Foundations of Artificial Intelligence

FAInt-07

Workshop at KI 2007

10th of Septermber, 2007

Organisers

Pascal Hitzler
Institut AIFB,
Universität Karlsruhe (TH)
phi@aifb.uni-karlsruhe.de

Thomas Roth-Berghofer
DFKI GmbH
Kaiserslautern
trb@dfki.uni-kl.de

Sebastian Rudolph
Institut AIFB,
Universität Karlsruhe (TH)
sru@aifb.uni-karlsruhe.de

1 Objectives

Within the course of the last 50 years, Artificial Intelligence has developed into a major field of research with a multitude of facets and application areas. While, in general, Artificial Intelligence research is driven by application needs, it is nevertheless a fact that foundational questions and theoretical insights have always been one of the driving forces behind its development. This includes the quest for realising intelligent behaviour in artificial systems as envisioned in the early days of AI research. But it also comprises biological inspirations, e.g., for robot design, artificial neural networks, or emergent intelligence, as well as logical underpinnings of automated deduction and knowledge representation.

Indeed, formal and foundational aspects of AI are being studied in many sub areas in order to serve application needs. It lies in the nature of such fundamental research that a critical mass of different formal perspectives can generate a cross-fertilization of ideas and applications. We, therefore, brought together researchers working on foundational aspects of Artificial Intelligence across different communities, in order to stimulate an exchange of ideas and methods between them.

The workshop is intended for researchers which contribute to the

- mathematical,
- logical,
- statistical,
- psychological,
- linguistic,
- cognitive,
- philosophical,
- biological,

and other foundations of AI research.

2 Programme

10th of September, 2007

14:00 – 15:20	**Keynote by Wolfgang Maass, TU Graz, Austria:** *The difficult search for intelligence in neural circuits.*
15:20 – 15:40	Kai-Uwe Kühnberger, Tonio Wandmacher, Angela Schwering, Ekaterina Ovchinnikova, Ulf Krumnack, Helmar Gust, Peter Geibel: *Modeling Human-Level Intelligence by Integrated Cognition in a Hybrid Architecture*
15:40 – 16:00	Sebastian Rudolph: *Relational Exploration – Reconciling Plato and Aristotle*
	coffee break
16:30 – 16:50	Matthias Knorr, Pascal Hitzler: *A Comparison of Disjunctive Well-founded Semantics*
16:50 – 17:10	Ander Altuna: *Imagining Contexts*
17:10 – 17:30	Pascal Hitzler, Andreas Eberhart: *Description Logic Programs: Normal Forms*

3 Committees

3.1 Workshop Organising Committee

PD Dr. Pascal Hitzler is assistant professor and project leader at the Institute for Applied Informatics and Formal Description Methods (AIFB) of the University of Karlsruhe in Germany. Beforehand, he graduated in Mathematics at Tübingen University, did his dissertation at the National University of Ireland in Cork, and worked as a postdoctoral researcher at the AI institute at TU Dresden. The focus of his research is foundations and applications of knowledge representation and reasoning, and his research record lists over 130 publications in such diverse areas as semantic web, neural-symbolic integration, knowledge representation and reasoning, lattice and domain theory, denotational semantics, and set-theoretic topology. He is currently writing a German textbook on Foundations of Semantic Web (jointly with Markus Krötzsch, Sebastian Rudolph, and York Sure). He leads AIFB's activities in the SmartWeb project funded by the German Ministry for Education and Research (BMBF), leads the ReaSem project funded by the German Research Council (DFG), contributes to several workpackages in the EU IST projects NeOn and X-Media, and in the EU Network of Excellence KnowledgeWeb. He is a steering committee member of the conference series Web Reasoning and Rules System (as vice-chair) and of the International Conference on Conceptual Structures, and also of the workshop series OWL – Experiences and Directions. He has also been an organiser of and teacher at international enhancement programmes for highly skilled students in Mathematics and Computer Science, and has served as an editor for several books in this area. For more information, please see http://www.pascal-hitzler.de.

Dr. Thomas Roth-Berghofer is Technical Director of the EU project NEPOMUK – The Social Semantic Desktop, Senior Researcher at the German Research Center for Artificial Intelligence DFKI GmbH, and Lecturer at the University of Kaiserslautern. His main research interests are in trustworthy personal knowledge-based systems with explanation capabilities. He initiated and organized several workshop series: on Modeling and Retrieval of Context, on Philosophy and Informatics, and on Explanation-aware Computing. He is program co-chair of the interdisciplinary conference CONTEXT'07, and he recently organized two other conferences, WM 2005 in Kaiserslautern, Germany, and ECCBR 2006 in Ölüdeniz/Fethiye, Turkey. For more information, please see http://thomas.roth-berghofer.de.

Dr. Sebastian Rudolph is a postdoctoral researcher at the Institute for Applied Informatics and Formal Description Methods (AIFB) of the University of Karlsruhe in Germany. He received his PhD in mathematics at the Technical University Dresden, Germany. His research interests comprise theoretical aspects of knowledge acquisition, representation and reasoning, complexity theory, lattice theory as well as linguistic and psychological dimensions of knowledge processing including e-learning. He is currently writing a German textbook on Foundations of Semantic Web (jointly with Pascal Hitzler, Markus Krötzsch, and York Sure). He is involved in AIFBs activities the ReaSem project funded by the German Research Council (DFG), in the EU IST project NeOn, and in the EU Network of Excellence KnowledgeWeb. He serves as Program Committee member for the International Conference on Formal Concept Analysis (ICFCA) and the International Conference on Conceptual Structures (ICCS) and as reviewer for Fundamenta Informaticae and numerous international conferences. He will be the local chair of the Second International Conference on Web Reasoning and Rule Systems (RR2008). Apart from university, he actively participates in various projects on classical vocal music both in choirs and as a solist. For more information, see http://www.sebastian-rudolph.de.

3.2 Programme Committee

- Jürgen Dix, TU Clausthal, Germany
- Thomas Eiter, TU Vienna, Austria
- Ulrich Furbach, University of Koblenz, Germany
- Dov Gabbay, King's College London, UK
- Bernhard Ganter, TU Dresden, Germany
- Ruth Hagengruber, Universität Paderborn, Germany
- Hans Kamp, University of Stuttgart, Germany
- Kai-Uwe Kühnberger, University of Osnabrück, Germany
- Markus Krötzsch, University of Karlsruhe (TH), Germany
- David Leake, Indiana University Bloomington, USA
- Benedikt Löwe, Universiteit van Amsterdam, The Netherlands
- Claudio Masolo, LOA CNR Trento, Italy
- Rainer Osswald, Fernuniversität Hagen, Germany
- Anthony K. Seda, University College Cork, Ireland
- Matthias Wendt, University of Leipzig, Germany
- Guo-Qiang Zhang, CWRU Cleveland, USA

Modeling Human-Level Intelligence by Integrated Cognition in a Hybrid Architecture

Kai-Uwe Kühnberger, Tonio Wandmacher, Angela Schwering,
Ekaterina Ovchinnikova, Ulf Krumnack, Helmar Gust, Peter Geibel

Institute of Cognitive Science, University of Osnabrück
D-49076 Osnabrück, Germany

Abstract. Various forms of reasoning, the profusion of knowledge, the gap between neuro-inspired approaches and conceptual representations, the problem of inconsistent data input, and the manifold of computational paradigms for solutions of these problems challenge AI models for higher cognitive abilities. We propose the I-Cog architecture as a step towards a solution for these problems. I-Cog is a modular system that is composed of a reasoning device based on analogical reasoning, a rewriting mechanism for the ontological knowledge base, and a neuro-symbolic interface for robust learning from noisy and inconsistent data.

1 Introduction

Since the origins of Artificial Intelligence – based on the fundamental work of Alan Turing [44], the first architecture for neural networks by McCulloch & Pitts [35], the development of higher programming languages like LISP [34], and finally the creation of AI as a discipline at the Dartmouth conference – artificial intelligence has (more or less) strongly been committed to interdisciplinary research and the modeling of higher cognitive abilities.[1] Several important achievements can be identified during the last 50 years with respect to modeling (or supporting) cognitive challenging tasks of humans: state-of-the-art computer programs beat world-class chess champions and intelligent programs support our daily life in various respects, for example, when driving a car, flying a plane, creating an engineer's CAD constructions, or searching the web for information.

Despite these apparent examples for the success of AI, there are severe problems of AI which can provocatively be described as follows: there is not even an idea of how human-level intelligence[2] (HLI) in the large can be achieved, taking into account the various forms of capabilities of human beings, for example,

[1] The term *higher cognitive abilities* can be identified with all forms of cognition which essentially include a deliberative aspect like reasoning, planning, game playing, learning, problem solving etc. In particular, purely reactive behaviors or behaviors which can be reduced to mere conditioning are not higher cognitive abilities.

[2] The term *human-level intelligence* is used in the sense of [7], namely as the problem to integrate many different types of representation formats, reasoning devices, computational paradigms etc., in order to approximate a breadth of intelligence usually ascribed to humans.

concerning reasoning, problem solving, learning, adapting, acting, using natural language etc. In the following we list three classes of such problems.

- The profusion of knowledge [7] and its constant updates.
- The variety of types of reasoning and computational paradigms for modeling human reasoning abilities (compare textbooks in AI).
- The gap between neuro-inspired learning approaches to cognition and symbolic representational approaches [4].

We think that these challenges are at the heart of achieving human-level intelligence, because of the following fundamental problem: The more fine-grained the methods are in developing tools for particular (and isolated) AI applications, the more we depart from the goal of achieving HLI and a unified model of higher cognition.[3] This paper aims to propose an architecture that provides a possible solution to model higher cognitive abilities by integrated cognition. We think that an integrated architecture can be considered as a device for achieving HLI.

This paper has the following structure. Section 2 discusses obvious problems in modeling a variety of higher cognitive abilities. Section 3 presents the I-Cog architecture consisting of a reasoning module, a background knowledge rewriting module, and a neuro-symbolic integration module. These modules interact in a non-trivial way described in Section 4. Finally Section 5 summarizes related work and Section 6 concludes the paper.

2 Problems for Modeling Higher Cognition in AI Systems

2.1 Knowledge

Knowledge representation is classically connected with coding entities in the environment by symbolic frameworks. Although such a straightforward logical representation is universal for most knowledge representation formalisms, and appropriate logical calculi ensure that many types of inferences for applications can be performed, there are non-trivial challenges for such a logical approach:

- *Problem of expressive strength:* For many applications first-order logic is simply too expressive. Examples are terminological hierarchies for ontological knowledge [43] or representations of STRIPS-like plans [11]. For other applications first-order logic is simply not expressive enough. Examples are forms of modeling common ground [6], or the usage of standard arithmetic.
- *Dynamic updates of background knowledge:* Whereas background knowledge is commonly considered to be static, human agents constantly update, modify, and learn new knowledge. Furthermore, they can overwrite existing knowledge easily without being threatened by inconsistencies.

[3] This claim clearly does not mean that other difficulties for modeling cognition in the large are simple or in some way straightforward to solve. Obviously challenges in computer vision, the modeling of autonomous agents and motor control, or natural language processing are also hard problems. But except for natural language processing, they concern lower cognitive abilities and are not considered here.

Due to the first challenge a profusion of representation formalisms emerged. Currently there is no idea how to reduce these paradigms significantly. The second challenge seems to be recently detected as a problem [24].

2.2 Reasoning

Reasoning abilities of humans can be classified into many types: Just to mention some of them, humans can perform deductions, inductions, and abductions. Furthermore, they are able to perform analogical reasoning steps, non-monotonic inferences, and frequency-based inferences (at least to a certain extent). Additionally, human agents are able to reason with vague and uncertain knowledge and they have the ability to associate certain situations with other similar situations. As a natural consequence of this variety of reasoning types, AI developed a tremendous number of frameworks for the computation of inferences. Unfortunately, these computational paradigms are not fully compatible with each other.

2.3 Neuro-Symbolic Integration

The gap between robust neural learning and symbolic representation formalisms is obvious: whereas symbolic theories are based on recursion and compositionality allowing the computation of (potentially) infinitely many meanings from a finite basis, such principles are not available for connectionist networks. On the other hand, neural networks have been proven to be a robust tool for learning from noisy data, pattern recognition, and handling vague knowledge – classical domains with which symbolic theories usually encounter problems. A potential solution for achieving HLI would require an integration of both approaches.

3 The Modules of the I-Cog Architecture

3.1 Analogical Reasoning

It is a crucial hypothesis of this paper that the establishment of analogical relations between a source and a target domain can be used for many forms of classical and non-classical reasoning tasks [14]. Examples for application domains of analogies are string domains [28], geometric figures [41], problem solving [1], naive physics [10], or metaphoric expressions [21]. Furthermore, analogies are a source of creativity [29] and a possibility to learn from sparse data [20]. Deductions and abductions are implicitly modeled in several systems (e.g. [13]).

In this paper, heuristic-driven theory projection (HDTP) will be used for sketching the expressive power of analogy making [21]. HDTP represents the source and target domains by sets of first-order formulas. The corresponding source theory Th_S and target theory Th_T are then generalized using an extension of anti-unification [40]. Here are the key elements of HDTP:

– Two formulas $p_1(a, b)$ and $p_2(a, c)$ can be anti-unified by $P(a, X)$, with substitutions $\Theta_1 = \{P \rightarrow p_1, X \rightarrow b\}$ and $\Theta_2 = \{P \rightarrow p_2, X \rightarrow c\}$.

Table 1. A simplified description of the algorithm HDTP-A omitting formal details. A precise specification of this algorithm can be found in [21].

Input: A theory Th_S of the source domain and a theory Th_T of the target domain represented in a many-sorted predicate logic language $\mathcal{L}$.
Output: A generalized theory Th_G such that the input theories Th_S and Th_T can be re-established by substitutions.
Selection and generalization of fact and rules.
Select an axiom from the target domain (according to a heuristic h).
Select an axiom from the source domain and construct a generalization (together with corresponding substitutions).
Optimize the generalization w.r.t. a given heuristic h'.
Update the generalized theory w.r.t. the result of this process.
Transfer (project) facts of the source domain to the target domain provided they are not generalized yet.
Test (using an oracle) whether the transfer is consistent with the target domain.

– A theorem prover allows the re-representation of formulas.
– Whole theories can be generalized, not only single terms or formulas.

