases, the assumption that queries can be processed independently clearly does not always hold in practical applications where the requesters are in direct competition for information. Let us consider for instance a multi-agent setting, with requester agents concurrently demanding information from a provider agent (example scenarios include military applications, news agencies, intelligence services, etc.). Of course, in this context, requester agents will not be willing to share "sensitive" information with other agents.

A structurally related problem is the multi-agent resource allocation (MARA) setting [2]. However, in such a setting (i) the agents ask for resources (not knowledge) and (ii) agents *a priori* know the pool of available resources. Work in this field either aims at bidding language expressiveness or algorithmic aspects of the allocation problem (see for instance [5, 1, 4] and others). The notion of multiplicity of resources, or resources used exclusively or shared has also been recently investigated in a logic-based language [6].

In the proposed multi-agent knowledge allocation (MAKA) setting, the n requester agents, at some given time (in a single-step), ask for knowledge (and not resources). They express their requests in the form of conjunctive queries that are endowed with exclusivity constraints and valuations, which indicate the subjective value of potentially allocated answers. Knowledge allocation poses interesting inherent problems not only from a bidding and query answering viewpoint, but also in terms of mechanism design.

The aim of this paper is to motivate and introduce the novel problem of Multi-Agent Knowledge Allocation and lay down future work directions opened by this setting: increased expressivity, dynamic allocations, fairness, multiple providers etc.

2. QUERYING WITH EXCLUSIVITY CONSTRAINTS

In [3] we fully introduce our framework of exclusivity-aware querying as a basis for the MAKA bidding formalism. In the following, we will just provide an intuitive overview of this work by the means of an example. Consider the following predicates: actor, director, singer (all unary), marriage and act (binary) and five constants AJ (Angelina Jolie), BP (Brad Pitt), MMS (Mr. and Ms. Smith), JB (Jessica Biel), JT (Justin Timberlake). A *knowledge base* consists of ground facts such as:

actor(AJ)	director(AJ)	marriage(AJ, BP)
actor(BP)	singer(JT)	act(AJ, MMS)
actor(JB)		act(BP, MMS)

If we consider a set of variables $V = \{x, y\}$ and the set of constants $C = \{AJ, BP, MMS, JB, JT\}$, then $actor(x)$, $act(y, MMS)$, $marriage(AJ, BP)$ are all *atoms* over the sets P and C.

Since in the MAKA scenario, requesters might be competing for certain pieces of knowledge, we have to provide them with the possibility of asking for an atom exclusively (exclusive) or not (shared). This additional information is captured by the notion of *exclusivity-annotated* atoms, ground facts and queries.

Some exclusivity-annotated atoms would for instance be: $\langle actor(x), sh \rangle$, $\langle marriage(AJ, BP), exc \rangle$ etc.

Note that the idea of exclusivity annotation is a novel concept going beyond the classical query answering framework. We assume an order exclusive $\succeq$ shared being used for query answering. It allows to specify concisely that an answer delivered exclusively is

suitable for a knowledge requester who demanded that information shared (but not vice versa).

For example, a query asking exclusively for marriages between actors and directors (where only the "marriage" itself is required as exclusive information, but the "actor" and "director" knowledge is sharable with other knowledge requester agents) is:

$$\langle\texttt{marriage}(x,y),\textsf{exclusive}\rangle\wedge$$
$$\big((\langle\texttt{actor}(x),\textsf{shared}\rangle\wedge\langle\texttt{director}(y),\textsf{shared}\rangle)\vee$$
$$(\langle\texttt{actor}(y),\textsf{shared}\rangle\wedge\langle\texttt{director}(x),\textsf{shared}\rangle)\big).$$

There is only one answer to this query w.r.t. our previously introduced knowledge base: $\mu = \{x \mapsto \texttt{AJ}, y \mapsto \texttt{BP}\}$. This means that $\texttt{marriage}(\texttt{AJ},\texttt{BP})$ can only be exclusively allocated (as $\langle\texttt{marriage}(\texttt{AJ},\texttt{BP}),\textsf{exclusive}\rangle$) but the $\texttt{director}(\texttt{AJ})$ and $\texttt{actor}(\texttt{BP})$ atoms can be either "shareably" allocated with other requesters ($\langle\texttt{actor}(\texttt{BP}),\textsf{shared}\rangle$) or exclusively allocated only to one requester agent ($\langle\texttt{director}(\texttt{AJ}),\textsf{exclusive}\rangle$).

3. THE KNOWLEDGE ALLOCATION PROBLEM DEFINED

Multi Agent Knowledge Allocation (MAKA) can be interpreted as an abstraction of a *market-based* centralized distributed knowledge-based system for query answering. In such a MAKA system, there is central node a, the *auctioneer* (or the *knowledge provider*), and a set of n nodes, $I = \{1, \ldots, n\}$, the *bidders* (or the *knowledge requesters*), which express their information need (including exclusivity requirements) via queries, which are to be evaluated against a knowledge base $\mathcal{K}$, held by the auctioneer.

The auctioneer asks bidders to submit in a specified common language, their *knowledge request*: $\langle q, \varphi \rangle$ where q is an exclusivity-annotated query and $\varphi : \mathbb{N} \to \mathbb{R}_+$ is a valuation function.

Following the ongoing example in the paper, a knowledge request for an exclusively known marriage between a known actor and a known director, where each such marriage information is paid 30 units for would be the singleton set $\{\langle q, \varphi \rangle\}$ with

$$q = \langle\texttt{marriage}(x,y),\textsf{exclusive}\rangle\wedge$$
$$\big((\langle\texttt{actor}(x),\textsf{shared}\rangle\wedge\langle\texttt{director}(y),\textsf{shared}\rangle)\vee$$
$$(\langle\texttt{actor}(y),\textsf{shared}\rangle\wedge\langle\texttt{director}(x),\textsf{shared}\rangle)\big),$$
$$\varphi = k \mapsto 30 \cdot k.$$

The valuation function $\varphi : \mathbb{N} \to \mathbb{R}_+$ can be defined in several ways. Assuming that $val_q^i \in \mathbb{R}_+$ denotes a bidder i's interest to obtain a single answer to a query q, standard valuation options are:

- naive valuation: $\varphi^n(|S|) = |S| \cdot val_q^i$,
- threshold valuation: $\varphi^t(|S|) = |S| \cdot val_q^i$ if $| S | \leq threshold_{q_i}^i$ and $|S| \cdot (val_q^i - discount_q^i)$ otherwise,
- budget valuation: $\varphi^b(|S|) = min\{\varphi_i(|S|), budget_i\}$ where φ_i can either be φ_i^n or φ_i^t.

Based on bidders' valuations, the auctioneer will determine a *knowledge allocation*, specifying for each bidder her obtained knowledge bundle and satisfying the *exclusivity constraints* (expressing that exclusivity annotations associated to atoms in the respective bundle are indeed complied with).

Given a knowledge base and a set of n bidders, a *knowledge allocation* is an n-tuple of subsets of the exclusivity-enriched knowledge base (i.e., the knowledge base atoms annotated with both exclusive and shared). An allocation needs to satisfy two conditions: First, we cannot allocate the same atom as both shared and exclusive. Second, an exclusive atom can only be allocated to one agent.

Given a knowledge allocation, one can compute its *global value* by summing up the individual prizes paid by the bidders for the share they receive. Obviously, the knowledge allocation problem aims at an *optimal allocation*, which maximizes this value.

Please see [3] providing more details and a full formalisation of the above intuitions, as well as a network representation of the problem, such that the winner determination can be cast into a max flow problem on the proposed graph structure.

4. CONCLUSION AND FUTURE WORK

We have introduced the problem of Multi-Agent Knowledge Allocation by drawing from the fields of query answering in information systems and combinatory auctions. To this end, we have sketched a bidding language based on exclusivity-annotated conjunctive queries. This approach opens up interesting work directions such as:

- **Extending the bidding language**: One straightforward extension would be to allow not just for ground facts (like $\texttt{marriage}(\texttt{AJ},\texttt{BP})$) to be delivered to the requester but also for "anonymized" facts (like $\texttt{marriage}(\texttt{AJ},*)$ or, more formally $\exists x.\texttt{marriage}(\texttt{AJ},x)$), which require handling adaption.

- **Extending knowledge base expressivity**: On one hand, the knowledge base formalism could be extended to cover not just ground facts but more advanced logical statements such as Datalog rules (used in deductive databases) or ontology languages. In that case, a distinction has to be made between propositions which are explicitly present in the knowledge base and those entailed by it.

- **Covering Dynamic Aspects of Knowledge Allocation**: In particular, in the area of news, dynamic aspects are of paramount importance: news items are annotated by time stamps and their value usually greatly depends on their timeliness. Moreover we can assume the information provider's knowledge pool to be continuously updated by incoming streams of new information.

- **Multiple Providers**: Finally, it might be useful to extend the setting to the case where multiple agents offer knowledge; in that case different auctioning and allocation mechanisms would have to be considered. This would also widen the focus towards distributed querying as well as knowledge-providing web-services.

5. REFERENCES

[1] C. Boutilier and H. Holger. Bidding languages for combinatorial auctions. In *Proc. IJCAI'01*, pages 1211–1217, 2001.

[2] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. MIT Press, 2006.

[3] M. Croitoru and S. Rudolph. Towards multi-agent knowledge allocation. Technical Report RR-12003(18), LIRMM, http://hal-lirmm.ccsd.cnrs.fr/lirmm-00657422/en/, 2012.

[4] A. Giovannucci, J. Rodriguez-Aguilar, J. Cerquides, and U. Endriss. Winner determination for mixed multi-unit combinatorial auctions via petri nets. In *Proc. AAMAS'07*, 2007.

[5] N. Nisan. Bidding and allocations in combinatorial auctions. In *Proc. EC-2000*, 2000.

[6] D. Porello and U. Endriss. Modelling combinatorial auctions in linear logic. In *Proc. KR'10*, 2010.

1250

How Much Should You Pay for Information?

Ioannis A. Vetsikas
National Center for Scientific Research "Demokritos",
Greece
ivetsikas@iit.demokritos.gr

Madalina Croitoru
University Montpellier 2,
France
croitoru@lirmm.fr

Abstract—The amount of data available greatly increases every year and information can be quite valuable in the right hands. The existing mechanisms for selling goods, such as VCG, cannot handle sharable goods, such as information. To alleviate this limitation we present our preliminary work on mechanisms for selling goods that can be shared or copied. We present and analyze efficient incentive compatible mechanisms for selling a single sharable good to bidders who are happy to share it.

I. INTRODUCTION

The evolution of the Web, and thus the facility of sharing data and putting data online has greatly improved, at least in the last decade. The data deluge can be noted in many day to day use cases: electronic journals access, music sharing, videos, social networks, open data initiatives etc. In the knowledge representation community (in a broad sense, and mainly in the database community) it is implicitly assumed that every answer to a query will be simply allocated to the user (unless constrained due to privacy restrictions); not to mention that the multiplicity of knowledge requesters was simply regarded as a simple extension of the individual case. However, in today's Web (Web of Data, Web of Science, Web of Knowledge, Semantic Web, Web 2.0 etc) information being given freely clearly does not always hold in practical applications where the requesters are in direct competition for information. The bottom line is that data, seen as an allocatable good, has the property of high cost production but negligible cost to copy. Studying implications of pricing information and allocating it thus becomes highly timely [1], [2]. The pricing information question on the web has also been investigated from a Linked Data perspective where information markets are being created [3], [4], [5].

A very important issue is that data can easily be shared (for example, music or software). If we were to apply well known auction mechanisms (e.g. VCG) to selling pieces of information that can be shared, hence are infinitely copied, no profit would be made; this would happen because competition is what drives the prices up [6] and offering more items - here infinite - than buyers essentially removes any competition in this setting. And yet getting profit is usually the first goal of a seller. Now, [7] examines expanding auctions where more copies of the same good are offered as the competition increases. However, their mechanism does not handle infinite copies and it is likely not *incentive compatible (IC)*, meaning that bidder would have an incentive to lie which would break down the mechanism. For this reason, it is important to propose IC mechanisms for this problem. Some mechanisms have been proposed in [8], [9] but they analyze the mechanisms from a worse case view point. As shown by [3] this is not always

relevant in practice where the distributions are not always a-priori known. Taking inspirations from these algorithm in this paper we present and analyze several incentive compatible mechanisms for selling a single sharable good to bidders who are happy to share it, aiming at creating competition by restricting the number of winners.

The only related work the examines this problem is [9]. However, this work is only concerned with examining the performance of the proposed mechanisms in the worst case, meaning how poor the performance becomes for any input even if this poor performance only occurs for extremely unlikely input. For a practical application, what would interest a company or an individual selling the information is the expected revenue that can be obtained from each mechanism. In view of this, in this paper, we thoroughly study mechanisms for selling goods that can be shared or copied as many times as necessary. We further the analysis of incentive compatible mechanisms, characterize a whole family of such mechanisms that can be used and evaluate the revenue obtained and the efficiency of these mechanisms, comparing also against the mechanisms of [9], showing that our mechanisms are better in average performance in most cases.

II. INCENTIVE COMPATIBLE MECHANISMS

In this section, we present formally the setting that we will address in this paper. We then present several incentive compatible mechanisms, starting from two baseline ones (mechanism $\mathcal{M}^{k+1}$ and $\mathcal{M}^r$) and then characterizing a family of such mechanisms, which generalizes the basic mechanism $\mathcal{M}^A$.[1] We subsequently evaluate these mechanisms in Section III.

We consider a set of n bidders want shared access to a piece of information. The valuations of the bidders who want shared access are $\overrightarrow{v} = \{v_1, \ldots, v_n\}$.

Now, the good could be also allocated to all the buyers but in this case we would not be able to extract any profit. For example, if we apply the well known VCG mechanism to this setting, the goods will be sold to all the buyers at a price equal to 0, thus making no money at all! This essentially happens because offering more items than buyers essentially removes any competition which would drive the prices up. There is a couple of straight forward ways to deal with this issue:

• One is **restricting the number of winners**. If the number of winners is fixed to a number k which is less than the actual number n of interested buyers, then this immediately

---

[1]This is a very basic and straight forward mechanism and a variation of this has been presented in [9].

will create competition between the bidders. By running an $(m+1)^{th}$ price auction (with $m = k$), henceforth denoted as mechanism $\mathcal{M}^{k+1}$, we can guarantee that this is IC and provides some profit while ensuring that the bidders will all pay the same price.

- Another is **setting a reserve price**. Instead of fixing the number of winners, a reserve price r is set; any bidder with valuation higher than r will buy the good. Now, the only IC mechanism in this instance is to make every winner pay r, henceforth denoted as mechanism $\mathcal{M}^r$; otherwise if the price paid depends on their bids, these bidders (who have valuations higher than r) will bid $r + \epsilon, \epsilon > 0$ instead, as they know that any bid above r will guarantee that they win.

A serious shortcoming of both these mechanisms ($\mathcal{M}^{k+1}$ and $\mathcal{M}^r$) is that the number of winners and the reserve price, respectively, should be selected optimally beforehand in order to maximize the revenue of the seller. This would need to rely on information such as prior knowledge of the distribution of the valuations. It cannot depend on the actual bidders valuations as then the mechanism would not be IC.

To alleviate this shortcoming, we need to design a mechanism that chooses the number of winners and subsequently the price that they pay so as to maximize the total revenue of the seller. Essentially, we should maximize $\max_{j \in \{1,...,n\}} j v^{(j)}$ where $v^{(j)}$ are the valuations of $\vec{v}$ ordered from highest to lowest. A first attempt would be to select j as to maximize $\max_{j \in \{1,...,n\}} j v^{(j+1)}$ instead and the price paid by the winners would be equal to $v^{(j+1)}$, the top bid that did not win. This is essentially the main idea from both the Vickrey (i.e. second price) auction [10] and the VCG mechanism. However, in this case *this mechanism is not IC*. The bidders can manipulate the price that they pay, and whether they win or not, by submitting a bid which is not their true value thus changing the number of winners j. It is relatively easy to check that neither a winner nor a loser can gain by increasing her bid. However, as shown next, both a winner and the $(j+1)-th$ bidder whose bid sets the price can gain by lowering their bids:

Example 1: Assume valuations $\vec{v} = \{11, 9, 8, 5, 3\}$. When all bidders declare their true values, then the mechanism selects $j = 2$ winners and they both pay $v^{(3)} = 8$.

- Any of the two top bidders would be able to profit by lowering her bid to $v_1' = 3.5$. So if the first one lies, then the ordered set of valuations would be $\{9,8,5,3.5,3\}$ and then the mechanism would select $j = 4$ winners all paying $v^{(5)} = 3$.
- The third highest bidder can also profit by lowering her bid to $v' = 3.5$, because then the ordered set of valuations would be $\{11,9,5,3.5,3\}$ and then the mechanism would again select $j = 4$ winners all paying a price of $v^{(5)} = 3$.

Why does this happen? While the price paid by a winner does not depend on her bid, the number of winners does, and therefore it is possible to indirectly manipulate the price paid. In fact, this is the reason why an IC mechanism must essentially ignore the bid of a bidder i when deciding whether bidder i is a winner and the price that she pays. The following mechanism $\mathcal{M}^A$ satisfies this requirement:

Definition 1 (IC Revenue Maximizing Mechanism $\mathcal{M}^A$):

For each bidder $i \in \{1, \ldots, n\}$ do:
  If $i > 1$ and $v^{(i)} = v^{(i-1)}$ then

decision is same as bidder with valuation $v^{(i-1)}$,
Else
  Compute j^* such that $j^* = \arg\max_j j v_{-i}^{(j)}$,[2]
    where $\overrightarrow{v_{-i}}$ is $\vec{v}$ without the valuation $v^{(i)}$
  If $v^{(i)} < v_{-i}^{(j^*)}$, bidder with value $v^{(i)}$ does not win
    otherwise, she is a winner and pays $v_{-i}^{(j^*)}$

Theorem 1: Mechanism $\mathcal{M}^A$ is IC.

As a variation of this mechanism has been presented in [9] and it is easy to prove that it is IC we will not do so here. *What we will focus on are the properties of this mechanism as they have not been analyzed in previous work* and they will be useful both in the experimental analysis we will conduct, as well as in generalizing it to the family of mechanisms we will later present.

We give two examples of how this mechanism works, the second of which contains tied valuations:

Example 2: Assume valuations $\vec{v} = \{11, 9, 7, 5, 3\}$.

- For bidder 1: $\overrightarrow{v_{-1}} = \{9, 7, 5, 3\}$, therefore $j^* = 3$ and since $v_1 = 11 \geq 5 = v_{-1}^{(3)}$ she wins with payment 5.
- For bidder 2: $\overrightarrow{v_{-2}} = \{11, 7, 5, 3\}$, therefore $j^* = 3$ and since $v_2 = 9 \geq 5 = v_{-2}^{(3)}$ she wins with payment 5.
- For bidder 3: $\overrightarrow{v_{-3}} = \{11, 9, 5, 3\}$, thus $j^* = 2$ and since $v_3 = 7 < 9 = v_{-3}^{(2)}$ she does not win.
- For bidder 4: $\overrightarrow{v_{-4}} = \{11, 9, 7, 3\}$, thus $j^* = 3$ and since $v_1 = 5 < 7 = v_{-4}^{(3)}$ she does not win.
- Bidder 5 (the bidder with the lowest valuation) can never win in this mechanism.

Therefore the two bidders with the highest valuations will win at a payment equal to the fourth highest bid of 5.

Example 3: Assume valuations $\vec{v} = \{10, 10, 7, 7, 7, 5\}$.

- For bidders 1 (and 2): $\overrightarrow{v_{-1}} = \{10, 7, 7, 7, 5\}$, therefore $j^* = 4$ and since $v_1 = 10 \geq 7 = v_{-1}^{(4)}$ they win with payment 7.
- For bidder 3 (**as well as** 4 **and** 5): $\overrightarrow{v_{-3}} = \{10, 10, 7, 7, 5\}$, therefore $j^* = 4$ and since $v_3 = 7 \geq 7 = v_{-3}^{(4)}$ they win (payment 7).
- Bidder 6 does not win.

Therefore the five bidders with the highest valuations will win at a payment equal to the fifth highest bid of 7.

Now, notice that had we been able to use all the information available the optimal allocation in Example 2 would have been to sell the good to the top three bidders at a price equal to 5. However, the third bidder is not a winner meaning that some efficiency has been lost. On the other hand, in Example 3, bidders 3, 4 and 5 pay their valuation. It makes sense to examine the properties of this mechanism, regarding which of the bidders win and the price that they pay. We will use the following lemma:

---

[2]In case that $\arg\max$ is a set, we define it to return the maximum element, meaning that if $\exists j_1 < j_2$ where $j_1 v_{-i}^{(j_1)} = j_2 v_{-i}^{(j_2)}$ are the maximizing terms, then $\arg\max$ will return j_2 and similarly if there are more maximizing terms.

Lemma 1: If the ith term is the maximizing one for the optimization problem solved by mechanism $\mathcal{M}^A$, when it disregards bidder i's bid, then it is also the maximizing term when the bid of bidder $i' = i+1$ is disregarded. Furthermore, if the maximizing term is instead the j-th one (where $j \neq i$), when disregarding bidder i's bid, then either it remains the maximizing term, when disregarding bidder i''s bid ($i' = i+1$), or the new i^{th} term is the maximizing one.

Proof: Due to space this proof as well as the proofs of the following theorem are omitted. They can be found in our technical report available at http://hal.inria.fr/lirmm-00830805/. ∎

We obtain the following theorem:

Theorem 2: When using mechanism $\mathcal{M}^A$ some number j' of the top bidders will win, where $j' \leq j^* + 1$ and $j^* = \arg\max_j j v^{(j+1)}$. The price that they all pay is equal to $v^{(j^*+1)}$. Furthermore $j' \leq j^*$, when there are no ties in the bidders' valuations.

We can observe that this mechanism has some very desirable properties beyond being simply IC: firstly, all the winners pay the same price, so there can be no envy really among them, and, secondly, they pay the price $v^{(j^*+1)}$ that is the one that maximizes the profit of the seller. However, in order for the profit of the seller and the efficiency of the system to be maximized it should be that j^* (or $j^* + 1$) bidders should win at this price. To alleviate this weakness of mechanism $\mathcal{M}^A$, we will examine some variations of it, *eventually generalizing it to a whole family of mechanisms*. Essentially, the maximization step $j^* = \arg\max_j j v_{-i}^{(j)}$ is replaced by a voting protocol.

First, notice that the desired optimization $\max_{j \in \{1,\dots,n\}} j v^{(j)}$ and the maximization step $j^* = \arg\max_j j v_{-i}^{(j)}$ of mechanism $\mathcal{M}^A$ are the same optimization problem when bidder i is the one with the highest valuation (her valuation is ranked first). In other cases though, they can lead to different results and this is the reason for the inefficiency. We cannot use any knowledge of the bidder's value when deciding when she's a winner or not, not even the rank (i.e. how many other bidders have a higher valuation). Therefore, we propose to examine all possible cases for the rank of the valuation of bidder i and then aggregate the "optimal" number of winners in each case via a voting protocol.

To illustrate what we mean we re-examine the setting of Example 2.

Example 4: Assume valuations $\overrightarrow{v} = \{11, 9, 7, 5, 3\}$.

- For bidder 1, it is $\overrightarrow{v_{-1}} = \{9, 7, 5, 3\}$. For her valuation we can assume the following cases:

  1) $v_1 \geq 9$. Then the set of valuations is $\{v_1, 9, 7, 5, 3\}$, and it would be optimal to have 3 winners.
  2) $9 > v_1 \geq 7$. Similarly 3 winners is the optimal.
  3) $7 > v_1 \geq 5$. Similarly 3 winners is the optimal.
  4) $5 > v_1 \geq 3$. Then the set of valuations is $\{9, 7, 5, v_1, 3\}$, and it would be optimal to either have 3 (when $v_1 \geq 4$) or 4 (when $v_1 < 4$) winners. Only in the second subcase would bidder 1 win.
  5) $v_1 < 3$. Then the whole set of valuations would be $\{9, 7, 5, 3, v_1\}$, and it would be optimal to either have

  4 (when $v_1 \geq 2.5$) or 2 (when $v_1 < 2.5$) winners. In either case bidder 1 would not be selected as a winner.

- For bidder 2, we obtain similar results as for bidder 1.
- For bidder 3: $\overrightarrow{v_{-3}} = \{11, 9, 5, 3\}$. For her valuation we can assume the following cases:

  1) $v_3 \geq 11$. Then the set of valuations is $\{v_3, 11, 9, 5, 3\}$, and it would be optimal to have 2 winners.
  2) $11 > v_3 \geq 9$. Similarly 2 winners is the optimal.
  3) $9 > v_3 \geq 5$. Then the whole set of valuations would be $\{11, 9, v_3, 5, 3\}$, and it would be optimal to have 2 (when $v_3 \geq 7.5$) or 3 (when $v_3 < 7.5$) winners. Only in the second subcase would bidder 3 win.
  4) $5 > v_3 \geq 3$. Then the whole set of valuations would be $\{11, 9, 5, v_3, 3\}$, and it would be optimal to either have 3 (when $v_3 \geq 4$) or 4 (when $v_3 < 4$) winners. Only in the second subcase would bidder 3 win.
  5) $v_3 < 3$. Then the whole set of valuations would be $\{11, 9, 5, 3, v_3\}$, and it would be optimal to either have 4 (when $v_1 \geq 2.5$) or 2 (when $v_1 < 2.5$) winners. In either case bidder 3 would not be selected as a winner.

- For bidder 4: $\overrightarrow{v_{-4}} = \{11, 9, 7, 3\}$. For her valuation we can assume the following cases:

  1) $v_4 \geq 11$. Then the set of valuations is $\{v_4, 11, 9, 7, 3\}$, and it would be optimal to have 3 winners.
  2) $11 > v_4 \geq 9$. Similarly 3 winners is the optimal.
  3) $9 > v_4 \geq 7$. Similarly 3 winners is the optimal.
  4) $7 > v_4 \geq 3$. Then the whole set of valuations would be $\{11, 9, 7, v_4, 3\}$, and it would be optimal to either have 3 (when $v_4 \geq 14/3$) or 2 (when $v_3 < 14/3$) winners.
  5) $v_4 < 3$. Then the whole set of valuations would be $\{11, 9, 7, 3, v_4\}$, and it would be optimal to either have 4 (when $v_1 \geq 2.5$) or 2 (when $v_1 < 2.5$) winners.

- Bidder 5 (the bidder with the lowest valuation) can never win (unless there is a tie which we choose not to consider when designing our mechanisms, as it happens with very low probability), therefore we do not analyze her case.

What can we observe from this example? Examining what happens each time we tried to solve the optimization problem for each bidder, the optimal number of winners changes as the valuation of that bidder is assumed to various ranges of values; of course the knowledge of this value is ignored in order to keep the mechanism IC, this is the reason why we need to examine all these possible cases. Now, note that considering only the case when this value is assumed to be higher than the highest among the remaining valuations and basing the decision on only that case, gives mechanism $\mathcal{M}^A$. However, this does not use the information from all the other cases where the bidder examined might still be a winner. Thus, we propose to use a voting protocol where, for each case where the bidder examined is selected to be a winner, votes would be cast for the number of winners that maximize the total profit.

We see that the decision regarding each bidders depends on the cases examined. Thus, we generalize the previous mechanism to consider all cases examined. To this end, we

propose the following family of mechanisms $\mathcal{M}^*$ where each case (i.e. when the rank of the missing valuation is k) casts votes with weight w_k:

Definition 2 (Family of IC Mechanisms $\mathcal{M}^*(\overrightarrow{w}, \delta)$):
Select the function $\delta(<profit>, <max\_profit>)$
For each bidder $i \in \{1, \ldots, n\}$ do:
  If $i > 1$ and $v^{(i)} = v^{(i-1)}$ then
    decision is same as bidder with valuation $v^{(i-1)}$,
  Else
    Set $\overrightarrow{v_{-i}}$ as $\overrightarrow{v}$ without the valuation $v^{(i)}$
    Set $\psi_k = 0, \forall k = 1, \ldots, n$
    For $k = 1, \ldots, n-1$ do
      Assume that the missing valuation (denoted v) is
      $v_{-i}^{(k-1)} > v \geq v_{-i}^{(k)}$, where $v_{-i}^{(0)} = \infty$ & $v_{-i}^{(n)} = 0$
      Set the weight w_k
      Define the terms $t_l = \begin{cases} (l-1)v_{-i}^{(l)}; l < k, \\ lv_{-i}^{(l)}; l \geq k. \end{cases}$
      Among these terms, find the highest: l_1
      and the second highest: l_2
      The min and max values of term $(k-1)v$ are resp.:
      $t_{min} = (k-1)v_{-i}^{(k)}$ and $t_{max} = (k-1)v_{-i}^{(k-1)}$
      If $t_{l_1} \geq t_{max}$ and $l_1 \geq k$ then
        $\psi_{l_1} = \psi_{l_1} + w_k$ (full vote for best - weighted)
      If $t_{l_2} \geq t_{max}$ and $l_1 \geq k$ then
        $\psi_{l_1} = \psi_{l_1} + w_k \delta(t_{l_2}, t_{l_1})$ (partial vote for 2nd best)
      If $t_{max} > t_{l_1} \geq t_{min}$ and $l_1 \geq k$ then
        $\psi_{l_1} = \psi_{l_1} + w_k \left(\frac{t_{l_1} - t_{min}}{t_{max} - t_{min}} + \int_{t_{l_1}}^{t_{max}} \frac{\delta(t_{l_1}, x)}{t_{max} - t_{min}} dx \right)$
      If $t_{max} > t_{l_2} \geq t_{min}$ and $l_2 \geq k$ then
        $\psi_{l_2} = \psi_{l_2} + w_k \frac{t_{l_2} - t_{min}}{t_{max} - t_{min}} \delta(t_{l_2}, t_{l_1})$
    Select $j^* = \arg\max \psi_j$
    If $v^{(i)} < v_{-i}^{(j^*)}$, bidder with value $v^{(i)}$ does not win
    otherwise, she is a winner and pays $v_{-i}^{(j^*)}$

The two lines that are presented in bold define the parameters that characterize the whole range of mechanisms that belong to this family of mechanisms $\mathcal{M}^*$. For example, mechanism $\mathcal{M}^A$, which we presented earlier, is derived from $\mathcal{M}^*$, by setting $\delta() = 0$, $w_1 = 1$ and $w_k = 0, \forall k > 1$. In this paper we will also use in our experiments, the following two mechanisms which are derived from $\mathcal{M}^*$:

- mechanism $\mathcal{M}^V$, in which $\delta(x, y) = \frac{x}{y}$ and $w_k = 1, \forall k$

- mechanism $\mathcal{M}^W$, in which $\delta(x, y) = \frac{x}{y}$ and $w_1 = 1$, while $w_k = (n-2)\frac{v_{-i}^{(k)} - v_{-i}^{(k-1)}}{v_{-i}^{(1)} - v_{-i}^{(n-1)}}, \forall k > 1$

In both these mechanisms, the best option gets 1 vote while the second best option (regarding the number of winners) gets votes equal to the ratio of the second highest and the highest profits. However, in the first mechanism, the weights for all cases are 1, while in the second the votes are weighted depending on how likely each case $v_{-i}^{(k-1)} > v \geq v_{-i}^{(k)}$ is, which depends on the distance between the values $v_{-i}^{(k-1)}$ and $\geq v_{-i}^{(k)}$.[3]

---

[3] Given that we do not use any prior information regarding the distribution of valuations, this is the most logical way to assign probabilities to each case. Essentially, as we do not assume any knowledge of the distribution of valuations, we will approximate it as a uniform distribution with bounds the lowest and highest values that are known at each step of the algorithm.
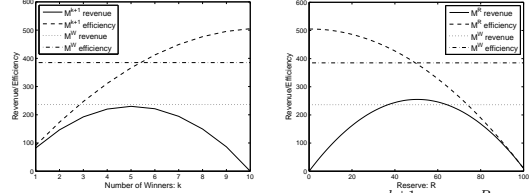
Fig. 4. Revenue and efficiency of mechanisms $\mathcal{M}^{k+1}$ and $\mathcal{M}^R$ as the number of winners k and the reserve price R varies (versus those of $\mathcal{M}^W$). The number of bidder $n = 10$.

Example 5: Assume valuations $\overrightarrow{v} = \{11, 9, 7, 5, 3\}$.

- For bidder 1, examining all cases ($k = 1 \ldots 4$) we get that having three winners gets the most votes ($\psi_3 = 3$). So the bidder wins and pays $v_{-i}^{(3)} = 5$.

- For bidder 2, we obtain the same results and price.

- For bidder 3: $\overrightarrow{v_{-3}} = \{11, 9, 5, 3\}$ and we examine cases:

  1) $k = 1 : v_3 \geq 11$. The valuations are $\{v_3, 11, 9, 5, 3\}$. The best choices are $l_1 = 2$ and $l_2 = 3$, thus $\psi_2 = \psi_2 + 1$ and $\psi_3 = \psi_3 + \frac{5}{6}$.
  2) $k = 2 : 11 > v_3 \geq 9$. Similarly, the mechanism updates $\psi_2 = \psi_2 + 1$ and $\psi_3 = \psi_3 + \frac{5}{6}$.
  3) $k = 3 : 9 > v_3 \geq 5$. The valuations are $\{11, 9, v_3, 5, 3\}$, thus $l_1 = 3$, $l_2 = 4$, $t_{max} = 18$ and $t_{min} = 10$. Therefore $\psi_3 = \psi_3 + 0.967$ and $\psi_4 = \psi_4 + \frac{1}{5}$.
  4) $k = 4 : 5 > v_3 \geq 3$. The valuations are $\{11, 9, 5, v_3, 3\}$, thus $l_1 = 4$, $l_2 = 3$ (but $l_2 < k = 4$ so votes are cast), while $t_{max} = 15$ and $t_{min} = 9$. Therefore $\psi_4 = \psi_4 + 0.9463$.

The tally of votes is: $\psi_2 = 2$, $\psi_3 = 2.63$ and $\psi_4 = 1.15$, hence $j^* = 3$. Hence bidder 3 wins and pays $v_{-i}^{(j^*)} = 5$.

- For bidder 4: $\overrightarrow{v_{-4}} = \{11, 9, 7, 3\}$. Following the same reasoning, $j^* = 3$ ($\psi_3 = 3$) and therefore she does not win.

To summarize, we find that under mechanism $\mathcal{M}^V$, the three bidders with the highest valuations (11, 9 and 7) would win and each pays a price equal to 5. However, this does not mean that mechanism $\mathcal{M}^V$ work always best; as we will see in the experimental evaluation, when the valuations can take only a couple of possible values then mechanism $\mathcal{M}^A$ is better!

III. EXPERIMENTAL EVALUATION

In this section, we conduct experiments to evaluate the performance of the mechanisms we presented in the previous section. Our goal is to compare the seller revenue and the efficiency (i.e. the sum of the valuations of all winners) of the different mechanisms $\mathcal{M}^A$, $\mathcal{M}^V$ and $\mathcal{M}^W$ as opposed to using the baseline mechanisms $\mathcal{M}^{k+1}$ and $\mathcal{M}^R$. We will also compare them with the Parameterized Random Sampling Optimal Price auction ($RSOP_r$) presented in Section 6.1 of [9]; we have implemented an improvement of this mechanism which does not allow one of the two randomized sets to be empty[4] and

---

[4] The original algorithm which did not impose this restriction had much worse performance, and it would always perform worse than the other algorithms, therefore it does not provide a good enough benchmark without this (minor) modification, as this significantly improves the mechanism's performance.

No Agents		3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
M^{k+1}	PR	50,98	80,98	101,45	130,08	151,95	179,78	202,38	229,90	253,11	279,98	303,60	330,25	353,91	380,60	404,35	430,86	454,91	481,00
	EF	126,04	140,96	201,44	215,75	277,02	290,89	352,38	366,17	428,10	441,51	503,53	516,90	578,95	592,24	654,45	667,66	729,90	743,01
M^R	PR	76,53	101,97	127,42	152,97	178,47	204,00	229,41	255,06	280,65	306,08	331,59	357,09	382,43	408,12	433,35	459,04	484,30	509,89
	EF	113,30	152,93	188,62	226,44	264,26	302,00	339,58	382,62	421,02	449,14	497,35	528,66	573,64	604,17	641,56	679,58	726,45	754,81
M^A	PR	57,81	81,51	105,81	130,34	154,72	179,36	203,96	228,79	253,53	278,37	303,12	327,94	352,84	378,01	402,78	427,87	452,62	477,78
	EF	92,54	124,87	160,07	195,82	231,47	267,38	303,31	339,75	376,06	412,44	449,06	485,54	522,18	559,23	595,79	632,76	669,56	706,52
M^V	PR	57,21	83,54	109,05	134,50	159,75	184,87	209,87	235,13	260,11	285,11	310,09	335,06	360,09	385,27	410,02	435,20	460,05	485,08
	EF	114,05	155,94	194,53	232,74	270,46	307,85	345,12	382,78	420,11	457,36	494,84	532,21	569,50	607,15	644,37	681,95	719,41	756,91
M^W	PR	57,21	83,48	109,91	135,54	160,81	185,99	211,05	236,28	261,27	286,20	311,15	336,10	361,06	386,18	410,89	435,99	460,84	485,81
	EF	114,05	158,31	196,53	234,76	272,50	310,03	347,34	385,00	422,37	459,60	497,05	534,41	571,67	609,27	646,44	683,99	721,39	758,82
M^P	PR	53,599	74,13	95,6032	118,02	141,12	164,48	188,79	213,61	238,09	262,87	287,13	311,43	336,55	361,8	386,59	411,55	435,85	461,7
	EF	98,477	128,76	159,13	190,18	222,28	254,92	319,48	349,3	379,57	410,63	441,89	474,62	538,46	568,07	598,27	629,67	661,45	725,99

Fig. 1. Table of experimental results for 3 up to 20 bidders. The valuation distribution used is Uniform$\{1, \ldots, 100\}$. (PR stands for profit; EF for efficiency) The error of the simulation is no larger than the first decimal point of each of the results presented in all the tables.