The underlying algorithm HDTP-A is computing candidates of generalizations relative to Th_S and Th_T (Table 1): first, axioms are chosen from the target according to a heuristic ordering. For these axioms generalizations are computed relative to chosen axioms from the source (also governed by a heuristic). If every axiom from the target is generalized, the algorithm allows a creative transfer of knowledge from the source to the target (governed by the computed generalizations already obtained). We consider the analogy between a water-pipe system and an electric circuit in order to clarify the framework:

(M1) *Current is the water in the electric circuit.*

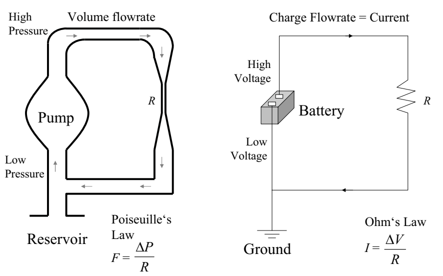

Fig. 1. The analogy between a water pipe system and an electric circuit in a diagrammatic representation. The Figure contains more information than is necessary for an interpretation of the metaphorical description (1).

Figure 1 depicts the situation represented by this analogy.[4] The analogy associates water-flow in a water pipe system with the flow of current in an electric circuit. An important new conceptualization about electricity can be learned by students using this analogy, namely that current is flowing in a circuit and that a battery has a function similar to a pump in the water pipe system.

We would like to achieve a modeling of metaphor (M1) using HDTP. Table 2 specifies the corresponding concepts in the target and the source domains that are associated with each other. The concepts of a closed water system and a closed electric system generalize to an abstract concept *closed*(A), where A is a variable. The terms *water* and *current* are associated

[4] The figure is based on http://hyperphysics.phy-astr.gsu.edu/hphys.html.

Table 2. Examples of corresponding concepts in the source and the target of the analogy between water-flow and the flow of current in an electric circuit. $ws1$ denotes an instance of a water pipe system and $es1$ an instance of an electric circuit.

Source	**Target**	**Generalization**
water_circuit(ws1,water,p1)	*electric_circuit(es1,current,b1)*	*Circuit(A,C,S1)*
closed(ws1)	*closed(es1)*	*closed(A)*
pump(p1)	*battery(b1)*	*Source(S1)*
pres(p1)>0→flow_in_c(w)	*pres(b1)>0→flow_in_c(c)*	*pres(S1)>0→flow_in_c(C)*
flow_in_circuit(water)	*flow_in_circuit(current)*	*flow_in_circuit(C)*

explicitly in the metaphoric expression (M1). From background knowledge a rule is available stating that if the pressure caused by the pump $p1$ in a water pipe system is different from 0, then water is flowing in the circuit (from high pressure to low pressure). This can be projected to the target side, inferring that due to the "pressure" of the battery $b1$ (realized by a positive voltage), current is flowing in the electric circuit. Hence, we end up with the conclusion (5 in Table 2) that current is flowing in the electric circuit (provided there is a "pressure" source). The substitutions Θ_1 and Θ_2 can be summarized as follows:

$$
\begin{array}{lrcl}
\Theta_1/\Theta_2: & A & \longrightarrow & ws1 \;/\; es1 \\
 & C & \longrightarrow & water \;/\; current \\
 & Source & \longrightarrow & pump \;/\; battery \\
 & S1 & \longrightarrow & p1 \;/\; b1 \\
 & Circuit & \longrightarrow & water\_circuit \;/\; electric\_circuit
\end{array}
$$

The following list sketches some reasons for the major claim of this subsection, namely that a large variety of human reasoning mechanisms can be modeled by analogies.

- Systems like HDTP allow the computation of analogical relations.
- Establishing analogical relations often requires the re-reprensentation of a domain. HDTP achieves this by a theorem prover that is included in the system and allows the application of rules (cf. Row 4 in Table 2).
- Learning generalizations is a first step towards an induction on given input data [20]. In the example, a new conceptualization of the target domain is learned.
- The fact that analogies are at most psychologically preferred, but never true or false, allows the extension of the system to model uncertainty.
- Non-monotonicity can be considered as a special case of a re-conceptualization of a given a domain very similar to a new conceptualization of a domain by an analogical inference.

3.2 Rewriting Ontological Background Knowledge

In Section 2, two major problems that are connected to knowledge representation and HLI were mentioned: first, the profusion of knowledge and second, the fact that human beings are able to dynamically adapt background knowledge on-the-fly. We sketch some ideas in this subsection, primarily addressing the second problem: we propose a rewriting system that is constantly adapting the ontological knowledge base (memory) focusing on the resolution of inconsistencies. Although the framework was developed primarily for text technological applications, the underlying logical basis is rather weak, and obviously not all types of inconsistencies can be automatically resolved, we think that proposals in this direction are crucial for achieving HLI.

Ontological knowledge is usually formalized within a logical framework, most importantly in the framework of Description Logics (DL) [2]. In the past years, a variety of successful systems have been developed that make use of markup standards based on DL with varying degrees of expressiveness.[5] However, the storage of ontological information within a logical framework has an undesirable side-effect: inconsistency problems can occur, because items of information may contradict each other, making the given ontology unsatisfiable and useless for reasoning purposes. Because HLI requires permanent updates of ontological knowledge, the problem of occurring inconsistencies becomes even more important. In this section, we sketch some ideas of how to address dynamic updates of ontologies leaving the problem of the profusion of knowledge aside.

Ontologies usually contain a terminological component and an assertion component. A description logic terminology consists of a set of terminological axioms defining concepts by formulas of the form $\forall x : C(x) \rightarrow D(x)$ or $\forall x : C(x) \leftrightarrow D(x)$, where C is a concept name and D is a concept description.[6] The assertion component mentioned above contains information about the assignment of the particular individuals to concepts and relations from the terminology. Axioms are interpreted model theoretically by an interpretation function mapping concept descriptions to subsets of the domain. A model of an ontology is an interpretation satisfying all axioms. An ontology is inconsistent if it does not have a model.

There are several possibilities why inconsistencies can occur in ontologies. In [24], structural inconsistencies, usage-defined inconsistencies, and logical inconsistencies are distinguished. The last type of inconsistency – potentially caused by dynamic updates of the knowledge base – is of particular interest in our context and is addressed by an automatic rewriting device allowing constant learning and updates of the ontological knowledge base. One aspect of logical inconsistency problems concerns polysemy: If an ontology is updated automatically, then it is hardly possible to distinguish between word senses. Suppose, the concept *tree* is declared to be a subconcept both of *plant* and of *data structure* (where *plant* and *data structure* are disjoint concepts). Both of these two interpretations of *tree* are correct, but it is still necessary to describe two different

[5] [2] provides an overview of different versions of description logics

[6] Compare [2] for an exhaustive definition of description logics.

concepts in the ontology with different identifiers (e.g. *TreePlant*, *TreeStructure*). Otherwise, the terminology remains unsatisfiable.

Another important aspect of logical inconsistency problems concerns generalization mistakes. Consider the following classical example:

Example 1 *Assume the following axioms are given:*

$\forall x : Bird(x) \rightarrow CanFly(x)$ $\quad \forall x : CanFly(x) \rightarrow CanMove(x)$
$\forall x : Canary(x) \rightarrow Bird(x)$ $\quad \forall x : Penguin(x) \rightarrow Bird(x) \wedge \neg CanFly(x)$

In Example 1, the statement *"birds can fly"* is too general. If an exception occurs (*penguin*), the ontology becomes unsatisfiable, since penguin is declared to be a bird, but it cannot fly. This type of inconsistency is the well-known problem of non-monotonicity, extensively discussed in the relevant AI literature.

The proposed approach – formally developed in [36], [37], and [38] – treats ontologies that are extended with additional axioms conflicting with the original knowledge base. Given a consistent ontology O (possibly empty) the procedure adds a new axiom A to O. If $O^+ = O \cup \{A\}$ is inconsistent, then the procedure tries to find a polysemy or an overgeneralization and repairs O^+.

We will illustrate the regeneralization of the overgeneralized concepts on the ontology in Example 2. Since the definition of the concept *Bird* is overgeneralized, it needs to be rewritten. We wish to retain as much information as possible in the ontology. The following solution is proposed:

Example 2 *Adapted ontology from Example 1:*
$\forall x : Bird(x) \rightarrow CanMove(x)$
$\forall x : CanFly(x) \rightarrow CanMove(x)$
$\forall x : Canary(x) \rightarrow FlyingBird(x)$
$\forall x : Penguin(x) \rightarrow Bird(x) \wedge \neg CanFly(x)$
$\forall x : FlyingBird(x) \rightarrow Bird(x) \wedge CanFly(x)$

We want to keep in the definition of the concept *Bird* (subsuming the unsatisfiable concept *Penguin*) a maximum of information that does not conflict with the definition of *Penguin*. The conflicting information is moved to the definition of the new concept *Flying bird*, which is declared to subsume all former subconcepts of *Bird* (such as *Canary* for example) except *Penguin*.

Our algorithm (cf. [36], [37], and [38] for a detailed description) detects problematic axioms that cause a contradiction, defines the type of contradiction (polysemy or overgeneralization) and automatically repairs the terminology by rewriting parts of the axioms that are responsible for the contradiction. Detected polysemous concepts are renamed and overgeneralized concepts are split into more general and more specific ones. This approach is knowledge preserving in the sense that it keeps as many entailments implied by the original terminology as possible.

The sketched solution for a constant adaptation process of background knowledge is a first step towards a general theory of dynamification and adaptation of background knowledge. The framework has been developed primarily for text

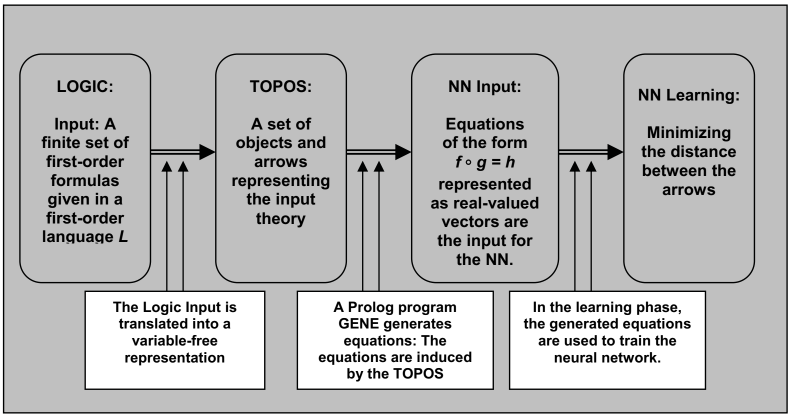

Fig. 2. The architecture for learning a first-order logical theory with neural networks.

technological applications. But the approach can straightforwardly be extended to a wider range of applications.[7]

3.3 Neuro-Symbolic Integration

In order to bridge the gap between symbolic and sub-symbolic approaches we sketch the theory presented in [19] and [22] based on the idea to translate first-order logical formulas into a variable-free representation in a topos [17]. A topos is a category theoretic structure consisting of objects *Obj* and arrows *Ar* having their domain and codomain in *Obj*. Certain construction principles allow to generate new arrows from old arrows. A fundamental theorem connects first-order logic and topos theory: a topos can be interpreted as a model of predicate logic [17]. The overall idea of learning symbolic theories with neural networks can be summarized as follows (compare also Figure 2):

- First, input data is given by a set of logical formulas (axioms and queries) relative to a given first-order logical language $\mathcal{L}$.
- Second, this set of formulas is translated into objects and arrows of a topos. The representation is variable-free and homogeneous, i.e. only objects and arrows are represented combined by the operation concatenation of arrows.
- Third, a PROLOG program generates equations in normal form $f \circ g = h$ identifying new arrows in the topos. This is possible because a topos allows several construction mechanisms.
- Last but not least, these equations are used as input for the training of a neural network. The network has a standard feedforward topology and learns by backpropagation: the network adapts the representations of arrows in such a way that the arrows representing "true" are approximating the arrow *true*.

[7] The crucial algorithms for resolving overgeneralization, undergeneralization, and polysemy problems, are implemented and prototypically tested in example domains [38].

The arrows *true* and *false* are the only hard-coded arrows, represented as $(1.0, 0.0, 0.0, \ldots 0.0)$ and $(0.0, 1.0, 0.0, \ldots 0.0)$ respectively.

Learning is possible, because the topos induces constructions that can be used for training the network. Clearly, infinitely many constructions are induced by the topos, but as it turns out a finite number is completely sufficient.

The details of the approach are rather complicated. We do not go into details here. The interested reader is referred to [19] and [22] for more information. The framework was tested with simple and also complex first-order logical theories.

4 The Integration of the Modules

4.1 A Hybrid Architecture for Higher Cognition

The three modules proposed in Section 3 – the neuro-symbolic integration module, the symbolic rewriting module, and the analogy reasoning module – attempt to learn a robust model of ontological background knowledge using a connectionist learning device, to dynamically rewrite ontologies on the symbolic level, and to perform various forms of reasoning, respectively. The task in this section is to integrate these modules into one single architecture called I-Cog (integrated cognition).

The integration of symbolic and sub-symbolic processes in a hybrid framework can be achieved, because the neuro-symbolic learning module is trained on symbolic data (i.e. on first-order logical expressions) and the fact that it learns a model of a logical theory. Although it is currently not possible to directly extract symbolic information from the neuro-symbolic learning device, an interaction and competition of the two modules can be implemented by querying both modules and evaluating their answers. Furthermore, both frameworks can interact with each other: queries of the rewriting module can be answered by the neuro-symbolic integration module. A similar remark holds for the integration of the analogy engine and the neuro-symbolic integration device.

Figure 3 depicts the overall architecture of the system. The architecture consists of the following modules (in the following paragraphs we will use the shortcuts *ORD* for the symbolic *Ontology Rewriting Device*, *NSLD* for the neural network-based *Neuro-Symbolic Learning Device*, *AE* for the *Analogy Engine*, and *CD* for the *Control Device*):

- The input may originate from various sources: input may be collected from resources based on structured data, unstructured data, or semi-structured data. The input needs to be available in an appropriate (subset) of a first-order language $\mathcal{L}$, in order to be in an appropriate format for the other modules. Therefore *ORD* generates appropriate logical formula from hypotheses.
- The input is used for feeding, updating, and training *ORD*.
- An important aspect is the interaction of *ORD* and *NSLD*: on the one hand, *ORD* trains *NSLD*, on the other hand *ORD* queries *NSLD*. Although *NSLD*

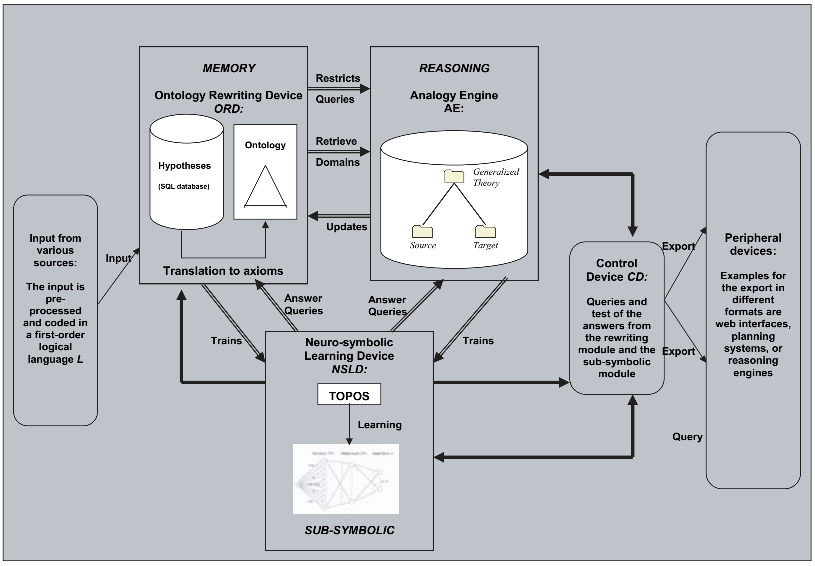

Fig. 3. The I-Cog architecture for an integration of the different modules. Whereas the modules *ORD* and *NSLD* are adapting new ontological axioms to an existing ontology, the analogy engine *AE* computes analogical relations based on background knowledge provided by the other two modules. The control device *CD* is intended to choose answers from all three modules.

can only give a tentative (or better approximate) answer in terms of a classification, this can improve the performance of the dynamic rewriting module in time-critical situations.

- With respect to the interaction of *AE* and *ORD*, ontological knowledge can naturally be used to constrain the computation of possible analogies [18]. Furthermore, newly generated analogies can be used to update and therefore rewrite background knowledge [23].
- Similarly to the relation between *ORD* and *NSLD*, *AE* is used to train *NSLD*, whereas query answering can be performed in the other direction.
- The control device *CD* of the two learning modules is intended to implement a competition of the feedback of the three modules with respect to queries. Feedback may be in accordance to each other or not. In the second case, the ranking of the corresponding hypotheses is decided by CD (see below).

We exemplify the interaction between *AE* and *ORD* in more detail (cf. [18], [23]): the establishment of an analogical relation of *AE*, if successful, provides a new conceptualization of the target domain. The example sketched in Subsection 3.1 results in a new conceptualization, where current is flowing in an electric circuit (triggered by a source). With respect to the ontological background knowledge *ORD* this means an update has to be performed, resulting in the introduction of a new (perhaps polysemous) concept, the update of a known concept using new relational constraints (flowing in an electric circuit), or even the generation of a conflict in the knowledge base (which has to be resolved). Additionally, the generalized theory of the anti-unification process introduces a new concept specifying an abstract circuit, where an entity is flowing caused by a source. On the other hand, *ORD* can be used to restrict possible analogical relations computed by *AE*: Due to the fact that *AE* can generalize arbitrary concepts, fact, rules etc., ontological knowledge may be used to restrict certain undesirable generalizations. For example, for a physics domain containing concepts like *time-point, real number, force, electric charge, pressure* etc., it is undesirable to generalize *force* with *real number* or *pressure* with *time-point*. But it is desirable to generalize different types of *force*, or different types of *pressure*. Such restrictions can be implemented by specifying an upper-level ontology in *ORD* which blocks certain (logically possible) generalizations.

A crucial problem of the presented approach concerns the control device *CD*. This module needs to assess possible answers of the three main modules and needs to implement a competition process. The natural way to realize such a control mechanism is to learn the behavior of the systems, based on certain heuristics. We exemplify possible situations with respect to *ORD* and *NSLD*: with respect to underdetermined situations, *ORD* is not able to answer queries, simply because the reasoning engine will not be able to prove anything without sufficient knowledge. In contrast to *ORD*, *NSLD* will be able to give an answer in any case. In such cases the usage of *NSLD* is clearly preferred by the heuristic. On the other hand, if *ORD* is able to prove a particular fact, for example, that a certain subsumption relation holds between two concepts A and B, then this result should be tentatively preferred by *CD* in comparison to the output of *NSLD*. In cases where time-critical reactions are necessary and *ORD* is not able to compute an answer in time, the natural heuristic would be to use *NSLD* instead. Finally, it could happen that the answers of *ORD* and *NSLD* are contradicting each other. In this case, *CD* cannot base the decision on *a priori* heuristics. A natural solution to this problem is to implement a reinforcement learning mechanism on *CD* itself, namely the learning of preferred choices (dependent on the particular domain) of the knowledge modules involved.

4.2 The Added-Value of a Hybrid Approach

The added-value of the overall architecture (as depicted in Figure 3) can be summarized as follows:

- The architecture is robust due to the fact that the trained neural network can give answers to queries even though noise might be contained in the training data.
- Even in time-critical situations the proposed framework is able to react and to provide relevant information, because the neural network can answer a query immediately without any processing time, although the symbolic rewriting module may be busy with computation tasks. This can be achieved by a heuristic governing the behavior of *CD* in cases of time-critical situations.

- The architecture gives a first idea how an interaction between a symbolic level and a sub-symbolic level of computation can be achieved. The crucial issue is the fact that *NSLD* is able to learn from highly structured training data on a neural level.
- The architecture is cognitively more plausible than pure symbolic or sub-symbolic approaches. Although the hard problem of cognitive science, namely how a one-to-one translation from the symbolic level to the corresponding neural correlate and vice versa can be defined is not resolved, at least a particular direction of communication between such devices can be achieved.

Besides the mentioned advantages of such an architecture for automatically learning and adapting ontologies, covering many aspects of different reasoning paradigms, and providing a hybrid architecture, there is the chance to integrate various forms of cognitive capacities into one framework that are often considered to be incompatible. Perhaps this provides an idea of how to bridge the obvious gap between symbolic and subsymbolic processes, as well as the corresponding differences in computing paradigms and capacities. Models of conceptual theories (in our case of logical theories) can be coded on the neural level in a trained neural network. Additionally, this is complemented by a symbolic representation of the semantic knowledge of the environment, allowing classical (and non-classical) deductions and reasoning processes. In total, we think that the proposed hybrid architecture seems to be cognitively more plausible than isolated approaches that are purely based on one computational reasoning mechanism and representation paradigm.

5 Related Work

Some application domains for analogical reasoning were already mentioned in Section 3. Concerning underlying methods for modeling analogies algebraic [29], graph-based [10], and similarity-based approaches [15] can be found.

A collection of approaches that aims at resolving inconsistencies in knowledge representation is related to non-monotonicity. Some examples are extensions by default sets [25] or by belief-revision processes [12]. In [9], inductive logic programming techniques are proposed to resolve ontological inconsistencies. A family of approaches is based on tracing techniques for detecting a set of axioms that are responsible for particular contradictions in an ontology [3], [30].

With respect to the problem of representing symbolic data structures with neural means, we mention as examples sign propagation [32], dynamic localist representations [5], tensor product representations [42], or holographic reduced representations [39]. Furthermore, researchers tried to solve the so-called inference problem: whereas symbolic approaches allow one to draw inferences from given representations, there is no neural correlate to this capacity. An example to solve this problem is described in [27] in which a logical deduction operator is approximated by a neural network. Another approach is [26], where category theoretic methods are used for neural constructions. In [8], tractable fragments of predicate logic are learned by connectionist networks.

Recently, some endeavor has been invested to approximate a solution to human-level intelligence. [7] proposes a so-called cognitive substrate in order to reduce higher cognition and the profusion of knowledge to a basis of low computational complexity. [13] propose to explain cognitive diversity of reasoning methods as a reduction to the well-known structure mapping theory [16]. Due to the combination of large knowledge bases, efficient retrieval, an analogy engine and learning modules, [13] is quite similar in spirit to the proposed architecture in this paper. Further approaches that resemble the integration idea presented here follow the tradition of cognitive architectures. Examples are the hybrid AMBR/DUAL model [31], which is modeling neuro-symbolic processing and analogical reasoning, the ICARUS architecture [33], which is focusing primarily on learning, or the NARS architecture [45], which is intended for integrating many different types of reasoning and representation formats.

6 Conclusions and Future Research

The paper proposes a hybrid architecture, based on analogical reasoning, an ontology rewriting device, and a module for neuro-symbolic integration, in order to model HLI. Although each module has been proven to be successfully applicable in theory and practice to the respective domains, many challenges remain open. Besides the fact that the overall architecture needs to be implemented and carefully evaluated, there are several theoretical questions that need to be addressed. One aspect concerns the control architecture, in particular, the question on which basis competing answers from the different modules are evaluated. Another issue concerns the interaction of the particular modules: for example, whereas the training of the *NSLD* module by *ORD* is more or less well-understood, the other direction, i.e. the input from *NSLD* to *ORD* is (at present) rather unclear. Consequently, it is currently only possible to query the neural network, because a direct extraction of symbolic knowledge from the trained network is an unsolved problem. Additionally, the problem of the profusion of knowledge and representation formalisms needs to be addressed. It may be a possibility to restrict ontological knowledge practically to hierarchical sortal restrictions that can be coded by relatively weak description logics, but in the long run, this is probably not sufficient. Last but not least, it would be desirable to add further devices to the system, e.g. planning systems and action formalisms.

The ultimate test scenario for the I-Cog architecture, as well as for HLI in general, would be a modified version of the Turing test: assume a robot operates an avatar in a virtual environment like "Second Life", where real humans operate their avatars, too. If a human cannot decide whether an avatar is controlled by a robot or a human, the robot shows HLI and higher cognition in the sense of this paper. It is essential that such systems are built for humans to interact with and not for robots. It is obvious that no isolated AI tool like a theorem prover, a knowledge base, a neural network, a planning system etc. is able to control the behavior of an avatar in a reasonable way in such a scenario. Although we do not claim that the presented architecture is sufficient to pass this type of "grand challenge", we believe that only integrated cognitive architectures like I-Cog will have a chance at all.

Acknowledgment

This work has been partially supported by the German Research Foundation (DFG) through the grants KU 1949/2-1 and MO 386/3-4.

References

1. Anderson, J., Thompson, R.: Use of analogy in a production system architecture, in: Similarity and analogical reasoning, editors: Vosniadou, Ortony, Cambridge (1989) 267–297.
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D. Patel-Schneider (eds.), P.: *Description Logic Handbook: Theory, Implementation and Applications.* Cambridge University Press (2003)
3. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning*, **14(1)** (1995) 149–180.
4. Bader, S., Hitzler, P.: Dimensions of neural-symbolic integration - a structured survey. In S. Artemov, H. Barringer, A. S. d'Avila Garcez, L. C. Lamb, J. Woods (eds.): We Will Show Them: Essays in Honour of Dov Gabbay vol. 1, 167-194. International Federation for Computational Logic, College Publications, 2005
5. Barnden, J. A.: Neural net implementation of complex symbol processing in a mental model approach to syllogistic reasoning. In Proceedings of the International Joint Conference on Artificial Intelligence (1989) 568–573.
6. Barwise, J.: On the Model Theory of Common Knowledge, in J. Barwise: The Situation in Logic, Cambridge 1990.
7. Cassimatis, N.: A Cognitive Substrate for Achieving Human-Level Intelligence, *AI Magazine* **27(2)** (2006) 45–56.
8. D'Àvila Garcez, A., Broda, K., Gabbay, D.: Neural-Symbolic Learning Systems: Foundations and Applications. Springer-Verlag (2002).
9. Fanizzi, N., Ferilli, S., Iannone, L., Palmisano, I., Semeraro, G.: Downward Refinement in the ALN Description Logic. *Proc. of the Fourth International Conference on Hybrid Intelligent Systems* (HIS'04) (2004) 68–73.
10. Falkenhainer, B., Forbus, K., Gentner, D.: The structure-mapping engine: Algorithm and example, Artificial Intelligence **41** (1989) 1–63.
11. Fikes, R., Nilson, N.: STRIPS: A new Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence* **2** (1971) 189-208.
12. Flouris, G., Plexousakis, D., Antoniou, G.: Updating DLs Using the AGM Theory: A Preliminary Study, in: Description Logics, 2006.
13. Forbus, K., Hinrichs, T.: Companion Cognitive Systems: A step towards human-level AI. *AI Magazine*, **27(2)** (2006) 83-95.
14. Gentner, D.: Why We're So Smart, in: D. Gentner & S. Goldin-Meadow: Language in mind: Advances in the study of language and thought, Cambridge MA: MIT Press (2003) 195–235.
15. Gentner, D.: The mechanisms of analogical learning, in: S. Vosniadou & A. Ortony (editors): *Similarity and Analogical Reasoning*, New York, Cambridge University Press.
16. Gentner, D.: Structure Mapping: A theoretical framework for analogy, *Cognitive Science* **7(2)** (1983) 155–170.
17. Goldblatt, R.: Topoi : The Categorial Analysis of Logic. Studies in Logic and the Foundations of Mathematics, **98**, North-Holland, Amsterdam (1979).
18. Gust, H., Kühnberger, K.-U., Schmid, U.: Ontological Aspects of Computing Analogies. Proceedings of the Sixth International Conference on Cognitive Modeling, Mahwah, NJ: Lawrence Erlbaum, pp. 350–351.
19. Gust, H., Kühnberger, K.-U.: Learning Symbolic Inferences with Neural Networks. In: Bara, B., Barsalou, L., Bucciarelli, M. (eds): CogSci 2005, XXVII Annual Conference of the Cognitive Science Society, Lawrence Erlbaum (2005) 875–880.
20. Gust, H., Kühnberger, K.-U.: Explaining Effective Learning by Analogical Reasoning, in: R. Sun, N. Miyake (eds.): CogSci/ICCS 2006: 28th Annual Conference of the Cognitive Science Society in coopera-tion with the 5th International Conference of the Cognitive Science (in the Asian-Pacific region), Lawrence Erlbaum (2006) 1417–1422.
21. Gust, H., Kühnberger, K.-U., Schmid, U.: Metaphors and Heuristic-Driven Theory Projection (HDTP), Theoretical Computer Science **354** (2006) 98-117.