No Agents		3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
M^{k+1}	PR	5,50	7,61	11,00	13,80	16,51	20,19	23,14	26,47	30,11	33,01	36,93	40,45	43,62	47,74	51,00	54,88	58,68	61,84
	EF	8,88	16,62	18,03	18,87	27,33	28,34	29,00	37,86	38,62	39,14	48,26	48,87	57,92	58,59	59,06	68,32	68,85	78,04
M^R	PR	15,01	19,99	25,00	29,98	35,00	39,98	44,99	50,04	54,98	60,01	65,02	69,99	75,01	80,04	85,01	90,00	95,00	99,97
	EF	15,01	19,99	25,00	29,98	35,00	39,98	44,99	50,04	54,98	60,01	65,02	69,99	75,01	80,04	85,01	90,00	95,00	99,97
M^A	PR	12,00	17,98	23,75	29,23	34,57	39,73	44,85	49,96	54,50	59,73	64,86	69,90	74,95	80,01	84,99	89,99	94,99	99,96
	EF	15,38	20,24	25,16	30,08	35,06	40,01	45,00	50,05	55,04	60,04	65,04	70,01	75,01	80,04	85,01	90,00	95,00	99,97
M^V	PR	12,00	11,98	18,75	25,48	31,95	32,74	39,77	46,46	47,42	54,56	61,22	67,41	73,29	78,91	84,28	89,54	94,71	99,79
	EF	15,38	20,99	25,78	30,55	35,39	40,89	45,64	50,49	55,93	60,69	65,50	70,32	75,22	80,18	85,10	90,05	95,03	99,99
M^W	PR	12,00	11,98	18,75	25,48	31,95	32,74	39,77	46,46	47,42	54,56	61,22	67,41	73,29	78,91	84,28	89,54	94,71	99,79
	EF	15,38	20,99	25,78	30,55	35,39	40,89	45,64	50,49	55,93	60,69	65,50	70,32	75,22	80,18	85,10	90,05	95,03	99,99
M^P	PR	10,00	14,56	19,58	24,86	30,38	35,92	41,49	47,11	52,54	57,99	63,37	68,64	73,89	79,11	84,22	89,30	94,38	99,39
	EF	15,63	20,67	25,68	30,62	35,58	40,49	45,42	50,41	55,29	60,26	65,23	70,16	75,14	80,15	85,11	90,08	95,07	100,04

Fig. 2. Table of experimental results for 3 up to 20 bidders. The values have a 50% chance of being 1 or 10.

No Agents		3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
M^{k+1}	PR	130,25	187,98	244,16	299,43	354,31	408,74	462,92	517,07	571,22	625,21	679,23	733,17	787,27	841,28	895,35	949,47	1003,55	1057,77
	EF	167,37	247,33	326,51	405,04	483,45	561,61	639,61	717,53	795,34	873,23	950,93	1028,57	1106,27	1183,89	1261,53	1339,12	1416,77	1494,28
M^R	PR	165,00	220,00	275,00	330,00	385,00	440,00	495,00	550,00	605,00	660,00	715,00	770,00	825,00	880,00	935,00	990,00	1045,00	1100,00
	EF	232,49	309,99	387,55	464,93	542,50	620,00	697,47	774,99	852,47	930,07	1007,53	1084,97	1162,50	1239,97	1317,49	1394,97	1472,52	1549,95
M^A	PR	140,25	192,74	246,78	301,91	355,85	409,94	463,81	519,89	573,39	629,98	684,54	738,76	793,01	845,24	899,57	954,10	1011,17	1061,95
	EF	177,37	247,13	318,75	392,68	466,62	537,40	609,97	689,65	761,12	842,80	918,95	985,07	1056,20	1123,85	1199,09	1276,60	1360,26	1425,02
M^V	PR	140,25	200,35	258,77	314,98	370,00	424,73	478,98	534,90	587,89	644,25	698,55	752,93	807,55	861,06	915,97	971,40	1028,09	1080,89
	EF	177,37	259,71	341,12	419,44	495,61	571,06	645,52	725,21	795,58	877,05	952,39	1025,53	1098,69	1171,27	1248,27	1328,42	1411,62	1481,12
M^W	PR	140,25	200,35	258,77	315,93	370,68	424,97	480,12	535,53	588,63	644,64	699,14	754,66	808,30	862,24	916,43	971,94	1028,38	1081,10
	EF	177,37	259,71	341,12	421,21	496,89	571,51	647,76	726,52	797,06	877,80	953,62	1031,55	1100,47	1173,99	1249,17	1329,42	1412,21	1481,52
M^P	PR	123,68	174,77	228,86	284,93	342,07	399,95	457,69	515,53	573,21	630,60	687,65	744,53	801,39	857,74	914,30	970,38	1026,55	1082,67
	EF	152,56	218,63	289,91	364,93	442,30	521,49	601,42	681,79	762,32	842,92	923,24	1003,60	1083,96	1163,91	1244,13	1323,87	1403,83	1483,58

Fig. 3. Table of experimental results for 3 up to 20 bidders. The values have a 10% chance of being $55, 60, 65, \ldots, 95, 100$.

we assume knowledge of the distribution in order to select the optimal value of the parameter r for this mechanism. In the experiments we call this mechanism $\mathcal{M}^P$ (i.e. probabilistic). Note that in [9], another algorithm is proposed for this problem: the Random Sampling Profit Extraction auction (RSPE); however the performance of this mechanism is very poor as it sacrifices half the profit (in most cases), therefore we chose not to include it in our experiments.

Now, there is very little research and knowledge on what real distributions of the valuations for data are like. Some information on current data markets is given in [3], however very little is known about the real values for such data. Obviously, different distributions would affect to some degree the performance of the different mechanisms. In view of this, we present here three sets of experiments each performed with a different valuation distribution. We explain why we select each, in turn, before presenting the results of the simulation.

Experiment Set 1: We simulate n bidders whose valuations are i.i.d. random variables drawn from the uniform distribution on $\{1, \ldots, 100\}$. The number of bidders n varies from 3 to 20. For the baseline mechanisms $\mathcal{M}^{k+1}$ and $\mathcal{M}^R$, we calculate beforehand the best values for k and R respectively that maximize the expected revenue using the knowledge of the distribution from which the bids are drawn; *for the other*

mechanisms no such knowledge is necessary.

The results of these experiments are presented in the table of Figure 1. The best revenue among the mechanisms $\mathcal{M}^A$, $\mathcal{M}^V$ and $\mathcal{M}^W$ (of family $\mathcal{M}^*$) is consistently obtained by mechanism $\mathcal{M}^W$. Its revenue is actually better than that of mechanism $\mathcal{M}^{k+1}$ (with k set optimally to maximize revenue), because mechanism $\mathcal{M}^W$ adjusts the number of winners based on the actual bids submitted rather than choosing the same number regardless of the input. On the other hand, mechanism $\mathcal{M}^R$ (with R set optimally) clearly outperforms the other mechanisms, because it uses its knowledge of the expected valuations to set a threshold (a reserve price) R which must be paid by all winning bidders. In this way it balances the revenue from each winner against the number of bidders. However, this mechanism is very dependent on knowing the distribution of valuations, as setting the reserve R to the wrong value will reduce very significantly the revenue obtained. In fact, to examine this effect we present in Figure 4, the revenue and efficiency of mechanisms $\mathcal{M}^{k+1}$ and $\mathcal{M}^R$ for different values of k and R respectively. We observe that the revenue obtained from $\mathcal{M}^R$ degrades significantly if R deviates by more than 15 from its optimal value. What is worse, if mistakenly the valuations were assumed to be between 1 and 50 (or, even worse, between 1 and 200), which would have approximately

halved (or doubled resp.) the value for R, then very little revenue would be obtained!

Regarding the efficiency of the mechanisms, we observe that all of them perform similarly. The only exception is mechanism $\mathcal{M}^A$, because, as we've seen in Example 2, sometimes the mechanism reduces the number of winners which significantly impacts its efficiency.

Finally, we notice that mechanism $\mathcal{M}^P$ performs worst than any other mechanism (even $\mathcal{M}^{k+1}$ in many cases). This happens because the mechanism splits the problem into two separate problems and uses the solution (i.e. the best price) of one to impose the cutoff price for the other; however, the solutions for these two problems are not always very close (or identical which would be the optimal case) and this leads to a loss of revenue and some efficiency.

Experiment Set 2: In the previous experiment set, we assumed that the valuations could take a continuum of values. The extreme opposite of this is that only two values are possible, thus we assume here that the values have a 50% chance of being either 1 or 10. While we do not believe that this could be realistic (to have so small a number of possible values), this case is suggested in [9] as the case when the deterministic algorithms would fail to produce good results.

The results of these experiments are presented in the table of Figure 2. We observe that the performance of the baseline mechanisms is similar in broad terms to the previous experiment set. Regarding the other mechanisms, now mechanism $\mathcal{M}^A$ performs best, better than mechanisms $\mathcal{M}^V$ and $\mathcal{M}^W$. This is not entirely surprising as the possible values are only two, which means that in almost all cases the optimal decision would be to select the bidder with value 10 and make them pay 10, which matches the maximum profit that can ever be extracted from any (not necessarily IC) mechanism; the other more complex mechanisms $\mathcal{M}^V$ and $\mathcal{M}^W$ try to be cleverer, but that is unnecessary and they suffer a bit because of this.

Furthermore, mechanism $\mathcal{M}^P$ actually shines in this case, even if it is outperformed by mechanism $\mathcal{M}^A$: because of having only two possible values, and the solutions for the two problems solved by the mechanism are almost always identical, therefore this leads to almost maximal revenue and efficiency.

Experiment Set 3: We mentioned that the second distribution is probably not realistic for real data markets. However, this does pose the question what happens in an intermediate case, where there are relatively few possible values (but still not as few as only two). To this end, we assumed for the third experiment set we conducted, that the values have a 10% chance of being $55, 60, 65, \ldots, 95, 100$ (10 possible values in total).

The results of these experiments are presented in the table of Figure 3. In this case, we notice that the observations of the different mechanisms performance are close to those made for the first experiment set. In particular, disregarding the baseline mechanisms, mechanism $\mathcal{M}^W$ performs best in this set closely followed by mechanism $\mathcal{M}^V$. Mechanism $\mathcal{M}^P$ lacks in performance to a substantial degree (the exception being when the number of bidders n approaches 20) and so does mechanism $\mathcal{M}^A$.

To summarize our observations from all the experiment sets, we notice that the baseline mechanism $\mathcal{M}^R$ is overall consistently the best, but it relies significantly on selecting the best reserve R. Our proposed mechanisms (and in particular $\mathcal{M}^W$ and $\mathcal{M}^V$) are typically the best among the other mechanisms. The exception to this is when there are very few (two or close) possible valuations when it is more advantageous to use mechanisms $\mathcal{M}^A$ primarily and $\mathcal{M}^P$ secondary. However, we remind the reader that mechanism $\mathcal{M}^P$ also relies on using knowledge of the valuation distribution in order to select the optimal parameter r, albeit to a lesser extend than mechanism $\mathcal{M}^R$.

IV. CONCLUSIONS

In this paper, we studied mechanisms for selling sharable information goods. We presented and analyzed several IC mechanisms, including a family of such mechanisms, for selling a single sharable good to bidders who are happy to share it; furthermore, we analyzed the properties of these mechanisms via simulations (for the most part).

There are still a number of avenues for future work. The most important extension is to examine whether we can generalize our mechanisms to the case where several goods are sold to bidder who want to buy bundles of these and are willing to share or would want each good exclusively. Furthermore, for the single unit case examined in this paper, the mechanisms of family $\mathcal{M}^*$ restrict the number of winners, in a similar manner to mechanism $\mathcal{M}^{k+1}$, the difference being not using prior knowledge; in this spirit, our second extension will examine new mechanisms that estimate a reserve price (like $\mathcal{M}^R$, the highest revenue mechanism we considered) without using prior knowledge about the valuation distribution.

REFERENCES

[1] H. R. Varian, "Buying, sharing and renting information goods," *The Journal of Industrial Economics*, vol. 48, no. 4, pp. 473–488, 2000.

[2] F. Linde and W. Stock, *Information Markets: A Strategic Guideline for the I-Commerce*. Walter de Gruyter, 2011.

[3] S. F. Stahl, F. and G. Vossen, "Marketplaces for data: An initial survey," no. 14, 2012.

[4] P. Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu, "Query-based data pricing," in *PODS*, M. Benedikt, M. Krötzsch, and M. Lenzerini, Eds. ACM, 2012, pp. 167–178.

[5] C. Li, D. Y. Li, G. Miklau, and D. Suciu, "A theory of pricing private data," in *ICDT*, W.-C. Tan, G. Guerrini, B. Catania, and A. Gounaris, Eds. ACM, 2013, pp. 33–44.

[6] V. Krishna, *Auction theory*. Academic Press, 2002.

[7] O. Shehory and E. Dror, "Computationally efficient and revenue optimized auctioneer's strategy for expanding auctions," in *Proceedings of AAMAS'06*, ser. AAMAS '06, 2006, pp. 1175–1182.

[8] M. Babaioff, R. Kleinberg, and R. Paes Leme, "Optimal mechanisms for selling information," in *Proceedings of the 13th ACM Conference on Electronic Commerce*, ser. EC '12. New York, NY, USA: ACM, 2012, pp. 92–109.

[9] A. V. Goldberg, J. D. Hartline, A. Karlin, M. Saks, and A. Wright, "Competitive auctions," *Games and Economic Behavior*, vol. 55, no. 2, pp. 242–269, 2006. A previous version appeared as: A. V. Goldberg, J. D. Hartline, and A. Wright, "Competitive auctions and digital goods," in *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2001, pp. 735–744.

[10] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of Finance*, vol. 16, no. 1, pp. 8–37, 1961.

What Can Argumentation Do for Inconsistent Ontology Query Answering?
–Technical Report–

Madalina Croitoru[1], Srdjan Vesic[2][*]

[1]INRIA, LIRMM, Univ. Montpellier 2, France; [2]CRIL - CNRS, France

Abstract. The area of inconsistent ontological knowledge base query answering studies the problem of inferring from an inconsistent ontology. To deal with such a situation, different semantics have been defined in the literature (e.g. AR, IAR, ICR). Argumentation theory can also be used to draw conclusions under inconsistency. Given a set of arguments and attacks between them, one applies a particular semantics (e.g. stable, preferred, grounded) to calculate the sets of accepted arguments and conclusions. However, it is not clear what are the similarities and differences of semantics from ontological knowledge base query answering and semantics from argumentation theory. This paper provides the answer to that question. Namely, we prove that: (1) sceptical acceptance under stable and preferred semantics corresponds to ICR semantics; (2) universal acceptance under stable and preferred semantics corresponds to AR semantics; (3) acceptance under grounded semantics corresponds to IAR semantics. We also prove that the argumentation framework we define satisfies the rationality postulates (e.g. consistency, closure).

1   Introduction

Ontological knowledge base query answering problem has received renewed interest in the knowledge representation community (and especially in the Semantic Web domain where it is known as the ontology based data access problem [17]). It considers a consistent ontological knowledge base (made from facts and rules) and aims to answer if a query is entailed by the knowledge base (KB). Recently, this question was also considered in the case where the KB is *inconsistent* [16, 8]. Maximal consistent subsets of the KB, called *repairs*, are then considered and different *semantics* (based on classical entailment on repairs) are proposed in order to compute the set of accepted formulae.

   Argumentation theory is also a well-known method for dealing with inconsistent knowledge [5, 2]. Logic-based argumentation [6] considers constructing arguments from

---

[*] The major part of the work on this paper was carried out while Srdjan Vesic was affiliated with the Computer Science and Communication Research Unit at the University of Luxembourg. During this period, Srdjan Vesic's project was supported by the National Research Fund, Luxembourg, and cofunded under the Marie Curie Actions of the European Commission (FP7-COFUND). At the time when the authors were finishing the work on this paper, Srdjan Vesic was a CRNS researcher affiliated with CRIL.

inconsistent knowledge bases, identifying attacks between them and selecting acceptable arguments and their conclusions. In order to know which arguments to accept, one applies a particular *argumentation semantics*.

This paper starts from the observation that both inconsistent ontological KB query answering and instantiated argumentation theory deal with the same issue, which is reasoning under inconsistent information. Furthermore, both communities have several mechanisms to select acceptable conclusions and they both call them *semantics*. The *research questions* one could immediately ask are: Is there a link between the semantics used in inconsistent ontological KB query answering and those from argumentation theory? Is it possible to instantiate Dung's ([15]) abstract argumentation theory in a way to implement the existing semantics from ontological KB query answering? If so, which semantics from ontological KB query answering correspond to which semantics from argumentation theory? Does the proposed instantiation of Dung's abstract argumentation theory satisfy the rationality postulates [10]?

There are several benefits from answering those questions. First, it would allow to *import some results* from argumentation theory to ontological query answering and vice versa, and more generally open the way to the Argumentation Web [19]. Second, it might be possible to use these results in order to *explain* to users how repairs are constructed and why a particular conclusion holds in a given semantics by constructing and evaluating arguments in favour of different conclusions [14]. Also, on a more theoretical side, proving a link between argumentation theory and the results in the knowledge representation community would be a step forward in understanding the *expressibility* of Dung's abstract theory for logic based argumentation [21].

The paper is organised as follows. In Section 2 the ontological query answering problem is explained and the logical language used throughout the paper is introduced. The end of this section introduces the existing semantics proposed in the literature to deal with inconsistent knowledge bases. Then, in Section 3, we define the basics of argumentation theory. Section 4 proves the links between the extensions obtained under different argumentation semantics in this instantiated logical argumentation setting and the repairs of the ontological knowledge base. We show the equivalence between the semantics from inconsistent ontological KB query answering area and those defined in argumentation theory in Section 5. Furthermore, the argumentation framework thus defined respects the rationality postulates (Section 6). The paper concludes with Section 7.

2 Ontological Conjunctive Query Answering

The main goal of section is to introduce the syntax and semantics of the $\mathcal{SRC}$ language [3, 4], which is used in this paper due to its relevance in the context of the ontological KB query answering.

Note that the goal of the present paper is not to change or criticise the definitions from this area; we simply present the existing work. Our goal is to study the link between the existing work in this area and the existing work in argumentation theory. In the following, we give a general setting knowledge representation language which can

then be instantiated according to properties on rules or constraints and yield equivalent languages to those used by [16] and [8].

A knowledge base is a 3-tuple $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ composed of three finite sets of formulae: a set $\mathcal{F}$ of facts, a set $\mathcal{R}$ of rules and a set $\mathcal{N}$ of constraints. Let us formally define what we accept as $\mathcal{F}$, $\mathcal{R}$ and $\mathcal{N}$.

Facts Syntax. Let $\mathbf{C}$ be a set of constants and $\mathbf{P} = P_1 \cup P_2 \ldots \cup P_n$ a set of predicates of the corresponding arity $i = 1, \ldots, n$. Let $\mathbf{V}$ be a countably infinite set of *variables*. We define the set of *terms* by $\mathbf{T} = \mathbf{V} \cup \mathbf{C}$. As usual, given $i \in \{1 \ldots n\}$, $p \in P_i$ and $t_1, \ldots, t_i \in \mathbf{T}$ we call $p(t_1, \ldots, t_i)$ an *atom*. If γ is an atom or a conjunction of atoms, we denote by $var(\gamma)$ the set of variables in γ and by $term(\gamma)$ the set of terms in γ. A *fact* is the existential closure of an atom or an existential closure of a conjunction of atoms. (Note that there is no negation or disjunction in the facts.) As an example, consider $\mathbf{C} = \{Tom\}$, $\mathbf{P} = P_1 \cup P_2$, with $P_1 = \{cat, mouse\}$, $P_2 = \{eats\}$ and $\mathbf{V} = \{x_1, x_2, x_3, \ldots\}$. Then, $cat(Tom)$, $eats(Tom, x_1)$ are examples of atoms and $\gamma = cat(Tom) \wedge mouse(x_1) \wedge eats(Tom, x_1)$ is an example of a conjunction of atoms. It holds that $var(\gamma) = \{x_1\}$ and $term(\gamma) = \{Tom, x_1\}$. As an example of a fact, consider $\exists x_1 (cat(Tom) \wedge mouse(x_1) \wedge eats(Tom, x_1))$.

An *interpretation* is a pair $I = (\triangle, .^I)$ where $\triangle$ is the interpretation domain (possibly infinite) and $.^I$, the interpretation function, satisfies:

1. For all $c \in \mathbf{C}$, we have $c^I \in \triangle$,
2. For all i and for all $p \in P_i$, we have $p^I \subseteq \triangle^i$,
3. If $c, c' \in \mathbf{C}$ and $c \neq c'$ then $c^I \neq c'^I$.

Note that the third constraint specifies that constants with different names map to different elements of $\triangle$.

Let γ be an atom or a conjunction of atoms or a fact. We say that γ is *true* under interpretation I iff there is a function ι which maps the terms (variables and constants) of γ into $\triangle$ such that for all constants c, it holds that $\iota(c) = c^I$ and for all atoms $p(t_1, \ldots t_i)$ appearing in γ, it holds that $(\iota(t_1), \ldots, \iota(t_i)) \in p^I$. For a set F containing any combination of atoms, conjunctions of atoms and facts, we say that F is *true* under interpretation I iff there is a function ι which maps the terms (variables and constants) of all formulae in F into $\triangle$ such that for all constants c, it holds that $\iota(c) = c^I$ and for all atoms $p(t_1, \ldots t_i)$ appearing in formulae of F, it holds that $(\iota(t_1), \ldots, \iota(t_i)) \in p^I$. Note that this means that for example sets $F_1 = \{\exists x(cat(x) \wedge dog(x))\}$ and $F_2 = \{\exists x(cat(x)), \exists x(dog(x))\}$ are true under exactly the same set of interpretations. Namely, in both cases, variable x is mapped to an object of $\triangle$. On the other hand, there are some interpretations under which set $F_3 = \{\exists x_1(cat(x_1)), \exists x_2(dog(x_2))\}$ is true whereas F_1 and F_2 are not.

If γ is true in I we say that I is a model of γ. Let γ' be an atom, a conjunction of atoms or a fact. We say that γ is a logical consequence of γ' (γ' entails γ, denoted $\gamma' \models \gamma$) iff all models of γ are models of γ'. If a set F is true in I we say that I is a model of F. We say that a formula γ is a logical consequence of a set F (denoted $F \models \gamma$) iff all models of F are models of γ. We say that a set G is a logical consequence of set F (denoted $F \models G$) if and only if all models of F are models of G. We say that two sets F and G are logically equivalent (denoted $F \equiv G$) if and only if $F \models G$ and $G \models F$.

Given a set of variables $\mathbf{X}$ and a set of terms $\mathbf{T}$, a *substitution* σ of $\mathbf{X}$ by $\mathbf{T}$ is a mapping from $\mathbf{X}$ to $\mathbf{T}$ (denoted $\sigma : \mathbf{X} \to \mathbf{T}$). Given an atom or a conjunction of atoms γ, $\sigma(\gamma)$ denotes the expression obtained from γ by replacing each occurrence of $x \in \mathbf{X} \cap var(\gamma)$ by $\sigma(x)$. If a fact F is the existential closure of a conjunction γ then we define $\sigma(F)$ as the existential closure of $\sigma(\gamma)$. Finally, let us define *homomorphism*. Let F and F' be atoms, conjunctions of atoms or facts (it is not necessarily the case that F and F' are of the same type, e.g. F can be an atom and F' a conjunction of atoms). Let σ be a substitution such that $\sigma : var(F) \to term(F')$. We say that σ is a homomorphism from F to F' if and only if the set of atoms appearing in $\sigma(F)$ is a subset of the set of atoms appearing in $\sigma(F')$. For example, let $F = cat(x_1)$ and $F' = cat(Tom) \wedge mouse(Jerry)$. Let $\sigma : var(F) \to term(F')$ be a substitution such that $\sigma(x_1) = Tom$. Then, σ is a homomorphism from F to F' since the atoms in $\sigma(F)$ are $\{cat(Tom)\}$ and the atoms in $\sigma(F')$ are $\{cat(Tom), mouse(Jerry)\}$.

Note that it well is known that $F' \models F$ if and only if there is a homomorphism from F to F' [12].

Rules. A rule R is a formula $\forall x_1, \ldots, \forall x_n \, \forall y_1, \ldots, \forall y_m \, (H(x_1, \ldots, x_n, y_1, \ldots, y_m) \to \exists z_1, \ldots \exists z_k \, C(y_1, \ldots, y_m, z_1, \ldots z_k))$ where H, the hypothesis, and C, the conclusion, are atoms or conjunctions of atoms, $n, m, k \in \{0, 1, \ldots\}$, $x_1, \ldots, x_n$ are the variables appearing in H, $y_1, \ldots, y_m$ are the variables appearing in both H and C and $z_1, \ldots, z_k$ the new variables introduced in the conclusion. As two examples of rules, consider $\forall x_1(cat(x_1) \to miaw(x_1))$ or $\forall x_1((mouse(x_1) \to \exists z_1(cat(z_1) \wedge eats(z_1, x_1)))$.

Reasoning consists of applying rules on the set and thus inferring new knowledge. A rule $R = (H, C)$ is *applicable* to set $\mathcal{F}$ if and only if there exists $\mathcal{F}' \subseteq \mathcal{F}$ such that there is a homomorphism σ from the hypothesis of $\mathcal{R}$ to the conjunction of elements of $\mathcal{F}'$. For example, rule $\forall x_1(cat(x_1) \to miaw(x_1))$ is applicable to set $\{cat(Tom)\}$, since there is a homomorphism from $cat(x_1)$ to $cat(Tom)$. If rule R is applicable to set F, the application of R to F according to π produces a set $F \cup \{\pi(C)\}$. In our example, the produced set is $\{cat(Tom), miaw(Tom)\}$. We then say that the new set (which includes the old one and adds the new information to it) is an *immediate derivation* of F by R. This new set is often denoted by $R(F)$. Thus, applying a rule on a set produces a new set.

Let F be a subset of $\mathcal{F}$ and let $\mathcal{R}$ be a set of rules. A set F_n is called an $\mathcal{R}$-*derivation* of F if there is a sequence of sets (called a *derivation sequence*) $(F_0, F_1, \ldots, F_n)$ such that:

- $F_0 \subseteq F$
- F_0 is $\mathcal{R}$-consistent
- for every $i \in \{1, \ldots, n-1\}$, it holds that F_i is an immediate derivation of F_{i-1}
- (no formula in $\mathcal{F}_n$ contains a conjunction and $\mathcal{F}_n$ is an immediate derivation of $\mathcal{F}_{n-1}$) or F_n is obtained from F_{n-1} by conjunction elimination.

Conjunction elimination is the following procedure: while there exists at least one conjunction in at least one formula, take an arbitrary formula φ containing a conjunction. If φ is of the form $\varphi = \psi \wedge \psi'$ then exchange it with two formulae ψ and ψ'. If φ is of the form $\exists x(\psi \wedge \psi')$ then exchange it with two formulae $\exists x(\psi)$ and $\exists x(\psi')$. The idea is just to start with an $\mathcal{R}$-consistent set and apply (some of the) rules. The only

technical detail is that the conjunctions are eliminated from the final result. So if the last set in a sequence does not contain conjunctions, nothing is done. Else, we eliminate those conjunctions. This technicality is needed in order to stay as close as possible to the procedures used in the literature in the case when the knowledge base is consistent.

Given a set $\{F_0, \ldots, F_k\} \subseteq \mathcal{F}$ and a set of rules $\mathcal{R}$, the closure of $\{F_0, \ldots, F_k\}$ with respect to $\mathcal{R}$, denoted $\text{Cl}_\mathcal{R}(\{F_0, \ldots, F_k\})$, is defined as the smallest set (with respect to $\subseteq$) which contains $\{F_0, \ldots, F_k\}$, and is closed for $\mathcal{R}$-derivation (that is, for every $\mathcal{R}$-derivation F_n of $\{F_0, \ldots, F_k\}$, we have $F_n \subseteq \text{Cl}_\mathcal{R}(\{F_0, \ldots, F_k\})$). Finally, we say that a set $\mathcal{F}$ and a set of rules $\mathcal{R}$ *entail* a fact G (and we write $\mathcal{F}, \mathcal{R} \models G$) iff the closure of the facts by all the rules entails F (i.e. if $\text{Cl}_\mathcal{R}(\mathcal{F}) \models G$).

As an example, consider a set of facts $\mathcal{F} = \{cat(Tom), small(Tom)\}$ and the rule set $\mathcal{R} = \{R_1 = \forall x_1(cat(x_1) \rightarrow miaw(x_1) \wedge animal(x_1)), R_2 = \forall x_1(miaw(x_1) \wedge small(x_1) \rightarrow cute(x_1))\}$. Then, F_0, F_1, F_2 is a derivation sequence, where $F_0 = \{cat(Tom), small(Tom)\}$, $F_1 = R_1(F_0) = \{cat(Tom), small(Tom), miaw(Tom) \wedge animal(Tom)\}$, $F_2 = \{cat(Tom), small(Tom), miaw(Tom) \wedge animal(Tom), cute(Tom)\}$ and $F_3 = \{cat(Tom), small(Tom), miaw(Tom), animal(Tom), cute(Tom)\}$.

We conclude the presentation on rules in $\mathcal{SRC}$ by a remark on performing union on facts when they are viewed as sets of atoms. In order to preserve semantics the union is done by renaming variables. For example, let us consider a fact $F_1 = \{\exists x cat(x)\}$ and a fact $F_2 = \{\exists x animal(x)\}$. Then the fact $F = F_1 \cup F_2$ is the union of the two fact after variable naming has been performed: $F = \{\exists x_1 cat(x_1), \exists x_2 animal(x_2)\}$.

Constraints. A constraint is a formula $\forall x_1 \ldots \forall x_n \ (H(x_1, \ldots, x_n) \rightarrow \bot)$, where H is an atom or a conjunction of atoms and $n \in \{0, 1, 2, \ldots\}$. Equivalently, a constraint can be written as $\neg(\exists x_1, \ldots, \exists x_n H(x_1, \ldots x_n))$. As an example of a constraint, consider $\forall x_1(cat(x_1) \wedge dog(x_1) \rightarrow \bot)$. $H(x_1, \ldots, x_n)$ is called the hypothesis of the constraint.

Given a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, a set $\{F_1, \ldots, F_k\} \subseteq \mathcal{F}$ is said to be *inconsistent* if and only if there exists a constraint $N \in \mathcal{N}$ such that $\{F_1, \ldots, F_k\} \models H_N$, where H_N denotes the existential closure of the hypothesis of N. A set is consistent if and only if it is not inconsistent. A set $\{F_1, \ldots, F_k\} \subseteq \mathcal{F}$ is $\mathcal{R}$-*inconsistent* if and only if there exists a constraint $N \in \mathcal{N}$ such that $\text{Cl}_\mathcal{R}(\{F_1, \ldots, F_k\}) \models H_N$, where H_N denotes the existential closure of the hypothesis of N.

A set of facts is said to be $\mathcal{R}$-*consistent* if and only if it is not $\mathcal{R}$-inconsistent. A knowledge base $(\mathcal{F}, \mathcal{R}, \mathcal{N})$ is said to be *consistent* if and only if $\mathcal{F}$ is $\mathcal{R}$-consistent. A knowledge base is *inconsistent* if and only if it is not consistent.

Example 1. Let us consider the following knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, with: $\mathcal{F} = \{cat(Tom), bark(Tom)\}$, $\mathcal{R} = \{\forall x_1(cat(x_1) \rightarrow miaw(x_1))\}$, $\mathcal{N} = \{\forall x_1(bark(x_1) \wedge miaw(x_1) \rightarrow \bot)\}$. The only rule in the knowledge base is applicable to the set $\{cat(Tom), bark(Tom)\}$ and its immediate derivation produces the set $\{cat(Tom), bark(Tom), miaw(Tom)\}$. We see that $\text{Cl}_\mathcal{R}(\mathcal{F}) \models \exists x_1(bark(x_1) \wedge miaw(x_1))$, thus the KB is inconsistent.

Given a knowledge base, one can ask a conjunctive query in order to know whether something holds or not. Without loss of generality we consider in this paper boolean conjunctive queries (which are facts). As an example of a query, take $\exists x_1 cat(x_1)$. The answer to query α is positive if and only if $\mathcal{F}, \mathcal{R} \models \alpha$.

2.1 Query Answering over Inconsistent Ontological Knowledge Bases

Notice that (like in classical logic), if a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is inconsistent, then everything is entailed from it. In other words, every query is true. Thus, the approach we described until now is not robust enough to deal with inconsistent information. However, there are cases when the knowledge base is inconsistent; this phenomenon has attracted particular attention during the recent years [8, 16]. For example, the set $\mathcal{F}$ may be obtained by combining several sets of facts, coming from different agents. In this paper, we study a general case when $\mathcal{K}$ is inconsistent without making any hypotheses about the origin of this inconsistency. Thus, our results can be applied to an inconsistent base independently of how it is obtained.

A common solution [8, 16] is to construct maximal (with respect to set inclusion) consistent subsets of $\mathcal{K}$. Such subsets are called *repairs*. Formally, given a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, define:

$$\mathcal{R}epair(\mathcal{K}) = \{\mathcal{F}' \subseteq \mathcal{F} \mid \mathcal{F}' \text{ is maximal for } \subseteq \mathcal{R}\text{-consistent set}\}$$

We now mention a very important technical detail. In some papers, a *set* of formulae is identified with the *conjunction* of those formulae. This is not of particular significance when the knowledge base is *consistent*. However, in case of an *inconsistent* knowledge base, this makes a big difference. Consider for example $\mathcal{K}_1 = (\mathcal{F}_1, \mathcal{R}_1, \mathcal{N}_1)$ with $\mathcal{F}_1 = \{dog(Tom), cat(Tom)\}$, $\mathcal{R}_1 = \emptyset$ and $\mathcal{N}_1 = \{\forall x_1(dog(x_1) \wedge cat(x_1) \rightarrow \bot)\}$, compared with $\mathcal{K}_2 = (\mathcal{F}_2, \mathcal{R}_2, \mathcal{N}_2)$ with $\mathcal{F}_2 = \{dog(Tom) \wedge cat(Tom)\}$, $\mathcal{R}_2 = \emptyset$ and $\mathcal{N}_2 = \{\forall x_1(dog(x_1) \wedge cat(x_1) \rightarrow \bot)\}$. In this case, according to the definition of a repair, $\mathcal{K}_1$ would have two repairs and $\mathcal{K}_2$ would have no repairs at all. We could proceed like this, but we find it confusing given the existing literature in this area. This is why, in order to be completely precise, from now on we suppose that $\mathcal{F}$ does not contain conjunctions. Namely, $\mathcal{F}$ is supposed to be a set composed of atoms and of existential closures of atoms. One could believe that this reduces the expressibility of the language, consider for example $\mathcal{F}_1 = \{\exists x(dog(x)), \exists x(black(x))\}$ as opposed to $\mathcal{F}_2 = \{\exists x(dog(x) \wedge black(x))\}$. Namely, in classical first order logic, $\mathcal{F}_1$ and $\mathcal{F}_2$ do not have the same models. However, in $\mathcal{SRC}$, $\mathcal{F}_1$ and $\mathcal{F}_2$ have the same models (see the definition of an interpretation).

Once the repairs calculated, there are different ways to calculate the set of facts that follow from an inconsistent knowledge base. For example, we may want to accept a query if it is entailed in *all repairs* (AR semantics).

Definition 1. *Let* $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *be a knowledge base and let* α *be a query. Then* α *is **AR-entailed** from* $\mathcal{K}$, *written* $\mathcal{K} \models_{AR} \alpha$ *iff for every repair* $A' \in \mathcal{R}epair(\mathcal{K})$, *it holds that* $\mathtt{Cl}_{\mathcal{R}}(A') \models \alpha$.

Another possibility is to check whether the query is entailed from the *intersection of closed repairs* (ICR semantics).

Definition 2. *Let* $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *be a knowledge base and let* α *be a query. Then* α *is **ICR-entailed** from* $\mathcal{K}$, *written* $\mathcal{K} \models_{ICR} \alpha$ *iff* $\bigcap_{A' \in \mathcal{R}epair(\mathcal{K})} \mathtt{Cl}_{\mathcal{R}}(A') \models \alpha$.

125

Example 2 (Example 1 Cont.). $\mathcal{R}epair(\mathcal{K}) = \{R_1, R_2\}$ with $R_1 = \{cat(Tom)\}$ and $R_2 = \{bark(Tom)\}\}$. $\mathtt{Cl}_{\mathcal{R}}(R_1) = \{cat(Tom), miaw(Tom)\}$, $\mathtt{Cl}_{\mathcal{R}}(R_2) = \{bark(Tom)\}$. It is *not* the case that $\mathcal{K} \models_{ICR} cat(Tom)$.