22. Gust, H., Kühnberger, K.-U. and Geibel, P. 2007. Learning Models of Predicate Logical Theories with Neural Networks based on Topos Theory. In P. Hitzler and B. Hammer (eds.): Perspectives of Neuro-Symbolic Integration, LNCS Springer (to appear).
23. Gust, H., Kühnberger, K.-U., Schmid, U.: Ontologies as a Cue for the Metaphorical Meaning of Technical Concepts, in A. Schalley, D. Khlentzos (Eds.): Mental States: Evolution, Function, Nature, John Benjamins Publishing Company, Amsterdam, Philadelphia (to appear).
24. Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A framework for handling inconsistency in changing ontologies. *Proc.of the Fourth Internation Semantic Web Conference*, LNCS, Springer (2005).
25. Heymans, S., Vermeir, D.: A Defeasible Ontology Language, In Meersman, R. et al. (eds): Confederated International Conferences: CoopIS, DOA, and ODBASE 2002 Springer 1033-1046.
26. Healy, M., Caudell, T.: Neural Networks, Knowledge and Cognition: A Mathematical Semantic Model Based upon Category Theory. University of New Mexico, (2004), EECE-TR-04-020.
27. Hitzler, P., Hölldobler, S., Seda, A.: Logic programs and connectionist networks. Journal of Applied Logic, **2(3)** (2004) 245-272.
28. Hofstadter, D. and The Fluid Analogies Research Group: Fluid concepts and creative analogies. New York: Basic Books, 1995.
29. Indurkhya, B.: Metaphor and Cognition, Dordrecht, the Netherlands, Kluver, 1992.
30. Kalyanpur, A.: Debugging and Repair of OWL Ontologies. Ph.D. Dissertation, 2006.
31. Kokinov, B., Petrov, A.: Integrating Memory and Reasoning in Analogy-Making: The AMBR Model, in D. Gentner, K. Holyoak, B. Kokinov (eds.): The Analogical Mind. Perspectives from Cognitive Science, Cambridge Mass. 2001.
32. Lange, T., Dyer, M. G.: High-level inferencing in a connectionist network. Technical report UCLA-AI-89-12, 1989.
33. Langley, P.: Cognitive Architectures and General Intelligent Systems, *AI Magazine*, **27(2)** (2006) 33-44.
34. McCarthy, J.: Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I. Communications of the ACM **3(4)** (1960) 184-195.
35. McCulloch, W., Pitts, W.: A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, **5** (1943) 115-133.
36. Ovchinnikova, E., Kühnberger, K.-U.: Adaptive $\mathcal{ALE}$-TBox for Extending Terminological Knowledge. In A. Sattar, B. H. Kang (eds.): *Proceedings of the 19th ACS Australian Joint Conference on Artificial Intelligence*, LNAI 4304, Springer, pp. 1111-1115, 2006.
37. Ovchinnikova, E., Kühnberger, K.-U. 2006b. Aspects of Automatic Ontology Extension: Adapting and Regeneralizing Dynamic Updates, in M. Orgun & T. Meyer (eds.): *Advances in Ontologies 2006, Conferences in Research and Practice in Information Technology*, **72** (2006) 52-60.
38. Ovchinnikova, E., Wandmacher, T. and Kühnberger, K.-U. 2007. Solving Terminological Inconsistency Problems in Ontology Design. *International Journal of Interoperability in Business Information Systems*, **2(1)** (2007) 65–80.
39. Plate, T.: Distributed Representations and Nested Compositional Structure. PhD thesis (1994), University of Toronto.
40. Plotkin, G.: A note of inductive generalization, *Machine Intelligence* **5** (1970) 153-163.
41. Schwering, A., Krumnack, U., Kühnberger, K.-U., Gust, H.: Using Gestalt Principles to Compute Analogies of Geometric Figures, in: Proceedings of the 29th Annual Conference of the Cognitive Science Society (2007) (to appear).
42. Smolenski, P.: Tensor product variable binding and the representation of symbolic structures in connectionist systems. Artificial Intelligence,**46(1–2)** (1996) 159–216.
43. Staab, S., Studer, R. (eds.): Handbook of Ontologies. Springer 2004.
44. Turing, A.: Computing Machinery and Intelligence. *Mind* **59** (1950) 433-460.
45. Wang, P.: Rigid Flexibility: The Logic of Intelligence Springer 2006.

Relational Exploration – Reconciling Plato and Aristotle*

Sebastian Rudolph

Universitt Karlsruhe
Institut AIFB
rudolph@aifb.uni-karlsruhe.de

Abstract. We provide an interactive method for knowledge acquisition combining approaches from description logic and formal concept analysis. Based on present data, hypothetical rules are formulated and checked against a description logic theory. We propose an abstract framework (Logical Domain Exploration) for this kind of exploration technique before presenting a concrete instantiation: Relational Exploration. We give a completeness result and provide an overview about some application fields for our approach: machine learning, data mining, and ontology engineering.

1 Introduction

A plethora of research fields is concerned with the question of finding specifications for a given domain. Research areas like machine learning, frequent pattern discovery, and data mining in general aim at extracting these description on the basis of (examplary or complete) data sets – following the Aristotelian paradigm, that every conceptualization has to start from entities actually present. Other approaches intend to deduce these specifications from pre-specified theories – being somehow more Platonic by assuming the primacy of abstract ideas. The latter is the usual *modus operandi* e.g. in description logic or theorem proving.

We reconcile these two antagonistic approaches by combining techniques from two fields of knowledge representation: description logic (DL) and formal concept analysis (FCA).

In our work, we use DL formalisms for defining FCA attributes and FCA exploration techniques to obtain or refine DL knowledge representation specifications. More generally, DL exploits FCA techniques for interactive knowledge acquisition and FCA benefits from DL in terms of expressing relational knowledge.

In most cases, the process of conceptually specifying a domain cannot dispense of human contribution. However, although all information needed in order to describe a domain is in general implicitly present in an expert's knowledge, the process of explicit formal specification may nevertheless be tedious and overstraining. Moreover, it might remain unclear whether a specification is complete, i.e., whether it covers all valid statements about the domain that can be expressed in the chosen specification language.

* This work has been supported by the European Commission under contract IST-2006-027595 NeOn, and by the Deutsche Forschungsgemeinschaft (DFG) under the ReaSem project.

Hence, we provide a method – called Relational Exploration (RE) – that organizes and structures the specification process by successively asking single questions to the domain expert in a way which minimizes the expert's effort (in particular, it does not ask redundant questions) and guarantees that the resulting specification will be complete in the sense stated above. To present our work, which generalize the results from [1] and [2], we will proceed as follows: Section 2 provides a general framework for this kind of procedure, called Logical Domain Exploration. In Section 3, we shortly sketch the FCA basics necessary for our work and give an overview about attribute exploration. Section 4 introduces the notions from description logics needed in this work. In Section 5, we establish the correspondence between DL models and certain formal contexts, which enables us to apply FCA to DL. In Section 6, the RE algorithm is described in detail. Section 7 shows certain completeness properties of the knowledge acquired via RE. Section 9 displays direction for further work. In Section 8, we discuss our results and consider in which fields the presented technique could be applied.

2 The Epistemic Framework: Logical Domain Exploration

Before engaging into the technical details, we sketch the overall setting for our approach, which helps conveying the underlying idea and identifying the contributing components. Doing this on an abstract level, we also give an opportunity to relate alternative approaches. This framework will be called Logical Domain Exploration.

Let Δ be the considered domain of interest the elements of which we will call (DOMAIN) INDIVIDUALS. Let $\mathcal{L}$ be a language the elements of which are called FORMULAE. We write $\Delta \models \varphi$ in order to state for a formula φ that it is valid in the domain. Moreover let the setting be well-behaved in the way that whenever $\Delta \models \varphi$ is *not* true, there is a finite individual set $\Gamma \subseteq \Delta$ witnessing this (we then write $\Gamma \dagger \varphi$ and say Γ SPOILS φ).

- The EXPERT is supposed to be "omniscient" wrt. the described domain and thus able to answer any question about it. In particular, he knows for all $\varphi \in \mathcal{L}$ and $\Gamma \subseteq \Delta$ whether $\Delta \models \varphi$ and whether $\Gamma \dagger \varphi$. Mostly, a human or a group of humans will take the role of the expert.
- The TERMINOLOGY consists of a theory $Th \subseteq \mathcal{L}^{\mathcal{T}}$ about the domain consisting of axioms in some language $\mathcal{L}^{\mathcal{T}} \supseteq \mathcal{L}$ and a reasoning functionality, i.e. for any statement $\varphi \in \mathcal{L}^{\mathcal{T}}$ it can be decided whether Th entails φ.[1]
- The DATA consists of a set of known or recorded individuals $\mathcal{D} \subseteq \Delta$ and is endowed with a special querying capability, i.e., a procedure providing for any $\varphi \in \mathcal{L}$ a set $\Gamma \subseteq \mathcal{D}$ with $\Gamma \dagger \varphi$ if there exists one.
- The SCHEDULER can be conceived as an automated procedure initiating and coordinating the "information flow". It links the other system components by asking questions, processing answers, and assuring that in the end all knowledge is acquired to quickly decide for any $\varphi \in \mathcal{L}$ whether $\Delta \models \varphi$.

[1] Hereby, entailment is as usual defined in a model-theoretic way: Th is said to entail φ if any domain Δ' wherein all formulae of Th are valid also satisfies $\Delta' \models \varphi$.

The system will operate as follows: We start with a (correct but in general incomplete) terminological theory $Th \subset \{\psi \in \mathcal{L}^{\mathcal{T}} \mid \Delta \models \psi\}$ and data $\mathcal{D} \subseteq \Delta$. The scheduler now comes up with hypothetical formulae. Every such hypothetical formula $\varphi \in \mathcal{L}$ is passed both to the terminology and the data. The reasoning service of the terminology component checks whether φ is entailed by Th. The data is queried for a spoiler of φ. Since – due to the starting conditions – the theory is consistent with the data, we get three disjoint possible results:

– φ is entailed by Th. In this case, φ is valid in Δ, which will be responded to the scheduler.
– $\Gamma \in \mathcal{D}$ spoils φ. Then, φ is not valid in Δ and the scheduler will be provided with this negative answer.
– Neither of the previous cases occurs. Then, the current specification leaves room for either possibility and the domain expert will have to be asked this about φ's validity in question. If he confirms the validity of φ in Δ, it will be added to Th. If he denies it, he has to provide a spoiler Γ for φ, which is then added to the data.

Note that querying the data and questioning the terminology can be done in either order or even in parallel. After finishing the procedure every formula $\varphi \in \mathcal{L}$ will either be a consequence of the resulting (updated) terminology or can be excluded via a spoiler present in the data (updated) data. The distinction between $\mathcal{L}$ (the EXPLORATION LANGUAGE) and $\mathcal{L}^{\mathcal{T}}$ (the TERMINOLOGICAL LANGUAGE) is motivated by the assumption that in most cases not all terminologically expressible axioms will be of interest but only those of a certain shape.

In the next chapters, we come down to an instance for the previously described framework for logical domain exploration: Relational Exploration.

3 Formal Concept Analysis

In our instantiation, the scheduler's task will be carried out by an extension of the attribute exploration algorithm well established in FCA. This necessitates to briefly introduce some basic FCA notions. We mainly follow the notation introduced in [3] being *the* reference for FCA theory.

The basic notion FCA is built on is that of a formal context. It is a common claim in FCA that any kind of grounded data can be represented in this way.

Definition 1. *A* FORMAL CONTEXT $\mathbb{K}$ *is a triple* (G, M, I) *with an arbitrary set* G *(called* OBJECTS*), an arbitrary set* M *(called* ATTRIBUTES*), and a relation* $I \subseteq G \times M$ *(called* INCIDENCE RELATION*). We read* gIm *as: "object* g *has attribute* m*." Furthermore, let* $g^I := \{m \mid gIm\}$*.*

The central means of expressing knowledge in FCA is via implications. Thus, in terms of the general framework from Section 2 the underlying language consists of implications on a fixed attribute set of atomic propositions.

Definition 2. *Let* M *be an arbitrary set. An* IMPLICATION *on* M *is a pair* (A, B) *with* $A, B \subseteq M$*. To support intuition, we write* $A \rightarrow B$ *instead of* (A, B)*.* $A \rightarrow B$ HOLDS *in* *a formal context* $\mathbb{K} = (G, M, I)$*, if for all* $g \in G$ *we have that* $A \subseteq g^I$ *implies* $B \subseteq g^I$*. We then write* $\mathbb{K} \models A \rightarrow B$*.*

For $C \subseteq M$ *and a set* $\mathfrak{I}$ *of implications on* M*, let* $C^{\mathfrak{I}}$ *denote the smallest set with* $C \subseteq C^{\mathfrak{I}}$ *that additionally fulfills*

$$A \subseteq C^I \quad \textit{implies} \quad B \subseteq C^I$$

for every implication $A \rightarrow B$ *in* $\mathfrak{I}$*.*[2] *If* $C = C^{\mathfrak{I}}$*, we call* C $\mathfrak{I}$-CLOSED*. We say* $\mathfrak{I}$ ENTAILS $A \rightarrow B$ *if* $B \subseteq A^{\mathfrak{I}}$*.*[3]

An implication set $\mathfrak{I}$ *will be called* NON-REDUNDANT*, if for any* $(A \rightarrow B) \in \mathfrak{I}$ *we have that* $B \not\subseteq A^{\mathfrak{I} \setminus \{A \rightarrow B\}}$*.*

An implication set $\mathfrak{I}$ *of a context* $\mathbb{K}$ *will be called* COMPLETE*, if every implication* $A \rightarrow B$ *holding in* $\mathbb{K}$ *is entailed by* $\mathfrak{I}$*.*

$\mathfrak{I}$ *will be called an* IMPLICATION BASE *of a formal context* $\mathbb{K}$ *if it is non-redundant and complete.*

Note that implication entailment is decidable in linear time ([4]). Therefore, knowing a domain's implication base allows fast handling of its whole implicational theory. Moreover, for every formal context, there exists a canonical implication base ([5]).

The attribute exploration algorithm our work is based on was introduced in [6]. Due to space reasons, we omit to display it in detail and refer the reader to the literature.

Essentially, the following happens: the domain to explore is formalized as a formal context $\mathbb{K} = (U, M, I)$. Usually, it is not known completely in advance. However, possibly, some entities of the universe $g \in U$ are already known, as well as their associated attributes g^I.

The algorithm now starts presenting questions of the form

"Does the implication $A \rightarrow B$ hold in the context $\mathbb{K} = (U, M, I)$?"

to the human expert. The expert might confirm this. In this case, $A \rightarrow B$ is archived as part of $\mathbb{K}$'s implicational base $\mathfrak{IB}$. The other case would be that $A \rightarrow B$ does not hold in (U, M, I). But then, there must exist a $g \in U$ with $A \in g^I$ and $B \notin g^I$. The expert is asked to input this g and g^I.[4] The procedure terminates when the implicational knowledge of the universe is completely acquired, i.e., the implications of the formal context built from the entered counterexamples coincide with those entailed by $\mathfrak{IB}$.

In the approach presented here, we will exploit the capability of attribute exploration to efficiently determine an implicational theory. Notwithstanding, we extend the underlying language[5] from purely propositional to certain DL expressions being introduced in the next section.

[2] Note, that this is well-defined, since the mentioned properties are closed wrt. intersection.

[3] Actually, this is a syntactic shortcut. Yet, it can be easily seen that this coincides with the usual entailment notion.

[4] Referring to the general framework we mention that in this special case the spoiler (called counterexample) is always a singleton set: $\{g\} \dagger A \rightarrow B$.

[5] There exist already other language extensions, e.g. to Horn-logic with a bounded variable set, see [7].

4 Description Logic

We recall basic notions from DL, following (and recommending for further reading) [8].

Unlike the way DL is normally conceived, we use DL expressions to describe or specify *one particular, fixed* domain.

Thus, we will start our considerations by formally defining the kind of relational structure that we want to "talk about."

Definition 3. *An* INTERPRETATION *for a set* $\mathcal{A}$ *of* (PRIMITIVE) CLASS NAMES *and a set* $\mathcal{R}$ *of* ROLE NAMES *is a pair* $\mathcal{I} = (\Delta^{\mathcal{I}}, (.)^{\mathcal{I}})$ *where* $\Delta^{\mathcal{I}}$ *is some set and* $(.)^{\mathcal{I}}$ *is a function mapping class names to subsets of* $\Delta^{\mathcal{I}}$ *and role names to subsets of* $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Verbally, for some primitive class name A, $\mathrm{A}^{\mathcal{I}}$ provides all members of that class and for some role name R, $\mathrm{R}^{\mathcal{I}}$ yields all ordered pairs "connected" by that role.

The DL languages introduced here provide constructors for defining new concept descriptions out of the primitive ones. Table 1 shows those constructors, their interpretation (as usual defined recursively), and their availabilities in the description logics considered here.

	name	interpretation	$\mathcal{FL}_0$	$\mathcal{EL}$	$\mathcal{FLE}$	$\mathcal{ALE}$
A	atomic concept	$\mathrm{A}^{\mathcal{I}}$	×	×	×	×
$\top$	universal concept	$\Delta^{\mathcal{I}}$	×	×	×	×
$\bot$	bottom concept	$\emptyset$	×	×	×	×
$\neg\mathrm{A}$	atomic negation	$\Delta^{\mathcal{I}} \setminus \mathrm{A}^{\mathcal{I}}$				×
$\mathrm{C} \sqcap \mathrm{D}$	conjunction	$\mathrm{C}^{\mathcal{I}} \cap \mathrm{D}^{\mathcal{I}}$	×	×	×	×
$\forall\mathrm{R.C}$	value restriction	$\{\delta \mid \forall\epsilon : (\delta, \epsilon) \in \mathrm{R}^{\mathcal{I}} \rightarrow \epsilon \in \mathrm{C}^{\mathcal{I}}\}$	×		×	×
$\exists\mathrm{R.C}$	existential quantification	$\{\delta \mid \exists\epsilon : (\delta, \epsilon) \in \mathrm{R}^{\mathcal{I}} \wedge \epsilon \in \mathrm{C}^{\mathcal{I}}\}$		×	×	×

Table 1. syntax and semantics of the DLs considered in this paper

In the sequel, we will in general speak of a description logic $\mathcal{DL}$ if the presented result or definition refers to any $\mathcal{DL} \in \{\mathcal{FL}_0, \mathcal{EL}, \mathcal{FLE}, \mathcal{ALE}\}$.

Definition 4. *Let* $\mathcal{I}$ *be an interpretation and* C, D *be* $\mathcal{DL}$ *concept descriptions. We say* C IS SUBSUMED BY D *in* $\mathcal{I}$ *(written:* $\mathrm{C} \sqsubseteq_{\mathcal{I}} \mathrm{D}$*) if* $\mathrm{C}^{\mathcal{I}} \subseteq \mathrm{D}^{\mathcal{I}}$. *This kind of subsumption statements is also called* GENERAL CONCEPT INCLUSION AXIOM *(GCI). Moreover, we say* C *and* D *are* EQUIVALENT *in* $\mathcal{I}$ *(written:* $\mathrm{C} \equiv_{\mathcal{I}} \mathrm{D}$*) if* $\mathrm{C}^{\mathcal{I}} = \mathrm{D}^{\mathcal{I}}$.

5 Subsumptions as Implications

Combinations of FCA and DL have already been described in several publications, e.g. in [9], [10], and [11]. Our approach is motivated by [9] insofar as we use the same way of transferring a DL setting into a formal context by considering the domain individuals as objects and DL concept expressions as attributes.

Definition 5. *Given an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, (.)^{\mathcal{I}})$ and a set M of $\mathcal{DL}$ concept descriptions, we define the corresponding $\mathcal{DL}$-CONTEXT*

$$\mathbb{K}^{\mathcal{I}}(M) := (\Delta^{\mathcal{I}}, M, I)$$

where $\delta I C :\Longleftrightarrow \delta \in \mathrm{C}^{\mathcal{I}}$, for all $\delta \in \Delta^{\mathcal{I}}$ and $\mathrm{C} \in M$.

The observation in the next theorem – though easy to see – is crucial for applying attribute exploration for the intended purpose.

Theorem 1. *Let $\mathfrak{I}$ be an arbitrary interpretation and $\mathbb{K}^{\mathcal{I}}(M)$ a corresponding $\mathcal{DL}$-context. Then for finite $\mathcal{C}, \mathcal{D} \subseteq M$, the implication*

$$\mathcal{C} \to \mathcal{D}$$

holds in $\mathbb{K}^{\mathcal{I}}(M)$ if and only if [6]

$$\bigsqcap \mathcal{C} \sqsubseteq_{\mathcal{I}} \bigsqcap \mathcal{D}.$$

In the sequel, we will exploit this correspondence in the following way: employing the FCA exploration method allows us to collect all information that is valid in a (not explicitly given) interpretation and can be expressed by $\mathcal{DL}$ subsumptions with restricted maximal role depth[7].

6 The Relational Exploration Algorithm

The algorithm we present here is an iterative one. In each step the maximal role depth of the considered $\mathcal{DL}$ concept descriptions will be incremented by one. In each step, the results from previous steps will be exploited in several ways.

In the worst case, the time needed for the attribute exploration algorithm is exponential with respect to the number of attributes. Thus, it is essential to see how the set of attributes can be reduced without losing completeness.

The first exploration step is aimed at clarifying the implicational interdependencies of $\mathcal{DL}$ concept descriptions with quantifier depth 0. Therefore, no roles occur yet and we start with

$$M_0 := \begin{cases} \{\bot\} \cup \{\mathrm{A}, \neg\mathrm{A} \mid \mathrm{A} \in \mathcal{A}\} & \text{if } \mathcal{DL} = \mathcal{ALE} \\ \{\bot\} \cup \mathcal{A} & \text{otherwise.} \end{cases}$$

In the actual exploration step – the interview-like procedure described in Section 3 – takes place with respect to the context $\mathbb{K}_i^{\mathcal{I}} = \mathbb{K}_i^{\mathcal{I}}(M_i)$. Every hypothetical implication $\mathcal{A} \to \mathcal{B}$ for $\mathcal{A}, \mathcal{B} \subseteq M_i$ presented to the expert has to be interpreted as question about the validity of $\bigsqcap \mathcal{A} \sqsubseteq_{\mathcal{I}} \bigsqcap \mathcal{B}$, and will be passed to the "answering components" as described in Section 2.

[6] We use $\bigsqcap\{\mathrm{C}_1, \ldots, \mathrm{C}_n\}$ to abbreviate $\mathrm{C}_1 \sqcap \ldots \sqcap \mathrm{C}_n$. Moreover, let $\bigsqcap\{\mathrm{C}\} := \mathrm{C}$ and $\bigsqcap\{\emptyset\} := \top$.

[7] As usual, a concept description's role depth indicates how deep quantifiers are nested in it.

The exploration step ends up with an implication base $\mathfrak{IB}_i$, which – as we will prove in Section 7 – represents the complete subsumptional knowledge of the considered domain up to role depth i.

For the next exploration step – incrementing the considered role depth – we have to stipulate the next attribute set M_{i+1}. In case of the concept descriptions preceded by an existential quantification, the previously acquired implication base $\mathfrak{IB}_i$ can be used to reduce the number of attributes to consider, keeping the completeness property.

$$\begin{aligned} M_{i+1} := M_0 & \\ & \cup\{\forall \mathrm{R.C} \mid \mathrm{R} \in \mathcal{R}, \mathrm{C} \in M_i\} \\ & \cup\{\exists \mathrm{R}. \textstyle\bigsqcap \mathcal{C} \mid \mathrm{R} \in \mathcal{R}, \mathcal{C} = \mathcal{C}^{\mathfrak{IB}_i}, \bot \notin \mathcal{C}\} \end{aligned}$$

If considering $\mathcal{EL}$ or $\mathcal{FL}_0$, simply discard the second resp. third line from the definition. In addition to minimizing the cardinality of M_{i+1}, we can accelerate the exploration process by providing implications on M_{i+1} that are already known to be valid. These are the following:

– $\{\bot\} \rightarrow M_{i+1}$,
– $\{(\mathrm{A})_{i+1} \mid \mathrm{A} \in \mathcal{A}\} \rightarrow \{(\mathrm{B})_{i+1} \mid \mathrm{B} \in \mathcal{B}\}$ for every implication $\mathcal{A} \rightarrow \mathcal{B}$ from $\mathfrak{IB}_i$ (i.e., translate[8] all known implications from M_i into M_{i+1}),
– $\{\forall \mathrm{R.A} \mid \mathrm{A} \in \mathcal{A}\} \rightarrow \{\forall \mathrm{R.B} \mid \mathrm{B} \in \mathcal{B}\}$ for every implication $\mathcal{A} \rightarrow \mathcal{B}$ from $\mathfrak{IB}_i$,
– $\{\exists \mathrm{R}. \bigsqcap \mathcal{A}\} \rightarrow \{\exists \mathrm{R}. \bigsqcap \mathcal{B}\}$ for all $\mathfrak{IB}_i$-closed sets $\mathcal{A}, \mathcal{B} \subseteq M_i$ with $\mathcal{A} \subsetneq \mathcal{B}$ where there is no $\mathfrak{IB}_i$-closed set $\mathcal{C}$ with $\mathcal{A} \subsetneq \mathcal{C} \subsetneq \mathcal{B}$, and
– $\{\exists \mathrm{R}. \bigsqcap \mathcal{A}, \forall \mathrm{R.A}\} \rightarrow \{\exists \mathrm{R}. \bigsqcap (\mathcal{A} \cup \{\mathrm{A}\})^{\mathfrak{IB}_i}\}$ for every $\mathfrak{IB}_i$-closed set $\mathcal{A} \subseteq M_i \setminus \{\mathrm{A}\}$ and every concept description $\mathrm{A} \in M_i$.

With this attribute set M_{i+1} and the a-priori implications we start the next exploration step.

In theory, this procedure can be continued to arbitrary role depths. In some but not in all cases a complete acquisition of knowledge can be achieved. Yet in practice, with increasing role depth, the questions brought up by the exploration procedure will be increasingly numerous as well as less intuitional and thus difficult to answer for a human expert. So in many cases, one will restrict to small role depths.

7 Verification of the Algorithm

Let $\mathcal{DL}_i$ denote the set of all $\mathcal{DL}$ concept descriptions with maximal role depth i. Now we show a way how the validity of any subsumption on $\mathcal{DL}_i$ can be checked by using just the attribute sets $M_0, \ldots, M_i$ as well as the corresponding implication bases $\mathfrak{IB}_0, \ldots, \mathfrak{IB}_i$ on those sets. First, we will define functions that provide for any concept description $\mathrm{C} \in \mathcal{DL}_i$ a set of attributes $\mathcal{C} \subseteq M_i$ such that $C \equiv_{\mathcal{I}} \bigsqcap \mathcal{C}$. The following definitions and proofs are carried out for $\mathcal{ALE}$ but can be easily adapted to the other DLs by simply removing the irrelevant parts.

[8] We will formally define and justify this translation $(.)_{i+1}$ in Section 7.

Definition 6. *Let $\mathcal{I}$ be an interpretation and the corresponding sequences $(M_i), (\mathbb{K}_i^{\mathcal{I}})$ defined as above. Given the according sequence $\mathfrak{IB}_0, \ldots, \mathfrak{IB}_n$ of implication bases, we define a sequence of functions $\tau_i : \mathcal{DL}_i \to \mathcal{P}(\mathcal{DL}_i)$ in a recursive way:*

$$\begin{aligned}
\tau_i(\mathrm{C}) &= \{\mathrm{C}\} \textit{ for } \mathrm{C} \in M_0 \\
\tau_i(\textstyle\bigsqcap \mathcal{C}) &= \textstyle\bigcup\{\tau_i(\mathrm{C}) \mid \mathrm{C} \in \mathcal{C}\} \\
\tau_i(\forall \mathrm{R.C}) &= \{\forall \mathrm{R.D} \mid \mathrm{D} \in \tau_{i-1}(\mathrm{C})\} \\
\tau_i(\exists \mathrm{R.C}) &= \begin{cases} \{\bot\} & \textit{if } \bot \in (\tau_{i-1}(\mathrm{C}))^{\mathfrak{IB}_{i-1}}, \\ \{\exists \mathrm{R}.\bigsqcap(\tau_{i-1}(\mathrm{C}))^{\mathfrak{IB}_{i-1}}\} & \textit{otherwise.} \end{cases}
\end{aligned}$$

Moreover, let $\bar{\tau}_i(\mathrm{C}) := (\tau_i(\mathrm{C}))^{\mathfrak{IB}_i}$ for all $\mathrm{C} \in \mathcal{DL}_i$.

Note that by this definition, we also have $\bar{\tau}_i(\top) = \bar{\tau}_i(\bigsqcap \emptyset) = \emptyset^{\mathfrak{IB}_i}$. Next, we have to show that the functions just defined behave in the desired way. The following lemma ensures that $\bar{\tau}_i$ and τ_i indeed map to M_i.