Finally, another possibility is to consider the *intersection of all repairs* and then close this intersection under the rules (IAR semantics).

Definition 3. *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and let α be a query. Then α is **IAR-entailed** from $\mathcal{K}$, written $\mathcal{K} \models_{IAR} \alpha$ iff $\mathtt{Cl}_{\mathcal{R}}(\bigcap_{A' \in \mathcal{R}epair(\mathcal{K})}) \models \alpha$.*

The three semantics can yield different results [16, 8], as illustrated by the next two examples.

Example 3. (ICR and IAR different from AR) Consider $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, with: $\mathcal{F} = \{havecat(Tom), haveMouse(Jerry)\}$, intuitively, we have a cat (called Tom) and a mouse (called Jerry); $\mathcal{R} = \{\forall x_1 (haveCat(x_1) \rightarrow haveAnimal(x_1)),$ $\forall x_2 (haveMouse(x_2) \rightarrow haveAnimal(x_2))\}$; $\mathcal{N} = \{\forall x_1 \forall x_2 (haveCat(x_1)$ $\wedge haveMouse(x_2) \rightarrow \bot)\}$, meaning that we cannot have both a cat and a mouse (since the cat would eat the mouse). There are two repairs: $R_1 = \{haveCat(Tom)\}$ and $R_2 = \{haveMouse(Jerry)\}$. $\mathtt{Cl}_{\mathcal{R}}(R_1) = \{haveCat(Tom), haveAnimal(Tom)\}$ and $\mathtt{Cl}_{\mathcal{R}}(R_2) = \{haveMouse(Jerry), haveAnimal(Jerry)\}$. Consider a query $\alpha = \exists x_1 \, haveAnimal(x_1)$ asking whether we have an animal. It holds that $\mathcal{K} \models_{AR} \alpha$ since $\mathtt{Cl}_{\mathcal{R}}(R_1) \models \alpha$ and $\mathtt{Cl}_{\mathcal{R}} \models \alpha$, but neither $\mathcal{K} \models_{ICR} \alpha$ (since $\mathtt{Cl}_{\mathcal{R}}(R_1) \cap \mathtt{Cl}_{\mathcal{R}}(R_2) = \emptyset$) nor $\mathcal{K} \models_{IAR} \alpha$ (since $R_1 \cap R_2 = \emptyset$).

Example 4. (AR and ICR different from IAR) Consider $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, with: $\mathcal{F} = \{cat(Tom), dog(Tom)\}$, $\mathcal{R} = \{\forall x_1 (cat(x_1) \rightarrow animal(x_1)),$ $\forall x_2 (dog(x_2) \rightarrow animal(x_2))\}$, $\mathcal{N} = \{\forall x(cat(x) \wedge dog(x) \rightarrow \bot)\}$.
We have $\mathcal{R}epair(\mathcal{K}) = \{R_1, R_2\}$ with $R_1 = \{cat(Tom)\}$ and $R_2 = \{dog(Tom)\}$. $\mathtt{Cl}_{\mathcal{R}}(R_1) = \{cat(Tom), animal(Tom)\}$, $\mathtt{Cl}_{\mathcal{R}}(R_2) = \{dog(Tom), animal(Tom)\}$.
It is *not* the case that $\mathcal{K} \models_{IAR} \exists x(animal(x))$ (since $R_1 \bigcap R_2 = \emptyset$). However, $\mathcal{K} \models_{AR} \exists x(animal(x))$. This is due to the fact that $\mathtt{Cl}_{\mathcal{R}}(R_1) \models \exists x(animal(x))$ and $\mathtt{Cl}_{\mathcal{R}}(R_2) \models \exists x(animal(x))$. Also, we have $\mathcal{K} \models_{ICR} \exists x(animal(x))$ since $\mathtt{Cl}_{\mathcal{R}}(R_1) \cap \mathtt{Cl}_{\mathcal{R}}(R_2) = \{animal(Tom)\}$.

3 Argumentation over Inconsistent Ontological Knowledge Bases

This section shows that it is possible to define an instantiation of Dung's abstract argumentation theory [15] that can be used to reason with an inconsistent ontological KB.

We first define the notion of an argument. For a set of formulae $\mathcal{G} = \{G_1, \ldots, G_n\}$, notation $\bigwedge G$ is used as an abbreviation for $G_1 \wedge \ldots \wedge G_n$.

Definition 4. *Given a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, an argument a is a tuple $a = (F_0, F_1, \ldots, F_n)$ where:*

– $(F_0, \ldots, F_{n-1})$ is a derivation sequence with respect to $\mathcal{K}$

– F_n *is an atom, a conjunction of atoms, the existential closure of an atom or the existential closure of a conjunction of atoms such that $F_{n-1} \models F_n$.*

Example 5 (Example 2 Cont.). Consider $a = (\{cat(Tom)\}, \{cat(Tom), miaw(Tom)\}, miaw(Tom))$ and $b = (\{bark(Tom)\}, bark(Tom))$ as two examples of arguments.

This is a straightforward way to define an argument when dealing with $\mathcal{SRC}$language, since this way, an *argument* corresponds to a *derivation*.

To simplify the notation, from now on, we suppose that we are given a fixed knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ and do not explicitly mention $\mathcal{F}$, $\mathcal{R}$ nor $\mathcal{N}$ if not necessary. Let $a = (F_0, ..., F_n)$ be an argument. Then, we denote $\text{Supp}(a) = F_0$ and $\text{Conc}(a) = F_n$. Let $S \subseteq \mathcal{F}$ a set of facts, $Arg(S)$ is defined as the set of all arguments a such that $\text{Supp}(a) \subseteq S$. Note that the set $\text{Arg}(S)$ is also dependent on the set of rules and the set of constraints, but for simplicity reasons, we do not write $\text{Arg}(S, \mathcal{R}, \mathcal{N})$ when it is clear to which $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ we refer to. Finally, let $\mathcal{E}$ be a set of arguments. The base of $\mathcal{E}$ is defined as the union of the argument supports: $\text{Base}(\mathcal{E}) = \bigcup_{a \in \mathcal{E}} \text{Supp}(a)$.

Arguments may attack each other, which is captured by a binary attack relation $\text{Att} \subseteq \text{Arg}(\mathcal{F}) \times \text{Arg}(\mathcal{F})$. Recall that the repairs are the subsets of $\mathcal{F}$ while the set $\mathcal{R}$ is always taken as a whole. This means that the authors of the semantics used to deal with an inconsistent ontological KB envisage the set of facts as inconsistent and the set of rules as consistent. When it comes to the attack relation, this means that we only need the so called "assumption attack" since, roughly speaking, all the inconsistency "comes from the facts".

Definition 5. *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and let a and b be two arguments. The argument a attacks argument b, denoted $(a, b) \in Att$, if and only if there exists $\varphi \in \text{Supp}(b)$ such that the set $\{Conc(a), \varphi\}$ is $\mathcal{R}$-inconsistent.*

This attack relation is not symmetric. To see why, consider the following example. Let $\mathcal{F} = \{p(m), q(m), r(m)\}$, $\mathcal{R} = \emptyset$, $\mathcal{N} = \{\forall x_1(p(x_1) \wedge q(x_1) \wedge r(x_1) \to \bot)\}$. Let $a = (\{p(m), q(m)\}, p(m) \wedge q(m))$, $b = (\{r(m)\}, r(m))$. We have $(a, b) \in \text{Att}$ and $(b, a) \notin \text{Att}$. Note that using attack relations which are not symmetric is very common in argumentation literature. Moreover, symmetric attack relation have been criticised for violating some desirable properties [1].

Definition 6. *Given a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, the* corresponding argumentation framework $\mathcal{AF}_K$ *is a pair* $(\mathcal{A} = \text{Arg}(\mathcal{F}), \text{Att})$ *where $\mathcal{A}$ is the set of arguments that can be constructed from $\mathcal{F}$ and Att is the corresponding attack relation as specified in Definition 5.*

Let $\mathcal{E} \subseteq \mathcal{A}$ and $a \in \mathcal{A}$. We say that $\mathcal{E}$ is conflict free *iff there exists no arguments $a, b \in \mathcal{E}$ such that $(a, b) \in Att$. $\mathcal{E}$ defends a iff for every argument $b \in \mathcal{A}$, if we have $(b, a) \in Att$ then there exists $c \in \mathcal{E}$ such that $(c, b) \in Att$.*

$\mathcal{E}$ is admissible *iff it is conflict free and defends all its arguments. $\mathcal{E}$ is a* complete *extension iff $\mathcal{E}$ is an admissible set which contains all the arguments it defends. $\mathcal{E}$ is a* preferred *extension iff it is maximal (with respect to set inclusion) admissible set. $\mathcal{E}$ is a* stable *extension iff it is conflict-free and for all $a \in \mathcal{A} \setminus \mathcal{E}$, there exists an argument $b \in \mathcal{E}$ such that $(b, a) \in Att$.*

$\mathcal{E}$ is a grounded extension *iff $\mathcal{E}$ is a minimal (for set inclusion) complete extension.*

If a semantics returns exactly one extension for every argumentation framework, then it is called a single-extension semantics.

For an argumentation framework $AS = (\mathcal{A}, Att)$ we denote by $Ext_x(AS)$ (or by $Ext_x(\mathcal{A}, Att)$) the set of its extensions with respect to semantics x. We use the abbreviations c, p, s, and g for respectively complete, preferred, stable and grounded semantics.

An argument is sceptically accepted if it is in all extensions, credulously accepted if it is in at least one extension and rejected if it is not in any extension.

Finally, we introduce two definitions allowing us to reason over such an argumentation framework. The output of an argumentation framework is usually defined [10, Definition 12] as the set of conclusions that appear in all the extensions (under a given semantics).

Definition 7 (Output of an argumentation framework). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and $\mathcal{AF}_K$ the corresponding argumentation framework. The output of $\mathcal{AF}_K$ under semantics x is defined as:*

$$\texttt{Output}_x(\mathcal{AF}_K) = \bigcap_{\mathcal{E} \in \texttt{Ext}_x(\mathcal{AF}_K)} \texttt{Concs}(\mathcal{E}).$$

In the degenerate case when $\texttt{Ext}_x(\mathcal{AF}_K) = \emptyset$, we define $\texttt{Output}(\mathcal{AF}_K) = \emptyset$ by convention.

Note that the previous definition asks for existence of a conclusion in every extension. This kind of acceptance is usually referred to as *sceptical* acceptance. We say that a query α is sceptically accepted if it is a logical consequence of the output of $\mathcal{AF}_K$:

Definition 8 (Sceptical acceptance of a query). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and $\mathcal{AF}_K$ the corresponding argumentation framework. A query α is sceptically accepted under semantics x if and only if $\texttt{Output}_x(\mathcal{AF}_K) \models \alpha$.*

It is possible to make an alternative definition, which uses the notion of universal acceptance instead of sceptical one. According to universal criteria, a query α is accepted if it is a logical consequence of conclusions of every extension:

Definition 9 (Universal acceptance of a query). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and $\mathcal{AF}_K$ the corresponding argumentation framework. A query α is universally accepted under semantics x if and only if for every extension $\mathcal{E}_i \in \texttt{Ext}_x(\mathcal{AF}_K)$, it holds that $\texttt{Concs}(\mathcal{E}_i) \models \alpha$.*

In general, universal and sceptical acceptance of a query do not coincide. Take for instance the KB from Example 3, construct the corresponding argumentation framework, and compare the sets of universally and sceptically accepted queries under preferred semantics.

Note that for single-extension semantics (e.g. grounded), the notions of sceptical and universal acceptance coincide. So we simply use word "accepted" in this context.

Definition 10 (Acceptance of a query). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $\mathcal{AF}_K$ the corresponding argumentation framework, x a single-extension semantics and let $\mathcal{E}$ be the unique extension of $\mathcal{AF}_K$. A query α is accepted under semantics x if and only if $\texttt{Concs}(\mathcal{E}) \models \alpha$.*

4   Equivalence between Repairs and Extensions

In this section, we prove two links between the repairs of an ontological KB and the corresponding argumentation framework: Theorem 1 shows that the repairs of the KB correspond exactly to the stable (and preferred, since in this instantiation the stable and the preferred semantics coincide) extensions of the argumentation framework; Theorem 2 proves that the intersection of all the repairs of the KB corresponds to the grounded extension of the argumentation framework.

Theorem 1. *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $\mathcal{AF}_K$ the corresponding argumentation framework and $x \in \{s, p\}$[1]. Then:*

$$\mathtt{Ext}_x(\mathcal{AF}_K) = \{\mathtt{Arg}(A') \mid A' \in \mathcal{R}epair(\mathcal{K})\}$$

Proof. The plan of the proof is as follows:

1. We prove that $\{\mathtt{Arg}(A') \mid A' \in \mathcal{R}epair(\mathcal{K})\} \subseteq \mathtt{Ext}_s(\mathcal{AF}_K)$.
2. We prove that $\mathtt{Ext}_p(\mathcal{AF}_K) \subseteq \{\mathtt{Arg}(A') \mid A' \in \mathcal{R}epair(\mathcal{K})\}$.
3. Since every stable extension is a preferred one [15], we can proceed as follows. From the first item, we have that $\{\mathtt{Arg}(A') \mid A' \in \mathcal{R}epair(\mathcal{K})\} \subseteq \mathtt{Ext}_p(\mathcal{AF}_K)$, thus the theorem holds for preferred semantics. From the second item we have that $\mathtt{Ext}_s(\mathcal{AF}_K) \subseteq \{\mathtt{Arg}(A') \mid A' \in \mathcal{R}epair(\mathcal{K})\}$, thus the theorem holds for stable semantics.

1. We first show $\{\mathtt{Arg}(A') \mid A' \in \mathcal{R}epair(\mathcal{K})\} \subseteq \mathtt{Ext}_s(\mathcal{AF}_K)$. Let $A' \in \mathcal{R}epair(\mathcal{K})$ and let $\mathcal{E} = \mathtt{Arg}(A')$. Let us prove that $\mathcal{E}$ is a stable extension of $(\mathtt{Arg}(\mathcal{F}), \mathtt{Att})$.
   We first prove that $\mathcal{E}$ is conflict-free. By means of contradiction we suppose the contrary, i.e. let $a, b \in \mathcal{E}$ such that $(a, b) \in \mathtt{Att}$. From the definition of attack, there exists $\varphi \in \mathtt{Supp}(b)$ such that $\{\mathtt{Conc}(a), \varphi\}$ is $\mathcal{R}$-inconsistent. Thus $\mathtt{Supp}(a) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent; consequently A' is $\mathcal{R}$-inconsistent, contradiction. Therefore $\mathcal{E}$ is conflict-free.
   Let us now prove that $\mathcal{E}$ attacks all arguments outside the set. Let $b \in \mathtt{Arg}(\mathcal{F}) \setminus \mathtt{Arg}(A')$ and let $\varphi \in \mathtt{Supp}(b)$, such that $\varphi \notin A'$. Let A'_c be the set obtained from A' by conjunction elimination and let $a = (A', A'_c, \bigwedge A'_c)$. We have $\varphi \notin A'$, so, due to the set inclusion maximality for the repairs, $\{\bigwedge A'_c, \varphi\}$ is $\mathcal{R}$-inconsistent. Therefore, $(a, b) \in \mathtt{Att}$. Consequently, $\mathcal{E}$ is a stable extension.
2. We now need to prove that $\mathtt{Ext}_p(\mathcal{AF}_K) \subseteq \{\mathtt{Arg}(A') \mid A' \in \mathcal{R}epair(\mathcal{K})\}$. Let $\mathcal{E} \in \mathtt{Ext}_p(\mathcal{AF}_K)$ and let us prove that there exists a repair A' such that $\mathcal{E} = \mathtt{Arg}(A')$. Let $S = \mathtt{Base}(\mathcal{E})$. Let us prove that S is $\mathcal{R}$-consistent. Aiming to a contradiction, suppose that S is $\mathcal{R}$-inconsistent. Let $S' \subseteq S$ be such that (1) S' is $\mathcal{R}$-inconsistent and (2) every proper set of S' is $\mathcal{R}$-consistent. Let us denote $S' = \{\varphi_1, \varphi_2, ..., \varphi_n\}$. Let $a \in \mathcal{E}$ be an argument such that $\varphi_n \in \mathtt{Supp}(a)$. Let S'_c be the set obtained from $S' \setminus \{\varphi\}$ by conjunction elimination and let $a' = (S' \setminus \{\varphi_n\}, S'_c, \bigwedge S'_c)$. We have that $(a', a) \in \mathtt{Att}$. Since $\mathcal{E}$ is conflict free, then $a' \notin \mathcal{E}$. Since $\mathcal{E}$ is an admissible set, there exists $b \in \mathcal{E}$ such that $(b, a') \in \mathtt{Att}$. Since b attacks a' then there exists

---
[1] Recall that s stands for *stable* and p for *preferred* semantics.

129

$i \in \{1, 2, ..., n-1\}$ such that $\{\text{Conc}(b), \varphi_i\}$ is $\mathcal{R}$-inconsistent. Since $\varphi_i \in \text{Base}(\mathcal{E})$, then there exists $c \in \mathcal{E}$ such that $\varphi_i \in \text{Supp}(c)$. Thus $(b, c) \in \text{Att}$, contradiction. So it must be that S is $\mathcal{R}$-consistent.

Let us now prove that there exists no $S' \subseteq \mathcal{F}$ such that $S \subsetneq S'$ and S' is $\mathcal{R}$-consistent. We use the proof by contradiction. Thus, suppose that S is not a maximal $\mathcal{R}$-consistent subset of $\mathcal{F}$. Then, there exists $S' \in \mathcal{R}epair(\mathcal{K})$, such that $S \subsetneq S'$. We have that $\mathcal{E} \subseteq \text{Arg}(S)$, since $S = \text{Base}(\mathcal{E})$. Denote $\mathcal{E}' = \text{Arg}(S')$. Since $S \subsetneq S'$ then $\text{Arg}(S) \subsetneq \mathcal{E}'$. Thus, $\mathcal{E} \subsetneq \mathcal{E}'$. From the first part of the proof, $\mathcal{E}' \in \text{Ext}_s(\mathcal{AF}_K)$. Consequently, $\mathcal{E}' \in \text{Ext}_p(\mathcal{AF}_K)$. We also know that $\mathcal{E} \in \text{Ext}_p(\mathcal{AF}_K)$. Contradiction, since no preferred set can be a proper subset of another preferred set. Thus, we conclude that $\text{Base}(\mathcal{E}) \in \mathcal{R}epair(\mathcal{K})$.

Let us show that $\mathcal{E} = \text{Arg}(\text{Base}(\mathcal{E}))$. It must be that $\mathcal{E} \subseteq \text{Arg}(S)$. Also, we know (from the first part) that $\text{Arg}(S)$ is a stable and a preferred extension, thus the case $\mathcal{E} \subsetneq \text{Arg}(s)$ is not possible.

3. Now we know that $\{\text{Arg}(A') \mid A' \in \mathcal{R}epair(\mathcal{K})\} \subseteq \text{Ext}_s(\mathcal{AF}_K)$ and $\text{Ext}_p(\mathcal{AF}_K) \subseteq \{\text{Arg}(A') \mid A' \in \mathcal{R}epair(\mathcal{K})\}$. The theorem follows from those two facts, as explained at the beginning of the proof.

To prove Theorem 2, we first prove the following lemma which says that if there are no rejected arguments under preferred semantics, then the grounded extension is equal to the intersection of all preferred extensions. Note that this result holds for every argumentation framework (not only for the one studied in this paper, where arguments are constructed from an ontological knowledge base). Thus, we only suppose that we are given a set and a binary relation on it (called attack relation).

Lemma 1. *Let* $AS = (\mathcal{A}, Att)$ *be an argumentation framework and* GE *its grounded extension.*

$$\text{If } \mathcal{A} \subseteq \bigcup_{\mathcal{E}_i \in \text{Ext}_p(AS)} \mathcal{E}_i \quad \text{then} \quad \text{GE} = \bigcap_{\mathcal{E}_i \in \text{Ext}_p(AS)} \mathcal{E}_i.$$

Proof. Let $\text{Iope} = \cap_{\mathcal{E}_i \in \text{Ext}_p(AS)} \mathcal{E}_i$ denote the intersection of all preferred extensions. It is known [15] the $\text{GE} \subseteq \text{Iope}$. Let us prove that in the case when there are no rejected arguments, it also holds the $\text{Iope} \subseteq \text{GE}$. Let $a \in \text{Iope}$. Let us show that no argument b attacks a. This holds since every argument b is in at least one preferred extension, say $\mathcal{E}_i$, and a is also in $\mathcal{E}_i$ (since a is in all preferred extensions) thus b does not attack a since both a and b are in $\mathcal{E}_i$ and $\mathcal{E}_i$ is a conflict-free set (since it is a preferred extension). All this means that arguments in Iope are not attacked. Consequently, they must all belong to the grounded extension. In other words, $\text{Iope} \subseteq \text{GE}$.

We can now, using the previous result, prove the link between the intersection of repairs and the grounded extension.

Theorem 2. *Let* $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *be a knowledge base and* $\mathcal{AF}_K$ *the corresponding argumentation framework. Denote the grounded extension of* $\mathcal{AF}_K$ *by* GE. *Then:*

$$\text{GE} = \text{Arg}\Big(\bigcap_{A' \in \mathcal{R}epair(\mathcal{K})} A' \Big).$$

Proof. Denote the intersection of all repairs by $\mathrm{Ioar} = \bigcap_{A' \in \mathcal{R}epair(\mathcal{K})} A'$ and the intersection of all preferred extensions by $\mathrm{Iope} = \bigcap_{\mathcal{E}_i \in \mathrm{Ext}_p(\mathcal{AF}_K)} \mathcal{E}_i$. From Theorem 1, we know that $\mathrm{Ext}_x(\mathcal{AF}_K) = \{\mathrm{Arg}(A') \mid A' \in \mathcal{R}epair(\mathcal{K})\}$. Consequently,

$$\mathrm{Iope} = \bigcap_{A' \in \mathcal{R}epair(\mathcal{K})} \mathrm{Arg}(A') \qquad (1)$$

Since every argument has an $\mathcal{R}$-consistent support, then its support is in at least one repair. From Theorem 1, that argument is in at least one preferred extension, (i.e. it is not rejected). From Lemma 1,

$$\mathrm{Iope} = \mathrm{GE} \qquad (2)$$

From (1) and (2), we obtain that

$$\mathrm{GE} = \bigcap_{A' \in \mathcal{R}epair(\mathcal{K})} \mathrm{Arg}(A') \qquad (3)$$

Note that for every collection $S_1, \ldots, S_n$ of of sets of formulae, we have $\mathrm{Arg}(S_1) \cap \ldots \cap \mathrm{Arg}(S_n) = Arg(S_1 \cap \ldots \cap S_n)$. By applying this rule on the set of all repairs, we obtain:

$$\bigcap_{A' \in \mathcal{R}epair(\mathcal{K})} \mathrm{Arg}(A') = \mathrm{Arg}(\mathrm{Ioar}) \qquad (4)$$

From (3) and (4), we obtain $\mathrm{GE} = \mathrm{Arg}(\mathrm{Ioar})$ which ends the proof.

5   Semantics Equivalence

This section presents the main result of the paper. It proves the links between semantics from argumentation theory (stable, preferred, grounded) and semantics from inconsistent ontology KB query answering (ICR, AR, IAR). More precisely, we show that: (1) sceptical acceptance under stable and preferred semantics corresponds to ICR semantics; (2) universal acceptance under stable and preferred semantics corresponds to AR semantics; (3) acceptance under grounded semantics corresponds to IAR semantics. The proof of Theorem 3 is based on Theorem 1 and the proof of Theorem 4 is derived from Theorem 2.

Theorem 3. *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, let $\mathcal{AF}_K$ be the corresponding argumentation framework and let α be a query. Let $x \in \{s, p\}$ be stable or preferred semantics. Then:*

- *$\mathcal{K} \models_{ICR} \alpha$ iff α is sceptically accepted under semantics x.*
- *$\mathcal{K} \models_{AR} \alpha$ iff α is universally accepted under semantics x.*

Proof. Theorem 1 implies $Ext_x(Arg(\mathcal{F}), Att) = \{Arg(A') \mid A' \in \mathcal{R}epair(\mathcal{K})\}$. In fact, the restriction of function Arg on $\mathcal{R}epair(\mathcal{K})$ is a bijection between $\mathcal{R}epair(\mathcal{K})$ and $\mathrm{Ext}_x(\mathcal{AF}_K)$. Note also that for every query α, for every repair A', we have that $\mathrm{Cl}_{\mathcal{R}}(A') \models \alpha$ if and only if $\mathrm{Concs}(\mathrm{Arg}(A')) \models \alpha$. By using those two facts, the result of the theorem can be obtained as follows:

– For every query α, we have: $\mathcal{K} \models_{ICR} \alpha$ if and only if $\bigcap_{A' \in \mathcal{R}epair(\mathcal{K})} \mathtt{Cl}_{\mathcal{R}}(A') \models \alpha$ if and only if $\bigcap_{\mathcal{E}_i \in \mathtt{Ext}_x(\mathcal{AF}_K)} \mathtt{Concs}(\mathcal{E}_i) \models \alpha$ if and only if $\mathtt{Output}_x(\mathcal{AF}_K) \models \alpha$ if and only if α is sceptically accepted.

– For every query α, we have: $\mathcal{K} \models_{AR} \alpha$ if and only if for every $A' \in \mathcal{R}epair(\mathcal{K})$, $\mathtt{Cl}_{\mathcal{R}}(A') \models \alpha$ if and only if for every $\mathcal{E}_i \in \mathtt{Ext}_x(\mathcal{AF}_K)$, $\mathtt{Concs}(\mathcal{E}_i) \models \alpha$ if and only if α is universally accepted.

Theorem 4. *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, let $\mathcal{AF}_K$ be the corresponding argumentation framework and let α be a query. Then:*

$$\mathcal{K} \models_{IAR} \alpha \text{ iff } \alpha \text{ is accepted under grounded semantics.}$$

Proof. Let us denote the grounded extension of $\mathcal{AF}_K$ by $\mathtt{GE}$ and the intersection of all repairs by $\mathtt{Ioar} = \bigcap_{A' \in \mathcal{R}epair(\mathcal{K})} A'$. From Definition 10, we have:

$$\alpha \text{ is accepted under grounded semantics iff } \mathtt{Concs}(\mathtt{GE}) \models \alpha. \tag{5}$$

From Theorem 2, we have:
$$\mathtt{GE} = \mathtt{Arg}(\mathtt{Ioar}). \tag{6}$$

Note also that for every set of facts $\{F_1, \ldots, F_n\}$ and for every query α, we have that $\mathtt{Cl}_{\mathcal{R}}(\{F_1, \ldots, F_n\}) \models \alpha$ if and only if $\mathtt{Concs}(\mathtt{Arg}(\{F_1, \ldots, F_n\})) \models \alpha$. Thus,

$$\mathtt{Cl}_{\mathcal{R}}(\mathtt{Ioar}) \models \alpha \text{ if and only if } \mathtt{Concs}(\mathtt{Arg}(\mathtt{Ioar})) \models \alpha. \tag{7}$$

From (6) and (7) we have that:

$$\mathtt{Cl}_{\mathcal{R}}(\mathtt{Ioar}) \models \alpha \text{ if and only if } \mathtt{Concs}(\mathtt{GE}) \models \alpha. \tag{8}$$

From Definition 3, one obtains:

$$\mathtt{Cl}_{\mathcal{R}}(\mathtt{Ioar}) \models \alpha \text{ if and only if } \mathcal{K} \models_{IAR} \alpha. \tag{9}$$

The theorem now follows from (5), (8) and (9).

6 Postulates

In this section, we prove that the framework we propose in this paper satisfies the rationality postulates for instantiated argumentation frameworks [10]. We first prove the indirect consistency postulate.

Proposition 1 (Indirect consistency). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $\mathcal{AF}_K$ the corresponding argumentation framework and $x \in \{s, p, g\}$. Then:*

– *for every $\mathcal{E}_i \in \mathtt{Ext}_x(\mathcal{AF}_K)$, $\mathtt{Cl}_{\mathcal{R}}(\mathtt{Concs}(\mathcal{E}_i))$ is a consistent set*
– *$\mathtt{Cl}_{\mathcal{R}}(\mathtt{Output}_x(\mathcal{AF}_K))$ is a consistent set.*

Proof.

132

– Let $\mathcal{E}_i$ be a stable or a preferred extension of $\mathcal{AF}_K$. From Theorem 1, there exists a repair $A' \in \mathcal{R}epair(\mathcal{K})$ such that $\mathcal{E}_i = \text{Arg}(A')$. Note that $\text{Concs}(\mathcal{E}_i) = \text{Cl}_\mathcal{R}(A') \cup \{\alpha \mid \text{Cl}_\mathcal{R}(A) \models \alpha\}$ (this follows directly from Definition 4). Consequently, the set of $\mathcal{R}$-derivations of $\text{Concs}(\mathcal{E}_i)$ and the set of $\mathcal{R}$-derivations of $\text{Cl}_\mathcal{R}(A')$ coincide. Formally, $\text{Cl}_\mathcal{R}(\text{Cl}_\mathcal{R}(A')) = \text{Cl}_\mathcal{R}(\text{Concs}(\mathcal{E}_i))$. Since $\text{Cl}_\mathcal{R}$ is idempotent, this means that $\text{Cl}_\mathcal{R}(A') = \text{Cl}_\mathcal{R}(\text{Concs}(\mathcal{E}_i))$. Since $\text{Cl}_\mathcal{R}(A')$ is consistent, then $\text{Cl}_\mathcal{R}(\text{Concs}(\mathcal{E}_i))$ is consistent.

Let us now consider the case of grounded semantics. Denote GE the grounded extension of $\mathcal{AF}_K$. We have just seen that for every $\mathcal{E}_i \in \text{Ext}_p(\mathcal{AF}_K)$, it holds that $\text{Cl}_\mathcal{R}(\text{Concs}(\mathcal{E}_i))$ is a consistent set. Since the grounded extension is a subset of the intersection of all the preferred extensions [15], and since there is at least one preferred extension, say $\mathcal{E}_1$, then GE $\subseteq \mathcal{E}_1$. Since $\text{Cl}_\mathcal{R}(\text{Concs}(\mathcal{E}_i))$ is consistent then $\text{Cl}_\mathcal{R}(\text{Concs}(\text{GE}))$ is also consistent.

– Consider the case of stable or preferred semantics. Let us prove $\text{Cl}_\mathcal{R}(\text{Output}_x(\mathcal{AF}_K))$ is a consistent set. Recall that $\text{Output}_x(\mathcal{AF}_K) = \bigcap_{\mathcal{E}_i \in \text{Ext}_x(\mathcal{AF}_K)} \text{Concs}(\mathcal{E}_i)$. Since every knowledge base has at least one repair then, according to Theorem 1, there is at least one stable or preferred extension $\mathcal{E}_i$. From Definition 7, we have that $\text{Output}_x(\mathcal{AF}_K) \subseteq \text{Concs}(\mathcal{E}_i)$. $\text{Concs}(\mathcal{E}_i)$ is $\mathcal{R}$-consistent thus $\text{Output}_x(\mathcal{AF}_K)$ is $\mathcal{R}$-consistent. In other words, $\text{Cl}_\mathcal{R}(\text{Output}_x(\mathcal{AF}_K))$ is consistent.

Note that in the case of grounded semantics the second part of the proposition follows directly from the first one, since $\text{Cl}_\mathcal{R}(\text{Output}_g(\mathcal{AF}_K)) = \text{Cl}_\mathcal{R}(\text{Concs}(\text{GE}))$.

Since our instantiation satisfies indirect consistency then it also satisfies direct consistency. This comes from $\mathcal{R}$-consistency definition; namely, if a set is $\mathcal{R}$-consistent, then it is necessarily consistent. Thus, we obtain the following corollary.

Corollary 1 (Direct consistency). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $\mathcal{AF}_K$ the corresponding argumentation framework and $x \in \{s, p, g\}$. Then:*

– *for every $\mathcal{E}_i \in \text{Ext}_x(\mathcal{AF}_K)$, $\text{Concs}(\mathcal{E}_i)$ is a consistent set*
– *$\text{Output}_x(\mathcal{AF}_K)$ is a consistent set.*

We now also prove that the present argumentation formalism also satisfies the closure postulate.

Proposition 2 (Closure). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $\mathcal{AF}_K$ the corresponding argumentation framework and $x \in \{s, p, g\}$. Then:*

– *for every $\mathcal{E}_i \in \text{Ext}_x(\mathcal{AF}_K)$, $\text{Concs}(\mathcal{E}_i) = \text{Cl}_\mathcal{R}(\text{Concs}(\mathcal{E}_i))$.*
– *$\text{Output}_x(\mathcal{AF}_K) = \text{Cl}_\mathcal{R}(\text{Output}_x(\mathcal{AF}_K))$.*

Proof.
– From the definition of $\text{Cl}_\mathcal{R}$, we see that $\text{Concs}(\mathcal{E}_i) \subseteq \text{Cl}_\mathcal{R}(\text{Concs}(\mathcal{E}_i))$. Let us prove that $\text{Cl}_\mathcal{R}(\text{Concs}(\mathcal{E}_i)) \subseteq \text{Concs}(\mathcal{E}_i)$. Suppose that $\alpha \in \text{Cl}_\mathcal{R}(\text{Concs}(\mathcal{E}_i))$. This means that there exists $\alpha_1, \ldots, \alpha_k \in \text{Concs}(\mathcal{E}_i)$ and that there exists a derivation sequence $F_0, \ldots, F_n$ such that $F_0 = \{\alpha_1, \ldots, \alpha_k\}$ and $\alpha \in F_n$. Note that from Proposition 1, we know that $\{\alpha_1, \ldots, \alpha_k\}$ is $\mathcal{R}$-consistent. Since $\alpha_1, \ldots, \alpha_k \in \text{Concs}(\mathcal{E}_i)$ then there exist $a_1, \ldots, a_k \in \mathcal{E}_i$ such that $\text{Conc}(a_1) = \alpha_1, \ldots, \text{Conc}(a_k) = \alpha_k$. Thus,

there exists an argument a such that $\mathtt{Supp}(a) = \mathtt{Supp}(a_1) \cup \ldots \cup \mathtt{Supp}(a_k)$ and $\mathtt{Conc}(a) = \alpha$. Since $\mathcal{E}_i$ is a preferred, a stable or the grounded extension, Theorems 1 and 2 imply that there exists a set of formulae S such that $\mathcal{E}_i = \mathtt{Arg}(S)$. Consequently, $\mathcal{E}_i = \mathtt{Arg}(\mathtt{Base}(\mathcal{E}_i))$. From this observation and since $\mathtt{Supp}(a) \subseteq \mathtt{Base}(\mathcal{E}_i)$, we conclude that $a \in \mathcal{E}_i$. Thus, $\alpha \in \mathtt{Concs}(\mathcal{E}_i)$, which ends the proof.

– In the case of grounded semantics, the result holds directly from the first part of the proposition. The reminder of the proof considers stable or preferred semantics. From the definition of $\mathtt{Cl}_{\mathcal{R}}$, $\mathtt{Output}_x(\mathcal{AF}_K) \subseteq \mathtt{Cl}_{\mathcal{R}}(\mathtt{Output}_x(\mathcal{AF}_K))$. So we only need to prove that $\mathtt{Cl}_{\mathcal{R}}(\mathtt{Output}_x(\mathcal{AF}_K)) \subseteq \mathtt{Output}_x(\mathcal{AF}_K)$.

Let $\alpha \in \mathtt{Cl}_{\mathcal{R}}(\mathtt{Output}_x(\mathcal{AF}_K))$. Then there exist $\alpha_1, \ldots, \alpha_k \in \mathtt{Output}_x(\mathcal{AF}_K)$ such that there is a derivation sequence $F_0, \ldots, F_n$ such that $F_0 = \{\alpha_1, \ldots, \alpha_k\}$ and $\alpha \in F_n$. Since $\alpha_1, \ldots, \alpha_k \in \mathtt{Output}_x(\mathcal{AF}_K)$ then for every $\mathcal{E}_i \in \mathtt{Ext}_x(\mathcal{AF}_K)$, we have $\alpha_1, \ldots, \alpha_k \in \mathcal{E}_i$. Therefore for every $\mathcal{E}_i \in \mathtt{Ext}_x(\mathcal{AF}_K)$, $\alpha \in \mathtt{Cl}_{\mathcal{R}}(\mathtt{Concs}(\mathcal{E}_i))$. From the first part of the proof, $\mathtt{Cl}_{\mathcal{R}}(\mathtt{Concs}(\mathcal{E}_i)) = \mathtt{Concs}(\mathcal{E}_i)$. Thus, for every $\mathcal{E}_i \in \mathtt{Ext}_x(\mathcal{AF}_K)$, $\alpha \in \mathtt{Concs}(\mathcal{E}_i)$. This means that $\alpha \in \mathtt{Output}_x(\mathcal{AF}_K)$.

7 Summary and Conclusion

This paper investigates the links between the semantics used in argumentation theory and those from the inconsistent ontological KB query answering.