Lemma 1. *Suppose $\mathrm{C} \in \mathcal{DL}_i$. Then we have $\tau_i(\mathrm{C}) \subseteq M_i$ and $\bar{\tau}_i(\mathrm{C}) \subseteq M_i$.*

Proof. Obviously, $\bar{\tau}_i(\mathrm{C}) \subseteq M_i$ whenever $\tau_i(\mathrm{C}) \subseteq M_i$. We show the latter by induction on the role depth considering four cases:

- $\mathrm{C} \in \{\mathrm{A}, \neg\mathrm{A} \mid \mathrm{A} \in \mathcal{A}\} \cup \{\bot\}$. Then by definition $\mathrm{C} \in M_i$.
- $\mathrm{C} = \exists\mathrm{R.D}$. If $\bot \in \bar{\tau}_{i-1}(\mathrm{D})$, we get $\tau_i(\mathrm{C}) = \tau_i(\exists\mathrm{R.D}) = \{\bot\} \subseteq M_i$.
  Now suppose $\bot \notin \bar{\tau}_{i-1}(\mathrm{D})$. As immediate consequence of the induction hypothesis we have $\bar{\tau}_{i-1}(\mathrm{D}) \subseteq M_{i-1}$. Since $\bar{\tau}_{i-1}$ gives an $\mathfrak{IB}_{i-1}$-closed set, we have also $\exists\mathrm{R}.\bigsqcap\bar{\tau}_{i-1}(\mathrm{D}) \in M_i$, as a look to the constructive definition of M_i immediately shows. Therefore, $\tau_i(\mathrm{C}) = \tau_i(\exists\mathrm{R.D}) = \{\exists\mathrm{R}.\bigsqcap\bar{\tau}_{i-1}(\mathrm{D})\} \subseteq M_i$
- $\mathrm{C} = \forall\mathrm{R.D}$. Again, our induction hypothesis yields $\tau_{i-1}(\mathrm{D}) \subseteq M_{i-1}$ which implies $\{\forall\mathrm{R.E} \mid \mathrm{E} \in \tau_{i-1}(\mathrm{D})\} \subseteq M_i$ due to the definition of M_i and therefore also $\tau_i(\mathrm{C}) = \tau_i(\forall\mathrm{R.D}) = \{\forall\mathrm{R.E} \mid \mathrm{E} \in \tau_{i-1}(\mathrm{D})\} \subseteq M_i$.
- $\mathrm{C} = \bigsqcap\mathcal{C}$. W.l.o.g., we presuppose that there is no conjunction outside the quantifier range in any $\mathrm{D} \in \mathcal{C}$. So we have $\tau_i(\mathrm{D}) \subseteq M_i$ due to the three cases above, and subsequently also $\tau_i(\mathrm{C}) = \tau_i(\bigsqcap\mathcal{C}) = \big(\bigcup\{\tau_i(\mathrm{D}) \mid \mathrm{C} \in \mathcal{C}\}\big) \subseteq M_i$. □

The next lemma and theorem show that in our fixed interpretation $\mathcal{I}$, for any concept description $\mathrm{C} \in \mathcal{DL}_i$, the entity sets fulfilling C on the one hand and $\bar{\tau}_i(\mathrm{C})$ as well as $\tau_i(\mathrm{C})$ on the other hand coincide.

Lemma 2. *For any $\mathcal{C} \subseteq M_i$, we have $\bigsqcap\mathcal{C} \equiv_{\mathcal{I}} \bigsqcap\mathcal{C}^{\mathfrak{IB}_i}$.*

Proof. First, observe $(\bigsqcap\mathcal{C})^{\mathcal{I}} = \bigcap\{(\mathrm{C})^{\mathcal{I}} \mid \mathrm{C} \in \mathcal{C}\} = \bigcap\{\mathrm{C}^{I_i} \mid \mathrm{C} \in \mathcal{C}\} = \{\delta \in \Delta^{\mathcal{I}} \mid \delta \in \mathrm{C}^{\mathcal{I}} \text{ for all } \mathrm{C} \in \mathcal{C}\}$. Now, consider $\mathbb{K}_i^{\mathcal{I}}$. Since $\mathfrak{IB}_i$ is an implication base of $\mathbb{K}_i^{\mathcal{I}}$, $\mathcal{C} \to \mathcal{C}^{\mathfrak{IB}_i}$ is an implication valid in $\mathbb{K}_i^{\mathcal{I}}$, ergo all objects of $\mathbb{K}_i^{\mathcal{I}}$ (being the individuals $\delta \in \Delta^{\mathcal{I}}$) fulfill $\mathcal{C} \subseteq \delta^{I_i} \Rightarrow \mathcal{C}^{\mathfrak{IB}_i} \subseteq \delta^{I_i}$. Therefore, one δ has all attributes from $\mathcal{C}$ exactly if it has all attributes from $\mathcal{C}^{\mathfrak{IB}_i}$. Finally, we have then $\{\delta \in \Delta^{\mathcal{I}} \mid \delta \in \mathrm{C}^{\mathcal{I}} \text{ for all } \mathrm{C} \in \mathcal{C}^{\mathfrak{IB}_i}\} = \bigcap\{\mathrm{C}^{\mathcal{I}} \mid \mathrm{C} \in \mathcal{C}^{\mathfrak{IB}_i}\} = (\bigsqcap\mathcal{C}^{\mathfrak{IB}_i})^{\mathcal{I}}$.

□

Theorem 2. *Let* $\mathrm{C} \in \mathcal{DL}_i$. *Then* $\mathrm{C} \equiv_{\mathcal{I}} \bigsqcap \tau_i(\mathrm{C}) \equiv_{\mathcal{I}} \bigsqcap \bar{\tau}_i(\mathrm{C})$.

Proof. The second equivalence is a direct consequence of Lemma 2. We show the first one again via induction on the role depth:

– $\mathrm{C} \in \{\mathrm{A}, \neg\mathrm{A} \mid \mathrm{A} \in \mathcal{A}\} \cup \{\bot\}$. Then, we trivially have $\mathrm{C}^{\mathcal{I}} = (\bigsqcap\{\mathrm{C}\})^{\mathcal{I}}$.
– $\mathrm{C} = \exists\mathrm{R.D}$. By induction hypothesis, we get $\mathrm{D}^{\mathcal{I}} = (\bigsqcap \bar{\tau}_{i-1}(\mathrm{D}))^{\mathcal{I}}$, therefore $(\exists\mathrm{R.D})^{\mathcal{I}} = (\exists\mathrm{R}.\bigsqcap \bar{\tau}_{i-1}(\mathrm{D}))^{\mathcal{I}}$ which by definition equals $(\bigsqcap \tau_i(\exists\mathrm{R.D}))^{\mathcal{I}}$.
– $\mathrm{C} = \forall\mathrm{R.D}$. Again, by induction hypothesis, we get $\mathrm{D}^{\mathcal{I}} = (\bigsqcap \bar{\tau}_{i-1}(\mathrm{D}))^{\mathcal{I}} = \bigcap\{\mathrm{E}^{\mathcal{I}} \mid \mathrm{E} \in \bar{\tau}_{i-1}(\mathrm{D})\}$. Now, observe that the statement $(\delta, \tilde{\delta}) \in \mathrm{R}^{\mathcal{I}} \to \tilde{\delta} \in \mathrm{D}^{\mathcal{I}}$ is equivalent to $\bigwedge_{\mathrm{E}\in\tau_{i-1}(\mathrm{D})} \big((\delta, \tilde{\delta}) \in \mathrm{R}^{\mathcal{I}} \to \tilde{\delta} \in \mathrm{E}^{\mathcal{I}}\big)$ and thus $(\forall\mathrm{R.D})^{\mathcal{I}} = \{\delta \mid (\delta, \tilde{\delta}) \in \mathrm{R}^{\mathcal{I}} \to \tilde{\delta} \in \bigcap\{\mathrm{D}^{\mathcal{I}}\}\} = \{\delta \mid \bigwedge_{\mathrm{E}\in\tau_{i-1}(\mathrm{D})} \delta \in (\forall\mathrm{R.E})^{\mathcal{I}}\} = \bigcap\{(\forall\mathrm{R.E})^{\mathcal{I}} \mid \mathrm{E} \in \tau_{i-1}(\mathrm{D})\} = (\bigsqcap\{\forall\mathrm{R.E} \mid \mathrm{E} \in \tau_{i-1}(\mathrm{D})\})^{\mathcal{I}}$ which by definition is just $(\bigsqcap \tau_i(\forall\mathrm{R.D}))^{\mathcal{I}}$.
– $\mathrm{C} = \bigsqcap\mathcal{C}$. Again, we can presume no conjunction outside the quantifier range in any $\mathrm{D} \in \mathcal{C}$. Then $(\bigsqcap\mathcal{C})^{\mathcal{I}} = \bigcap\{(\mathrm{D})^{\mathcal{I}} \mid \mathrm{D} \in \mathcal{C}\} = \bigcap\{(\bigsqcap \tau_i(\mathrm{D}))^{\mathcal{I}} \mid \mathrm{D} \in \mathcal{C}\}$ because of the cases shown before. Now, this is obviously the same as $\bigcap\{(\mathrm{E})^{\mathcal{I}} \mid \mathrm{E} \in \tau_i(\mathrm{D}), \mathrm{D} \in \mathcal{C}\} = (\bigsqcap(\bigcup\{\tau_i(\mathrm{D}) \mid \mathrm{D} \in \mathcal{C}\}))^{\mathcal{I}} = (\tau_i(\bigsqcap\mathcal{C}))^{\mathcal{I}}$. □

Using these propositions, we can easily provide a method to check – using only the closure operators $\mathfrak{IB}_0, \ldots, \mathfrak{IB}_i$ – the validity of any subsumption on $\mathcal{DL}_i$ with respect to a fixed (but not explicitly known) interpretation $\mathcal{I}$. It suffices to apply $\bar{\tau}_i$ on both sides and then check for inclusion.

Corollary 1. *Let* $\mathrm{C}_1, \mathrm{C}_2 \in \mathcal{DL}_i$. *Then* $\mathrm{C}_1 \sqsubseteq_{\mathcal{I}} \mathrm{C}_2$ *if and only if* $\bar{\tau}_i(\mathrm{C}_2) \subseteq \bar{\tau}_i(\mathrm{C}_1)$.

Proof. Due to Theorem 2, $\mathrm{C}_1 \sqsubseteq_{\mathcal{I}} \mathrm{C}_2$ is equivalent to $\bigsqcap \bar{\tau}_i(\mathrm{C}_1) \sqsubseteq_{\mathcal{I}} \bigsqcap \bar{\tau}_i(\mathrm{C}_2)$. According to Lemma 1, we have $\bar{\tau}_i(\mathrm{C}_1) \subseteq M_i$ and $\bar{\tau}_i(\mathrm{C}_2) \subseteq M_i$. In view of Theorem 1, this means the same as the validity of the implication $\bar{\tau}_i(\mathrm{C}_1) \to \bar{\tau}_i(\mathrm{C}_2)$ in $\mathbb{K}_i$. Now, since the application of $\bar{\tau}$ always gives a closed set with respect to all implications valid in $\mathbb{K}_i$, this is equivalent to $\bar{\tau}_i(\mathrm{C}_2) \subseteq \bar{\tau}_i(\mathrm{C}_1)$. □

Finally, consider the function τ_i from Definition 6. It is easy to see that for any $\mathrm{C} \in M_{i-1}$ by calculating $\tau_i(\mathrm{C})$ we get a singleton set $\{\mathrm{D}\}$ with $\mathrm{D} \in M_i$. We then have even $\mathrm{C} \equiv_{\mathcal{I}} \mathrm{D}$. For the sake of readability we will just write $\mathrm{D} = (\mathrm{C})_i$. Roughly spoken, D is just the "equivalent M_i-version" of C. Note that evaluating τ_i does not need the implication base $\mathfrak{IB}_i$ but only $\mathfrak{IB}_0, \ldots, \mathfrak{IB}_{i-1}$. So we have provided the translation function we promised in Section 6.

8 Conclusion

We have introduced an interactive knowledge acquisition technique for finding DL-style subsumption statements valid in a domain of interest. Its outstanding properties are

– minimal workload for the domain expert (i.e., no redundant questions will be posed) and

– completeness of the resulting specification (any statement from the exploration language is known to hold or not to hold).

Several current fields of AI can benefit from the results presented here.

Ontology engineering would be the first to mention. Since based on DL formalisms, our method can obviously contribute to the development and refinement of ontologies. RE can be used for an organized search for new GCIs[9] of a certain shape (namely those expressible by $\mathcal{DL}$ concept descriptions). Clearly, the description logics nowaday's ontology specifications are based on are much more complex than any of $\mathcal{DL}$. Nonetheless, our algorithm is still applicable since all of them incorporate the DLs considered as exploration language candidates. Hence, any of the existent reasoning algorithms for deciding subsumption (as for instance KAON2 [12] or FaCT [13], both capable of reasoning in $\mathcal{SHIQ}(D)$ – see [14]) can be used for the terminology part. All information beyond $\mathcal{DL}$ would then be treated as background knowledge and "hidden" from the exploration algorithm. As already pointed out, one major advantage of applying this technique is the guarantee that all valid axioms expressible as subsumption statements on $\mathcal{DL}$ with a certain role depth will certainly be found and specified.

Another topic RE can contribute to is machine learning. The supervised case corresponds almost directly to the RE algorithm – mostly one would have large data sets and (almost) empty theories in this setting. Yet also unsupervised machine learning can be carried out – by "short-circuiting" the expert such that every potential statement directed to him would be automatically confirmed. Essentially, the same would be the case for data mining tasks.

Finally, we are confident that an implementation of the RE algorithm will be a very helpful and versatile tool for eliciting information from various knowledge resources.

9 Future Work

So, as the very next step, we plan an implementation of the presented algorithm including interfaces for database querying as well as for DL reasoning. Applying this tool in the ontology engineering area will in turn enable us to investigate central questions concerning practical usability; in particular performance on real-life problems and scalability (being of unprecedented relevance in the Semantic Web Technologies sector), as well as issues concerning user acceptance will be of special interest for evaluation.

References

1. Rudolph, S.: An FCA method for the extensional exploration of relational data. In Ganter, B., de Moor, A., eds.: Using Conceptual Structures, Contributions to ICCS 2003, Dresden, Germany, Shaker, Aachen (2003) 197 – 210
2. Rudolph, S.: Exploring relational structures via FLE. In Wolff, K.E., Pfeiffer, H.D., Delugach, H.S., eds.: Conceptual Structures at Work: 12th International Conference on Conceptual Structures. Volume 3127 of LNCS., Huntsville, AL, USA, Springer (2004) 196 – 212

[9] I.e., GCIs not already logically entailed by the present specification.

3. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1997) Translator-C. Franzke.
4. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. J. Log. Program. **1** (1984) 267–284
5. Guigues, J.L., Duquenne, V.: Familles minimales d'implications informatives resultant d'un tableau de données binaires. Math. Sci Humaines **95** (1986) 5–18
6. Ganter, B.: Two basic algorithms in concept analysis. Technical Report 831, FB4, TH Darmstadt (1984)
7. Zickwolff, M.: Rule Exploration: First Order Logic in Formal Concept Analysis. PhD thesis, FB4, TH Darmstadt (1991)
8. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. In Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: Description Logic Handbook, Cambridge University Press (2003)
9. Prediger, S.: Terminologische Merkmalslogik in der formalen Begriffsanalyse. In Stumme, G., Ganter, B., eds.: Begriffliche Wissensverarbeitung: Methoden und Anwendungen, Springer (2000) 99–124
10. Baader, F.: Computing a minimal representation of the subsumption lattice of all conjunctions of concepts defined in a terminology. In: Proceedings of the International Symposium on Knowledge Retrieval, Use, and Storage for Efficiency, KRUSE 95, Santa Cruz, USA (1995) 168–178
11. Baader, F., Sertkaya, B.: Applying formal concept analysis to description logics. In: ICFCA. (2004) 261–286
12. Motik, B.: Reasoning in Description Logics using Resolution and Deductive Databases. PhD thesis, Universität Karlsruhe (TH), Germany (2006)
13. Horrocks, I.: Using an expressive description logic: FaCT or fiction? In Cohn, A.G., Schubert, L., Shapiro, S.C., eds.: Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR'98), Morgan Kaufmann Publishers, San Francisco, California (1998) 636–647
14. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for expressive description logics. In Ganzinger, H., McAllester, D., Voronkov, A., eds.: Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99). Number 1705, Springer-Verlag (1999) 161–180

Comparing Disjunctive Well-founded Semantics*

Matthias Knorr[1] and Pascal Hitzler[2]

[1] CENTRIA, Universidade Nova de Lisboa, Portugal
[2] AIFB, Universität Karlsruhe, Germany

Abstract. While the stable model semantics, in the form of Answer Set Programming, has become a successful semantics for disjunctive logic programs, a corresponding satisfactory extension of the well-founded semantics to disjunctive programs remains to be found. The many current proposals for such an extension are so diverse, that even a systematic comparison between them is a challenging task. In order to aid the quest for suitable disjunctive well-founded semantics, we present a systematic approach to a comparison based on level mappings, a recently introduced framework for characterizing logic programming semantics, which was quite successfully used for comparing the major semantics for normal logic programs. We extend this framework to disjunctive logic programs, which will allow us to gain comparative insights into their different handling of negation. Additionally, we show some of the problems occurring when trying to handle minimal models (and thus disjunctive stable models) within the framework.