Contribution of the paper. First, we show that it is possible to instantiate Dung's abstract argumentation theory in a way to deal with inconsistency in an ontological KB. Second, we formally prove the links between the semantics from ontological KB query answering and those from argumentation theory: ICR semantics corresponds to sceptical acceptance under stable or preferred argumentation semantics, AR semantics corresponds to universal acceptance under stable / preferred argumentation semantics and IAR semantics corresponds to acceptance under grounded argumentation semantics. Third, we show that the instantiation we define satisfies the rationality postulates. The fourth contribution of the paper is to make a bridge between the argumentation community and the knowledge representation community in this context, allowing for future exchanges.

Applications of our work. The first possible application of our work is to import some results about semantics and acceptance from argumentation to ontological KB query answering and vice versa. Second, arguments can be used for explanatory purposes. In other words, we can use arguments and counter arguments to graphically represent and explain why different points of view are conflicting or not and why certain argument is (not) in all extensions. However, we suppose that the user understands the notion of logical consequence under first order logic when it comes to *consistent* data. For example, we suppose that the user is able to understand that if $cat(Tom) \wedge miaw(Tom)$ is present in the set, then queries $cat(Tom)$ and $\exists x cat(x)$ are both true. To sum up, we suppose that the other methods are used to explain reasoning under *consistent* knowledge and we use argumentation to explain reasoning under *inconsistent* knowledge.

Related work. Note that this is the first work studying the link between semantics used in argumentation (stable, preferred, grounded) and semantics used in inconsistent ontological knowledge base query answering (AR, IAR, ICR). There is not much related work. However, we review some papers that study similar issues.

134

For instance, the link between maximal consistent subsets of a knowledge base and stable extensions of the corresponding argumentation system was shown by Cayrol [11]. That was the first work showing this type of connection between argument-based and non argument-based reasoning. This result was generalised [20] by studying the whole class of argumentation systems corresponding to maximal consistent subsets of the propositional knowledge base. The link between the ASPIC system [18] and the Argument Interchange Format (AIF) ontology [13] has recently been studied [7]. Another related paper comprises constructing an argumentation framework with ontological knowledge allowing two agents to discuss the answer to queries concerning their knowledge (even if it is inconsistent) without one agent having to copy all of their ontology to the other [9]. While those papers are in the area of our paper, none of them is related to the study of the links between different semantics for inconsistent ontological KB query answering and different argumentation semantics.s

Future work. We plan to answer different questions, like: Can other semantics from argumentation theory yield different results? Are those results useful for inconsistent ontological KB query answering? What happens in the case when preferences are present? What is the link between having preferences on databases and having preferences on arguments? More generally speaking, we want to examine how the knowledge representation community could benefit from other results from argumentation theory and whether the argumentation community could use some open problems in the knowledge representation as inspiration for future work.

References

1. L. Amgoud and P. Besnard. Bridging the gap between abstract argumentation systems and logic. In *International Conference on Scalable Uncertainty Management (SUM'09)*, pages 12–27, 2009.

2. L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning*, 29 (2):125–169, 2002.

3. J.-F. Baget and M.-L. Mugnier. The Complexity of Rules and Constraints. *JAIR*, 16:425–465, 2002.

4. J.-F. Baget, M.-L. Mugnier, S. Rudolph, and M. Thomazo. Walking the complexity lines for generalized guarded existential rules. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence, (IJCAI'11)*, pages 712–717, 2011.

5. S. Benferhat, D. Dubois, and H. Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proceedings of the 9th Conference on Uncertainty in Artificial intelligence (UAI'93)*, pages 411–419, 1993.

6. P. Besnard and A. Hunter. *Elements of Argumentation*. MIT Press, 2008.

7. F. J. Bex, S. J. Modgil, H. Prakken, and C. Reed. On logical specifications of the argument interchange format. *Journal of Logic and Computation*, page In Press, 2013.

8. M. Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proc of AAAI*, 2012.

9. E. Black, A. Hunter, and J. Z. Pan. An argument-based approach to using multiple ontologies. In *Proceedings of the 3rd International Conference on Scalable Uncertainty Management (SUM'09)*, pages 68–79. Springer-Verlag, 2009.

10. M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence Journal*, 171 (5-6):286–310, 2007.

11. C. Cayrol. On the relation between argumentation and non-monotonic coherence-based entailment. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1443–1448, 1995.

12. M. Chein and M.-L. Mugnier. *Graph-based Knowledge Representation and Reasoning— Computational Foundations of Conceptual Graphs*. Advanced Information and Knowledge Processing. Springer, 2009.

13. C. Chesnevar, J. McGinnis, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, and S. Willmott. Towards an argument interchange format. *Knowledge Engineering Review*, 21(4):293–316, 2006.

14. J. Dix, S. Parsons, H. Prakken, and G. R. Simari. Research challenges for argumentation. *Computer Science - R&D*, 23(1):27–34, 2009.

15. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence Journal*, 77:321–357, 1995.

16. D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. F. Savo. Inconsistency-tolerant semantics for description logics. In *Proc. of RR*, pages 103–117, 2010.

17. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS 2002*, 2002.

18. S. J. Modgil and H. Prakken. A general account of argumentation with preferences. *Artificial Intelligence Journal*, page In Press, 2013.

19. I. Rahwan, F. Zablith, and C. Reed. Laying the foundations for a world wide argument web. *Artificial Intelligence*, 171(10-15):897–921, 2007.

20. S. Vesic. Maxi-consistent operators in argumentation. In *20th European Conference on Artificial Intelligence (ECAI'12)*, pages 810–815, 2012.

21. S. Vesic and L. van der Torre. Beyond maxi-consistent argumentation operators. In *13th European Conference on Logics in Artificial Intelligence (JELIA'12)*, pages 424–436, 2012.

On Conceptual Graphs and Explanation of Query Answering Under Inconsistency

Abdallah Arioua, Nouredine Tamani, Madalina Croitoru

University of Montpellier II, France
{arioua,tamani,croitoru}@lirmm.fr

Abstract. Conceptual Graphs are a powerful visual knowledge representation language. In this paper we are interested in the use of Conceptual Graphs in the setting of Ontology Based Data Access, and, more specifically, in reasoning in the presence of inconsistency. We present different explanation heuristics of query answering under inconsistency and show how they can be implemented under the Conceptual Graphs editor COGUI.

1   Introduction

We place ourselves in a Rule-based Data Access (RBDA) setting that investigates how to query multiple data sources defined over the same ontology represented using a rule based language. The RBDA is a specific case of the Ontology Based Data Access (OBDA) setting. In RBDA we assume that the ontology is encoded using rules. The growing number of distinct data sources defined under the same ontology makes OBDA an important and timely problem to address. The input to the problem is a set of facts, an ontology and a conjunctive query. We aim to find if there is an answer to the query in the facts (eventually enriched by the ontology).

More precisely, the RBDA problem stated in reference to the classical *forward chaining* scheme is the following: "Can we find an answer to the query Q in a database F' that is built from F by adding atoms that can be logically deduced from F and the rule based ontology $\mathcal{R}$?"

In certain cases, the integration of factual information from various data sources may lead to inconsistency. A solution is then to construct maximal (with respect to set inclusion) consistent subsets of $\mathcal{F}$ called *repairs* [6, 20]. Once the repairs are computed, there are different ways to combine them in order to obtain an answer for the query.

In this paper we address the RBDA problem from a Conceptual Graphs perspective. Conceptual Graphs are powerful visual formalism for representing a subset of First Order Logic covered by the RBDA setting.

We make explicit these links and focus on the case where we want to perform query answering in presence of inconsistency. We present query answering explanation strategies inspired from the link between the ODBA inconsistent-tolerant semantics and argumentation acceptance semantics[11]. Our work is inspired by the argumentation explanation power [14, 25, 28].

2 Related work

There are two major approaches in order to represent an ontology for the OBDA problem and namely Description Logics (such as $\mathcal{EL}$([2]) and DL-Lite [9] families) and rule based languages (such as the Datalog$^+$ [8] language, a generalization of Datalog that allows for existentially quantified variables in the head of the rules). When using rules for representing the ontology we would denote the OBDA problem under the name of RBDA. Despite Datalog$^+$ undecidability when answering conjunctive queries, there exist decidable fragments of Datalog$^+$ which are studied in the literature [5]. These fragments generalize the above mentioned Description Logics families.

Here we follow the second method: representing the ontology via rules. We give a general rule based setting knowledge representation language equivalent to the Datalog$^+$ language and show how this language is equivalent to Conceptual Graphs with rules and negative constraints.

Within this language we are mainly interested in studying the question of "why an *inconsistent* KB entails a certain query α under an inconsistency-tolerant semantics". Indeed, many works focused on the following questions: "Why a concept C is subsumed (non-subsumed) by D" or "Why the KB is unsatisfiable and incoherent"? The need for explanation-aware methods stems from the desire to seek a comprehensive means that facilitates maintenance and repairing of inconsistent knowledge bases as well as understanding the underlying mechanism for reasoning services. In the field of databases there has been work on explaining answer and non-answer returned by database systems [1, 24, 23, 16, 15] using causality and responsibility or using a cooperative architecture to provide a cooperative answer for query failing.

In the area of DLs, the question was mainly about explaining either reasoning (subsumption and non-subsumption) or unsatisfiability and incoherence. In a seminal paper McGuinness et al. [22, 7] addressed the problem of explaining subsumption and non-subsumption in a coherent and satisfiable DL knowledge base using *formal proofs as explanation* based on a complete and sound deduction system for a fragment of Description Logics, while other proposals [27, 26, 3, 4] have used *Axiom pinpointing* and *Concept pinpointing* as explanation to highlight contradictions within an unsatisfiable and incoherent DL KB.

Another proposal [19, 18] is the so-called *justification-oriented proofs* in which the authors proposed a *proof-like* explanation without the need for deduction rules. The explanation then is presented as an acyclic proof graph that relates axioms and lemmas. Another work [12] in the same context proposes a resolution-based framework in which the explanation is constructed from a refutation graph.

3 Logical Language

We consider a (potentially inconsistent) knowledge base composed of the following:

– A set $\mathcal{F}$ of facts that correspond to existentially closed conjunctions of atoms. The atoms can contain n-ary predicates. The following facts are borrowed from [21]: $F_1 : directs(John, d_1)$, $F_2 : directs(Tom, d_1)$, $F_3 : directs(Tom, d_2)$, $F_4 : supervises(Tom, John)$, $F_5 : works\_in(John, d_1)$, $F_6 : works\_in(Tom, d_1)$.

- A set of negative constraints which represent the negation of a fact. Alternatively negative constraints can be seen as rules with the absurd conclusion. Negative constraints can also be n-ary. For example, $N_1 = \forall x, y, z \, (supervises(x, y) \wedge work\_in(x, z) \wedge directs(y, z)) \rightarrow \perp$ and
$N_2 = \forall x, y \, supervises(x, y) \wedge manager(y) \rightarrow \perp$ are negative constraints.
- An ontology composed of a set of rules that represent general implicit knowledge and that can introduce new variables in their head (conclusion). Please note that these variables, in turn, can trigger new rule application and cause the undecidability of the language in the general case. Different rule application strategies (chase), including the skolemized chase, are studied in the literature. For example
$R_1 = \forall x \forall d \, works\_in(x, d) \rightarrow emp(x)$
$R_2 = \forall x \forall d \, directs(x, d) \rightarrow emp(x)$
$R_3 = \forall x \forall d \, directs(x, d) \wedge works\_in(x, d) \rightarrow manager(x)$
$R_4 = \forall x \, emp(x) \rightarrow \exists y (\textit{office}(y) \wedge \textit{uses}(x, y))$

A rule is *applicable* to set of facts $\mathcal{F}$ if and only if the set entails the hypothesis of the rule. If rule R is applicable to the set F, the application of R on F produces a new set of facts obtained from the initial set with additional information from the rule conclusion. We then say that the new set is an *immediate derivation* of F by R denoted by $R(F)$. For example $R_1(F_5) = works\_in(John, d_1) \wedge emp(John)$.

Let F be a set of facts and let $\mathcal{R}$ be a set of rules. A set F_n is called an $\mathcal{R}$-*derivation* of F if there is a sequence of sets (*derivation sequence*) $(F_0, F_1, \ldots, F_n)$ such that: (i) $F_0 \subseteq F$, (ii) F_0 is $\mathcal{R}$-consistent, (iii) for every $i \in \{1, \ldots, n - 1\}$, it holds that F_i is an immediate derivation of F_{i-1}.

Given a set $\{F_0, \ldots, F_k\}$ and a set of rules $\mathcal{R}$, the closure of $\{F_0, \ldots, F_k\}$ with respect to $\mathcal{R}$, denoted $\mathtt{Cl}_{\mathcal{R}}(\{F_0, \ldots, F_k\})$, is defined as the smallest set (with respect to $\subseteq$) which contains $\{F_0, \ldots, F_k\}$, and is closed for $\mathcal{R}$-derivation (that is, for every $\mathcal{R}$-derivation F_n of $\{F_0, \ldots, F_k\}$, we have $F_n \subseteq \mathtt{Cl}_{\mathcal{R}}(\{F_0, \ldots, F_k\})$). Finally, we say that a set $\mathcal{F}$ and a set of rules $\mathcal{R}$ *entail* a fact G (and we write $\mathcal{F}, \mathcal{R} \models G$) iff the closure of the facts by all the rules entails F (i.e. if $\mathtt{Cl}_{\mathcal{R}}(\mathcal{F}) \models G$).

Given a set of facts $\{F_1, \ldots, F_k\}$, and a set of rules $\mathcal{R}$, the set of facts is called $\mathcal{R}$-*inconsistent* if and only if there exists a constraint $N = \neg \mathcal{F}$ such that $\mathtt{Cl}_{\mathcal{R}}(\{F_1, \ldots, F_k\}) \models \mathcal{F}$. A set of facts is said to be $\mathcal{R}$-*consistent* if and only if it is not $\mathcal{R}$-inconsistent.

A knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, composed of a set of facts (denoted by $\mathcal{F}$), a set of rules (denoted by $\mathcal{R}$) and a set of negative constraints (denoted by $\mathcal{N}$), is said to be *consistent* if and only if $\mathcal{F}$ is $\mathcal{R}$-consistent. A knowledge base is *inconsistent* if and only if it is not consistent.

The above facts $\{F_1, \ldots, F_6\}$ are $\mathcal{R}$-inconsistent with $\mathcal{R} = \{R_1, \ldots, R_4\}$ since $\{F_1, F_4, F_6\}$ activate together N_1. Moreover, R_3 can be applied on F_1 and F_5 delivering the new fact $manager(John)$ which put together with F_4 activate N_2.

3.1 Conceptual Graphs Representation

Conceptual Graphs are a visual, logic-based knowledge representation knowledge representation formalism. They encode a part of the ontological knowledge in a structure called *support*. The support consists of a number of taxonomies of the main concepts

(unary predicates) and relations (binary or more predicates) used to describe the world. Note that these taxonomies correspond to certain rules in Datalog. More complex rules (for instance representing transitivity or symmetry of relations) or rules that introduce existential variables in the conclusion are represented using Conceptual Graphs rules. Finally, negative constraints represent rules with the conclusion the absurd operator (or, logically equivalent, negation of facts).

The factual information is described using a bipartite graph in which the two classes of the partition are the concepts, and the relations respectively.

We recall the definition of support and fact following [10]. We consider here a simplified version of a support $S = (T_C, T_R, \mathcal{I})$, where: $(T_C, \leq)$ is a finite partially ordered set of *concept types*; $(T_R, \leq)$ is a partially ordered set of *relation types*, with a specified *arity*; $\mathcal{I}$ is a set of *individual markers*. A (simple) CG is a triple CG= $[S, G, \lambda]$, where:

– S is a support;
– $G = (V_C, V_R, E)$ is an ordered bipartite graph ; $V = V_C \cup V_R$ is the node set of G, V_C is a finite nonempty set of *concept nodes*, V_R is a finite set of *relation nodes*; E is the set of edges $\{v_r, v_c\}$ where the edges incident to each relation node are ordered and this ordering is represented by a positive integer label attached to the edge; if the edge $\{v_r, v_c\}$ is labeled i in this ordering then v_c is the i-neighbor of v_r and is denoted by $N_G^i(v_r)$;
– $\lambda : V \to S$ is a labeling function; if $v \in V_C$ then $\lambda(v) = (type_v, ref_v)$ where $type_v \in T_C$ and $ref_v \in \mathcal{I} \cup \{*\}$; if $r \in V_R$ then $\lambda(r) \in T_R$.

We denote a conceptual graph CG= $[S, G, \lambda]$ by G, keeping support and labeling implicit. The order on $\lambda(v)$ preserves the (pair-wise extended) order on T_C (T_R), considers $\mathcal{I}$ elements mutually incomparable, and $* \geq i$ for each $i \in \mathcal{I}$. Usually, CGs are provided with a logical semantics via the function Φ, which associates to each CG a FOL formula (Sowa (1984)). If S is a support, a constant is associated to each individual marker, a unary predicate to each concept type and a n-ary predicate to each n-ary relation type. We assume that the name for each constant or predicate is the same as the corresponding element of the support. The partial orders specified in S are translated in a set of formulae $\Phi(S)$ by the following rules: if $t_1, t_2 \in T_C$ such that $t_1 \leq t_2$, then $\forall x(t_2(x) \to t_1(x))$ is added to $\Phi(S)$; if $t_1, t_2 \in T_R$, have arity k and $t_1 \leq t_2$, then $\forall x_1 \forall x_2 \ldots \forall x_k(t_2(x_1, x_2, \ldots, x_k) \to t_1(x_1, x_2, \ldots, x_k))$ is added to $\Phi(S)$.

If CG= $[S, G, \lambda]$ is a conceptual graph then a formula $\Phi(\text{CG})$ is constructed as follows. To each concept vertex $v \in V_C$ a term a_v and a formula $\phi(v)$ are associated: if $\lambda(v) = (type_v, *)$ then $a_v = x_v$ (a logical variable) and if $\lambda(v) = (type_v, i_v)$, then $a_v = i_v$ (a logical constant); in both cases, $\phi(v) = type_v(a_v)$. To each relation vertex $r \in V_R$, with $\lambda(r) = type_r$ and $deg_G(r) = k$, the formula associated is $\phi(r) = type_r(a_{N_G^1(r)}, \ldots, a_{N_G^k(r)})$.

$\Phi(\text{CG})$ is the existential closure of the conjunction of all formulas associated with the vertices of the graph. That is, if $V_C(*) = \{v_{i_1}, \ldots, v_{i_p}\}$ is the set of all concept vertices having generic markers, then $\Phi(\text{CG})= \exists v_1 \ldots \exists v_p(\wedge_{v \in V_C \cup V_R} \phi(v))$.

If (G, λ_G) and (H, λ_H) are two CGs (defined on the same support S) then $G \geq H$ (G subsumes H) if there is a *projection* from G to H. A projection is a mapping π

from the vertices set of G to the vertices set of H, which maps concept vertices of G into concept vertices of H, relation vertices of G into relation vertices of H, preserves adjacency (if the concept vertex v in V_C^G is the ith neighbour of relation vertex $r \in V_R^G$ then $\pi(v)$ is the ith neighbour of $\pi(r)$) and furthermore $\lambda_G(x) \geq \lambda_H(\pi(x))$ for each vertex x of G. If $G \geq H$ then $\Phi(S), \Phi(H) \models \Phi(G)$ (*soundness*). *Completeness* (if $\Phi(S), \Phi(H) \models \Phi(G)$ then $G \geq H$) only holds if the graph H is in *normal form*, i.e. if each individual marker appears at most once in concept node labels.

A CG rule $(Hyp, Conc)$ expresses implicit knowledge of the form "if *hypothesis* then *conclusion*", where hypothesis and conclusion are both basic graphs. This knowledge can be made explicit by applying the rule to a specific fact: intuitively, when the hypothesis graph is found in a fact, then the conclusion graph can be added to this fact. There is a one to one correspondence between some concept nodes in the hypothesis with concept nodes in the conclusion. Two nodes in correspondence refer to the same entity. These nodes are said to be *connection nodes*. A rule can be represented by a bicolored graph or by a pair of two CGs (represented, for instance, on the right and respectively left hand side of the screen).

A rule R can be applied to a CG H if there is a homomorphism from its hypothesis to H. Applying R to H according to such a homomorphism π consists of "attaching" to H the conclusion of R by merging each connection node in the conclusion with the image by π of the corresponding connection node in the hypothesis. When a knowledge base contains a set of facts (say $\mathcal{F}$) and a set of rules (say $\mathcal{R}$), the query mechanism has to take implicit knowledge coded in rules into account. The knowledge base answers a query Q if a CG F' can be derived from $\mathcal{F}$ using the rules of $\mathcal{R}$ such that Q maps to F'.

We note that using the $(\mathcal{F}, \mathcal{R}, \mathcal{N})$ Datalog$^+$ notation or the rule based Conceptual Graphs with negative constraints has the same logical expressivity. However, the added value of using Conceptual Graphs comes from the visual depiction of the knowledge. This aspect is shown next, where the previous example knowledge base is depicted using COGUI, a Conceptual Graphs editor developed by the LIRMM, University of Montpellier 2.

3.2 COGUI CG Editor

All figures depict graphs drawn using the conceptual graph editor Cogui [1]. CoGui is a Conceptual Graphs editor. Please note that Cogui is also fully integrated with the conceptual graph engine Cogitant [2] to perform reasoning on the above mentioned graphs.

Let us consider again the knowledge base previously considered:

- $\mathcal{F}$: $F_1 : directs(John, d_1)$, $F_2 : directs(Tom, d_1)$, $F_3 : directs(Tom, d_2)$, $F_4 :$ $supervises(Tom, John)$, $F_5 : works\_in(John, d_1)$, $F_6 : works\_in(Tom, d_1)$.
- $N_1 = \forall x, y, z\, (supervises(x, y) \land work\_in(x, z) \land directs(y, z)) \rightarrow \bot$ and $N_2 = \forall x, y\, supervises(x, y) \land manager(y) \rightarrow \bot$.
- The set of rules: $R_1 = \forall x \forall d\, works\_in(x, d) \rightarrow emp(x)$ $R_2 = \forall x \forall d\, directs(x, d) \rightarrow emp(x)$
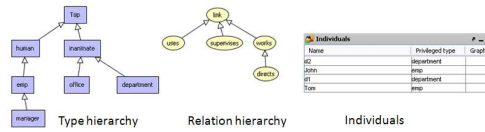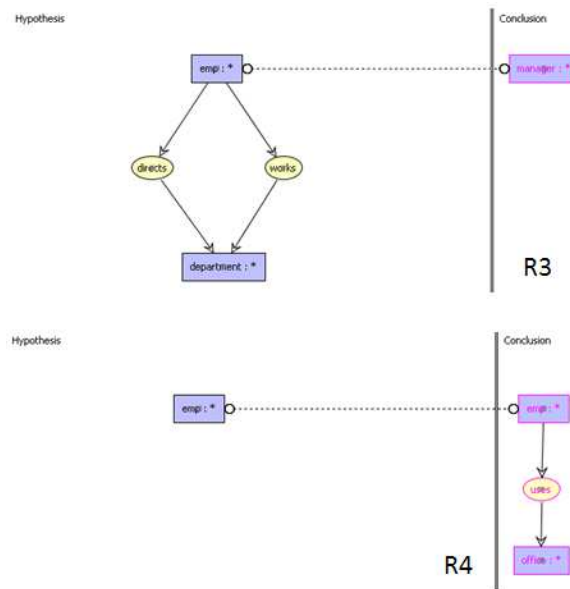

Fig. 1: Visualisation of a support (vocabulary) using CoGui

Fig. 2: Visualisation of two rules using CoGui

$$R_3 = \forall x \forall d \, directs(x,d) \wedge works\_in(x,d) \rightarrow manager(x)$$
$$R_4 = \forall x \, emp(x) \rightarrow \exists y(\textit{office}(y) \wedge uses(x,y))$$

Figure 1 presents the concept type hierarchy, the relation type hierarchy and the list of individuals. Please note that the rule hierarchy encodes the rules R_1 and R_2.

Rules R_3 and R_4 respectively are depicted in Figure 2. The negative constraints N_1 and N_2 are depicted in Figure 3.

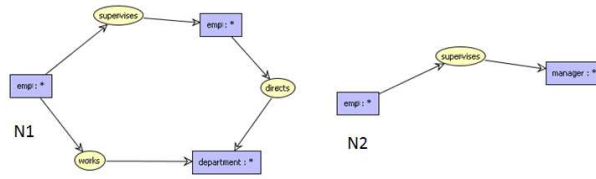Finally, the set of facts is represented in Figure 4.

142

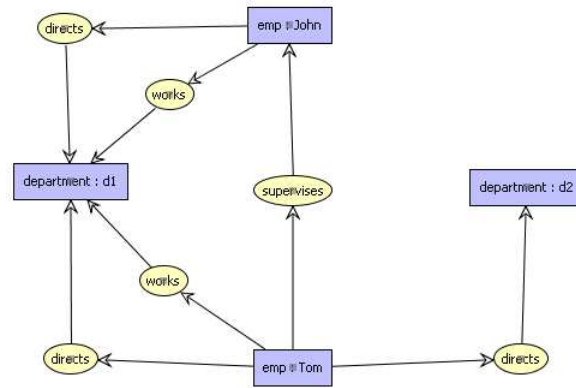Fig. 3: Visualisation of negative constraints using CoGui

Fig. 4: Visualisation of factual knowledge using CoGui

4 Dealing with inconsistency

We recall the definition of inconsistency. Given a set of facts $\{F_1, \ldots, F_k\}$, and a set of rules $\mathcal{R}$, the set of facts is called $\mathcal{R}$-*inconsistent* if and only if there exists a constraint $N = \neg \mathcal{F}$ such that $\texttt{Cl}_{\mathcal{R}}(\{F_1, \ldots, F_k\}) \models \mathcal{F}$.

In Figure 5 we can see that there is a negative constraint entailed by the facts enriched by the rules. The image of the negative constraint by homomorphism is represented in red (if color is available) or darker shade of grey (greyscale).

Like in classical logic everything can be entailed from an inconsistent knowledge base. Different semantics have been introduced in order to allow query answering in the presence of inconsistency. Here we only focus on the ICR (**I**ntersection of **C**losed **R**epair) semantics defined as follows:
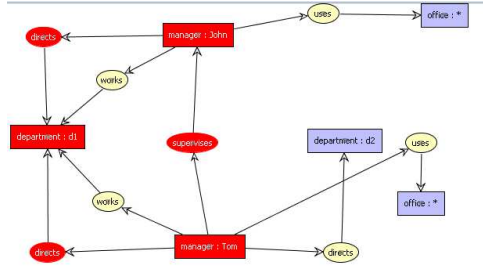
143

Fig. 5: Visualisation of factual knowledge using CoGui

Definition 1. *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and let α be a query. Then α is* ***ICR-entailed*** *from $\mathcal{K}$, written $\mathcal{K} \models_{ICR} \alpha$ if and only if $\bigcap_{A' \in \mathcal{R}epair(\mathcal{K})} \mathrm{Cl}_{\mathcal{R}}(A') \models \alpha$.*

In the above example, we obtain 6 repairs. The following are one of them (closed under set of rules):

$A_1 = \{directs(John, d_1), directs(Tom, d_1), directs(Tom, d_2),$
$supervises(Tom, John), emp(John), emp(Tom), \exists y_1(\textit{office}(y_1) \wedge uses(Tom, y_1)),$
$\exists y_2(\textit{office}(y_2) \wedge uses(John, y_2))\}$

The intersection of the closed repairs is:

$\bigcap_{\mathrm{Cl}_{\mathcal{R}}(A)} = \{directs(Tom, d_1), directs(Tom, d_2), emp(Tom),$
$\exists y\, uses(Tom, y), \exists y\textit{office}(y)\}$.

Another possibility to deal with an inconsistent knowledge base in the OBDA setting is to define an instantiation [11] of Dung's abstract argumentation theory [13]. An argumentation framework is composed of a set of arguments and a binary relation defined over arguments, the attack.

Definition 2 (Argument). *[11] An argument A in a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is a tuple $A = (F_0, \ldots, F_n)$ where:*

- *$(F_0, \ldots, F_{n-1})$ is a derivation sequence w.r.t $\mathcal{K}$.*
- *F_n is an atom, a conjunction of atoms, the existential closure of an atom or the existential closure of a conjunction of atoms such that $F_{n-1} \models F_n$.*

We can extract from each argument its sub-arguments.

Definition 3 (Sub-argument). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and $A = (F_0, F_1, \ldots, F_n)$ be an argument. $A' = (F_0, \ldots, F_k)$ with $k \in \{0, ..., n-1\}$ is a sub-argument of A iff (i) $A' = (F_0, \ldots, F_k)$ is an argument and (ii) $F_k \in F_{k+1}$.*

Let $A = (F_0, ..., F_n)$ be an argument, then $\mathrm{Supp}(A) = F_0$ and $\mathrm{Conc}(A) = F_n$. Let $S \subseteq \mathcal{F}$ a set of facts, $Arg(S)$ is defined as the set of all arguments A such that $\mathrm{Supp}(A) \subseteq S$.

An argument corresponds to a rule derivation. Therefore we can use the Cogui editor in order to depict arguments (via the depiction of rule derivations). In Figure 6 an example of a derivation is depicted. The added information by the rule is visible due to the changed color (pink in color, darker shade of grey on grey scale).
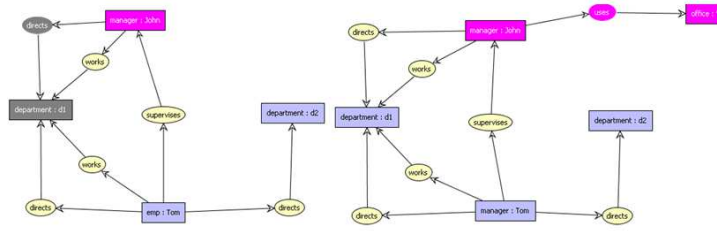

Fig. 6: Visualisation of a rule derivation using CoGui

Definition 4 (Attack). *[11] Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and let $a, b \in \mathcal{A}$. The argument a attacks b, denoted by $(a, b) \in Att$, iff there exists $\varphi \in \text{Supp}(b)$ such that the set $\{Conc(a), \varphi\}$ is $\mathcal{R}$-inconsistent.*

Definition 5 (Argumentation framework). *[11] Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, the corresponding argumentation framework $\mathcal{AF}_K$ is a pair $(\mathcal{A} = \text{Arg}(\mathcal{F}), Att)$ where $\mathcal{A}$ is the set of arguments that can be built from $\mathcal{F}$ and Att is the attack relation. Let $\mathcal{E} \subseteq \mathcal{A}$ and $a \in \mathcal{A}$.*

• *We say that $\mathcal{E}$ is* conflict free *iff there exists no arguments $a, b \in \mathcal{E}$ such that $(a, b) \in Att$.*

• *$\mathcal{E}$ defends a iff for every argument $b \in \mathcal{A}$, if we have $(b, a) \in Att$ then there exists $c \in \mathcal{E}$ such that $(c, b) \in Att$.*

• *$\mathcal{E}$ is* admissible *iff it is conflict free and defends all its arguments.*

• *$\mathcal{E}$ is a* preferred extension *iff it is maximal (with respect to set inclusion) admissible set.*

• *$\mathcal{E}$ is a* stable extension *iff it is conflict-free and $\forall a \in \mathcal{A} \backslash \mathcal{E}$, there exists an argument $b \in \mathcal{E}$ such that $(b, a) \in Att$.*

• *$\mathcal{E}$ is a* grounded extension *iff $\mathcal{E}$ is a minimal (for set inclusion) complete extension.*

We denote by $\text{Ext}(\mathcal{AF}_K)$ the set of extensions of $\mathcal{AF}_K$. We use the abbreviations p, s, and g for respectively preferred, stable and grounded semantics. An argument is skeptically accepted if it is in all extensions, credulously accepted if it is in at least one extension and rejected if it is not in any extension.

The following results are then showed by [11]:

145

Theorem 1. *[11] Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, let $\mathcal{AF}_K$ be the corresponding argumentation framework, α be a query, and $x \in \{s, p\}$ be stable or preferred semantics. Then $\mathcal{K} \models_{ICR} \alpha$ iff α sceptically accepted under semantics x.*

5 Argumentative Explanation

In this section we define two different heuristics for explanation of inconsistency tolerant semantics. Since these heuristics work under inconsistent knowledge bases the Cogui editor is not yet adapted to implement them. We note that explanations correspond to the notion of argument, thus, the Cogui visual power could be easily adapted for our case. Moreover, in section 5.1 we show the equivalence between one type of explanation and a visual rule depiction in Cogui. This could be a starting point for the explanation of queries under inconsistency using Cogui.

When handling inconsistent ontological knowledge bases we are interested in the *explanation* of query answers conforming to a given semantics. More precisely we are interested in explaining why a query α is ICR-entailed by an inconsistent knowledge base $\mathcal{K}$. By explanation we mean a *structure* that has to incorporate minimal set of facts (w.r.t $\subseteq$) and general rules that, if put together, will lead to the entailment of the query α. According to this intuition (which coincides with the definition of [17]) and the link between inconsistent ontological knowledge bases and logic-based argumentation framework, a first candidate of explanation is an argument. However, an argument as defined in definition 2 can be cumbersome and difficult to understand, because the information of how these derivations have been achieved and how they lead to the conclusion are missing. Therefore we propose a refined explanation that incorporates rules as a crucial component.

Definition 6 (Explanation). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be an inconsistent knowledge base, let α be query and let $\mathcal{K} \models_{ICR} \alpha$. An explanation for α in $\mathcal{K}$ is a 3-tuple $E = (A, G, C)$ composed of three finite sets of formulae such that: (1) $A \subseteq \mathcal{F}$, $G \subseteq \mathcal{R}$, (2) $C \models \alpha$, (3) $\mathtt{Cl}_G(A) \not\models \bot$ (consistency), (4) For every formula β in A, $\mathtt{Cl}_G(A - \beta) \not\models C$ (minimality). Such that $\mathtt{Cl}_G$ represents the closure w.r.t to the set of rules G.*

We denote by $\mathcal{EXP}$ the universe of explanations and by $\mathcal{EXP}_\alpha$ the set of all explanations for α. We denote the sets A, G and C as antecedents, general laws and conclusions respectively. Here the definition specifies three important components for explaining query α. First, the set A of antecedent conditions which is a minimal subset of facts that entails the query α. Second, the set of general laws G (from now on, rules) that produce the query α, the reason for integrating rules is that the user is often interested in knowing how we achieved the query. Finally, the third component is the conclusion C (the answer for the query α). The definition also imposes a central concept, namely *explanation consistency*.

An explanation can be computed directly from $\mathcal{K}$ or from an argumentation framework using the following mapping $\mathbb{A}$.

Definition 7 (Mapping $\mathbb{A}$). *Given an inconsistent knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ and $\mathcal{AF}_K = (\mathcal{A}, \mathtt{Att})$ the corresponding argumentative framework. The mapping $\mathbb{A}$ is a total function defined on $\mathcal{A} \longrightarrow \mathcal{EXP}$ as $E = \mathbb{A}((F_0, ..., F_n))$ with:*

- *The set of antecedent conditions $A = F_0$,*
- *The set of rules $G \subseteq \mathcal{R}$, such that $\forall r \in \mathcal{R}$, $r \in G$ iff for all F_i in x the rule r is applicable to F_i.*
- *The conclusion $C = F_n$ iff $\mathtt{Cl}_{\mathcal{G}}(A) \vDash F_n$.*

Proposition 1 (Bijection of $\mathbb{A}$). *For any argument $a \in \mathcal{A}$, the mapping $\mathbb{A}$ is a bijection.*

The proposition follows from the definition of the mapping because for every argument we can construct one explanation. Since the mapping is a bijection, we call the argument $x_e = \mathbb{A}^{-1}(e)$ the corresponding argument of an explanation e. We say the argument x_e supports the explanation e. The following proposition states that there is always an explanation for an ICR-entailed query.

Proposition 2 (Existence of explanation). *For every query α such that $\mathcal{K} \vDash_{ICR} \alpha$, the set $\mathcal{EXP}_\alpha$ is not empty.*

Proof 1 *On the one hand, if $\mathcal{K} \vDash_{ICR} \alpha$ then the query α is sceptically accepted. That means $\forall \mathcal{E} \in \mathtt{Ext}(\mathcal{AF}_K), \mathcal{E} \vDash \alpha$. Hence there is an argument $a \in \mathcal{E}$ such that $Cons(a) \vDash \alpha$. On the other hand, using the mapping $\mathbb{A}$ we have $e = \mathbb{A}(a)$ is an explanation for α, namely $e \in \mathcal{EXP}_\alpha$. Consequently $\mathcal{EXP}_\alpha \neq \emptyset$*

Example 1 (Corresponding Argument). Let us explain $\alpha = \exists x\, emp(x)$. We can build the following argument for α:
$a_\alpha^+ = (\{works\_in(Tom, d_1)\}, \{works\_in(Tom, d_1), emp(Tom)\}, emp(Tom))$,
and the delivered explanation is:
$E_\alpha = (\{directs(Tom, d_1)\}, \{\forall x \forall d\, works\_in(x, d) \rightarrow emp(x)\}, emp(Tom))$.

There could be cases where the user wants to know how the set of *rules and facts* interact in order to explain a query α. Put differently, a user-invoked explanation that makes explicit any relation between the facts and the rules which lead to α. Notice that, this type of user-invoked explanation is called *deepened* explanation and it should not confounded with a proof-like explanation, because we are considering an inconsistent and incomplete settings. For this reason the explanation below has not yet been implemented as a stand alone plugin for Cogui (Cogui only deals with querying consistent knowledge).

5.1 Deepened Explanation (d-explanation)

Definition 8 (d-explanation). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be an inconsistent knowledge base, let α be a query and let $\mathcal{K} \vDash_{ICR} \alpha$. Then, the finite sequence of tuples $d = \langle t_1, t_2, ..., t_n \rangle$ is a d-explanation for α iff:*

1. *For every tuple $t_i = (a_i, r_i) \in d$ such that $i \in \{1, ..., n\}$, it holds that $a_i \subseteq \mathtt{Cl}_{\mathcal{R}}(\mathcal{F})$ and $r_i \in \mathcal{R}$.*
2. *For every tuple $t_i = (a_i, r_i) \in d$ such that $i \in \{2, ..., n\}$ we have $a_i = a_i' \cup a_i''$ where (i) $r_{i-1}(a_{i-1}) \vDash a_i'$, (ii) $a_i'' \subseteq \mathtt{Cl}_{\mathcal{R}}(\mathcal{F})$ and (iii) r_i is applicable to a_i. Note that if $i = 1$ then $a_i' = \emptyset$.*
3. *The tuple (a_n, r_n) entails α (i.e $r_n(a_n) \vDash \alpha$).*

147

4. $\text{Cl}_{\mathcal{R}}(\cup_{i=0}^{n} a_i) \nvDash \perp$ *(consistency).*

We denote by $\mathcal{D}$ the universe of d-explanations and by $\mathcal{D}_{\alpha}$ the set of all d-explanations for a query α.

The intuition about the d-explanation d is as follows: tuples in d represent $\langle fact, applicable\,rule \rangle$, and the sequence of tuples represents the order by which we achieve the answer of the query. Think of it as a chain where each a_i has a link with the previous a_{i-1} through the rule r_{i-1}. This is similar to the notion of derivation depicted in Figure 6.

Example 2 (Deepened explanation). The deepened explanation associated to α is the same as E and doesn't provide more information. Let us consider the explanation of $\alpha_2 = \exists x\,office(x)$. A possible argument for α_2 is:

$a_{\alpha_2}^{+} = (\{works\_in(Tom, d_1)\}, \{works\_in(Tom, d_1), emp(Tom)\},$
$\{works\_in(Tom, d_1), emp(Tom), \exists y(office(y) \wedge uses(Tom, y))\},$
$\exists y(office(y) \wedge uses(Tom, y)))$.

So $E_{\alpha_2} = (\{works\_in(Tom, d_1)\}, \{\forall x \forall d\,directs(x, d) \rightarrow emp(x), \forall x\,emp(x) \rightarrow \exists y(office(y) \wedge uses(x, y))\}, \exists y office(y))$.

$D_E = (\langle works\_in(Tom, d_1), \forall x \forall d\,directs(x, d) \rightarrow emp(x) \rangle,$
$\langle emp(Tom), \forall x\,emp(x) \rightarrow \exists y(office(y) \wedge uses(x, y)) \rangle)$.

There is a bijection between an explanation e and a d-explanation d represented here by the following mapping.

Definition 9 (Mapping $\mathbb{D}$). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be an inconsistent knowledge base, α be a query, $e = (A, G, C) \in \mathcal{EXP}$ be an explanation for α and $d = \langle t_1, t_2, ..., t_n \rangle \in \mathcal{D}$ be a d-explanation for α. The mapping $\mathbb{D}$ is total function $\mathbb{D} : \mathcal{EXP} \longrightarrow \mathcal{D}, e \longrightarrow d$ defined as follows:*
1. *For every tuple $t_i = (a_i, r_i)$ such that $i \in \{1, ..., n\}$, it holds that $r_i \in G$.*
2. *For every tuple $t_i = (a_i, r_i)$ in D such that $i \in \{2, ..., n\}$ we have $a_i = a'_i \cup a''_i$ where $r_{i-1}(a_{i-1}) \vDash a'_i, a''_i \subseteq A$ and r_i is applicable to a_i. Note that if $i = 1$ then $a_i = A$ and e_i is applicable to a_i.*
3. *The tuple (a_n, r_n) entails α (i.e $r_n(a_n) \vDash \alpha$) and $C \vDash \alpha$.*

Since the mapping is a bijection the existence of the inverse function is guaranteed. Thereby we consider the mapping $\mathbb{D}(e)$ as deepening the explanation e and the inverse mapping $\mathbb{D}^{-1}(d)$ as simplifying the d-explanation d. The advantage of such a mapping is that it gives the users the freedom to shift from an explanation to another which complies better with their level of understanding and their experiences. Also it guarantees that every explanation can be deepened. As done before, we also define the corresponding argument x_d for a d-explanation d, as the corresponding argument x_e for an explanation $e = \mathbb{D}^{-1}(d)$. This can be achieved by the following composition of function: $x_d = (\mathbb{D} \circ \mathbb{A})^{-1}(d)$.

6 Conclusion

In this paper we have presented an argumentative approach for explaining user query answers in a particular setting, Namely, an inconsistent ontological knowledge base

where inconsistency is handled by inconsistency-tolerant semantics (ICR) and it is issued from the set of facts. In this paper we have exploited the relation between ontological knowledge base and logical argumentation framework to establish different levels of explanation ranging from an explanation based on the notion of argument to a user-invoked explanation called deepened explanation. We have also shown the relation between every type of explanation using a one-to-one correspondence which gives the user the possibility to deepen (or simplify) the explanation in hand. Future works aims at studying the proposed explanation in the context of other inconsistency-tolerant semantics. We are currently working on a Cogui based plug-in that only deals with reasoning under inconsistency and the above mentioned semantics.

7   Acknowledgments

References

1. T. Arora, R. Ramakrishnan, W. Roth, P. Seshadri, and D. Srivastava. Explaining program execution in deductive systems. In S. Ceri, K. Tanaka, and S. Tsur, editors, *Deductive and Object-Oriented Databases*, volume 760 of *Lecture Notes in Computer Science*, pages 101–119. Springer Berlin Heidelberg, 1993.
2. F. Baader, S. Brandt, and C. Lutz. Pushing the el envelope. In *Proc. of IJCAI 2005*, 2005.
3. F. Baader, R. Peñaloza, and B. Suntisrivaraporn. Pinpointing in the description logic el. In J. Hertzberg, M. Beetz, and R. Englert, editors, *KI 2007: Advances in Artificial Intelligence*, volume 4667 of *Lecture Notes in Computer Science*, pages 52–67. Springer Berlin Heidelberg, 2007.
4. F. Baader and B. Suntisrivaraporn. Debugging snomed ct using axiom pinpointing in the description logic el+. In *KR-MED*, volume 410, 2008.
5. J.-F. Baget, M.-L. Mugnier, S. Rudolph, and M. Thomazo. Walking the complexity lines for generalized guarded existential rules. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence, (IJCAI'11)*, pages 712–717, 2011.
6. M. Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proc of AAAI*, 2012.
7. A. Borgida, E. Franconi, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider. Explaining ALC subsumption. In P. Lambrix, A. Borgida, M. Lenzerini, R. Möller, and P. F. Patel-Schneider, editors, *Proceedings of the 1999 International Workshop on Description Logics 1999*, volume 22, Linköping, Sweden, July 1999.
8. A. Calì, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In *Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 77–86. ACM, 2009.
9. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The dl-lite family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
10. M. Chein and M.-L. Mugnier. *Graph-based Knowledge Representation and Reasoning—Computational Foundations of Conceptual Graphs*. Advanced Information and Knowledge Processing. Springer, 2009.

11. M. Croitoru and S. Vesic. What can argumentation do for inconsistent ontology query answering? In W. Liu, V. Subrahmanian, and J. Wijsen, editors, *Scalable Uncertainty Management*, volume 8078 of *Lecture Notes in Computer Science*, pages 15–29. Springer Berlin Heidelberg, 2013.

12. X. Deng, V. Haarslev, and N. Shiri. A framework for explaining reasoning in description logics. In *IN: PROCEEDINGS OF THE AAAI FALL SYMPOSIUM ON EXPLANATION-AWARE COMPUTING*, pages 189–204. AAAI Press, 2005.

13. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-persons games. *Artificial Intelligence*, 77(2):321–357, 1995.

14. A. Garcıa, C. I. Chesnevar, N. D. Rotstein, and G. R. Simari. An abstract presentation of dialectical explanations in defeasible argumentation. *ArgNMR07*, pages 17–32, 2007.

15. P. Godfrey. Minimization in cooperative response to failing database queries. *International Journal of Cooperative Information Systems*, 06(02):95–149, 1997.

16. P. Godfrey, J. Minker, and L. Novik. An architecture for a cooperative database system. In W. Litwin and T. Risch, editors, *Applications of Databases*, volume 819 of *Lecture Notes in Computer Science*, pages 3–24. Springer Berlin Heidelberg, 1994.

17. C. G. Hempel and P. Oppenheim. Studies in the logic of explanation. *Philosophy of Science*, 15(2):pp. 135–175, 1948.

18. M. Horridge, B. Parsia, and U. Sattler. Explaining inconsistencies in owl ontologies. In L. Godo and A. Pugliese, editors, *Scalable Uncertainty Management*, volume 5785 of *Lecture Notes in Computer Science*, pages 124–137. Springer Berlin Heidelberg, 2009.

19. M. Horridge, B. Parsia, and U. Sattler. Justification oriented proofs in owl. In *Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part I*, ISWC'10, pages 354–369, Berlin, Heidelberg, 2010. Springer-Verlag.

20. D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. F. Savo. Inconsistency-tolerant semantics for description logics. In *Proc. of RR*, pages 103–117, 2010.

21. T. Lukasiewicz, M. V. Martinez, G. I. Simari, et al. Inconsistency handling in datalog+/-ontologies. In *ECAI*, pages 558–563, 2012.

22. D. L. McGuinness and A. T. Borgida. Explaining subsumption in description logics. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 816–821, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

23. A. Meliou, W. Gatterbauer, J. Y. Halpern, C. Koch, K. F. Moore, and D. Suciu. Causality in databases. *IEEE Data Eng. Bull.*, 33(3):59–67, 2010.

24. A. Meliou, W. Gatterbauer, K. F. Moore, and D. Suciu. Why so ? or why no ? functional causality for explaining query answers. In *Proceedings of the Fourth International VLDB workshop on Management of Uncertain Data (MUD 2010) in conjunction with VLDB 2010, Singapore, September 13, 2010*, 2010.

25. S. Modgil and M. Caminada. Proof theories and algorithms for abstract argumentation frameworks. In *Argumentation in artificial intelligence*, pages 105–129. Springer, 2009.

26. S. Schlobach. Debugging and semantic clarification by pinpointing. In *Proceedings of the Second European Conference on The Semantic Web: Research and Applications*, ESWC'05, pages 226–240, Berlin, Heidelberg, 2005. Springer-Verlag.

27. S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pages 355–360, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

28. D. Seselja and C. Strasser. Abstract argumentation and explanation applied to scientific debates. *Synthese*, 190(12):2195–2217, 2013.

Chapter 6

Application Papers

Conflicting Viewpoint Relational Database Querying: an Argumentation Approach

(Extended Abstract)

Nouredine Tamani
INRIA/LIRMM GraphiK
161 rue Ada
Montpellier, France
tamani@lirmm.fr

Madalina Croitoru
LIRMM Graphik UM2
161 rue Ada
Montpellier, France
croitoru@lirmm.fr

Patrice Buche
UMR IATE INRA
2 place Pierre Viala
Montpellier, France
buche@supagro.inra.fr

ABSTRACT

Within the framework of the European project EcoBioCap, we model a real world use case aiming at conceiving the next generation of food packagings. The objective is to select packaging materials according to possibly conflicting requirements expressed by the involved parties (food and packaging industries, health authorities, consumers, waste management authority, etc.). The requirements and user preferences are modeled by several ontological rules provided by the stakeholders expressing their viewpoints and expertise. Since several aspects need to be considered (CO2 and O2 permeance, interaction with the product, sanitary, cost, end of life, etc.) in order to select objects, an argumentation process can be used to express/reason about different aspects or criteria describing the packagings. We define then in this paper an argumentation approach which combines a description logic (DLR-Lite) within ASPIC framework for relational database querying. The argumentation step is finally used to express and/or enrich a bipolar query employed for packaging selection.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architecture

Keywords

Argumentation; decision support system; description logics and DLR-Lite; application within the EcoBioCap project.

1. INTRODUCTION

Within the framework of the European project EcoBioCap (www.ecobiocap.eu) about the design of next generation packagings using advanced composite structures based on constituents derived from the food industry, we aim at developing a Decision Support System (DSS) for packaging material selection. The DSS will consist of two steps: (1) aggregating possibly conflicting needs expressed by several parties involved in the considered field and (2) querying a database of packagings with the resulting aggregation obtained at point (1). The problem at hand does not simply

consist in addressing a multi-criteria optimization problem [4]: the domain experts would need to be able to justify why a certain packaging (or set of possible packagings) are chosen. Argumentation theory in general [8, 3, 11] is actively pursued in the literature, some approaches even combining argumentation and multi criteria decision making [2].

We rely in this work on a logical structured argumentation system [1, 10, 9] since it (i) allows the expression of logical arguments as a combination of facts and inference rules, (ii) defines attacks and defeat relations between arguments based on a logical conflict notion. Stakeholder's set of arguments i is then modeled as concepts, facts and rules to build a partial knowledge bases $\mathcal{K}_{\mathcal{I}_i}$. The union of every stakeholder knowledge base $\mathcal{K} = \bigcup_{i=1,\ldots,n} \mathcal{K}_{\mathcal{I}_i}$ will be used to instantiate the ASPIC [1] argumentation system. The solution developed in this paper is to instantiate for each criterion, called viewpoint or aspect, an argumentation system to reason about arguments solely expressed on it. This will then be used to generate the query on the packaging database. The main contribution of this paper is to demonstrate the use of argumentation in a real world industrial scenario within the EcoBioCap project. To this aim we show how to instantiate ASPIC with the DLR-Lite logic modeling expert ontologies in this real world scenario.

2. CONTRIBUTIONS

The main contributions of the paper are the following:

1. A DLR-Lite [7, 5] ontology extended to a negation to express stakeholders' arguments about packaging characteristics as combination of concepts (defined as m-ary relations connected to a database) and inference rules (specified as subsumptions). The language is detailed in the technical report [12],

2. An instantiation of ASPIC argumentation system AS with the proposed DLR-Lite logical language. The instantiated ASPIC AS satisfies the rationality postulates [6], please see details in [12],

3. The study of the influence of the modeling rules on the argumentation results. We showed the limitation of the crisp split of the inference rules into *defeasible* and *strict*, and we propose to overcome this limitation a viewpoint approach in which arguments are gathered according to packaging aspects. Each viewpoint delivers subsets of non-conflicting arguments supporting or
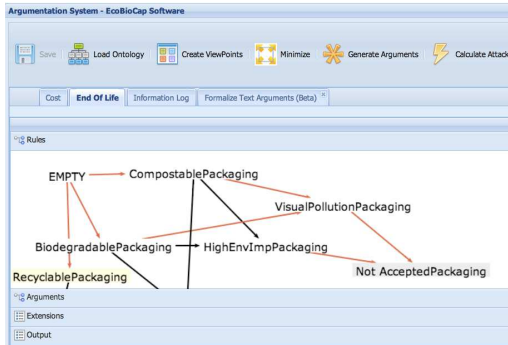
Figure 1: The user interface of the system.

opposing a kind of packaging according to a single aspect (respiration parameters, cost, materials, sanitary, end elf life, etc.),

4. The use of the argumentation results for a bipolar querying of the packaging database. Indeed, we can gather the results onto positive and negative collections. We can then deduce automatically such queries from the collections the users formed during the argumentation process. We can also carry out an analogical reasoning by generalizing results obtained from an argumentation process applied upon instances, where an instance of the sought objects can help to better understand the involved stakeholders' needs and then to be able to express, based on arguments pros and cons, a query reflecting the way objects should be selected from a database,

5. Implementation of the approach within the EcoBio-Cap project (www.ecobiocap.eu). A java GXT/GWT web interface was developed and a open version is accessible on *pfl.grignon.inra.fr/EcoBioCapProduction/*. The main difficulties encountered were the translation of text arguments into DLR-Lite formal representation. In the freely available version, stakeholders' arguments are provided as a manually built XML file specifying viewpoints and rules. The system generates then arguments and attacks and computes the extensions (stable, preferred, admissible, grounded, naive, etc. semantics) inside each view. Figure 1 shows the main interface of the application and a fragment of rules formalizing an argumentation scenario about the aspect *end of life* of packagings. Stakeholders argued about biodegradability, recyclability and compostability (the test XML file is accessible on *https://docs.google.com/ file/d/0B0DPgJDRNwbLR2RjWWhwMjgwVEU/edit? usp=sharing).*

3. CONCLUSION

We applied in this paper an argumentation approach on a real use case from the industry, based on a combination of ASPIC *AS* with a DLR-Lite specifications allowing stakeholders to express their preferences and providing the system with concepts and subsumptions in the packaging domain.

We have proposed an argumentation system in which each criterion is considered as a viewpoint in which stakeholders express their arguments in homogenous way. The set of non conflicting viewpoints are then gathered according goals, to form consistent collections which support/oppose them.

We plan to extend the proposed approach to fuzzy argumentation to make it possible to deal with vague and uncertain concepts and rules by exploiting the fuzzy interpretation of the fuzzy DLR-Lite. Another line to develop consists of studying the bipolarity in our context of argumentation.

5. REFERENCES

[1] L. Amgoud, L. Bodenstaff, M. Caminada, P. McBurney, S. Parsons, H. Prakken, J. Veenen, and G. Vreeswijk. Final review and report on formal argumentation system.deliverable d2.6 aspic. Technical report, 2006.

[2] L. Amgoud and H. Prade. Using arguments for making and explaining decisions. *Artificial Intelligence*, 173(3-4):413–436, 2009.

[3] P. Besnard and A. Hunter. *Elements of Argumentation*. The MIT Press, 2008.

[4] D. Bouyssou, D. Dubois, M. Pirlot, and H. Prade. *Decision-making process – Concepts and Methods*. Wiley, 2009.

[5] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *KR*, pages 260–270, 2006.

[6] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171:286–310, 2007.

[7] S. Colucci, T. D. Noia, A. Ragone, M. Ruta, U. Straccia, and E. Tinelli. *Semantic Web Information Management*, chapter 19 : Informative Top-k retrieval for advanced skill management, pages 449–476. Springer-Verlag Belin Heidelberg, 2010.

[8] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-persons games. *Artificial Intelligence*, 77(2):321–357, 1995.

[9] S. Modgil and H. Prakken. A general account of argumentation with preferences. *Artificial Intelligence*, 195:361–397, 2013.

[10] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2011.

[11] I. Rahwan and G. Simari. *Argumentation in Artificial Intelligence*. Springer, 2009.

[12] N. Tamani, M. Croitoru, and P. Buche. A viewpoint approach to structured argumentation. https://docs.google.com/file/d/0B0DPgJDRNwbLRml qUVh4cGFrSVk/edit?usp=sharing. Technical report, INRA-SupAgro, 2013.

Eco-Efficient Packaging Material Selection for Fresh Produce: Industrial Session

Nouredine Tamani[1], Patricio Mosse[2], Madalina Croitoru[1], Patrice Buche[2],
Valérie Guillard[2], Carole Guillaume[2], Nathalie Gontard[2]

[1]LIRMM, University Montpellier 2, France
[2]UMR IATE INRA 2 place Pierre Viala Montpellier, France

Abstract. Within the framework of the European project EcoBioCap (ECOefficient BIOdegradable Composite Advanced Packaging), we model a real world use case aiming at conceiving the next generation of food packagings. The objective is to select packaging materials according to possibly conflicting requirements expressed by the involved parties (food and packaging industries, health authorities, consumers, waste management authority, etc.). The requirements and user preferences are modeled by several ontological rules provided by the stakeholders expressing their viewpoints and expertise. To deal with these several aspects (CO_2 and O_2 permeance, interaction with the product, sanitary, cost, end of life, etc.) for packaging selection, an argumentation process has been introduced.

1 Introduction

Within the framework of the European project EcoBioCap (www.ecobiocap.eu) about the design of next generation packagings using advanced composite structures based on constituents derived from the food industry, we aim at developing a Decision Support System (DSS) to help parties involved in the packaging design to make rational decisions based on knowledge expressed by the experts of the domain.

The DSS is made of two parts, as depicted in Figure 1:

1. a flexible querying process which is based on a bipolar approach dealing with imprecise data [8] corresponding to the characteristics related to the food product to pack like the optimal permeance, the dimension of the packaging, its shape, etc.,
2. an argumentation process which aims at aggregating several stakeholders (researchers, consumers, food industry, packaging industry, waste management policy, etc.) requirements expressed as simple textual arguments, to enrich the querying process by stakeholders' justified preferences. Each argument supports/opposes a choice justified by the fact that it either meets or not a requirement according to a particular aspect of the packagings.

We implementated of the second part of the DSS, called argumentation system, which aims at aggregating preferences associated with justifications expressed by stakeholders about the characteristics of a packaging. This module
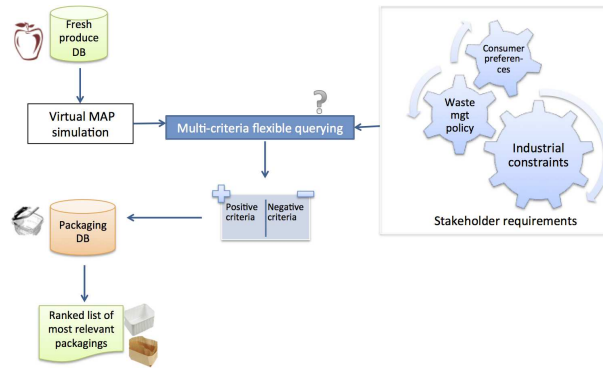
Fig. 1. Global insight of the EcoBioCap DSS.

has as inputs stakeholders' arguments supporting or opposing a packaging choice which could be seen as preferences combined with their justifications, and returns consensual preferences which may be candidates to enrich the bipolar querying system.

The DSS consists of two steps: (i) aggregating possibly conflicting needs expressed by the involved several parties (ii) querying a database of packagings with the resulting aggregation obtained at point (i).

In this real case, packagings have to be selected according to several aspects or criteria (permeance, interaction with the packed food, end of life, etc.), highlighted by the expressed stakeholders' arguments. The problem at hand does not simply consist in addressing a multi-criteria optimization problem [4]: the domain experts would need to be able to justify why a certain packaging (or set of possible packagings) are chosen. Argumentation theory in general [9, 3, 11] is actively pursued in the literature, some approaches even combining argumentation and multi criteria decision making [2].

2 Approach

Stakeholder's set of arguments i is then modeled as concepts, facts and rules to build a partial knowledge bases $\mathcal{K}_{\mathcal{I}_i}$. The union of every stakeholder knowledge base $\mathcal{K} = \bigcup_{i=1,...,n} \mathcal{K}_{\mathcal{I}_i}$ will be used to instantiate the ASPIC [1] argumentation system, as shown on Figure 2.

The main salient points of our work in the EcoBioCap project are the following:

1. A DLR-Lite [7, 5] ontology extended to a negation to express stakeholders' arguments about packaging characteristics as combination of concepts

155
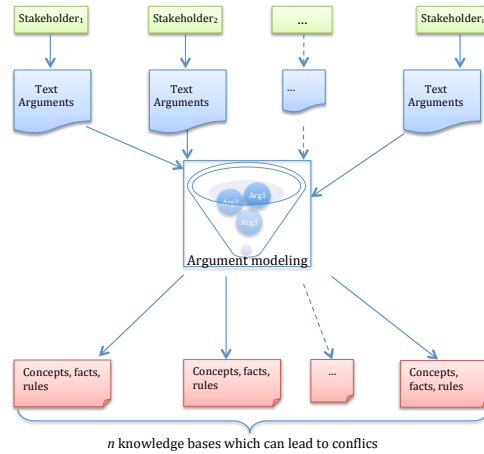
Fig. 2. The global knowledge base of the system.

(defined as m-ary relations connected to a database) and inference rules (specified as subsumptions). The language is detailed in the technical report [12],

2. An instantiation of ASPIC argumentation system (AS) with the proposed DLR-Lite logical language. The instantiated ASPIC AS satisfies the rationality postulates [6], please see details in [12],

3. The study of the influence of the modeling rules on the argumentation results. We showed the limitation of the crisp split of the inference rules into *defeasible* and *strict*, and we propose to overcome this limitation a viewpoint approach in which arguments are gathered according to packaging aspects. Each viewpoint delivers subsets of non-conflicting arguments supporting or opposing a kind of packaging according to a single aspect (shelf life parameters, cost, materials, sanitary, end elf life, etc.),

4. The use of the argumentation results for a bipolar querying of the packaging database. Indeed, we can gather the results onto positive and negative collections. We can then deduce automatically such queries from the collections the users formed during the argumentation process.

5. Implementation of the approach. A java GXT/GWT web interface was developed and a open version is accessible on
http://pfl.grignon.inra.fr/EcoBioCapProduction/.

156

3 Architecture of the argumentation system

As illustrated in Figure 3, the proposed argumentation system relies on 5 main modules, described below.

Fig. 3. The architecture of the argumentation system.

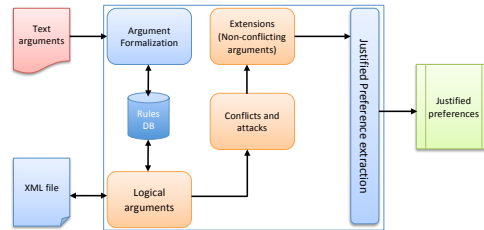- *Argument formalization module*: this module implements a user-friendly interface for a semi-automatic translation of text arguments into a formal representation made of concepts and rules (claims and hypothesis). A graphical representation of the expressed rules is also built as the users formalize their text arguments. The formal representation obtained is finally saved in a database for a persistent storage allowing to reload argumentation projects without rebuilding all the arguments and to reuse also the already formatted rules in other projects.
- *Logical arguments*: this module receives as inputs the list of concepts and rules corresponding to text arguments. This list can be the result of the formalization module or given by the user as an XML file. Then, by a derivation process, this module builds all possible arguments according to the logical process defined in ASPIC/ASPIC+ logic-based argumentation frameworks [1, 10] and reused in [13, 14]. This modules implements also a function to export the argument list into an XML document.
- *Conflicts and attacks*: this module relies on the logical arguments built by the previous module. According to the negation operator, it detects all the conflicts among arguments and models them as attacks with respect to the definition of attacks introduced in [13, 14]. The output of this module is an argumentation graph made of arguments (nodes) and attacks (edges).
- *Extensions*: an extension is a subset of non-conflicting (consistent) arguments which defend themselves from attacking arguments. The computation of extensions is made under one semantics (preferred, stable, grounded, etc.) as defined in [9]. This module allows the computation of one or all semantics considered (preferred, stable, grounded, eager, semi-stable). We notice that

theoretically we can get empty extensions under any semantics. This situation occurs when a user expresses at least one self-defeated argument, which is not attacked by any other argument, but attacks all the others. This kind of arguments are called contaminating arguments [15]. The current version of the system detects the rules leading to such arguments and discards them before performing the process of extension computations.

– *Extraction of the justified preferences*: the computation of extensions delivers one or several extensions. In the case of several extensions, the system lets the users selecting the more suitable one according to their objectives. The selected extension is then used to extract corresponding preferences underlying the contained concepts. These preferences are expressed as a list of couples ($attribut, value$), where $attribute$ stands for a packaging attribute as defined in the packaging database schema of the flexible querying system part of the DSS, and $value$ is the preferred value expressed for the considered attribute.

4 Conclusion

We applied an argumentation approach on a real use case from the industry allowing stakeholders to express their preferences and providing the system with stable concepts and subsumptions of a domain. We have proposed an argumentation system in which each criterion (attribute or aspect) is considered as a viewpoint in which stakeholders express their arguments in homogenous way. Each viewpoint delivers extensions supporting or opposing certain choices according one packaging aspect, which are then used in the querying process. The approach is implemented as freely accessible web application.

References

1. L. Amgoud, L. Bodenstaff, M. Caminada, P. McBurney, S. Parsons, H. Prakken, J. Veenen, and G. Vreeswijk. Final review and report on formal argumentation system.deliverable d2.6 aspic. Technical report, 2006.
2. L. Amgoud and H. Prade. Using arguments for making and explaining decisions. *Artificial Intelligence*, 173(3-4):413–436, 2009.
3. P. Besnard and A. Hunter. *Elements of Argumentation*. The MIT Press, 2008.
4. D. Bouyssou, D. Dubois, M. Pirlot, and H. Prade. *Decision-making process – Concepts and Methods*. Wiley, 2009.
5. D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *KR*, pages 260–270, 2006.
6. M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171:286–310, 2007.
7. S. Colucci, T. D. Noia, A. Ragone, M. Ruta, U. Straccia, and E. Tinelli. *Semantic Web Information Management*, chapter 19 : Informative Top-k retrieval for advanced skill management, pages 449–476. R. De Virgilio et al., springer-verlag belin heidelberg edition, 2010.

8. S. Destercke, P. Buche, and V. Guillard. A flexible bipolar querying approach with imprecise data and guaranteed results. *Fuzzy sets and Systems*, 169:51–64, 2011.

9. P. M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-persons games. *Artificial Intelligence*, 77(2):321–357, 1995.

10. H. Prakken. An abstract framework for argumentation with structured arguments. Technical report, Department of Information and Computing Sciences. Utrecht University., 2009.

11. I. Rahwan and G. Simari. *Argumentation in Artificial Intelligence*. Springer, 2009.

12. N. Tamani and M. Croitoru. Fuzzy argumentation system for decision making. https://drive.google.com/file/d/0B0DPgJDRNwbLdE5wdzFQekJocXM/edit?usp=sharing. Technical report, INRIA LIRMM, 2013.

13. N. Tamani, M. Croitoru, and P. Buche. A viewpoint approach to structured argumentation. In M. Bramer and M. Petridis, editors, *The Thirty-third SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 265–271, 2013.

14. N. Tamani, M. Croitoru, and P. Buche. Conflicting viewpoint relational database querying: an argumentation approach. In L. Scerri and B. Huhns, editors, *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014),*, 2014. to appear.

15. Y. Wu. *Between argument and conclusion. Argument-based approaches to discussion. Inference and Uncertainty*. PhD thesis, Université du Luxembourg, 2012.

159

Decision Support for Agri-Food Chains: A Reverse Engineering Argumentation-Based Approach

Rallou Thomopoulos[a,b,*], Madalina Croitoru[b], Nouredine Tamani[b]

[a]IATE Joint Research Unit, UMR1208, CIRAD-INRA-Supagro-Univ. Montpellier II, 2 place P. Viala, F-34060 Montpellier cedex 1, France
[b]INRIA GraphIK, LIRMM, 161 rue Ada, F-34392 Montpellier cedex 5, France

Abstract

Evaluating food quality is a complex process since it relies on numerous criteria historically grouped into four main types: nutritional, sensorial, practical and hygienic qualities. They may be completed by other emerging preoccupations such as the environmental impact, economic phenomena, etc. However, all these aspects of quality and their various components are not always compatible and their simultaneous improvement is a problem that sometimes has no obvious solution, which corresponds to a real issue for decision making. This paper proposes a decision support method guided by the objectives defined for the end products of an agrifood chain. It is materialized by a backward chaining approach based on argumentation.

Keywords: decision support, knowledge representation, argumentation, reverse engineering, backward chaining, agrifood chain control, goal, viewpoint

*Corresponding author
  Email addresses: `rallou.thomopoulos@supagro.inra.fr` (Rallou Thomopoulos), `madalina.croitoru@lirmm.fr` (Madalina Croitoru), `nouredine.tamani@lirmm.fr` (Nouredine Tamani)

1. Introduction

In agrifood chains, the products traditionally go through the intermediate stages of processing, storage, transport, packaging and reach the consumer (the demand) from the producer (the supply). More recently, due to an increase in quality constraints, several parties are involved in production process, such as consumers, industrials, health and sanitary authorities, etc. expressing their requirements on the final product as different point of views which could be conflicting. The notion of reverse engineering control, in which the demand (and not the supply) sets the specifications of desired products and it is up to the supply to adapt and find its ways to respond, can be considered in this case.

In this article, we discuss two aspects of this problem. First, we accept the idea that specifications cannot be established and several complementary points of view - possibly contradictory - can be expressed (nutritional, environmental, taste, etc.). We then need to assess their compatibility (or incompatibility) and identify solutions satisfying a maximum set of viewpoints. To this end we propose a logical framework based on argumentation and introduce a method of decision making based on backward chaining for the bread industry.

Since a joint argumentation - decision support approach is highly relevant to the food sector (Thomopoulos et al., 2009), the contribution of the paper is twofold. First we present a real use case of an argumentation process in the agrifood domain. Second we introduce the notion of viewpoint / goal in this setting based on the notion of backwards chaining reasoning and show how to use those techniques in a concrete application.

2

The main alternative method to deal with the problem is the multicriteria decision approach. However multicriteria decision aims at evaluating several alternative options, whereas argumentation-based decision focuses on whether several options make sense together, which is a different perspective, addressed in this paper. Moreover, multicriteria decision is not connected to the backward chaining procedure as the argumentative approach is, by construction of the arguments, as will be explained in Section 5.2.

In Section 2, we introduce the real scenario considered in the application. In Section 3, we motivate our technical and modeling choices. In Section 4, the developed approach is introduced. It relies on an instantiation of a logic based argumentation framework based on a specific fragment of first order logic. In Section 5, we explain the technical results that ensure the soundness and completeness of our agronomy application method. In Section 6, some evaluation results are presented. Finally, Section 7 concludes the paper.

2. Scenario

The case of study considered in this paper relates to the debate around the change of ash content in flour used for common French bread. Various actors of the agronomy sector are concerned, in particular the Ministry for Health through its recommendations within the framework of the PNNS ("National Program for Nutrition and Health"), the millers, the bakers, the nutritionists and the consumers.

The PNNS recommends to privilege the whole-grain cereal products and in particular to pass to a common bread of T80 type, i.e made with flour containing an ash content (mineral matter rate) of 0.8%, instead of the type

3

T65 (0.65% of mineral matter) currently used. Increasing the ash content comes down to using a more complete flour, since mineral matter is concentrated in the peripheral layers of the wheat grain, as well as a good amount of components of nutritional interest (vitamins, fibers). However, the peripheral layers of the grain are also exposed to the phytosanitary products, which does not make them advisable from a health point of view, unless one uses organic flour.

Other arguments (and of various nature) are in favour or discredit whole-grain bread. From an organoleptic point of view for example, the bread loses out in its "being crusty". From a nutritional point of view, the argument according to which the fibers are beneficial for health is discussed, some fibers could irritate the digestive system. From an economic point of view, the bakers fear selling less bread, because whole-grain bread increases satiety – which is beneficial from a nutritional point of view, for the regulation of the appetite and the fight against food imbalances and pathologies. However whole-grain bread requires also less flour and more water for its production, thus reducing the cost. The millers also fear a decrease in the quality of the technical methods used in the flour production.

Beyond the polemic on the choice between two alternatives (T65 or T80), one can take the debate further by distinguishing the various points of view concerned, identifying the desirable target characteristics, estimating the means of reaching that point. The contribution of this paper is showing how using argumentation can help towards such practical goals.

4

3. Motivation

In this paper we will elicit the points of view and the desirable target characteristics by the means of interviews with agronomy experts. Once the target characteristics identified, finding the means of reaching them will be done automatically by a combination of reverse engineering and argumentation. The reverse engineering will be used in order to find the complete set of actions to take towards a given characteristic, for all characteristics. In certain cases the actions to take will be inconsistent. Argumentation will then be employed in order to identify actions that can be accepted together.

3.1. Reverse Engineering

While reverse engineering has been widely employed in other Computer Science domains such as multi agent systems or requirements engineering (e.g. Brunelière et al. (2014)), it is quite a novel methodology when applied in agronomy. In agrifood chains, the products traditionally go through the intermediate stages of processing, storage, transport, packaging and reach the consumer (the demand) from the producer (the supply). It is only recently, due to an increase in quality constraints, that the notion of reverse engineering control has emerged (Perrot et al., 2011). In this case the demand (and not the supply) sets the specifications of desired products and it is up to the supply to adapt and find its ways to respond. In what follows, starting from the desired target criteria for the final product, the methods allowing one to identify ways to achieve these criteria (by intervention on the various stages of the supply chain) are named "reverse engineering".

Reverse engineering is known to be challenging from a methodological

5

viewpoint. This is due to two main aspects. First, the difficulty of defining the specifications for the expected finished product. The desired quality criteria are multiple, questionable, and not necessarily compatible. The second difficulty lies in the fact that the impact of different steps of food processing and their order is not completely known. Some steps are more studied than others, several successive steps can have opposite effects (or unknown effects), the target criteria may be outside of the characteristics of products. Second, reconciling different viewpoints involved in the food sector still raises unaddressed questions. The problem does not simply consist in addressing a multi-criteria optimisation problem (Bouyssou et al., 2009): the domain experts would need to be able to justify why a certain decision (or set of possible decisions) is taken.