1 Introduction

Two semantics are nowadays considered to be the most important ones for normal logic programs. Stable model semantics [1] is the main two-valued approach whereas the major three-valued semantics is the well-founded semantics [2]. These two semantics are well-known to be closely related. However, enriching normal logic programs with indefinite information by allowing disjunctions in the head[3] of the clauses separates these two approaches. While disjunctive stable models [5] are a straightforward extension of the stable model semantics, the issue of disjunctive well-founded semantics remains unresolved, although several proposals exist.

Even a comparison of existing proposals is difficult due to the large variety of completely different constructions on which these semantics are based. In [6], Ross introduced the strong well-founded semantics (SWFS) based on a top-down procedure using derivation trees. The generalized disjunctive well-founded

* This research has been partly funded by the European Commission within the 6th Framework Programme projects REWERSE number 506779 (cf. http://rewerse.net/) and NeOn (IST-2006-027595, cf. http://www.neon-project.org/), and by the Deutsche Forschungsgemeinschaft under project ReaSem.

[3] For an overview of semantics for disjunctive logic programs we refer to [3] and [4].

semantics (GDWFS) was defined by Baral, Lobo, and Minker in [7], built on several bottom-up operators and the extended generalized closed world assumption [8]. Brass and Dix proposed the disjunctive well-founded semantics (D-WFS) in [9] based on two operators iterating over conditional facts, respectively some general program transformations.

In order to allow for easier comparison of different semantics, a methodology has recently been proposed for uniformly characterizing semantics by means of level mappings, which allow for describing syntactic and semantic dependencies in logic programs [10]. This results in characterizations providing easy comparisons of the corresponding semantics.

In this paper, we attempt to utilize this approach and present level mapping characterizations for the three previously mentioned semantics, namely SWFS, GDWFS and D-WFS. The obtained uniform characterizations will allow us to compare the semantics in a new and more structured way. It turns out, however, that even under the uniform level-mapping characterizations the three semantics differ widely, such that there is simply not enough resemblance between the approaches to obtain a coherent picture. We can thus, basically, only confirm in a more formal way what has been known beforehand, namely that the issue of a good definition of well-founded semantics for disjunctive logic programs remains widely open. We still believe that our approach delivers structural insights which can help to guide the quest.

The paper is structured as follows. In Section 2, basic notions are presented and we recall shortly the well-founded semantics. Then we devote one section to each of the three semantics recalling the approach itself and presenting the level mapping characterization. We start with SWFS in Section 3, continue with GDWFS in Section 4 and end with D-WFS in Section 5. After that, in Section 6 we compare the characterizations looking for common conditions which might be properties for an appropriate well-founded semantics for disjunctive programs, and consider some of the difficulties occurring when applying the framework to minimal models. We conclude with Section 7.

The formal proofs required for the level-mapping characterizations of the semantics reported in this paper are very involved and technical. Due to space limitations, it was obviously not possible to include them. They can be found in the publicly available Technical Report [11].

2 General Notions and Preliminaries

A *disjunctive logic program* Π consists of finitely many universally quantified *clauses* of the form $H_1 \vee \cdots \vee H_l \leftarrow A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m$ where H_k, A_i, and B_j, for $k = 1, \ldots, l$, $i = 1, \ldots, n$, and $j = 1, \ldots, m$, are atoms of a given first order language, consisting of predicate symbols, function symbols, constants and variables. The symbol $\neg$ is representing default negation. A clause c can be divided into the *head* $H_1 \vee \cdots \vee H_l$ and the *body* $A_1, \ldots, A_n$, $\neg B_1, \cdots, \neg B_m$. If the body is empty then c is called a *fact*. We also abbreviate c by $H \leftarrow A, \neg B$, where H, A and B are sets of pairwise distinct atoms and, likewise, we sometimes handle disjunctions D and conjunctions C as sets. A *normal* (*definite*) clause contains exactly one atom in H (no atom in B) and we call a program consisting only of normal (definite) clauses a *normal* (*definite*) logic program. We denote normal programs by P to distinguish from disjunctive ones represented by Π. Any expression is called *ground* if it contains no variables. The *Herbrand base* B_Π is the set of all ground atoms that can be formed by using the given language from Π. A *literal* is either a *positive literal*, respectively an atom, or a *negative literal*, a negated atom, and usually we denote by $A, B, \ldots$ atoms and by $L, M, \ldots$ literals. Moreover, a *disjunction literal* is a disjunction or a negated disjunction. The *extended Herbrand base* EB_Π (*conjunctive Herbrand base* CB_Π) is the set of all disjunctions (conjunctions) that can be formed using pairwise distinct atoms from B_Π. Finally, $\mathsf{ground}(\Pi)$ is the set of all ground instances of clauses in Π with respect to B_Π.

We continue by recalling three-valued semantics based on the truth values true ($\mathbf{t}$), undefined ($\mathbf{u}$), and false ($\mathbf{f}$). A *(partial) three-valued interpretation* I of a normal program P is a set $A \cup \neg B$, for $A, B \subseteq B_P$ and $A \cap B = \emptyset$, where elements in A, B respectively, are $\mathbf{t}$, $\mathbf{f}$, and the remaining wrt. B_P are $\mathbf{u}$. The set of three-valued interpretations is denoted by $I_{P,3}$. Given a three-valued interpretation I, the body of a ground clause $H \leftarrow L_1, \ldots, L_n$ is true in I if and only if $L_i \in I$, $1 \leq i \leq n$, or false in I if and only if $L_i \notin I$ for some i, $1 \leq i \leq n$. Otherwise the body is undefined. The ground clause $H \leftarrow \mathtt{body}$ is true in I if and only if the head H is true in I or $\mathtt{body}$ is false in I or $\mathtt{body}$ is undefined and H is not false in I. Moreover, I is a *three-valued model* for P if and only if all clauses in $\mathsf{ground}(P)$ are true in I. The *knowledge ordering* is recalled which, given two three-valued interpretations I_1 and I_2, is defined as $I_1 \leq_k I_2$ if and only if $I_1 \subseteq I_2$. For a program P and a three-valued interpretation $I \in I_{P,3}$ an *I-partial level mapping* for P is a partial mapping $l : B_P \rightarrow \alpha$ with domain $\mathsf{dom}(l) = \{A \mid A \in I \text{ or } \neg A \in I\}$, where α is some (countable) ordinal. Every such mapping is extended to literals by setting $l(\neg A) = l(A)$ for all $A \in \mathsf{dom}(l)$. Any ordinal α is identified with the set of ordinals β such that $\alpha > \beta$. Thus, any mapping $f : X \rightarrow \{\beta \mid \beta < \alpha\}$ is represented by $f : X \rightarrow \alpha$. Given two ordinals α, β, the lexicographic order $(\alpha \times \beta)$ is also an ordinal with $(a, b) \geq (a', b')$ if and only if $a > a'$ or $a = a'$ and $b \geq b'$ for all $(a, b), (a', b') \in \alpha \times \beta$. This order can be split into its components, namely $(a, b) >_1 (a', b')$ if and only if $a > a'$ for all $(a, b), (a', b') \in \alpha \times \beta$ and $(a, b) \geq_2 (a', b')$ if and only if $a = a'$ and $b \geq b'$ for all $(a, b), (a', b') \in \alpha \times \beta$. Additionally we allow the order $\succ$ which given an ordinal $(\alpha \times \beta)$ is defined as $(a, b) \succ (a', b')$ if and only if $b > b'$ for all (a, b), $(a', b') \in (\alpha \times \beta)$.

We shortly recall the level mapping characterization of the well-founded semantics and refer for the original bottom-up operator to [2].

Definition 2.1. *([10]) Let P be a normal logic program, let I be a model for P, and let l be an I-partial level mapping for P. We say that P* satisfies *(WF)* with respect to I and l *if each $A \in \mathsf{dom}(l)$ satisfies one of the following conditions.*

(WFi) $A \in I$ and there is a clause $A \leftarrow L_1, \ldots, L_n$ in $\mathsf{ground}(P)$ such that $L_i \in I$ and $l(A) > l(L_i)$ for all i.

(WFii) $\neg A \in I$ *and for each clause* $A \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ *in* $\mathsf{ground}(P)$ *one (at least) of the following conditions holds:*

(WFiia) There exists i *with* $\neg A_i \in I$ *and* $l(A) \geq l(A_i)$*.*

(WFiib) There exists j *with* $B_j \in I$ *and* $l(A) > l(B_j)$*.*

If $A \in \mathsf{dom}(l)$ satisfies (WFi), then we say that A *satisfies* (WFi) *with respect to* I *and* l, and similarly if $A \in \mathsf{dom}(l)$ satisfies (WFii).

Theorem 2.1. *([10]) Let* P *be a normal logic program with well-founded model* M*. Then, in the knowledge ordering,* M *is the greatest model amongst all models* I *for which there exists an* I*-partial level mapping* l *for* P *such that* P *satisfies (WF) with respect to* I *and* l*.*

Example 2.1. Consider the program $P = \{p \leftarrow \neg q; q \leftarrow q; r \leftarrow \neg p\}$. We obtain the well-founded model $M = \{p, \neg q, \neg r\}$ with $l(p) = 1$, $l(q) = 0$ and $l(r) = 2$. Note that, for $I = \emptyset$ and arbitrary l, P satisfies (WF) wrt. I and l as well but I is not the greatest such model wrt. $\leq_k$ and thus not the well-founded model.

We continue extending some of the previous notions to the disjunctive case. Let I be a set of disjunction literals. The *closure* of I, $\mathsf{cl}(I)$, is the least set I' with $I \subseteq I'$ satisfying the following conditions: if $D \in I'$ then $D' \in I'$ for all D' with $D \subseteq D'$, and for all disjunctions D_1 and D_2, $\neg D_1 \in I'$ and $\neg D_2 \in I'$ if and only if $\neg(D_1 \vee D_2) \in I'$. Then, I is *consistent* if there is no $D \in \mathsf{cl}(I)$ with $\neg D \in \mathsf{cl}(I)$ as well[4]. A *disjunctive three-valued interpretation* I of a disjunctive program Π is a consistent set $A \cup \neg B$, $A, B \subseteq EB_\Pi$, where elements in A are $\mathbf{t}$, elements in B are $\mathbf{f}$, and the remaining wrt. EB_Π are $\mathbf{u}$. The body of a ground clause $H \leftarrow A, \neg B$ is true in I if and only if all literals in the body are true in I, or false in I if and only if there is a D such that either $D \subseteq A$ with $\neg D \in I$ or $D \subseteq B$ with $D \in I$[5]. Otherwise the body is undefined. The truth of a ground clause $H \leftarrow \mathtt{body}$ is identical to normal programs and I is a *disjunctive three-valued model* of Π if every clause in $\mathsf{ground}(\Pi)$ is true in I. The *disjunctive knowledge ordering* $\preceq_k$ is defined as $I_1 \preceq_k I_2$ if and only if $I_1 \subseteq I_2$ and the corresponding level mapping is extended as follows.

Definition 2.2. *For a disjunctive program* Π *and a disjunctive interpretation* I *a* disjunctive I-partial level mapping *for* Π *is a partial mapping* $l : EB_\Pi \rightarrow \alpha$ *with domain* $\mathsf{dom}(l) = \{D \mid D \in I \text{ or } \neg D \in I\}$*, where* α *is some (countable) ordinal. Every such mapping is extended to negated disjunctions by setting* $l(\neg D) = l(D)$ *for all* $D \in EB_\Pi$*.*

Another way of representing disjunctive information are *state-pairs* $A \cup \neg B$, where A is a subset of EB_Π such that for all D' if $D \in A$ and $D \subseteq D'$ then $D' \in A$, and B is a subset of CB_Π such that for all C' if $C \in B$ and $C \subseteq C'$

[4] Here, a consistent set is not automatically closed, in contrast with the assumption made in [6].

[5] The extension is necessary since we might e.g. know the truth of some disjunction without knowing which particular disjunct is true.

then $C' \in B$. Disjunctions in A are $\mathbf{t}$, conjunctions in B are $\mathbf{f}$, and all remaining are $\mathbf{u}$. A state-pair is consistent if whenever $D \in A$ then there is at least one disjunct $D' \in D$ such that $D' \notin B$ and whenever $C \in B$ then there is at least one conjunct $C' \in C$ such that $C' \notin A$. The notions of models and the disjunctive knowledge ordering can easily be adopted. Note that a state-pair is not necessarily consistent and that it contains indefinite positive and negative information in opposite to disjunctive interpretations where negative information will be precise. Level mappings are adjusted to state-pairs in the following and now we do not extend the mapping to identify $l(D) = l(\neg D)$ since in a state-pair D is a disjunction and $\neg D$ a negated conjunction.

Definition 2.3. *For a disjunctive program Π and a state-pair I a disjunctive I-partial level mapping for Π is a partial mapping $l : (EB_\Pi \cup \neg CB_\Pi) \to \alpha$ with domain $\mathsf{dom}(l) = \{D \mid D \in I \text{ or } \neg D \in I\}$, where α is some (countable) ordinal.*

3 Strong Well-founded Semantics

We start with SWFS which was introduced by Ross [6] and based on disjunctive interpretations. The derivation rules of the applied top-down procedure are the following. Given a set of disjunction literals I and a disjunctive program Π the *derivate* I' is *strongly derived* from I ($I \Leftarrow I'$) if I contains a disjunction D and $\mathsf{ground}(\Pi)$ a clause $H \leftarrow A_1, \ldots, A_n, \neg B$ such that either

(S1) $H \subseteq D$ and $I' = (I \setminus \{D\}) \cup \{A_1 \vee D, \ldots, A_n \vee D\} \cup \neg B$ or
(S2) $H \not\subseteq D$, $H \cap D \neq \emptyset$, $C = H \setminus D$, and $I' = (I \setminus \{D\}) \cup A \cup \neg B \cup \neg C$.