3.2. Argumentation

Argumentation is a reasoning model based on the construction and the evaluation of interacting arguments. It has been applied to nonmonotonic reasoning, decision making, or for modeling different types of dialogues including negotiation. Most of the models developed for these applications are grounded on the abstract argumentation framework proposed by Dung in Dung (1995). This framework consists of a set of arguments and a binary relation on that set, expressing conflicts among arguments. An argument gives a reason for believing a claim, for doing an action.

Argumentation theory in general (Dung, 1995; Besnard and Hunter, 2008; Rahwan and Simari, 2009) is actively pursued in the literature. Some approaches combine argumentation and multi criteria decision making (Amgoud and Prade, 2009).

6

Value based Argumentation Frameworks (Bench-Capon, 2003a) have been proposed, where the strength of an argument corresponds to the values it promotes. What we call viewpoint later on in this paper would then correspond to the notion of audience in such setting. Although intuitive, this approach is not adapted in the case of the considered application. Here a value can be "split" into several audiences: there could be contradictory goals even from the same viewpoint. The notion of viewpoint and goals introduced in this setting also remind those proposed by (Assaghir et al., 2011).

3.2.1. Logic-based Argumentation

In this paper we present a methodology combining reverse engineering and logical based argumentation for selecting the actions to take towards the agronomy application at hand. The logical instantiation language is a subset of first order logic denoted in this paper $\mathcal{SRC}$ equivalent to Datalog+- (Calì et al., 2010), Conceptual Graphs or Description Logics (more precisely the $\mathcal{EL}$ fragment (Baader et al., 2005) and DL-Lite families (Calvanese et al., 2007)). All above mentioned languages are logically equivalent in terms of representation or reasoning power. The reason why this application is using $\mathcal{SRC}$ is the graph based representation proper to $\mathcal{SRC}$ (and not to the other languages). This graph based representation (implemented in the Cogui tool (Chein and Mugnier, 2009; Chein et al., 2013)) makes the language suitable for interacting with non computing experts (Chein et al., 2013).

Here we use the instantiation of (Croitoru and Vesic, 2013) for defining what an argument and an attack are. While other approaches such as (García and Simari, 2004), (Besnard and Hunter, 2005), (Muller and Hunter, 2012) etc. address first order logic based argumentation, the work of (Croitoru and

7

Figure 1: The Cogui visual graph based interface

Vesic, 2013) uses the same $\mathcal{SRC}$ syntax and graph reasoning foundations. In Figure 1 the visual interface of Cogui is depicted: knowledge is represented as graph which is enriched dynamically by rule application. More on the visual appeal of Cogui for knowledge representation and reasoning can be found in (Chein et al., 2013).

4. Approach

As mentioned above, in this paper we use an instantiation of logic based argumentation based on a specific fragment of first order logic. This subset is equivalent to Datalog+- (Calì et al., 2010), Conceptual Graphs or Description Logics (the $\mathcal{EL}$ fragment (Baader et al., 2005) and the DL-Lite families (Calvanese et al., 2007)). The reason for which our application required this

8

specific logic fragment is related to the information capitalisation needs of the food sector. The long term aim is to enrich ontologies and data sources based on these ontologies and join the Open Data movement. This entails that the language used by the food applications needs to be compatible with the Semantic Web equivalent languages as mentioned before.

The choice of the $\mathcal{SRC}$ syntax and graph reasoning mechanism is justified by the visual appeal of this language for non computing experts.

In a nutshell our methodology is as follows. The set of goals, viewpoints as well as the knowledge associated with the goals / viewpoints is elicited either by the means of interviews with the domain experts or manually from different scientific papers. This step of the application is the most time consuming but the most important. If the knowledge elicited is not complete, sound or precise the outcome of the system is compromised. Then, based on the knowledge elicited from the knowledge experts and the goals of the experts, we enrich the knowledge bases using reverse engineering (implemented using backwards chaining algorithms). Putting together the enriched knowledge bases obtained by backwards chaining from the different goals will lead to inconsistencies. The argumentation process is used at this step and the extensions yield by the applications computed. Based on the extensions and the associated viewpoints we can use voting functions to determine the application choice of viewpoints.

4.1. Use Case Real Data

Expressing the target characteristics − or goals − according to various points of view consists of identifying the facets involved in the construction of product quality: points of view, topics of concern such as nutrition, envi-

9

ronment, technology, etc. In addition, such viewpoints have to be addressed according to their various components (fibers, minerals, vitamins, etc). Desirable directions need to be laid down, and in a first step we consider them independent one from another.

The considered sources of information include, from most formal to less formal: (1) peer reviewed scientific papers; (2) technical reports or information posted on websites; (3) conferences and scientific meetings around research projects; (4) expert knowledge obtained through interviews. The scientific articles we have analysed – with the supervision of experts in agrifood – include: (Bourre et al., 2008; Slavin and Green, 2007; Dubuisson-Quellier, 2006; Ginon et al., 2009; Layat, 2011). (Bourre et al., 2008) compares the different types of flour from a nutritional point of view. (Slavin and Green, 2007) explores the link between fiber and satiety. (Dubuisson-Quellier, 2006; Ginon et al., 2009) deal with consumer behaviour and willingness to pay. They focus on French baguette when information concerning the level of fibers is provided, and they base their results on statistical studies of consumer panels. (Layat, 2011) provides a summary of the nutritional aspects of consumption of bread and the link with technological aspects.

We also reviewed technical reports available on official websites on health policy: the public PNNS (National Program for Nutrition and Health, www.mangerbouger.fr/pnns) (PNNS (documents statutaires), 2010), the European project Healthgrain (looking at improving nutrition and health through grains) (Dean et al., 2007; HEALTHGRAIN, 2009), as well as projects and symposia on sanitary measures regarding the nutritional, technological and organoleptic properties of breads (DINABIO, 2008; CADINNO, 2008; AQUA-

10

169

NUP, 2009; FCN, 2009). Finally, several interviews were conducted to collect domain expert knowledge, in particular for technology specialists in our laboratory.

A summary of the results obtained in the baking industry is synthesised in Figure 2 regarding nutritional and organoleptic aspects. Figure 2(a) shows the main identified goals to reach for a nutritionally optimised bread (for instance, containing a high level of soluble fibers, vitamins and minerals, low salt, etc.), whereas Figure 2(b) sums up the main goals to achieve for an enjoyable bread regarding sensorial concerns (for example, crusty, etc.).

5. Technical Soundness

In this section we explain the technical results that ensure the soundness and completeness of our agronomy application method. The section is composed of three parts. A first subsection explains the logical subset of first order logic language employed in the paper. The second subsection shows how to construct arguments and attacks in order to obtain extensions when a knowledge base expressed under this language is inconsistent. Last, the third section shows how we used reverse engineering to complete the knowledge base with all possible actions and how argumentation can be used in order to select consistent subsets of knowledge which support given actions.

5.1. The Logical Language

In the following, we give the general setting knowledge representation language used throughout the paper.

A knowledge base is a 3-tuple $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ composed of three finite sets of formulae: a set $\mathcal{F}$ of facts, a set $\mathcal{R}$ of rules and a set $\mathcal{N}$ of constraints.

11

The « ideal » healthy bread

- Conserved minerals, vitamins, soluble fibers & nitrogenous matter
- Digestibility: limited cellulose fibers
- Contaminant-free
- Diversified trace element intake
- Bioavailable micronutriens
- Satiety and low glycemic index
- Low-salt

(a)

Some factors of a « good » bread

- Crusty
- Grilled or smoked flour flavor
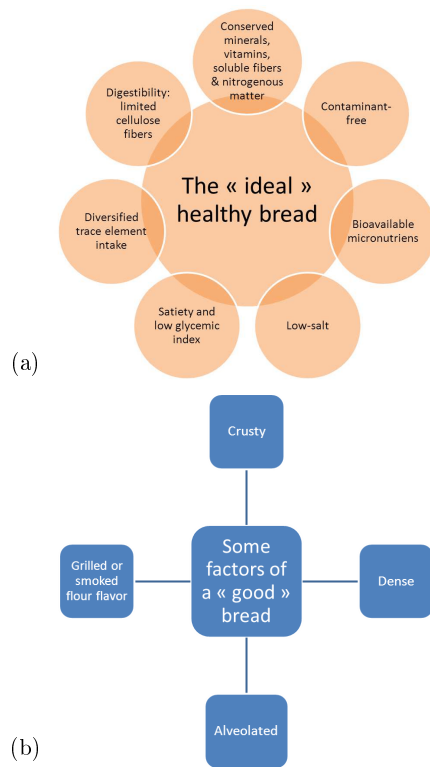- Dense
- Alveolated

(b)

Figure 2: Nutritional (a) and organoleptic (b) goals

12

Let us formally define what we accept as $\mathcal{F}$, $\mathcal{R}$ and $\mathcal{N}$.

Facts Syntax. Let $\mathbf{C}$ be a set of constants and $\mathbf{P} = P_1 \cup P_2 \ldots \cup P_n$ a set of predicates of the corresponding arity $i = 1, \ldots, n$. Let $\mathbf{V}$ be a countably infinite set of *variables*. We define the set of *terms* by $\mathbf{T} = \mathbf{V} \cup \mathbf{C}$. As usual, given $i \in \{1 \ldots n\}$, $p \in P_i$ and $t_1, \ldots, t_i \in \mathbf{T}$ we call $p(t_1, \ldots, t_i)$ an *atom*. A *fact* is the existential closure of an atom or an existential closure of a conjunction of atoms. (Note that there is no negation or disjunction in the facts and that we consider a generalised notion of facts that can contain several atoms.)

- *Bread, Cereal, LowSalt, ContaminantFree* are examples of unary predicates (arity 1) and *IsIngredientOf* is a binary predicate (arity 2).

- *Wheat, oats, rye, barley* are constant examples.

- *Cereal (wheat)* is an atom.

- $\exists$ *x (Bread(x) $\wedge$ IsIngredientOf(wheat, x))* is a fact.

Due to lack of space we do not show the full semantic definitions of facts (or rules and constraints in the following section). For a complete semantic depiction of this language please check (Chein and Mugnier, 2009; Chein et al., 2013; Croitoru and Vesic, 2013). It is well known that $F' \models F$ (read the fact F' entails the fact F) if and only if there is a homomorphism from F to F' (Chein and Mugnier, 2009).

Rules. A rule R is a formula of the form

$$\forall x_1, \ldots, \forall x_n \; \forall y_1, \ldots, \forall y_m \; (H(x_1, \ldots, x_n, y_1, \ldots, y_m) \rightarrow$$
$$\exists z_1, \ldots \exists z_k \; C(y_1, \ldots, y_m, z_1, \ldots z_k))$$

where H, the hypothesis, and C, the conclusion, are atoms or conjunctions of atoms, $n, m, k \in \{0, 1, \ldots\}$, $x_1, \ldots, x_n$ are the variables appearing in H,

13

$y_1, \ldots, y_m$ are the variables appearing in both H and C and $z_1, \ldots, z_k$ the new variables introduced in the conclusion. An example of a rule is the following:

$\forall \ x \ (Bread(x) \wedge PesticideFree(x) \wedge MycotoxinFree(x)$

$$\rightarrow ContaminantFree(x)).$$

In the following we will consider rules without new existential variables in the conclusion.

Reasoning consists of applying rules on the set $\mathcal{F}$ and thus inferring new knowledge. A rule $R = (H, C)$ is *applicable* to set $\mathcal{F}$ if and only if there exists $\mathcal{F}' \subseteq \mathcal{F}$ such that there is a homomorphism σ from the hypothesis of $\mathcal{R}$ to the conjunction of elements of $\mathcal{F}'$. A rule $R = (H, C)$ is inversely applicable to a fact F if there is a homomorphism π from C to F. In this case, the inverse application of R to F according to π produces a new fact F' such that $R(F') = F$. We then say that the new fact is an immediate inverse derivation of F by R, abusively denoted $R^{-1}(F)$.

Note that this technique is commonly used, for example, for backward chaining query answering (Baget and Salvat, 2006; Konig et al., 2012) where a query is rewritten according to the rules. The same mechanism is also discussed by abductive reasoning algorithms (Klarman et al., 2011) where minimal sets of facts (in the set inclusion sense) are added to the knowledge base in order to be able to deduct a query.

Let $F = Bread(bleuette) \wedge PesticideFree(bleuette) \wedge MycotoxinFree(bleuette)$ and R the rule $\forall \ x \ (Bread(x) \wedge PesticideFree(x) \wedge MycotoxinFree(x) \rightarrow ContaminantFree(x))$.

R is applicable to F and produces by derivation the following fact: *Bread (bleuette) $\wedge$ PesticideFree(bleuette) $\wedge$ MycotoxinFree(bleuette) $\wedge$ Contaminant-*

14

Free(bleuette).

Let $F = Bread(bleuette) \wedge ContaminantFree(bleuette)$ and R the rule $\forall x$ *(Bread(x) $\wedge$ PesticideFree(x) $\wedge$ MycotoxinFree(x) $\rightarrow$ ContaminantFree(x)).*

R inversely applicable to F and produces by inverse derivation the fact: $F' = Bread(bleuette) \wedge PesticideFree(bleuette) \wedge MycotoxinFree(bleuette).$

Let F be a subset of $\mathcal{F}$ and let $\mathcal{R}$ be a set of rules. A set F_n is called an $\mathcal{R}$-*derivation* of F if there is a sequence of sets (called a *derivation sequence*) $(F_0, F_1, \ldots, F_n)$ such that $F_0 \subseteq F$, F_0 is $\mathcal{R}$-consistent, for every $i \in \{1, \ldots, n-1\}$, it holds that F_i is an immediate derivation of F_{i-1}.

Given a set $\{F_0, \ldots, F_k\} \subseteq \mathcal{F}$ and a set of rules $\mathcal{R}$, the closure of $\{F_0, \ldots, F_k\}$ w.r.t. $\mathcal{R}$, denoted $\mathtt{Cl}_\mathcal{R}(\{F_0, \ldots, F_k\})$, is defined as the smallest set (with respect to $\subseteq$) which contains $\{F_0, \ldots, F_k\}$, and is closed for $\mathcal{R}$-derivation (that is, for every $\mathcal{R}$-derivation F_n of $\{F_0, \ldots, F_k\}$, we have $F_n \subseteq \mathtt{Cl}_\mathcal{R}(\{F_0, \ldots, F_k\})$). Finally, we say that a set $\mathcal{F}$ and a set of rules $\mathcal{R}$ *entail* a fact G (and we write $\mathcal{F}, \mathcal{R} \models G$) iff the closure of the facts by all the rules entails F (i.e. if $\mathtt{Cl}_\mathcal{R}(\mathcal{F}) \models G$).

Constraints. A constraint is a formula $\forall x_1 \ldots \forall x_n$ $(H(x_1, \ldots, x_n) \rightarrow \bot)$, where H is an atom or a conjunction of atoms and $n \in \{0, 1, 2, \ldots\}$. Equivalently, a constraint can be written as $\neg(\exists x_1, \ldots, \exists x_n H(x_1, \ldots x_n))$. As an example of a constraint, consider $N = \neg(\exists x (Growth(x) \wedge Decrease(x)))$.

Given a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, a set $\{F_1, \ldots, F_k\} \subseteq \mathcal{F}$ is said to be *inconsistent* if and only if there exists a constraint $N \in \mathcal{N}$ such that $\{F_1, \ldots, F_k\} \models H_N$, where H_N denotes the existential closure of the hypothesis of N. A set is consistent if and only if it is not inconsistent. A set $\{F_1, \ldots, F_k\} \subseteq \mathcal{F}$ is $\mathcal{R}$-*inconsistent* if and only if there exists a constraint

$N \in \mathcal{N}$ such that $\mathtt{Cl}_{\mathcal{R}}(\{F_1, \ldots, F_k\}) \models H_N$, where H_N denotes the existential closure of the hypothesis of N.

Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ where:

- $\mathcal{F}$ contains the following facts:

  - $F_1 = Bread(bleuette) \wedge ContaminantFree(bleuette)$

  - $F_2 = \exists\ e\ ExtractionRate(e, bleuette)$

  - $F_3 = \exists\ f\ (FiberContent(f, bleuette) \wedge High(f))$

- $\mathcal{R}$ consists of the following rules:

  - $R_1 = \forall\ x, y\ (Bread(x) \wedge ExtractionRate(y, x) \wedge PesticideFree(x)$
    $$\rightarrow Decrease(y))$$

  - $R_2 = \forall\ x, y, z\ (Bread(x) \wedge ExtractionRate(y, x) \wedge FiberContent(z, x) \wedge$
    $$High(z) \rightarrow Growth(y))$$

  - $R_3 = \forall\ x\ (Bread(x) \wedge ContaminantFree(x)$
    $$\rightarrow PesticideFree(x) \wedge MycotoxinFree(x))$$

- $\mathcal{N}$ contains the following negative constraint:

  - $N = \neg(\exists\ x\ (Growth(x) \wedge Decrease(x)))$

$\mathcal{K}$ is inconsistent since $(\mathcal{F}, \mathcal{R}) \models N$. Indeed, F_1 and R_3 allow to deduce *PesticideFree(bleuette)*. Combined to F_2 and R_1 we obtain *Decrease(e)*. F_3 and R_2 deduce *Growth(e)*, violating the negative constraint N.

Given a knowledge base, one can ask a conjunctive query in order to know whether something holds or not. Without loss of generality we consider

16

boolean conjunctive queries (which are facts). As an example of a query, take $\exists x_1 cat(x_1)$. The answer to query α is positive if and only if $\mathcal{F}, \mathcal{R} \models \alpha$.

Answering Q, traditionally, has two different algorithmic approaches: either forward chaining or backwards chaining. The two approaches come to either (1) finding an answer of Q in the $\mathcal{R}$-derivations of the facts in the knowledge base or (2) computing the inverse $\mathcal{R}$-derivations of the query and finding if there is a match in the facts. We will focus on the latter approach in the following.

5.2. Arguments and Attacks

This section shows that it is possible to define an instantiation of Dung's abstract argumentation theory (Dung, 1995) that can be used to reason with an inconsistent ontological KB.

We first define the notion of an argument. For a set of formulae $\mathcal{G} = \{G_1, \ldots, G_n\}$, notation $\bigwedge G$ is used as an abbreviation for $G_1 \wedge \ldots \wedge G_n$.

Definition 1. *Given a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, an argument a is a tuple $a = (F_0, F_1, \ldots, F_n)$ where:*

- *$(F_0, \ldots, F_{n-1})$ is a derivation sequence with respect to $\mathcal{K}$*

- *F_n is an atom, a conjunction of atoms, the existential closure of an atom or the existential closure of a conjunction of atoms such that $F_{n-1} \models F_n$.*

This definition, following the definition of (Croitoru and Vesic, 2013) is a straightforward way to define an argument, since an *argument* corresponds to a *derivation*.

17

To simplify the notation, from now on, we suppose that we are given a fixed knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ and do not explicitly mention $\mathcal{F}$, $\mathcal{R}$ nor $\mathcal{N}$ if not necessary. Let $a = (F_0, ..., F_n)$ be an argument. Then, we denote $\mathtt{Supp}(a) = F_0$ and $\mathtt{Conc}(a) = F_n$.

Arguments may attack each other, which is captured by a binary attack relation $\mathtt{Att} \subseteq \mathtt{Arg}(\mathcal{F}) \times \mathtt{Arg}(\mathcal{F})$.

Definition 2. *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and let a and b be two arguments. The argument a attacks argument b, denoted $(a, b) \in Att$, if and only if there exists $\varphi \in \mathtt{Supp}(b)$ such that the set $\{Conc(a), \varphi\}$ is $\mathcal{R}$-inconsistent.*

This attack relation is not symmetric. To see why, consider the following example. Let $\mathcal{F} = \{p(m), q(m), r(m)\}$, $\mathcal{R} = \emptyset$, $\mathcal{N} = \{\forall x_1 (p(x_1) \wedge q(x_1) \wedge r(x_1) \rightarrow \bot)\}$. Let $a = (\{p(m), q(m)\}, p(m) \wedge q(m))$, $b = (\{r(m)\}, r(m))$. We have $(a, b) \in \mathtt{Att}$ and $(b, a) \notin \mathtt{Att}$. This will ensure that the naive extension is different, at least in theory, from the preferred, stable, etc. semantics. However, in our application they all entail the same information as shown later on.

Definition 3. *Given a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, the corresponding argumentation framework $\mathcal{AF}_K$ is a pair $(\mathcal{A} = \mathtt{Arg}(\mathcal{F}), \mathtt{Att})$ where $\mathtt{Arg}(\mathcal{F})$ is the set of all arguments that can be constructed from $\mathcal{F}$ and $\mathtt{Att}$ is the corresponding attack relation as specified in Definition 2.*

Let $\mathcal{E} \subseteq \mathcal{A}$ and $a \in \mathcal{A}$. We say that $\mathcal{E}$ is conflict free *iff there exists no arguments $a, b \in \mathcal{E}$ such that $(a, b) \in Att$. $\mathcal{E}$ defends a iff for every argument $b \in \mathcal{A}$, if we have $(b, a) \in Att$ then there exists $c \in \mathcal{E}$ such that $(c, b) \in Att$.*

18

$\mathcal{E}$ is admissible *iff it is conflict free and defends all its arguments. $\mathcal{E}$ is a* complete extension *iff $\mathcal{E}$ is an admissible set which contains all the arguments it defends. $\mathcal{E}$ is a* preferred extension *iff it is maximal (with respect to set inclusion) admissible set. $\mathcal{E}$ is a* stable extension *iff it is conflict-free and for all $a \in \mathcal{A} \setminus \mathcal{E}$, there exists an argument $b \in \mathcal{E}$ such that $(b, a) \in Att$.*

$\mathcal{E}$ is a grounded extension *iff $\mathcal{E}$ is a minimal (for set inclusion) complete extension.*

For an argumentation framework $AS = (\mathcal{A}, Att)$ we denote by $Ext_x(AS)$ (or by $Ext_x(\mathcal{A}, Att)$) the set of its extensions with respect to semantics x. We use the abbreviations c, p, s, and g for respectively complete, preferred, stable and grounded semantics.

An argument is sceptically accepted if it is in all extensions, credulously accepted if it is in at least one extension and rejected if it is not in any extension.

Based on this definition of arguments and attacks in (Croitoru and Vesic, 2013) was also shown that the rationality postulates of (Caminada and Amgoud, 2007) are respected. This instantiation respects the direct, indirect consistency as well as the closure.

5.3. Formalising the use case

In this subsection we formalise the notions presented in section 4.

Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a consistent knowledge base. This is the knowledge base that all actors share and agree upon. In this paper we assume that the rules and negative constraints are common to everybody.

The goals of the different actors can be seen as a set of existentially closed conjuncts. We denote them by $G_1, G_2, ..., G_n$.

19

Let G_i be a goal and $\mathcal{K}$ the knowledge base. $\mathcal{K}$ is consistent and $\mathcal{K}$ does not entail G_i. We compute the inverse $\mathcal{R}$-derivations of G_i (where $\mathcal{R}$ is the set of rules of the knowledge base). We add all of the $\mathcal{R}^{-1}(G_i)$ to the facts. We thus obtain a new knowledge base $\mathcal{K}_i$ which differs from $\mathcal{K}$ solely by its facts set (which now also includes $\mathcal{R}^{-1}(G_i)$): $\mathcal{K} = (\mathcal{F} \cup \mathcal{R}^{-1}(G_i), \mathcal{R}, \mathcal{N})$. We also impose that $\mathcal{K}_i$ is consistent.

Given $\mathcal{G} = \{G_1, G_2, ..., G_n\}$, the goals correspond to a set of viewpoints $\mathcal{V}$ (there exists a function $\kappa : \mathcal{G} \to 2^{\mathcal{V}}$). This function can assign a goal to one or more viewpoints and each viewpoint can be associated with one or more goals. Given a goal G_i, the (set of) viewpoint(s) associated with this goal is denoted by $\kappa(G_i)$. Similarly, given a viewpoint v_i, the set of goals associated with it is denoted by $\kappa^{-1}(v_i)$.

Example 1. *Let the set of viewpoints $\mathcal{V} = \{nutrition, sanitary, organoleptic\}$ and $\mathcal{G}$ consisting of the following goals: $G_1 = \exists\, x\ (Bread(x) \wedge LowSalt(x))$, $G_2 = \exists\, x\ (Bread(x) \wedge ContaminantFree(x))$, $G_3 = \exists\, x\ (Bread(x) \wedge Crusty(x))$, $G_4 = \exists\, x\ (Bread(x) \wedge TraceElementRich(x))$.*

We have $\kappa(G_1) = \kappa(G_4) = nutrition$, $\kappa(G_2) = sanitary$ and $\kappa(G_3) = organoleptic$. Conversely $\kappa^{-1}(nutrition) = \{G_1, G_4\}$, $\kappa^{-1}(sanitary) = \{G_2\}$ and $\kappa^{-1}(organoleptic) = \{G_3\}$.

The rules will correspond to the set of sufficient conditions needed for the goal G_i. In the context of our practical application this is illustrated in Figure 3 (with respect to nutrition goals).

Example 2. *To reach the goal $G_1 = \exists\, x\ (Bread(x) \wedge LowSalt(x))$, the*
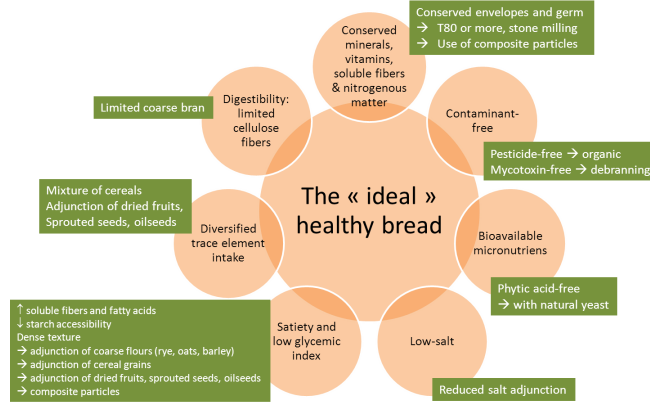
20

Figure 3: Ways to reach nutritional goals

knowledge base $\mathcal{K}$ contains the following rule: $\forall$ x,y (Bread(x) $\wedge$ SaltAdjunction(y,x) $\wedge$ Decrease(y) $\rightarrow$ LowSalt(x))

Let us now consider the set of goals $\mathcal{G} = \{G_1, G_2, ..., G_n\}$ and the initial knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$. As described above we compute the n knowledge bases, corresponding to each goal: $\mathcal{K}_i = (\mathcal{F} \cup \mathcal{R}^{-1}(G_i), \mathcal{R}, \mathcal{N})$ for each $i = 1, ..., n$. We consider the union of all these knowledge bases:

$$\mathcal{K}_{agg} = (\mathcal{F} \bigcup_{i=1,...,n} \mathcal{R}^{-1}(G_i), \mathcal{R}, \mathcal{N})$$

Example 3. Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ where :

- $\mathcal{F} = \{F_1\} = \{CurrentExtractionRate(T65)\}$

- $\mathcal{R}$ contains the following rules:

21

- $R_1 = \forall\ x,y\ (Bread(x) \wedge ExtractionRate(y,x) \wedge Decrease(y)$

$$\rightarrow Digestible(x))$$

- $R_2 = \forall\ x,z\ (Bread(x) \wedge SaltAdjunction(z,x) \wedge Decrease(z)$

$$\rightarrow LowSalt(x))$$

- $R_3 = \forall\ x,y\ (Bread(x) \wedge ExtractionRate(y,x) \wedge Growth(y)$

$$\rightarrow TraceElementRich(x))$$

- $R_4 = \forall\ x,y\ (Bread(x) \wedge ExtractionRate(y,x) \wedge Decrease(y)$

$$\rightarrow PesticideFree(x))$$

- $\mathcal{N}$ contains the following negative constraint:

  - $N = \neg(\exists\ x\ (Growth(x) \wedge Decrease(x)))$

Let the goal set $\mathcal{G}$ as follows:

- $G_1 = \exists\ p\ (Bread(p) \wedge Digestible(p)),$ \qquad where $\kappa(G_1) = nutrition$
- $G_2 = \exists\ p\ (Bread(p) \wedge LowSalt(p)),$ \qquad where $\kappa(G_2) = nutrition$
- $G_3 = \exists\ p\ (Bread(p) \wedge TraceElementRich(p)),$ where $\kappa(G_3) = nutrition$
- $G_4 = \exists\ p\ (Bread(p) \wedge PesticideFree(p)),$ \qquad where $\kappa(G_4) = sanitary.$

Then:

- $\mathcal{K}_1 = (\mathcal{F}_1, \mathcal{R}, \mathcal{N})$ where $\mathcal{F}_1 = \mathcal{F} \cup \mathcal{R}^{-1}(G_1)$ contains the following facts:
  - $F_1 = CurrentExtractionRate(T65)$
  - $F_2 = Bread(p) \wedge ExtractionRate(\tau,p) \wedge Decrease(\tau)$
- $\mathcal{K}_2 = (\mathcal{F}_2, \mathcal{R}, \mathcal{N})$ where $\mathcal{F}_2 = \mathcal{F} \cup \mathcal{R}^{-1}(G_2)$ contains the following facts:
  - $F_1 = CurrentExtractionRate(T65)$
  - $F_3 = Bread(p) \wedge SaltAdjunction(s,p) \wedge Decrease(s)$
- $\mathcal{K}_3 = (\mathcal{F}_3, \mathcal{R}, \mathcal{N})$ where $\mathcal{F}_3 = \mathcal{F} \cup \mathcal{R}^{-1}(G_3)$ contains the following facts:

22

– $F_1 = CurrentExtractionRate(T65)$

– $F_4 = Bread(p) \wedge ExtractionRate(\tau,p) \wedge Growth(\tau)$

• $\mathcal{K}_4 = (\mathcal{F}_4, \mathcal{R}, \mathcal{N})$ where $\mathcal{F}_4 = \mathcal{F} \cup \mathcal{R}^{-1}(G_4)$ contains the following facts:

– $F_1 = CurrentExtractionRate(T65)$

– $F_2 = Bread(p) \wedge ExtractionRate(\tau,p) \wedge Decrease(\tau)$

Finally $\mathcal{K}_{agg} = (\mathcal{F} \bigcup_{i=1,...,n} \mathcal{R}^{-1}(G_i), \mathcal{R}, \mathcal{N})$ where

$\mathcal{F} \bigcup_{i=1,...,n} \mathcal{R}^{-1}(G_i) = \{F_1, F_2, F_3, F_4\}$.

As observed in the previous example, it may happen that $\mathcal{K}_{agg}$ is inconsistent (and it does so even for goals belonging to the same viewpoint). We then use argumentation, which, by the means of extensions will isolate subsets of facts we can accept together (called extensions). Furthermore, the extensions will allow us to see which are the viewpoints associated to each maximal consistent subset of knowledge (by the means of the function κ). A choice procedure then has to be used (see example below).

The argument framework we can construct from the above knowledge base is $(\mathcal{A}, Att)$ where $\mathcal{A}$ contains the following:

• $a = (\{F_2\}, F_2, R_1(F_2))$ where $R_1(F_2) = Bread(p) \wedge ExtractionRate(\tau,p) \wedge Decrease(\tau) \wedge Digestible(p)$.

• $b = (\{F_4\}, F_4, R_3(F_4))$ where $R_3(F_4) = Bread(p) \wedge ExtractionRate(\tau,p) \wedge Growth(\tau) \wedge TraceElementRich(p)$.

• $c = (\{F_2\}, F_2, R_4(F_2))$ where $R_4(F_2) = Bread(p) \wedge ExtractionRate(\tau,p) \wedge Decrease(\tau) \wedge PesticideFree(p)$.

• $d = (\{F_3\}, F_3, R_2(F_3))$ where $R_2(F_3) = Bread(p) \wedge SaltAdjunction(s,p) \wedge Decrease(s) \wedge LowSalt(p)$ and

$Att = \{(a,b), (b,a), (b,c), (c,b)\}$.

23

In this argumentation system defined we now obtain:

$\bullet\, Ext_{stable}(\mathcal{A}, Att) = Ext_{semi-stable}(\mathcal{A}, Att) = Ext_{prefered}(\mathcal{A}, Att) = \{\{a, c, d\}, \{b, d\}\}$.

Starting from the extensions $Ext_x(\mathcal{A}, Att)$, the proposed decision support system functions as follows: for every extension $\varepsilon \in Ext_x(\mathcal{A}, Att)$:

- Consider the facts occurring in the arguments of ε ;

- Identify the knowledge bases $\mathcal{K}_i$ where these facts occur;

- Obtain the goals G_i which are satisfied by the extension;

- Using the κ function to obtain the viewpoints corresponding to these goals;

- Show domain experts the set of goals, and compatible viewpoints corresponding to the given extension.

This method allows us to obtain a set of options equal to the cardinality of $Ext_x(\mathcal{A}, Att)$. For taking a final decision several possibilities can be considered and presented to the experts:

- Maximise the number of goals satisfied;

- Maximise the number of viewpoints satisfied;

- Use preference relations of experts on goals and / or viewpoints.

In the previous example (please recall that the goals G_1 and G_2 are associated with the nutritional viewpoint while G_4 is associated with the sanitary viewpoint) we have:

24

183

- The first extension $\{a, c, d\}$ is based on the facts F_2 and F_3 obtained from $\mathcal{K}_1$, $\mathcal{K}_2$ and $\mathcal{K}_4$ that satisfy the goals G_1, G_2 and G_4.

- The second extension $\{b, d\}$ is based on F_3 and F_4 obtained from $\mathcal{K}_2$ and $\mathcal{K}_3$ satisfying G_2 and G_3 both associated with the nutritional viewpoint.

One first possibility (corresponding to the extension $\{a, c, d\}$) consists of accomplishing F_2 and F_3 and allows to satisfy the biggest number of goals and viewpoints.

The second possibility (corresponding to the extension $\{b, d\}$) consists of accomplishing F_3 and F_4. It would satisfy two goals and one viewpoint. It could be considered though if the goal G_3 (not satisfied by the first option) is preferred to the others.

6. Evaluation

The evaluation of the implemented system was done via a series of interviews with domain experts. The above knowledge and reasoning procedures were implemented using the Cogui knowledge representation tool (Chein et al., 2013), with an extension of 2000 lines of supplemental code. Three experts have validated our approach: two researchers in food science and cereal technologies of the French national institute of agronomic research, specialists respectively of the grain-to-flour transformation process and of the breadmaking process, and one industrial expert - the Director of the French National Institute of Bread and Pastry.

The first meeting dealt with the delimitation of the project objectives and addressed fundamental questions such as: Is it possible to uniquely define a

25

"good" bread? Which scenarii of "good bread" should be considered? How could they be defined from a nutritional, sanitary, sensorial and economic point of view? Which are the main known ways to achieve them?

Then a series of individual interviews constituted the elicitation phase. Each expert gave more arguments which were complementing one each other. In parallel, the writing of specifications for the demonstrator and the definition of the knowledge base structure were conducted.

In the following plenary meeting the real potential of the approach was shown. The experts were formulating goals and viewpoints they were interested in and the Cogui system together with the argumentation extension was yielding the associated possible propositions. Figure 6 shows a screenshot of the demonstrator answers for a two-goal query: a nutritional goal (high fiber content) and an organoleptic goal (crusty bread). Two sets of compatible actions are proposed, some choices (such as increasing or decreasing the extraction rate) being incompatible for both goals, and thus separated in the two alternative sets.

Four scenarii were more specifically evaluated. These scenarii concern four kinds of consumers: obeses (fiber preference), people with iron deficiency (micronutrient preference), people with cardivascular disease (decreased salt preference) and vegetarians (limited phytic acid), which produces different sets of goals. For each scenario, the system proposes several outputed recommendations. The audience for decreasing salt tips the balance in favour of a recommendation for the T80 bread, while the audience for decreasing phytic acid pushes to specify recommendations towards a natural sourdough bread or a conservative T65 bread. Other audiences are in favor of a status
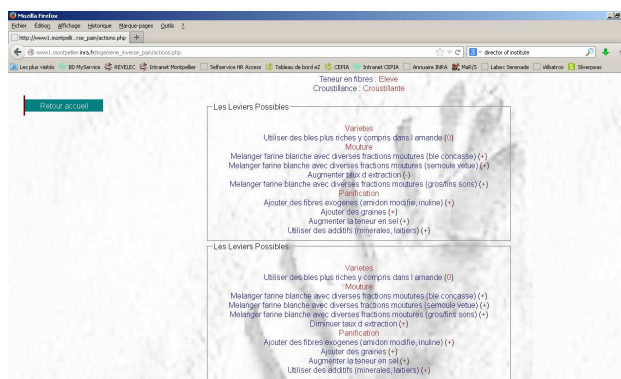
26

Figure 4: Demonstrator screenshot showing two sets of possible actions

quo. The results were considered as explanable by experts, but not obvious, since many considerations had to be taken into account.

Two interests of the approach were more particularly highlighted. They concern cognitive considerations. Firstly, experts were conscious that the elicitation procedure was done according to their thought processes, that is, in a forward way which is more natural and intuitive. The system was thus able to restitute the knowledge in a different manner than the experts usually do. Secondly, from a problem that could initially seem simple, the experts realized that it covered a huge complexity that a human mind could hardly address alone. The tool is currently available to them under restricted access.

The knowledge modeling task can be a very time-consuming step. As presented in Section 4.1, several sources of information were used, from peer reviewed scientific papers and technical reports, to conference meetings and expert interviews. On the one hand, expert interviews appeared to be the

27

least expensive ones in terms of time. A one-day period allows both eliciting knowledge through an interview and formalizing it in the software system – which constitutes the longest part of the work. However, this relatively short time hides a strong prerequisite: having already a clear view of the case study, a synopsis of the questions to ask the expert and an implemented knowledge model. On the other hand, websites, technical reports and scientific articles are more costly to analyze. For instance, the critical reading a scientific paper of the domain may require a one-day period on its own, for a discerning reader. However they allow one to to grasp the ins and outs of the question.

During the evaluation step, the experts raised the question of the importance attached to the different pieces of knowledge modeled in the system. Moreover, in some cases experts may hesitate on the relevance of some facts or rules. A possibility would thus be to adopt a preference-based argumentation system, as proposed in several works such as (Amgoud and Cayrol, 2002; Bench-Capon, 2003b; Kaci and van der Torre, 2008; Amgoud et al., 2000; Bourguet et al., 2013a), able to take into account different levels of importance among arguments.

7. Conclusion

Even if argumentation based decision making methods applied to the food industry were also proposed by (Bourguet, 2010; Bourguet et al., 2013b), this paper addresses a key point in the context of current techniques used by the food sector and namely addressing reverse engineering. Also, in this approach, an argument is used here as a method computing compatible objectives in the sector. This case study represents an original application and

28

an introspective approach in the agronomy field by providing an argumentation based decision-support system for the various food sectors. It requires nevertheless the very expensive task of knowledge modeling. Such task, in its current state cannot be automated. It strongly depends on the quality of expert opinion and elicitation (exhaustiveness, certainty, etc). The current trend for decision-making tools includes more and more methods of argumentation as means of including experts in the task of modeling and the decision-making processes. Another element to take into account, not discussed in this paper, is the difficulty of technologically (from an agronomy viewpoint) putting in place the facts of each option. Modeling this aspect in the formalism is still to be studied.

Amgoud, L., Cayrol, C., 2002. A reasoning model based on the production of acceptable arguments. Annals of Mathematics and Artificial Intelligence 34, 197–216.

Amgoud, L., Parsons, S., Perrussel, L., 2000. An argumentation framework based on contextual preferences. In: Proceedings of the International Conference on Formal and Applied and Practical Reasoning. pp. 59–67.

Amgoud, L., Prade, H., 2009. Using arguments for making and explaining decisions. Artificial Intelligence 173 (3-4), 413–436.

AQUANUP, 2009. `http://www.inra.fr/inra_cepia/vous_recherchez/des_projets/france/aquanup`.

Assaghir, Z., Napoli, A., Kaytoue, M., Dubois, D., Prade, H., 2011. Numeri-

29

cal information fusion: Lattice of answers with supporting arguments. In: ICTAI. pp. 621–628.

Baader, F., Brandt, S., Lutz, C., 2005. Pushing the el envelope. In: Proc. of IJCAI 2005.

Baget, J.-F., Salvat, E., 2006. Rules dependencies in backward chaining of conceptual graphs rules. In: Conceptual Structures: Inspiration and Application, 14th International Conference on Conceptual Structures. Vol. 4068 of LNCS. Springer, pp. 102–116.

Bench-Capon, T. J., 2003a. Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation 13 (3), 429–448.

Bench-Capon, T. J. M., 2003b. Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation 13 (3), 429–448.

Besnard, P., Hunter, A., 2005. Practical first-order argumentation. In: Proc. of AAAI. pp. 590–595.

Besnard, P., Hunter, A., 2008. Elements of Argumentation. The MIT Press.

Bourguet, J.-R., 2010. Contribution aux methodes d'argumentation pour la prise de decision. application a l'arbitrage au sein de la filiere cerealiere. Thèse de doctorat, Université Montpellier II, Montpellier, France.

Bourguet, J.-R., Thomopoulos, R., Mugnier, M.-L., Abécassis, J., September 2013a. An artificial intelligence-based approach to deal with argumentation

30

applied to food quality in a public health policy. Expert Systems with Applications 40 (11), 4539–4546.

Bourguet, J.-R., Thomopoulos, R., Mugnier, M.-L., Abécassis, J., 2013b. An artificial intelligence-based approach to deal with argumentation applied to food quality in a public health policy. Accepted for publication in: Expert Systems With Applications.

Bourre, J.-M., Bégat, A., Leroux, M.-C., Mousques-Cami, V., Pérandel, N., Souply, F., 2008. Valeur nutritionnelle (macro et micro-nutriments) de farines et pains français. Médecine et Nutrition 44 (2), 49–76.

Bouyssou, D., Dubois, D., Pirlot, M., Prade, H., 2009. Decision-making process – Concepts and Methods. Wiley.

Brunelière, H., Cabot, J., Dupé, G., Madiot, F., 2014. Modisco: A model driven reverse engineering framework. Information and Software Technology 56 (8), 1012 – 1032.

CADINNO, 2008. Information, choix, consommateurs responsables : des leviers pour un développement durable ? `http://www.melissa.ens-cachan.fr/IMG/pdf/Colloque_CadInno_FR.pdf`.

Calì, A., Gottlob, G., Lukasiewicz, T., Marnette, B., Pieris, A., 2010. Datalog+/-: A family of logical knowledge representation and query languages for new applications. In: LICS. pp. 228–242.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R., 2007. Tractable reasoning and efficient query answering in description logics: The dl-lite family. J. Autom. Reasoning 39 (3), 385–429.

31

Caminada, M., Amgoud, L., 2007. On the evaluation of argumentation formalisms. Artificial Intelligence 171 (5), 286–310.

Chein, M., Mugnier, M.-L., 2009. Graph-based Knowledge Representation and Reasoning—Computational Foundations of Conceptual Graphs. Advanced Information and Knowledge Processing. Springer.

Chein, M., Mugnier, M.-L., Croitoru, M., 2013. Visual reasoning with graph-based mechanisms: the good, the better and the best. The Knowledge Engineering Review 28, 249–271.

Croitoru, M., Vesic, S., 2013. What can argumentation do for inconsistent ontology query answering? In: Proc. of SUM 2013 (to appear).

Dean, M., Sheperd, R., Arvola, A., Lampila, P., Lahteenmaki, L., Vassalo, M., Saba, A., Claupein, E., Winkelmann, M., 2007. Report on consumer expectations of health benefits of modified cereal products. Tech. rep., University of Surrey, UK.

DINABIO, 2008. Proceedings of dinabio développement et innovation en agriculture biologique. `http://www.inra.fr/ciag/revue_innovations_agronomiques/volume_4_janvier_2009`.

Dubuisson-Quellier, S., 2006. De la routine à la délibération. les arbitrages des consommateurs en situation d'achat. Réseaux 135/136, 253–284.

Dung, P. M., 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence Journal 77, 321–357.

32

FCN, 2009. Fibres, céréales et nutrition. `http://www.inra.fr/content/view/full/24670029`.

García, A. J., Simari, G. R., 2004. Defeasible logic programming: An argumentative approach. Theory and practice of logic programming 4, 95–138.

Ginon, E., Lohérac, Y., Martin, C., Combris, P., Issanchou, S., 2009. Effect of fibre information on consumer willingness to pay for french baguettes. Food Quality and Preference 20, 343–352.

HEALTHGRAIN, 2009. `http://www.healthgrain.org`.

Kaci, S., van der Torre, L., 2008. Preference-based argumentation: Arguments supporting multiple values. International Journal of Approximate Reasoning 48 (3), 730–751.

Klarman, S., Endriss, U., Schlobach, S., 2011. Abox abduction in the description logic *alc*. J. Autom. Reasoning 46 (1), 43–80.

Konig, M., Leclere, M., Mugnier, M.-L., Thomazo, M., 2012. A sound and complete backward chaining algorithm for existential rules. In: Krotzsch, M., Straccia, U. (Eds.), Web Reasoning and Rule Systems. Vol. 7497 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 122–138.

Layat, T., 2011. Place du pain dans l'équilibre alimentaire. Pratiques en nutrition 7 (26), 45–50.

Muller, J., Hunter, A., 2012. An argumentation-based approach for decision making. In: Proc. of ICTAI. pp. 564–571.

33

Perrot, N., Trelea, I., Baudrit, C., Trystram, G., Bourgine, P., 2011. Modelling and analysis of complex food systems: State of the art and new trends. Trends in Food Science & Technology 22 (6), 304 − 314.

PNNS (documents statutaires), 2010. `http://www.sante.gouv.fr/htm/pointsur/nutrition/pol_nutri4.htm`.

Rahwan, I., Simari, G., 2009. Argumentation in Artificial Intelligence. Springer.

Slavin, J., Green, H., 2007. Diatery fibre and satiety. British Nutrition Foundation 32(1), 32–42.

Thomopoulos, R., Charnomordic, B., Cuq, B., Abécassis, J., September 2009. Artificial intelligence-based decision support system to manage quality of durum wheat products. Quality Assurance and Safety of Crops & Foods 1 (3), 179–190.

34

Query Failure Explanation in Inconsistent Knowledge Bases Using Argumentation

Abdallah ARIOUA [a] Nouredine TAMANI [b] Madalina CROITORU [b] and
Patrice BUCHE [a]

[a] *IATE, INRA, University of Montpellier 2*
[b] *LIRMM, INRIA, University of Montpellier 2*

Abstract.
We address the problem of explaining Boolean Conjunctive Query (BCQ) failure in the presence of inconsistency within the Ontology-Based Data Access (OBDA) setting, where inconsistency is handled by the intersection of closed repairs semantics (ICR) and the ontology is represented by Datalog+/- rules. Our proposal relies on an interactive and argumentative approach where the processes of explanation takes the form of a dialogue between the *User* and the *Reasoner*. We exploit the equivalence between argumentation and ICR-semantics to prove that the *Reasoner* can always provide an answer for user's questions.

1. Introduction

In the popular ONTOLOGY-BASED DATA ACCESS setting the domain knowledge is represented by an ontology facilitating query answering over existing data [15]. In practical systems involving large amounts of data and multiple data sources, data inconsistency with respect to the ontology is unavoidable. Many inconsistency-tolerant semantics [3,2,12,13] have been proposed that rely on the notion of data repairs i.e. subsets of maximally consistent data with respect to the ontology. Query answering under these semantics may not be intuitively straightforward and can lead to loss of user's trust, satisfaction and may affect the system's usability [14]. As argued by Calvanese et al.[6] *explanation facilities* should not just account for user's "Why Q ?" question (why a query holds under a given inconsistency-tolerant semantics) but also for question like "Why not Q ?" (why a query does not hold under a given inconsistency-tolerant semantics).

The research problem addressed by this paper is the boolean conjunctive query failure explanation in inconsistent knowledge bases, precisely: "*Given an inconsistent KB and a boolean conjunctive query Q, why Q is not entailed from KB under the ICR-semantics?*". We use argumentation as an approach for explanation. We consider the logical instantiation of Dung's [10] abstract argumentation framework for OBDA in [8] and we exploit the equivalence result shown by the authors between the ICR-semantics and sceptical acceptance under preferred semantics to guarantee the existence of an explanation for any failed query. The explanation takes the form of a dialogue between the *User* and the *Reasoner* with the purpose of explaining the query failure. At each level of the dialogue, we use language-based introduced primitives such as *clarification* and

194

deepening to further refine the answer. The added value of our contribution lies in its significance and originality. We are the first to propose query failure explanation in the context of OBDA for **inconsistent knowledge bases** by means of **argumentation**. Our approach differs from [4,6] in handling query failure since we consider an inconsistent setting within OBDA. In addition, the work presented in [11] is neither applied to an OBDA context nor to the Datalog+/- language.

2. Background and Overview

In this section, we introduce the motivation and the context of our work and a formal definition of the addressed problem. Consider a knowledge base about university staff and students which contains inconsistent knowledge. This inconsistency is handled by ICR-semantics. The *User* might be interested in knowing why the knowledge base *does not entail* the query Q:"Luca is a student". Observe that the individual δ (e.g. Luca in the example above) is a negative answer for a conjunctive query Q (e.g. get me all the students in the example above) if and only if the boolean conjunctive query $Q(\delta)$ (e.g. student(Luca) in the example above) has failed. Hence, in this paper we concentrate only explaining the failure of a boolean conjunctive query. Let us formally introduce the problem of *Query Failure Explanation* in inconsistent knowledge bases.

Definition 1 (Query Failure Explanation Problem $\mathscr{P}$) *Let $\mathscr{K}$ be an inconsistent knowledge base, Q a Boolean Conjunctive Query such that $\mathscr{K} \nvDash_{ICR} Q$. We then call $\mathscr{P} = \langle \mathscr{K}, Q \rangle$ a Query Failure Explanation Problem (QFEP).*

To address the *Query Failure Explanation Problem*, we use a logical instantiation of Dung's [10] abstract argumentation framework for OBDA in [8] ensuring that the argumentation framework used respects the rationality postulates [7].

Let us first introduce the OBDA setting and inconsistency-tolerant semantics. We consider *the positive existential* conjunctive fragment of first-order logic, denoted by FOL($\wedge, \exists$), which is composed of formulas built with the connectors $(\wedge, \rightarrow)$ and the quantifiers $(\exists, \forall)$. For more details about the language please check [8]. In this paper, for lack of space we simply give an example of a knowledge base $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ is composed of finite set of facts $\mathscr{F}$ and finite set of existential rules $\mathscr{R}$ and a finite set of negative constrains $\mathscr{N}$.

Example 1 *Let us consider an example inspired from [5]. In an enterprise, employees work in departments and use offices which are located in departments, some employees direct departments, and supervise other employees. In addition, a supervised employee cannot be a manager. A director of a given department cannot be supervised by an employee of the same department, and two employees cannot direct the same department, and an employee cannot work in more than one department. The following sets of (existential) rules $\mathscr{R}$ and negative constraints $\mathscr{N}$ model the corresponding ontology:*

$$\mathscr{R} = \begin{cases} \forall x \forall y \, (works\_in(x,y) \rightarrow emp(x)) & (r_1) \\ \forall x \forall y \, (directs(x,y) \rightarrow emp(x)) & (r_2) \\ \forall x \forall y \, (directs(x,y) \wedge works\_in(x,y) \rightarrow manager(x)) & (r_3) \\ \forall x \forall y \forall z \, locate\_office(y,z) \wedge uses\_office(x,y) \rightarrow works\_in(x,z) & (r_4) \end{cases}$$

$$\mathcal{N} = \begin{cases} \forall x \forall y \, (supervises(x,y) \wedge manager(y)) \rightarrow \bot & (n_1) \\ \forall x \forall y \forall z \, (supervises(x,y) \wedge works\_in(x,z) \wedge directs(y,z)) \rightarrow \bot & (n_2) \\ \forall x \forall y \forall z \, (works\_in(x,y) \wedge works\_in(x,z)) \rightarrow \bot & (n_3) \end{cases}$$

Let us suppose the following set of facts $\mathcal{F}$ that represent explicit knowledge:

$$\mathcal{F} = \begin{cases} directs(John,d_1) & (f_1) & directs(Tom,d_1) & (f_2) \\ directs(Tom,d_2) & (f_3) & supervises(Tom,John) & (f_4) \\ works\_in(John,d_1) & (f_5) & works\_in(Tom,d_1) & (f_6) \\ works\_in(Carlo,Statistics) & (f_7) & works\_in(Luca,Statistics) & (f_8) \\ works\_in(Jane,Statistics) & (f_9) & works\_in(Linda,Statistics) & (f_{10}) \\ uses\_office(Linda,o_1) & (f_{11}) & locate\_office(o_1,Accounting) & (f_{12}) \end{cases}$$

Let $F \subseteq \mathcal{F}$ be a set of facts and $\mathcal{R}$ be a set of rules. An $\mathcal{R}$-derivation of F in $\mathcal{K}$ is a finite sequence $\langle F_0,...,F_n \rangle$ of sets of facts s.t $F_0 = F$, and for all $i \in \{0,...,n\}$ there is a rule $r_i = (H_i,C_i) \in \mathcal{R}$ and a logical entailment from the H_i to F_i. For a set of facts $F \subseteq \mathcal{F}$ and a query Q and a set of rules $\mathcal{R}$, we say $F,\mathcal{R} \models Q$ iff there exists an $\mathcal{R}$-derivation $\langle F_0,...,F_n \rangle$ such that $F_n \models Q$. Given a set of facts $F \subseteq \mathcal{F}$ and a set of rules $\mathcal{R}$, the closure of F with respect to $\mathcal{R}$, denoted by $\mathtt{Cl}_\mathcal{R}(F)$ is the minimal set of all the knowledge that can be derived from a set of facts F by applying all the rules of $\mathcal{R}$. Finally, we say that a set of facts $F \subseteq \mathcal{F}$ and a set of rules $\mathcal{R}$ *entail* a fact f (and we write $F,\mathcal{R} \models f$) iff the closure of F by all the rules entails f (i.e. $\mathtt{Cl}_\mathcal{R}(F) \models f$).

Given a knowledge base $\mathcal{K} = (\mathcal{F},\mathcal{R},\mathcal{N})$, a set $F \subseteq \mathcal{F}$ is said to be *inconsistent* iff there exists a constraint $n \in \mathcal{N}$ such that $F \models H_n$, where H_n is the hypothesis of the constraint n. A set of facts is consistent iff it is not inconsistent. Notice that (like in classical logic) one can entail everything from an inconsistent set. A common solution [2, 12] is to construct maximal (with respect to set inclusion) consistent subsets of $\mathcal{K}$. Such subsets are called *repairs* and denoted by $\mathcal{R}epair(\mathcal{K})$. Once the repairs are computed, different semantics can be used for query answering over the knowledge base. In this paper we focus on (**I**ntersection of **C**losed **R**epairs semantics) [2] and we will denote ICR entailment as $\mathcal{K} \models_{ICR} Q$.

Example 2 *The knowledge base in Example 1 is inconsistent because the set of facts $\{f_1,f_4,f_6\} \subseteq \mathcal{F}$ is inconsistent since it violates the negative constraint n_2. To be able to reason in presence of inconsistency one has to construct first the repairs and intersect their closure. The following is of the repairs:*
$\mathscr{A}_1 = \{directs(John,d_1), supervises(Tom,John), works\_in(Linda,Statistic), uses\_office(Linda,o_1), directs(Tom,d_1), directs(Tom,d_2), works\_in(Carlo,Statistic), works\_in(Jane,Statistic), works\_in(Luca,Statistic), emp(John), emp(Tom), emp(Carlo), emp(Luca), emp(Jane), emp(Linda)\}.$
The intersection of all closed repairs is:
$\bigcap_{A \in \mathcal{R}epair(\mathcal{K})} \mathtt{Cl}_\mathcal{R}(A) = \{directs(Tom,d_1), directs(Tom,d_2), works\_in(Carlo,Statistics), works\_in(Luca,Statistics), works\_in(Jane,Statistics), emp(Carlo), emp(Jane), emp(Luca), emp(Tom), emp(John), emp(Linda)\}.$

Observe that in the intersection of all closed repairs there is $works\_in(Luca,Statistics)$. That means that $works\_in(Luca,Statistics)$ is ICR-entailed from the knowledge base. Whereas, $works\_in(Linda,Statistics)$ is not ICR-entailed since the facts about Linda are conflicting (because she works also for the department of Accounting).

3. Argumentation Framework, Deepening and Clarification

In what follows we quickly recall the definition of argumentation framework in the context of rule-based languages. We use the definition of argument of [8] and extend it to the notions of deepened and clarified arguments. Given a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, the *corresponding argumentation framework* $\mathcal{AF}_{\mathcal{K}}$ is a pair $(\texttt{Arg}, \texttt{Att})$ where $\texttt{Arg}$ is the set of arguments that can be constructed from $\mathcal{F}$ and $\texttt{Att}$ is an asymmetric binary relation called *attack* defined over $\texttt{Arg} \times \texttt{Arg}$. Given an argument a we denote by $Supp(a)$ the support of the argument and by $Conc(a)$ the conclusion.

Example 3 (Argument) *The following argument indicates that John is an employee because he directs department d_1:*
$$a = (\{directs(John, d_1)\}, \{directs(John, d_1), emp(John)\}, emp(John)).$$

Example 4 (Attack) *Consider the argument a of Example 3, the following argument $b = (\{supervises(Tom, John), works\_in(Tom, d_1)\}, supervises(Tom, John) \wedge works\_in(Tom, d_1))$ attacks a, because $\{supervises(Tom, John) \wedge works\_in(Tom, d_1), directs(John, d_1)\}$ is $\mathcal{R}$-inconsistent since it violates the constraint n_2.*

The results of [8] show the equivalence between sceptically acceptance under preferred semantics and ICR-entailment. Let us now propose functionalities that give the *User* the possibility to manipulate arguments to gain clarity for query answering and namely: deepening and clarification. Deepening aims at showing the reason why an argument attacks another. In our knowledge base the attack is justified by the violation of a constraint. Put differently, an argument attacks another argument if the conclusion of the former and the hypothesis of the latter are mutually exclusive. Thus deepening amounts to explain the attack between two arguments by showing the violated constraint.

Definition 2 (Deepening $\mathbb{D}$) *Given two arguments $a, b \in \mathscr{A}$. The mapping deepening denoted by $\mathbb{D}$ is a total function from the set $\texttt{Att}$ to $2^{\mathscr{N}}$ defined as follows:* $\mathbb{D}(b, a) = \{n|$
> *1. $n \in \mathscr{N}$ and ,*
> *2. $\exists f \in \texttt{Supp}(a)$ such that $\texttt{Cl}_{\mathscr{R}}(\{Conc(b), f\}) \models H_n$.*
> *}*
Note that H_n is the hypothesis of the constraint n.

Example 5 (Deepening) *Consider the argument a of Example 3, the argument $b = (\{supervises(Tom, John), works\_in(Tom, d_1)\}, supervises(Tom, John) \wedge works\_in(Tom, d_1))$ attacks a, hence deepening is $\mathbb{D}(b, a) = \{\forall x \forall y \forall z (supervises(x, y) \wedge work\_in(x, z) \wedge directs(y, z)) \rightarrow \bot\}$. This explains why the argument b attacks a.*

The information carried by the argument would be more useful if the structure exhibits the line of reasoning leading to the conclusion, called *clarifying* the argument.

Definition 3 (Clarifying $\mathbb{C}$) *Given an argument $a \in \mathscr{A}$. The mapping clarification denoted by $\mathbb{C}$ is a total function from the set $\mathscr{A}$ to $2^{\mathscr{R}}$ such that:* $\mathbb{C}(a = \langle F_0, ..., F_n \rangle) = \{r | r \in \mathscr{R}$ *s.t r is applicable to F_i and the application of r on F_i yields F_{i+1} for all $i \in \{0, ..., n\}\}$.*

Definition 4 (Clarified Argument) *Given an argument $a \in \mathscr{A}$. The corresponding clarified argument C_a is a 3-tuple $\langle \texttt{Supp}(a), \mathbb{C}(a), \texttt{Conc}(a) \rangle$ such that $\mathbb{C}(a) \subseteq \mathscr{R}$ are the rules used to derive the conclusion $\texttt{Conc}(a)$.*

Example 6 (Clarification count. Example 3) *A clarified version of the argument a is* $C_a = (\{diretcs(John, d_1)\}, \{\forall x \forall d\, directs(x, d) \rightarrow emp(x)\}, emp(John))$ *such that* $\mathrm{Supp}(C_a) = \{directs(John, d_1)\}$, $\mathbb{C}(C_a) = \{\forall x \forall d\, directs(x, d) \rightarrow emp(x)\}$ *and* $\mathrm{Conc}(C_a) = emp(John)$.

4. Dialectical Explanation for Query Failure

In what follows, we describe a simple dialectical system of explanation based on the work of [9]. Our system is custom-tailored for the problem of *Query Failure Explanation* under ICR-semantics in inconsistent knowledge bases with rule-based language. Our *dialectical explanation* involves two parties: the *User* and the *Reasoner*. The *User* wants to understand why the query is not ICR-entailed and the *Reasoner* provides a respond aiming at showing the reason why the query is not ICR entailed. We model this explanation through a dialogue composed of moves (speech acts) put forward by both the *User* and the *Reasoner*. This dialogue is governed by rules (pre/post conditions rules, termination rules, success rules) that specify what type of moves should follow the other, the conditions under which the dialogue terminates, and when and under which conditions the explanation has been successfully achieved (success rules).

We denote by $Arg^+(Q)$ the set of all arguments that support the query Q, namely $a \in Arg^+(Q)$ iff $\mathrm{Conc}(a) \models Q$. In what follows we define types of moves that can be used in the dialogue.

Definition 5 (Moves) *A move is a 3-tuple $m = \langle ID, I, \omega \rangle$ such that:*

1. *m is an **explanation request**, denoted by $m^{\mathtt{ERQ}}$ iff $ID = User$, I is a query Q and ω is an argument supporting Q.*
2. *m is an **explanation response**, denoted by $m^{\mathtt{ERP}}$ iff $ID = Reasoner$, I is an argument supporting Q and ω is an argument such that ω attacks I.*
3. *m is a **follow-up question**, denoted by $m^{\mathtt{FQ}}$ iff $ID = User$, I is an argument and ω is either $\mathrm{Conc}(I)$ or an argument ω_1 that supports Q s.t $(\omega, \omega_1) \in \mathtt{Att}$.*
4. *m is a **follow-up answer**, denoted by $m^{\mathtt{FA}}$ iff $ID = Reasoner$, I is an argument and ω is either a deepening $\mathbb{D}$ or a clarified argument $\mathbb{C}(I)$.*

The explanation request $m^{\mathtt{ERQ}} = \langle User, Q, \omega \rangle$ is an explanation request made by the *User* asking *"why the query Q is not ICR-entailed while there is an argument ω asserts the entailment of Q"*, an explanation response $m^{\mathtt{ERP}} = \langle Reasoner, \omega, \omega_1 \rangle$ made by the *Reasoner* is an explanation for the previous inquiry by showing that the argument ω (that supports Q) is the subject of an attack made by ω_1. The *User* also can ask a follow-up question if the *Reasoner* provides an explanation. The follow-up question $m^{\mathtt{FQ}} = \langle User, \omega_1, \omega \rangle$ is a compound move, it can represent a need for *deepening* (the *User* wants to know why the argument ω_1 is attacking the argument ω) or the need for clarification (how the argument ω_1 comes to a certain conclusion). To distinguish them, the former has $\omega = \mathrm{Conc}(\omega_1)$ and the latter has ω as an argument. A follow-up answer $m^{\mathtt{FA}} = \langle Reasoner, \omega_1, \omega_1' \rangle$ is also a compound move. Actually, it depends on the follow-up question. It shows the argument ω_1 that needs to be deepened (resp. clarified) and its deepening (resp. clarification) by the deepening mapping $\mathbb{D}(\omega_1, \omega)$ (resp. clarification mapping $\mathbb{C}(\omega)$) in Definition 4 (resp. Definition 6). An example is provided afterward.

In what follows we specify the structure of dialectical explanation and the rules that have to be respected throughout the dialogue.

Definition 6 (Dialectical Explanation) *Given a QFEP $\mathscr{P}$. A dialectical explanation $\mathscr{D}_{exp}$ for $\mathscr{P}$ is a non-empty sequence of moves $\langle m_1^s, m_2^s, ..., m_n^s \rangle$ where $s \in \{\text{ERQ}, \text{FQ}, \text{ERP}, \text{FA}\}$ and $i \in \{1, ..., n\}$ such that:*

1. *The first move is always an explanation request m_1^{ERQ}, we call it an opening.*
2. *$s \in \{\text{ERQ}, \text{FQ}\}$ iff i is odd, $s \in \{\text{ERP}, \text{FA}\}$ iff i is even.*
3. *For every explanation request $m_i^{\text{ERQ}} = \langle User, I_i, \omega_i \rangle$, I_i is the query Q to be explained and ω_i is an argument supporting Q and for all m_j^{ERQ} s.t $j < i$ $\omega_i \neq \omega_j$.*
4. *For every explanation response $m_i^{\text{ERP}} = \langle Reasoner, I_i, \omega_i \rangle$ s.t $i \geq 1$, $I_i = \omega_{i-1}$ and $\omega_i = \omega'$ s.t $(\omega', I_i) \in \text{Att}$.*
5. *For every follow-up question $m_i^{\text{FQ}} = \langle User, I_i, \omega_i \rangle$, $i \geqslant 1$, $I_i = \omega_{i-1}$ and ω is either I_{i-1} or $\text{Conc}(\omega_{i-1})$.*
6. *For every follow-up answer $m_i^{\text{FA}} = \langle Reasoner, I_i, \omega_i \rangle$, $i \geqslant 1$, $I_i = I_{i-1}$ and $\omega_i = \mathbb{D}(I_i, \omega_{i-1})$ or $\omega = \mathbb{C}(I_i)$.*

We denote by $Arg_{user}(\mathscr{D}_{exp})$ the set of all arguments put by the User in the dialogue.

Every dialogue has to respect certain rules (protocol). Theses rules organize the way the *Reasoner* and the *User* should put the moves. For each move we specify the conditions that have to be me for the move to be valid (preconditions). We also specify the conditions that identify the next moves (postconditions).

Definition 7 (Pre/Post Condition Rules) *Given a QFEP $\mathscr{P}$ and a dialectical explanation $\mathscr{D}_{exp}$ for $\mathscr{P}$. Then, $\mathscr{D}_{exp}$ has to respect the following rules:*

Explanation request:

- *Preconditions: The beginning of the dialogue or the last move of the Reasoner was either an explanation response or a follow-up answer.*
- *Postconditions: The next move must be an explanation answer.*

Explanation response:

- *Preconditions: The last move by the User was an explanation request.*
- *Postconditions: The next move must be either another explanation request (it may implicitly means that the User had not understood the previous explanation) or a follow-up question.*

Follow-up question:

- *Preconditions: The last move by the Reasoner was an explanation response or this follow-up question is not the second in a row.*
- *Postconditions: The next move must be a follow-up answer.*

Follow-up answer:

- *Preconditions: The last move by the User was a follow-up question.*
- *Postconditions: The next move must be an explanation request (it may implicitly means that the User had not understood the previous explanation).*

Beside the previous rules, there are termination rules that indicate the end of a dialectical explanation.

Definition 8 (Termination Rules) *Given a QFEP $\mathscr{P}$ and a dialectical explanation $\mathscr{D}_{exp}$ for $\mathscr{P}$. Then, $\mathscr{D}_{exp}$ terminates when the User puts an empty explanation request $m_i^{\mathtt{ERQ}} = \langle User, \emptyset, \emptyset \rangle$ or when $Arg_{User}(\mathscr{D}_{exp}) = Arg^+(Q)$.*

The rules in Definition 7 & 8 state that the *Reasoner* is always **committed** to respond with an explanation response, the *User* then may indicate the end of the dialogue by an *empty explanation request* (Definition 8) declaring his/her understanding, otherwise starts another explanation request (this indicates that he/she has not understood the last explanation) or asks a follow-up question, the *User* cannot ask more than two successive follow-up questions. If the *User* asks a follow-up question then the *Reasoner* is **committed** to a follow-up answer. When the *User* asks for another explanation he/she cannot use an argument that has already been used. If the *User* ran out of arguments and he/she has not yet understood, the dialogue ends (Definition 8) and the explanation is judged unsuccessful. It is important to notice that when the *Reasoner* wants to answer the *User* there may be more than one argument to chose. There are many "selection strategies" that can be used in such case (for instance, the shortest argument, the least attacked argument...etc), but their study is beyond the scope of the paper.

In what follows we elaborate more on the success and the failure of an explanation.

Definition 9 (Success Rules) *Given a QFEP $\mathscr{P}$ and a dialectical explanation $\mathscr{D}_{exp}$ for $\mathscr{P}$. Then, $\mathscr{D}_{exp}$ is successful when it terminates with an empty explanation request $m_i^{\mathtt{ERQ}} = \langle User, \emptyset, \emptyset \rangle$, otherwise it is unsuccessful.*

A dialectical explanation is judged to be successful if the *User* terminates the dialogue voluntarily by putting an empty explanation request. If the *User* has used all arguments supporting Q then he/she is forced to stop without indicating his/her understanding, in this case we consider the explanation unsuccessful. By virtue of the equivalence between ICR-semantics and argumentation presented in Section 3, the existence of response is always guaranteed. This property is depicted in the following proposition.

Proposition 1 (Existence of response) *Given a QFEP $\mathscr{P}$ and a dialectical explanation $\mathscr{D}_{exp}$ for $\mathscr{P}$. Then, For every $m_i^s \in \mathscr{D}_{exp}$ s.t $s \in \{\mathtt{ERQ}, \mathtt{FQ}\}$ and $1 \le i \le |\mathscr{D}_{exp}|$, the next move m_{i+1}^s s.t $s \in \{\mathtt{ERP}, \mathtt{FA}\}$ always exists.*

5. Conclusion

In this paper, we have presented a dialectical approach for explaining boolean conjunctive queries failure, designated by Query Failure Explanation Problem (QFEP), in an inconsistent ontological knowledge base where inconsistency is handled by inconsistency-tolerant semantics (ICR) and issued from the set of facts. The introduced approach relies on both (i) the relation between ontological knowledge base and logical argumentation framework and (ii) the notions of argument deepening and clarifications. So, through a dialogue, the proposed approach explains to the *User* how and why his/her query is not entailed under ICR semantics.

We currently investigate the explanation problem not only for Query Failure but also for Query Answering. We have proposed a Query Explanation framework under the CoGui editor[1] and plan to test the two approaches within the DUR-DUR ANR project which investigates the use of argumentation in agri-food chains.

References

[1] A. Arioua, N. Tamani, and M. Croitoru. On conceptual graphs and explanation of query answering under inconsistency. In N. Hernandez, R. Jaesche, and M. Croitoru, editors, *Proceedings of the 2014 International Conference of Conceptual Structures (ICCS)*, volume LNAI 8577, pages 51–64. Springer, 2014.

[2] M. Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proc of AAAI*, 2012.

[3] M. Bienvenu and R. Rosati. Tractable approximations of consistent query answering for robust ontology-based data access. In *Proceedings of IJCAI'13*, pages 775–781. AAAI Press, 2013.

[4] A. Borgida, D. Calvanese, and M. Rodriguez-Muro. Explanation in the dl-lite family of description logics. In R. Meersman and Z. Tari, editors, *On the Move to Meaningful Internet Systems: OTM 2008*, volume 5332 of *LNCS*, pages 1440–1457. Springer Berlin Heidelberg, 2008.

[5] A. Calì, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 14:57–83, 2012.

[6] D. Calvanese, M. Ortiz, M. Šimkus, and G. Stefanoni. Reasoning about explanations for negative query answers in dl-lite. *Journal of Artificial Intelligence Research*, 48:635–669, 2013.

[7] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5):286–310, 2007.

[8] M. Croitoru and S. Vesic. What can argumentation do for inconsistent ontology query answering? In W. Liu, V. Subrahmanian, and J. Wijsen, editors, *Scalable Uncertainty Management*, volume 8078 of *LNCS*, pages 15–29. Springer Berlin Heidelberg, 2013.

[9] Douglas Walton. A Dialogue System Specification for Explanation . *Synthese, 182(3), 2011,*, Volume 182(3):349–374, 2011.

[10] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

[11] A. J. Garcia, C. I. Chesñevar, N. D. Rotstein, and G. R. Simari. Formalizing dialectical explanation support for argument-based reasoning in knowledge-based systems. *Expert Systems with Applications*, 40(8):3233 – 3247, 2013.

[12] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. F. Savo. Inconsistency-tolerant semantics for description logics. In *Proceedings of the Fourth International Conference on Web Reasoning and Rule Systems*, RR'10, pages 103–117, Berlin, Heidelberg, 2010. Springer-Verlag.

[13] T. Lukasiewicz, M. V. Martinez, and G. I. Simari. Complexity of inconsistency-tolerant query answering in datalog+/-. In R. Meersman, H. Panetto, T. Dillon, J. Eder, Z. Bellahsene, N. Ritter, P. D. Leenheer, and D. Dou, editors, *Proceedings of the 12th International Conference on Ontologies, Databases, and Applications of Semantics, September 10-11, 2013*, volume 8185 of *LNCS*, pages 488–500. Springer, 2013.

[14] D. L. McGuinness and P. F. Patel-Schneider. Usability issues in knowledge representation systems. In *In Proc. of AAAI-98*, pages 608–614, 1998.

[15] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. In *Journal on data semantics X*, pages 133–173. Springer, 2008.

An analysis of the SUDOC bibliographic knowledge base from a link validity viewpoint

Léa Guizol*, Olivier Rousseaux**, Madalina Croitoru*,
Yann Nicolas**, Aline Le Provost**

*LIRMM (University of Montpellier II & CNRS), INRIA Sophia-Antipolis, France
**ABES, France

Abstract. In the aim of evaluating and improving link quality in bibliographical knowledge bases, we develop a decision support system based on partitioning semantics. The novelty of our approach consists in using symbolic values criteria for partitioning and suitable partitioning semantics. In this paper we evaluate and compare the above mentioned semantics on a real qualitative sample. This sample is issued from the catalogue of French university libraries (SUDOC), a bibliographical knowledge base maintained by the University Bibliographic Agency (ABES).

1   Introduction

Real World Context. The SUDOC (catalogue du Système Universitaire de Documentation) is a large bibliographical knowledge base managed by ABES (Agence Bibliographique de l'Enseignement Supérieur). The SUDOC contains **bibliographic notices** (document descriptions $\approx$ 10.000.000), and **authorship notices** (person descriptions $\approx$ 2.000.000). An authorship notice possesses some attributes (ppn[1], appellation set, date of birth...). A bibliographic notice also possesses some attributes (title, ppn[1], language, publication date...) and **link**(s) to authorship notices. A link is labeled by a **role** (as *author*, *illustrator* or *thesis advisor*) and means that the person described by the authorship notice has participated as the labeled role to the document described by the bibliographic notice.

One of the most important tasks for ABES experts is to reference a new book in SUDOC. To this end, the expert has to register the title, number of pages, types of publication domains, language, publication date, and so on, in a new bibliographic notice. This new bibliographic notice represents the physical books in the librarian hands which he/she is registering. He/she also has to register people which participated to the book's creation (namely the **contributors**). In order to do that, for each contributor, he/she selects every authorship notice (named *candidates*) which has an appellation similar to the book contributor. Unfortunately, there is not that much information in authorship notices because the librarian politics is to give minimal information, solely in order to distinguish two authorship notices which have the same appellation, and nothing more (they reference books, not people!). So the librarian has to look at bibliographic notices which are linked to authorship notices candidates (the *bibliography* of candidates) in

---

[1] A ppn identifies a notice.

order to see whether the book in his/her hands seems to be a part of the bibliography of a particular candidate. If it is the case, he/she links the new bibliographic notice to this candidate and looks at the next unlinked contributor. If there is no good candidate, he/she creates a new authorship notice to represent the contributor.

This task is fastidious because it is possible to have a lot of candidates for a single contributor (as much as 27 for a contributor named "BERNARD, Alain"). This creates errors, which in turn can create new errors since linking is an incremental process. In order to help experts to repair erroneous links, we proposed two **partitioning semantics** in [11] which enables us to detect erroneous links in bibliographic knowledge bases. A partitioning semantics evaluates and compares **partitions**[2].

Contribution. The contribution of this paper is to practically evaluate the results quality of partitioning semantics [11] on a real SUDOC sample. We recall the semantics in section 3, clearly explain on which objects and with which criteria the semantics have been applied in section 2, and present qualitative results in section 4. We discuss the results and conclude the paper in section 5.

2 Qualitative experiments

In this section, we first adapt the **entity resolution problem**[3][4] to investigate link quality in SUDOC in section 2.1. This problem is known in literature under very different names (as record linkage [16], data deduplication [2], reference reconciliation [14]...). Then we define (section 2.3) and detail (section 2.4) criteria used in order to detect erroneous links in SUDOC. Those criteria are used on SUDOC subsets defined in section 2.2.

2.1 Contextual entities: from erroneous links to entity resolution

In order to detect and repair erroneous links, we represent SUDOC links into **contextual entity** (the i contextual entity is denoted Nc_i). A contextual entity represents a bibliographic notice Nb_j from the viewpoint of one of its contributor, named the C contributor of Nc_i and denoted $C(Nc_i)$. The contextual entities are compared together with an entity resolution method, in order to see which ones have a contributor representing a same real-word person. As explained in [8], traditional entity resolution methods cannot be directly applied. This entity resolution method is supposed to group (in a same class of the created partition) the contextual entities such as their C contributor represents a same real-word person, and to separate the other ones (to put them in distinct partition classes). A contextual entity Nc_i has several attributes. Most of them are $Nb(Nc_i)$ attributes (as title, publication date, publication language, publication domain codes list) and others depend on the C contributor:

---

[2] A **partition** P of an object set X is a set of **classes** (X subsets) such as each object of X is in one and only one P class.

[3] The **entity resolution problem** is the problem of identifying as equivalent two objects representing the same real-world entity.

– role of the C contributor (there is a set of typed roles as "thesis_advisor"),
– list of the possible appellations of the C contributor. An appellation is composed of a name and a surname, sometimes abbreviated (as "J." for surname),
– list of contributors which are not C. For each of them, we have the identifier of the authorship notice which represents it, and the role.

The publication language attribute is typed (for example, "eng" for English language, "fre" for French language and so on). The publication date is most of the time the publication year ("1984"). Sometimes information is missing and it only gives the century or decade ("19XX" means that the document has been published last century). A publication domain is not a describing string but a code with 3 digits which represent a domain area.

Example 1 (Contextual entity attributes). The authorship notice of ppn **026788861**, which represents "CHRISTIE, Agatha" is linked as "author" to the bibliographic notice of ppn 121495094, which represents "*Evil under the sun*" book. The contextual entity which represents this links has the following attributes:

– title: "*Evil under the sun*"
– publication date: "2001"
– publication language: "eng"
– publication domain codes list: {} (they have not been given by a librarian)
– list of the possible appellations of the C contributor: {"CHRISTIE, Agatha","WEST-MACOTT, Mary","MALLOWAN, Agatha","MILLER, Agathe Marie Clarissa"}
– role of the C contributor: "author"
– list of contributors which are not C: {} (there is no other contributors in this case)

Let Nc_i be the contextual entity identified by i. As any contextual entity, it has been constructed because of a link between an authorship notice and a bibliographic notice, which are respectively denoted $Na(Nc_i)$ and $Nb(Nc_i)$. We define two particular partitions: the initial one and the human one.

The **initial partition** (denoted Pi) of contextual entities is the one such as two contextual entities Nc_i, Nc_j are in a same class if and only if $Na(Nc_i) = Na(Nc_j)$. This represents the original organization of links in SUDOC.

The **human partition** (denoted Ph) of contextual entities is a partition based on an human expert's advice: two contextual entities Nc_i, Nc_j are in a same class if and only if the expert thinks that their C contributor corresponds to a same real word person.

The goal of this paper's work is to distinguish SUDOC subsets constructed as in the following section 2.2 with or without erroneous links. We make the hypothesis that the human partition has to be a best one (because it is the good one according to expert) and that the initial partition has to not be a best partition except if $Pi = Ph$. So, partitioning semantics are approved if Ph is a best partition according to the semantics, but not Pi. Let us determine what is a SUDOC contextual entities subset to partition.

2.2 Selecting contextual entities on appellation

A SUDOC subset $\mathbb{O}$ selected for an appellation A contains all contextual entities which represent a link between any SUDOC bibliographic notice and a SUDOC authorship

notice which has an appellation close to the appellation A. To select a SUDOC subset for a given appellation (as "BERNARD, Alain") is a way to separate SUDOC in subsets which can be treated separately, This is also a simulation of how experts select a SUDOC subset to work on it, as explained in part 1. In the following, we will only be interested into partitioning SUDOC subsets selected for an appellation. Let us define and describe criteria used in order to compare contextual entities together.

2.3 Symbolic criteria

In the general case, a **criterion** is a function which compares two objects and returns a comparison value. Let c be a criterion, and o_i, o_j are two objects. We denote $c(o_i, o_j)$, the comparison values according to c between o_i and o_j.

In this work case, we use **symbolic criteria** which can return *always*, *never*, *neutral*, a **closeness value** or a **farness value** as comparison value. *always* (respectively *never*) means that objects have to be in a same (respectively distinct) partition class[2]. Closeness (respectively farness) values are more or less intense and far from the *neutral* value, meaning that objects should be in a same (respectively distinct) partition class. Closeness (respectively farness) values are strictly ordered between themselves, specific to a criterion and less intense than *always* (respectively *never*). Those values are denoted $+,++$ and so on (respectively $-,--$) such as the more $+$ (respectively $-$) symbols they have, the more intense and the further from *neutral* the value is. For a criterion, *always* is more intense than $+++++$, which is more intense than $++$ which is more intense than $+$. $+$ is only more intense than *neutral*. *neutral* means that the criterion has no advice about whether to put objects in a same class or not.

2.4 Criteria for detecting link issues in SUDOC

In order to simulate human expert behaviour, nine symbolic criteria have been developed. Some are closeness-criteria[4] (*title*, *otherContributors*), farness-criteria[4] (*thesis*, *thesisAdvisor*, *date*, *appellation*, *language*) and others are both (*role*, *domain*). Each of these criteria give the *neutral* comparison value when a required attribute of a compared contextual entity is unknown and by default. Let Nc_i, Nc_j be two contextual entities.

– *appellation* criterion is a particular farness-criterion. Indeed, it compares appellation lists to determine which contextual entities can not have a same contributor C. When it is certain (as when appellations are "CONAN DOYLE, Arthur" and "CHRISTIE, Agatha"), it gives a *never* comparison value, which forbids other criteria to compare the concerned authorship notices together. This is also used to divide SUDOC in subsets which should be evaluated separately.
– *title* criterion is a closeness-criterion. This criterion can give an *always* value and 3 closeness comparison values. It is based on a Levenshtein comparison [13]. It is useful to determine which contextual entities represent a same work, edited several times. This is used by the *thesis* criterion.

---

[4] A closeness-criterion (respectively a farness-criterion) c is a criterion which can give a closeness or *always* (respectively a farness or *never*) comparison value to two objects.

- *otherContributors* criterion is a closeness-criterion. It counts the others contributors in common, by comparing their authorship notices. One (respectively several) other common contributor gives a $+$ (respectively $++$) comparison value.
- *thesis* criterion is a farness-criterion. $thesis(Nc_i, Nc_j) = -$ means that Nc_i, Nc_j are contextual entities which represent distinct thesis (recognized thanks to the *title* criterion) from their "author" point of view. $thesis(Nc_i, Nc_j) = --$ means that Nc_i, Nc_j have also been submitted simultaneously.
- *thesisAdvisor* criterion is a farness-criterion. $thesisAdvisor(Nc_i, Nc_j) = --$ (respectively $-$) means that Nc_i and Nc_j have a same contributor C if and only if this contributor has supervised a thesis before (respectively two years after) submitting his/her own thesis.
- *date* criterion is a farness-criterion. For 100 (respectively 60) years at least between publication dates, it gives a $--$ (respectively $-$) comparison value.
- *language* criterion is a farness-criterion. When publication languages are distinct and none of them is English, *language* returns a $-$ value.
- *role* criterion returns $+$ when contributor C roles are the same (except for current roles as "author", "publishing editor" or "collaborator"), or $-$ when they are distinct (except for some pairs of roles as "thesis advisor" and "author").
- *domain* criterion compares list of domain codes. Domain codes are pair-wise compared. $domain(Nc_i, Nc_j)$ gives closeness (respectively farness) comparison values if every Nc_i domain codes is close (respectively far) from a Nc_j domain code and the other way around.

Let us resume global and local semantics before to evaluate their relevance with respect to the above mentioned criteria on real SUDOC subsets.

3 Partitioning semantics

Let us summarize partitioning semantics detailed in [11]. A partitioning semantics evaluates and compares partitions on a same object set. The following partitioning semantics (in sections 3.1 and 3.2) are based on symbolic criteria.

3.1 Global semantics

In this section we define what is a a best partition on the object set $\mathbb{O}$ (with respect to the $\mathbb{C}$ criteria set) according to global semantics. A partition has to be **valid**[5][2] in order to be also a best one. A partition P has also an **intra value** and an **inter value** per criterion of $\mathbb{C}$. The intra value of a criterion c depends of the most intense (explained in section 2.3) farness or *never* value of c such as it compares two objects in a same class (should not be the case according to c). In the same way, the inter value of c depends of the most intense closeness or *always* value of c such as it compares two

---

[5] A partition P is **valid** if and only if there is no two objects o_i, o_j such as: (i) they are in a same class of P and they *never* have to be together according to a criterion (expressed by *never* comparison value), or (ii) they are in distinct P classes but *always* have to be together according to at least a criterion.

objects in distinct P classes. The inter value measures proximity between classes and the intra value measures distance between objects in a class [10]. We note that the *neutral* comparison value does not influence partition values.

A partition P on an object set $\mathbb{O}$ is a best partition according to a criteria set $\mathbb{C}$ if P is valid and P has a best value, meaning that it is impossible to improve an inter or intra value of any criterion $C \in \mathbb{C}$ without decreasing inter or intra value of a criterion $C' \in \mathbb{C}$ (it is a Pareto equilibrium [15]).

id	title	date	domains [...]	appellations
Nc_1	"*Letter to a Christian nation*"		religion	"HARRIS, Sam"
Nc_2	"*Surat terbuka untuk bangsa kristen*"	2008	religion	"HARRIS, Sam"
Nc_3	"*The philosophical basis of theism*"	1883	religion	"HARRIS, Samuel"
Nc_4	"*Building pathology*"	2001	building	"HARRIS, Samuel Y."
Nc_5	"*Building pathology*"	1936	building	"HARRIS, Samuel Y."
Nc_6	"*Aluminium alloys 2002*"	2002	physics	"HARRIS, Sam J."

Table 1. Example of objects set

Example 2 (Global semantics evaluating a partition on an object set $\mathbb{O}$).

Let us represent an object set $\mathbb{O} = \{Nc_1, Nc_2, Nc_3, Nc_4, Nc_5, Nc_6\}$ in table 1. Each object is a contextual entity and represents a link between a bibliographic notice and an authorship notice (here, an "author" of a book). Id is the object identity. For each of them, title, date of publication, publication domain and appellation of the contributor C are given as attributes.

Nc_1 and Nc_2 represent a same person, as Nc_4, Nc_5 does. The human partition on $\mathbb{O}$ is: $Ph = \{\{Nc_1, Nc_2\}, \{Nc_3\}, \{Nc_4, Nc_5\}, \{Nc_6\}\}$. This partition, according to global semantics and with respect to the criteria set $\mathbb{C} = \{appellation, title, domain, date\}$ (criteria are detailed in section 2.4) is not coherent with some of $\mathbb{C}$ criteria. The Ph value is such that:

– inter classes $domain$ value is very bad ($always$) because Nc_1 and Nc_2 are in distinct classes but are both about religion.
– intra classes $date$ value is bad ($--$) because Nc_4 and Nc_5 are in a same class, but with publication dates distant of more than 60 years and less than 100 years.

Ph has a best partition value because increasing an inter or intra criterion value (as inter $domain$ value by merging $\{Nc_1, Nc_2\}$ and $\{Nc_3\}$ classes) is not possible without decreasing an other inter or intra criterion value (Nc_2 and Nc_3 have publication dates distant more than 100 years, so put them in a same class will decrease $date$ intra value).

3.2 Local semantics

The local semantics, when evaluating a partition on an object set $\mathbb{O}$ with respect to a criteria set $\mathbb{C}$, gives a partition value per parts of $\mathbb{O}$. Parts of $\mathbb{O}$ can be coherent or incoherent. An **incoherent part** $\mathbb{O}_a$ is a subset of $\mathbb{O}$ such as:

– there is no $c(o_i, o_j)$, an *always* or closeness value with $Nc_i \in \mathbb{O} - \mathbb{O}_a$, $Nc_j \in \mathbb{O}_a$, and $c \in \mathbb{C}$;
– there is no subset of $\mathbb{O}_a$ for which the previous property is true;
– there is $b(o_k, o_l)$, a farness or *never* value such as $o_k, o_l \in \mathbb{O}_a$, and $b \in \mathbb{C}$.

An *incoherent part partition value* is based on every comparison between objects which are in it. The **coherent part** of an object set $\mathbb{O}$ is a $\mathbb{O}$ subset containing every $\mathbb{O}$ object which is not in a incoherent part of $\mathbb{O}$. The *coherent part partition value* of $\mathbb{O}$ is based on every comparison between objects which are not in the same incoherent part of $\mathbb{O}$.

Example 3 (Incoherent and coherent parts).
Let us identify incoherent parts of the object set $\mathbb{O}$ according to $\mathbb{C}$ given in example 2. Nc_1, Nc_2, Nc_3 are close together due to *domain* criterion: they are about religion. Nc_1, Nc_2, Nc_3 are not close to Nc_4, Nc_5 or Nc_6 according to any of $\mathbb{C}$ criteria and Nc_2, Nc_3 are far according to *date* criterion ($date(Nc_2, Nc_3) = --$) so $\{Nc_1, Nc_2, Nc_3\}$ is an incoherent part of $\mathbb{O}$. The same way, Nc_4, Nc_5 are close together according to *title* and *domain* criteria, but not close to Nc_6. Nc_4, Nc_5 are also far according to *date* criterion ($date(Nc_4, Nc_5) = -$) so $\{Nc_4, Nc_5\}$ is also an incoherent part.

So, there are 2 incoherent parts in $\mathbb{O}$: $\{Nc_1, Nc_2, Nc_3\}$ and $\{Nc_4, Nc_5\}$. Nc_6 is not in an incoherent part so Nc_6 is in the coherent part of $\mathbb{O}$.

A partition on $\mathbb{O}$ is a *best partition according to local semantics* if it has best partition values for each incoherent part of $\mathbb{O}$ and for the $\mathbb{O}$ coherent part.

Example 4 (Local semantics evaluating a partition on an object sets $\mathbb{O}$).
In example 3, we identified the incoherent parts of the object set $\mathbb{O} = \{Nc_1, Nc_2, Nc_3, Nc_4, Nc_5, Nc_6\}$ according to the criteria set $\mathbb{C} = \{appellation, title, domain, date\}$.

The partition on $\mathbb{O}$ given in example 2: is $Ph = \{\{Nc_1, Nc_2\}, \{Nc_3\}, \{Nc_4, Nc_5\}, \{Nc_6\}\}$. According to local semantics, Ph has 3 values, one for the coherent part and 2 for incoherent parts (1 per incoherent part):

– a perfect value for the coherent part of $\mathbb{O}$;
– the incoherent part $\{Nc_1, Nc_2, Nc_3\}$ has a very bad inter value for the *domain* criterion (*always*);
– the incoherent part $\{Nc_4, Nc_5\}$ has an bad intra value for the *date* criterion ($--$);

This semantics enables us to split an object set into several parts which can be evaluated separately. We explained local and global semantics in this part, which are a way to solve the entity resolution problem. Let us evaluate them on a real SUDOC sample.

4 Results

ABES experts have selected 537 contextual entity divided into 7 SUDOC subsets selected for an appellation. The table 2 shows for each SUDOC subset selected for an appellation A (please see section 2.2):

1. $|Nc|$ is the number of contextual entities which represent a link between a bibliographic notice and an authorship notice which has a close appellation to A,
2. $|Na|$ is the number of authorities notices according to human partitions (corresponding to class number of human partition),
3. "Ph best" (respectively "Pi best") shows whether the human partition Ph (respectively initial partition Pi) has a best value according to global semantics and with respect to all 9 criteria detailed in part 2.4,
4. $Ph \succ Pi$ is true if and only if Ph has a better value than Pi.

Appellation	$|Nc|$	$|Na|$	Ph best	Pi best	$Ph \succ Pi$	Ph' best	Repairs
"BERNARD, Alain"	165	27	no	not valid	yes	yes	
"DUBOIS, Olivier"	27	8	no	no	yes	no	1
"LEROUX, Alain"	59	6	no	not valid	yes	yes	
"ROY, Michel"	52	9	yes	not valid	yes	yes	
"NICOLAS, Maurice"	20	3	yes	no	yes	yes	
"SIMON, Alain"	63	13	no	no	yes	no	1
"SIMON, Daniel"	151	16	no	not valid	yes	yes	

Table 2. Human and initial partitions with respect to 9 criteria and global semantics

Local semantics, has the same results than global semantics on this sample.

For global semantics, Pi is never a best partition. 5 times out of 7, Ph does not have a best value (each time, it is due to the *domain* and *language* criteria, and two times *thesisAdvisor* is also involved), but it is all the time valid and better than Pi, which is encouraging for erroneous link detection. Erroneous links are particularly obvious when Pi is not even valid (4 times out of 7). It is due to the *title* criterion detailed in part 2.4. We regret that Ph is not all the time a best partition, but the global semantics is able to distinguish Pi from Ph in 5 cases out of 7: when Pi is not valid, or when Ph is a best partition but not Pi.

Because the *domain* and *language* criteria often considers that Ph is not a good enough partition, Ph was also evaluated for global semantics according to all criteria without *domain* and *language* (shown in table 2 in column "Ph' best") and that increases the human partition which obtains a best value in 3 more cases (for "BERNARD, Alain", "SIMON, Daniel" and "LEROUX, Alain" appellations). This tells us that *domain* and *language* criteria are not reasonably accurate.

In order to evaluate if Ph is far from having a best partition value, we enumerate the number of repairs to transform Ph' into a partition Ph'' which has a best value according to all criteria except *domain* and *language*. We show this repair number in the "Repairs" column of table 2. An atomic repair could be:

– merging two partition classes (corresponds to merging two contextual entities which represent a same real word person), or

– splitting a partition class in two classes (corresponds to separate books which are attributed to a same real word person but belong to two distinct real word persons).

We can see that only a few repairs are needed compared to the number of classes (corresponding to $|Na|$ column in the table): 1 repair for "Dubois, Olivier" and for "Bernard, Alain" appellations.

Let us highlight that observing human partition values has permitted to *detect and correct an erroneous link* (for "Roy, Michel" appellation) in the human reference set, validated with experts. The global semantics does not always consider that the human partition is a best partition, but in the worst case the human partition is very close to be one according to repairs number, and global semantics allow us to detect that initial partitions are much worse than human partitions. This last point is encouraging. This means that the semantics can also be useful to help in criteria tuning, by showing which criteria are bad according to human partitions, and for which authorship notices comparison. For example, the fact that the human partition value is often bad according to the $domain$ criterion shows that this criterion is actually not an accurate criterion. Let us talk about other entity resolution methods and conclude.

5 Discussion

The entity resolution problem [4][16][14][6] is the problem of identifying as equivalent two objects representing the same real-world entity. The causes of such mismatch can be due to homonyms (as in people with the same name), errors that occurred at data entry (like "Léa Guizo" for "Léa Guizol"), missing attributes (e.g publication date = XXXX), abbreviations ("L. Guizol") or attributes having different values for two objects representing the same entity (change of address).

The entity resolution problem can be addressed as a rule based pairwise comparison rule approach. Approaches have been proposed in literature [12] using a training pairs set for learning such rules. Rules can be then be chained using different constraints: transitivity [3], exclusivity [12] and functional dependencies [1] [9].

An alternative method for entity resolution problem is partitioning (hierarchical partitioning [5], closest neighbor-based method [7] or correlation clustering [3]). Our work falls in this last category. Due to the nature of treating criteria values, the closest approach to our semantics are [3] and [2]. We distinguish ourself to [3] and [2] because of (1) the lack of $neutral$ values in these approaches, (2) the numerization of symbolic values (numerically aggregated into -1 and $+1$ values), and (3) the use of numerical aggregation methods on these values.

Conclusion. In this paper we proposed a practical evaluation of the global and local semantics proposed in [11]. The conclusions of this evaluation are:

– For SUDOC subsets selected by appellation, *both semantics* are effective to distinguish a human partition from the initial partition; however it is not perfect with respect to our set of criteria (if the human partition is not a best partition, it has a close value).
– *Both semantics* could be useful to detect meaningless criteria.

As immediate next steps to complete this our work we mention using global or local semantics to improve implemented criteria.

References

1. R. Ananthakrishna, S. Chaudhuri, and V. Ganti. Eliminating fuzzy duplicates in data warehouses. In *Proceedings of the 28th international conference on Very Large Data Bases*, VLDB '02, pages 586–597. VLDB Endowment, 2002.
2. A. Arasu, C. Ré, and D. Suciu. Large-scale deduplication with constraints using dedupalog. In *Proceedings of the 25th International Conference on Data Engineering (ICDE)*, pages 952–963, 2009.
3. N. Bansal, A. Blum, and S. Chawla. Correlation clustering. volume 56, pages 89–113. Springer, 2004.
4. I. Bhattacharya and L. Getoor. *Entity Resolution in Graphs*, pages 311–344. John Wiley & Sons, Inc., 2006.
5. M. Bilenko, S. Basil, and M. Sahami. Adaptive product normalization: Using online learning for record linkage in comparison shopping. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.
6. P. Bouquet, H. Stoermer, and B. Bazzanella. An entity name system (ens) for the semantic web. In *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*, ESWC'08, pages 258–272, Berlin, Heidelberg, 2008. Springer-Verlag.
7. S. Chaudhuri, V. Ganti, and R. Motwani. Robust identification of fuzzy duplicates. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 865–876. IEEE, 2005.
8. M. Croitoru, L. Guizol, and M. Leclère. On Link Validity in Bibliographic Knowledge Bases. In *IPMU'2012: 14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, volume Advances on Computational Intelligence, pages 380–389, Catania, Italie, July 2012. Springer.
9. W. Fan. Dependencies revisited for improving data quality. In *Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 159–170. ACM, 2008.
10. A. Guénoche. Partitions optimisées selon différents critères: évaluation et comparaison. *Mathématiques et sciences humaines. Mathematics and social sciences*, (161), 2003.
11. L. Guizol, M. Croitoru, and M. Leclere. Aggregation semantics for link validity. *AI-2013: Thirty-third SGAI International Conference on Artificial Intelligence*, page to appear, 2013.
12. R. Gupta and S. Sarawagi. Answering table augmentation queries from unstructured lists on the web. *Proceedings of the VLDB Endowment*, 2(1):289–300, 2009.
13. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.
14. F. Saïs, N. Pernelle, and M.-C. Rousset. Reconciliation de references : une approche logique adaptee aux grands volumes de donnees. In *EGC*, pages 623–634, 2007.
15. S. Wang. Existence of a pareto equilibrium. *Journal of Optimization Theory and Applications*, 79(2):373–384, 1993.
16. W. E. Winkler. Overview of record linkage and current research directions. Technical report, BUREAU OF THE CENSUS, 2006.

Bibliography

[1] L. Amgoud, L. Bodenstaff, M. Caminada, P. McBurney, S. Parsons, H. Prakken, J. Veenen, and G. Vreeswijk. Final review and report on formal argumentation system.deliverable d2.6 aspic. Technical report, 2006.

[2] L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning*, 29 (2):125–169, 2002.

[3] L. Amgoud and H. Prade. Reaching agreement through argumentation: a possiblistic approach. In *Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning*, pages 175–182, 2004.

[4] A. Arioua, N. Tamani, and M. Croitoru. On conceptual graphs and explanation of query answering under inconsistency. In N. Hernandez, R. Jaesche, and M. Croitoru, editors, *Proceedings of the 2014 International Conference of Conceptual Structures (ICCS)*, volume LNAI 8577, pages 51–64. Springer, 2014.

[5] A. Arioua, N. Tamani, M. Croitoru, and P. Buche. Query failure explanation in inconsistent knowledge bases using argumentation. In *Proc. of COMMA 2014*, 2014 (To appear).

[6] F. Baader, S. Brandt, and C. Lutz. Pushing the el envelope. In *Proc. of IJCAI 2005*, 2005.

[7] J.-F. Baget, M. Croitoru, A. Gutierrez, M. Leclere, and M.-L. Mugnier. Translations between rdf(s) and conceptual graphs. In *Proceedings of ICCS 2010*, pages 28–41, 2010.

[8] J.-F. Baget, M. Croitoru, and B. Paiva Lima da Silva. Alaska for ontology based data access. In *Proceedings of ESWC 2014 (Satellite Events)*, pages 157–161, 2013.

[9] S. Benferhat, D. Dubois, and H. Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proceedings of the 9th Conference on Uncertainty in Artificial intelligence (UAI'93)*, pages 411–419, 1993.

[10] P. Besnard and A. Hunter. *Elements of Argumentation.* The MIT Press, 2008.

[11] M. Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proc of AAAI*, 2012.

[12] D. Bouyssou, D. Dubois, M. Pirlot, and H. Prade. *Decision-making process – Concepts and Methods.* Wiley, 2009.

[13] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The dl-lite family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

[14] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Dl-lite: Tractable description logics for ontologies. In *American Association for Artificial Intelligence*, pages 602–607, 2005.

[15] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence Journal*, 171 (5-6):286–310, 2007.

[16] M. Chein and M. Mugnier. *Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs.* Springer, 2009.

[17] M. Chein, M.-L. Mugnier, and M. Croitoru. Visual reasoning with graph-based mechanisms: the good, the better and the best. *Knowledge Eng. Review*, 28(3):249–271, 2013.

[18] S. Colucci, T. D. Noia, A. Ragone, M. Ruta, U. Straccia, and E. Tinelli. *Semantic Web Information Management*, chapter 19 : Informative Top-k retrieval for advanced skill management, pages 449–476. R. De Virgilio et al., springer-verlag belin heidelberg edition, 2010.

[19] M. Croitoru, E. Compatangelo, and C. Mellish. Hierarchical knowledge integration using layered conceptual graphs. In *Proc of ICCS 2005*, pages 267–280.

[20] M. Croitoru, C. Croitoru, and P. Lewis. Bidflow: a new graph-based bidding language for combinatorial auctions. In *Proc. of ECAI 2008*, pages 871–872, 2008.

[21] M. Croitoru, B. Hu, S. Dashmapatra, P. Lewis, D. Dupplaw, and L. Xiao. A conceptual graph description of medical data for brain tumour classification. In *Proc of ICCS 2007*, 140-153.

[22] M. Croitoru and N. Oren. Information revelation strategies in abstract argument frameworks using graph based reasoning. In *Proc. of GKR @ IJCAI 2013*, pages 13–20, 2013.

[23] M. Croitoru, N. Oren, S. Miles, and M. Luck. Graphical norms via conceptual graphs. *Knowl.-Based Syst.*, 29:31–43, 2013.

[24] M. Croitoru and R. Rodriguez. Update operators for inconsistent query answering: A new point of view. In *Proc. of SUM 2014*, 2014. to appear.

[25] M. Croitoru and S. Rudolph. Exclusivity-based allocation of knowledge. In *Proc. of AAMAS 2012*, pages 1249–1250.

[26] M. Croitoru and R. Thomopoulos. Representing and reasoning about different viewpoints: An agronomy application. In *Proc of ICCS 2009*, pages 128–140.

[27] M. Croitoru and K. Van-Deemter. A conceptual graph approach for the generation of referring expressions. In *Proc. of IJCAI 2007*, pages 2456–2461, 2007.

[28] M. Croitoru and S. Vesic. What can argumentation do for inconsistent ontology query answering? In W. Liu, V. Subrahmanian, and J. Wijsen, editors, *Proceedings of SUM 2013*, pages 15–29. Springer, 2013.

[29] S. Destercke, P. Buche, and V. Guillard. A flexible bipolar querying approach with imprecise data and guaranteed results. *Fuzzy sets and Systems*, 169:51–64, 2011.

[30] J. Dix, S. Parsons, H. Prakken, and G. R. Simari. Research challenges for argumentation. *Computer Science - R&D*, 23(1):27–34, 2009.

[31] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-persons games. *Artificial Intelligence*, 77(2):321–357, 1995.

[32] D. Dupplaw, M. Croitoru, S. Dasmahapatra, A. Gibb, G. Gonzalez-Valez, M. Lurgi, B. Hu, P. Lewis, and A. Peet. A knowledge-rich distributed decision support framework: a case study for brain tumour diagnosis. *Knowledge Eng. Review*, 26(3):247–260, 2011.

[33] L. Guizol, M. Croitoru, and M. Leclere. Aggregation semantics for link validity. In *Proceedings of SGAI 2013*, pages 359–372, 2013.

[34] L. Guizol, O. Rousseaux, M. Croitoru, Y. Nicolas, and A. Le Provost. An analysis of the sudoc bibliographic knowledge base from a link validity viewpoint. In *Proc. of IPMU 2014*, 2014 (To appear).

[35] B. Haslhofer, E. M. Roochi, B. Schandl, and S. Zander. Europeana RDF store report. Technical report, University of Vienna, Vienna, Mar. 2011.

[36] P. Hayes. RDF Semantics. W3C Recommendation. http://www.w3.org/TR/2004/REC-rdf-mt-20040210/, 2004.

[37] H. Horst. Extending the RDFS Entailment Lemma. In *Proc. of the Third International Semantic Web ConferenceISWC'04*, volume 3298 of *LNCS*. Springer, 2004.

[38] B. Hu, M. Croitoru, S. Dasmahapatra, P. Lewis, and N. Shadbolt. Indexing ontologies with semantics-enhanced keywords. In *Proc of K-CAP*, pages 119–126, 2007.

[39] B. Hu, M. Croitoru, R. Roset, D. Dupplaw, M. Lurgi, S. Dasmahapatra, P. Lewis, J. Martínez-Miranda, and C. Saez. The healthagents ontology: knowledge representation in a distributed decision support system for brain tumours. *Knowledge Eng. Review*, 26(3):303–328, 2011.

[40] S. Ieong and Y. Shoham. Marginal contribution nets: A complact representation scheme for coalitional games. *ACM EC-06*, pages 170–179, 2006.

[41] A. C. Kakas, N. Maudet, and P. Moraitis. Layered strategies and protocols for argumentation-based agent interaction. In I. Rahwan, P. Moraitis, and C. Reed, editors, *Proceedings of the First International Workshop on Argumentation in Multi-agent Systems*, volume 3366 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 66–79, New York, 2004. Springer-Verlag.

[42] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. F. Savo. Inconsistency-tolerant semantics for description logics. In *Proceedings of the Fourth International Conference on Web Reasoning and Rule Systems*, RR'10, pages 103–117, Berlin, Heidelberg, 2010. Springer-Verlag.

[43] M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS 2002*, 2002.

[44] D. L. McGuinness and P. F. Patel-Schneider. Usability issues in knowledge representation systems. In *In Proc. of AAAI-98*, pages 608–614, 1998.

[45] S. Modgil and H. Prakken. A general account of argumentation with preferences. *Artificial Intelligence*, 195:361–397, 2013.

[46] N. Moreau, M. Leclere, and M. Croitoru. Distinguishing answers in conceptual graph knowledge bases. In *Proc. of ICCS 2009*, pages 233–246.

[47] N. Nisan. Bidding and allocations in combinatorial auctions. In *ACM Conference on Electronic Commerce (EC-2000)*, 2000.

[48] N. Oren, T. J. Norman, and A. D. Preece. Arguing with confidential information. In *Proc of ECAI 2006*, pages 280–284, 2006.

[49] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2011.

[50] I. Rahwan, F. Zablith, and C. Reed. Laying the foundations for a world wide argument web. *Artificial Intelligence*, 171(10-15):897–921, 2007.

[51] T. Rahwan and N. R. Jennings. Coalition structure generation: Dynamic programming meets anytime optimisation. In *In Proceedings of AAAI*, pages 156–161, 2008.

[52] T. Rahwan and N. R. Jennings. An improved dynamic programming algorithm for coalition structure generation. In *In Proceedings of AAMAS*, pages 1417–1420, 2008.

[53] T. Rahwan, T. Nguyen, T. Michalak, M. Polukarov, M. Croitoru, and N. Jennings. Coalitional games via network flows. In *Proc of IJCAI 2013*, 2013.

[54] T. Rahwan, S. D. Ramchurn, V. D. Dang, and N. R. Jennings. Near-optimal anytime coalition structure generation. In *In Proceedings of IJCAI*, pages 2365–2371, 2007.

[55] T. Rienstra, M. Thimm, and N. Oren. Opponent models with uncertainty for strategic argumentation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, page to appear, 2013.

[56] J. F. Sowa. Conceptual Graphs. *IBM Journal of Research and Development*, 20(4):336–375, 1976.

[57] J. F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.

[58] N. Tamani and M. Croitoru. Fuzzy argumentation system for decision support. In A. e. a. Laurent, editor, *Proceedings of IPMU 2014*, volume CCIS 442, pages 77–86. Springer, 2014.

[59] N. Tamani and M. Croitoru. A quantitative preference-based structured argumentation system for decision support. In *Proceedings of Fuzz-IEEE (to appear)*, 2014.

[60] N. Tamani, M. Croitoru, and P. Buche. Conflicting viewpoint relational database querying: An argumentation approach. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 1553–1554, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems.

[61] N. Tamani, M. Patricio, M. Croitoru, P. Buche, V. Guillard, C. Guillaume, and N. Gontard. Eco-efficient packaging material selection for fresh produce: Industrial session. In N. Hernandez, R. Jaesche, and M. Croitoru, editors, *Proceedings of the 2014 International Conference of Conceptual Structures ICCS 2014*, volume LNAI 8577, pages 305–310. Springer, 2014.

[62] R. Thomopoulos, M. Croitoru, and N. Tamani. Decision support for agrifood chains: A reverse engineering argumentation-based approach. *Ecological Informatics*, 2014 (To Appear).

[63] S. Vesic and L. van der Torre. Beyond maxi-consistent argumentation operators. In *13th European Conference on Logics in Artificial Intelligence (JELIA '12)*, pages 424–436, 2012.

[64] I. Vetsikas and M. Croitoru. How much should you pay for information? In *Proceedings of ICEBE 2013*, pages 44–49, 2013.

[65] L. Xiao, S. Dasmahapatra, P. Lewis, B. Hu, A. Peet, A. Gibb, D. Dupplaw, M. Croitoru, F. Estanyol, J. Martínez-Miranda, H. Gonzalez-Valez, and M. Lluch. The design and implementation of a novel security model for healthagents. *Knowledge Eng. Review*, 26(3):261–282, 2011.