Consider a ground disjunction D, let $I_0 = \{D\}$ and suppose that $I_0 \Leftarrow I_1 \Leftarrow I_2 \ldots$, then $I_0, I_1, I_2 \ldots$ is a *(strong) derivation sequence* for D. An *active* (strong) derivation sequence for D is a finite derivation sequence for D whose last element, also called a *basis* of D, is either empty or contains only negative literals. A basis $I = \{\neg l_1, \ldots, \neg l_n\}$ is turned into a disjunction $\bar{I} = l_1 \vee \cdots \vee l_n$ and if I is empty, denoting $\mathbf{t}$, then $\bar{I}$ denotes $\mathbf{f}$. Thus, a *strong global tree* Γ_D^S for a given disjunction $D \in EB_\Pi$ contains the root D and its children are all disjunctions of the form $\bar{I}$, where I ranges over all bases for D. The *strong well-founded model* of a disjunctive program Π is called $M_{WF}^S(\Pi)$ and $D \in M_{WF}^S(\Pi)$, i.e. D is true, if some child of D is false and $\neg D \in M_{WF}^S(\Pi)$, i.e. D is false, if every child of D is true. Otherwise, D is undefined and neither D nor $\neg D$ occur in $M_{WF}^S(\Pi)$. In [6], it was shown that $M_{WF}^S(\Pi)$ is a consistent interpretation and that, for normal programs, SWFS coincides with the well-founded semantics[6].

Example 3.1. The following program Π will be used to demonstrate the behavior of the three semantics.

$p \vee q \leftarrow \neg q$	$b \vee l \leftarrow \neg r$	$c \leftarrow \neg l, \neg r$	$f \leftarrow \neg e$
$q \leftarrow \neg q$	$l \vee r \leftarrow$	$e \leftarrow \neg f, c$	$g \leftarrow e$

[6] More precisely, the disjunctive model has to be restricted to (non-disjunctive) literals.

We obtain a sequence $\{l \vee r\} \Leftarrow \{\}$ and $l \vee r$ is true as expected. Furthermore, there is a finite sequence in Γ_e^S, namely $\{e\} \Leftarrow \{\neg f, e \vee c\} \Leftarrow \{\neg f, \neg l, \neg c\}$ with the only (true) child and e is false. Thus, we have that $M_{WF}^S(\Pi) = \{l \vee r, f, \neg b, \neg c, \neg e, \neg g\}$ while p and q remain undefined. Literally, this is only a small part of the model and we might close the model (e.g. $\neg(e \vee g) \in M_{WF}^S$) for this example, but the strong well-founded is not necessarily closed which does not allow us to add this implicit information in general.

The level mapping framework is based on bottom-up operators and SWFS is a top-down-procedure so we introduced a bottom-up operator on derivation trees defined on $\boldsymbol{\Gamma_\Pi^S}$ which is the power set of Γ_Π^S - the set of all strong global trees with respect to Π.

Definition 3.1. *Let Π be a disjunctive logic program, $M_{WF}^S(\Pi)$ the strong well-founded model, and $\Gamma \in \boldsymbol{\Gamma_\Pi^S}$. We define:*

- $T_\Pi^S(\Gamma) = \{\Gamma_D^S \in \Gamma_\Pi^S \mid \Gamma_D^S$ *contains an active strong derivation sequence* $\{D\}, I_1, \ldots, I_r$ *with child* $C = \bar{I}_r$ *and* $I_1 = \{D_1, \ldots, D_n, \neg D_{n+1}, \ldots, \neg D_m\}$ *where* $\neg C \in M_{WF}^S(\Pi)$, $\Gamma_C^S \in \Gamma$ *if* $C \neq \{\}$, $\Gamma_{D_i}^S \in \Gamma$, $D_i \in M_{WF}^S$, $\Gamma_{D_j}^S \in \Gamma$, $\neg D_j \in M_{WF}^S$ *for all* $i = 1, \ldots, n$ *and* $j = n+1, \ldots, m\}$
- $U_\Pi^S(\Gamma) = \{\Gamma_D^S \in \Gamma_\Pi^S \mid$ *for all active strong derivation sequences in* Γ_D^S *the corresponding child* C *is true in* $M_{WF}^S(\Pi)$ *and* $\Gamma_C^S \in \Gamma\}$

The information is joined by $W_\Pi^S(\Gamma) = T_\Pi^S(\Gamma) \cup U_\Pi^S(\Gamma)$ and iterated: $W_\Pi^S \uparrow 0 = \emptyset$, $W_\Pi^S \uparrow n+1 = W_\Pi^S(W_\Pi^S \uparrow n)$, and $W_\Pi^S \uparrow \alpha = \bigcup_{\beta<\alpha} W_\Pi^S \uparrow \beta$ for limit ordinal α. It was shown in [11] that W_Π^S is monotonic, allowing to apply the Tarski fixed-point theorem which yields that the operator W_Π^S always has a least fixed point, and that this least fixed point coincides with M_{WF}^S. This was used to derive the following alternative characterization of SWFS.

Definition 3.2. *Let Π be a disjunctive logic program, let I be a model for Π, and let l be a disjunctive partial level mapping for Π. We say that Π satisfies (SWF) with respect to I and l if each $D \in \mathsf{dom}(l)$ satisfies one of the following conditions:*

(SWFi) $D \in I$ *and* Γ_D^S *contains an active strong derivation sequence with child* C, $\neg C \in I$ *and* $l(D) > l(C)$ *if* $C \neq \{\}$, *and there is a clause* $H \leftarrow A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m$ *in* $\mathsf{ground}(\Pi)$ *which is used for the first derivation of that sequence such that* $\neg B_j \in I$ *and* $l(D) > l(B_j)$, $1 \leq j \leq m$, *and one of the following conditions holds:*

(SWFia) $H \subseteq D$ *such that there is* $D_i \subseteq D$ *with* $(D_i \vee A_i) \in I$ *and* $l(D) > l(D_i \vee A_i)$, $1 \leq i \leq n$.

(SWFib) $H \not\subseteq D$, $H \cap D \neq \emptyset$, $\{C_1, \ldots, C_l\} = H \setminus D$, $A_i \in I$ *and* $l(D) > l(A_i)$, $1 \leq i \leq n$, *and* $\neg C_k \in I$ *and* $l(D) > l(C_k)$, $1 \leq k \leq l$.

(SWFii) $\neg D \in I$ *and for each active strong derivation sequence in* Γ_D^S *with child* $C \in I$ *there is a clause* $H \leftarrow A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m$ *in* $\mathsf{ground}(\Pi)$ *which is used for the first derivation of that sequence such that (at least) one of the following conditions holds:*

(SWFiia') $H \subseteq D$ and there exists i, $1 \leq i \leq n$, with $\neg(A_i \vee D) \in I$, $l(D) \geq l(A_i \vee D)$.
(SWFiia") $H \not\subseteq D$, $H \cap D \neq \emptyset$, and there exists i with $\neg A_i \in I$, $l(D) \geq l(A_i)$, $1 \leq i \leq n$.
(SWFiib') $H \subseteq D$ and there exists D' with $D' \subseteq B$, $D' \in I$ and $l(D) > l(D')$.
(SWFiib") $H \not\subseteq D$, $H \cap D \neq \emptyset$, $C = (H \setminus D)$, and there exists D' with $D' \subseteq (B \cup C)$, $D' \in I$ and $l(D) > l(D')$.
(SWFiic) $l(D) > l(C)$.

Theorem 3.1. *Let Π be a disjunctive program with strong well-founded model M. Then, in the disjunctive knowledge ordering, M is the greatest model amongst all models I for which there exists a disjunctive I-partial level mapping l for Π such that Π satisfies (SWF) with respect to I and l.*

The characterization is obviously more involved than Definition 2.1. In fact, even though it appears that for every true disjunction there are a sequence and a clause satisfying (SWFia), we were unable to show that due to the missing closure property of the strong well-founded model. Thus, we have to keep the condition (SWFib). Moreover, it can be checked that all the cases for negated disjunctions yield that (SWFiic) holds as well. We therefore could have formulated (SWFii) just using (SWFiic), but for a better comparison to the characterization of well-founded semantics and the following semantics we separate the case. We continue with the example.

Example 3.2. (Example 3.1 continued) As shown in [11], we obtain $l(D) = \alpha$, where α is the least ordinal such that $\Gamma_D^S \in (W_\Pi^S \uparrow (\alpha + 1)) = W_\Pi^S(W_\Pi^S \uparrow \alpha)$. Thus, we have $l(l \vee r) = 0$ by (SWFia) and $l(e) = 1$ by (SWFiia') and therefore $l(f) = 2$ by (SWFia). Moreover, $l(b) = 1$ by (SWFiib') whereas $l(c) = 1$ by (SWFiib"). Note that in case of b, c, and e also (SWFiic) is satisfied.

It should be mentioned that the reference to derivation sequences in the conditions is also necessary because of the missing closure property of $M_{WF}^S(\Pi)$.

4 Generalized Disjunctive Well-founded Semantics

Baral, Lobo, and Minker introduced GDWFS [7] based on state-pairs. They applied various operators for calculating the semantics and we recall at first $\mathcal{T}_S^D$ and $\mathcal{F}_S^D$ for disjunctive programs.

Definition 4.1. *Let S be a state-pair and Π be a disjunctive program. Let $T \subseteq EB_\Pi$ and $F \subseteq CB_\Pi$.*

$\mathcal{T}_S^D(T) = \{D \in EB_\Pi \mid D$ undefined in S, $H \leftarrow A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m$ in $\mathsf{ground}(\Pi)$ such that for all i, $1 \leq i \leq n$, $(A_i \vee D_i) \in S$ or $(A_i \vee D_i) \in T$, D_i might be empty, $\neg B_j \in S$ for all j, $1 \leq j \leq m$, and $(H \cup \bigcup_i D_i) \subseteq D.\}$

$\mathcal{F}_S^D(F) = \{C \in CB_\Pi \mid C$ is undefined in S, $A \in C$, and for all clauses $H \leftarrow A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m$ in $\mathsf{ground}(\Pi)$, with $A \in H$, at least one of the following three cases holds: $(B_1 \vee \cdots \vee B_m) \in S$, $\neg(A_1 \wedge \cdots \wedge A_n) \in S$, or $\neg(A_1 \wedge \cdots \wedge A_n) \in F\}$

T_S^D is bottom-up and F_S^D is top-down: $\mathcal{T}_S^D \uparrow 0 = \emptyset$, $\mathcal{T}_S^D \uparrow (n+1) = \mathcal{T}_S^D(\mathcal{T}_S^D \uparrow n)$, $\mathcal{T}_S^D = \bigcup_{n<\omega} \mathcal{T}_S^D \uparrow n$, and $\mathcal{F}_S^D \downarrow 0 = CB_\Pi$, $\mathcal{F}_S^D \downarrow (n+1) = \mathcal{F}_S^D(\mathcal{F}_S^D \downarrow n)$, $\mathcal{F}_S^D = \bigcap_{n<\omega} \mathcal{F}_S^D \downarrow n$.

There are two more operators defined for definite programs which necessitates the following program transformations. Given a disjunctive program Π and a state-pair S, $DIS(\Pi)$ is obtained by transferring all negated atoms in the body of each clause of Π as atoms to its head. Then, $Dis(\Pi, S)$ results from $DIS(\Pi)$ by reducing the clauses in $DIS(\Pi)$ as follows: remove atoms from the body of a clause if they are true in S, remove a clause if its head is true in S, and remove atoms from the head of a clause if they are false in S. This is similar to the construction used for stable models and we recall $T_\Pi^D(T)$, a simplification of $\mathcal{T}_S^D(T)$ to definite programs. Given a definite (disjunctive) program Π and T, a subset of EB_Π, we have that $T_\Pi^D(T) = \{D \in EB_\Pi \mid H \leftarrow A_1, \ldots, A_n$ in $\mathsf{ground}(\Pi)$ such that for all i, $1 \leq i \leq n$, $(A_i \vee D_i) \in T$, D_i might be empty, and $(H \cup \bigcup_i D_i) \subseteq D.\}$ We then iterate $T_\Pi^D \uparrow 0 = \emptyset$, $T_\Pi^D \uparrow (n+1) = T_\Pi^D(T_\Pi^D \uparrow n)$, and $T_\Pi^D = \bigcup_{n<\omega} T_\Pi^D \uparrow n$.

For deriving indefinite false conjunctions the Extended Generalized Closed World Assumption (EGCWA) [8] is applied. It intuitively says that a conjunction can be inferred to be false from Π if and only if it is false in all minimal models of Π where a minimal model [12] is a two-valued model M of Π such that no subset of it is a model as well.

The previous two constructions yield the operators $T_S^{ED} = \{D \mid D \in T^D_{Dis(\Pi,S)}$ and $D \notin S\}$ and $F_S^{ED} = \{C \mid C \in \text{EGCWA}(Dis(\Pi, S) \cup S)$ and $C \notin S\}$. Now we can combine all the operators and obtain

$$\mathcal{S}^{ED}(S) = S \cup \mathcal{T}_S^D \cup \neg\mathcal{F}_S^D \cup T_S^{ED} \cup \neg F_S^{ED}.$$

The iteration is done via $M_0 = \emptyset$, $M_{\alpha+1} = \mathcal{S}^{ED}(M_\alpha)$, and $M_\alpha = \bigcup_{\beta<\alpha} M_\beta$ for limit ordinal α, and has a fixed point [7]. The fixed point corresponds to the generalized disjunctive well-founded model M_Π^{ED} which is consistent [7]. However, for normal logic programs in general, the well-founded semantics and GDWFS do not coincide.

Example 4.1. Recall the program from Example 3.1. We have $M_\Pi^{ED} = \{l \vee r, q, f, \neg p, \neg b, \neg c, \neg e, \neg g, \neg(l \wedge r)\}$. Note that M_Π^{ED} is closed in so far that any superset of a true disjunction (false conjunction) is true (false) as well. Moreover, the semantics concludes q to be true from $q \leftarrow \neg q$. This is exactly the kind of reasoning which is not applied for the well-founded semantics, thus being the cause for GDWFS and WFS not to coincide on normal programs.

We continue with the level mapping characterization of GDWFS.

Definition 4.2. *Let Π be a disjunctive logic program, let the state-pair I be a model for Π, and let l_1, l_2 be disjunctive I-partial level mappings for Π. We say that Π satisfies (GDWF) with respect to I, l_1, and l_2 if each $D \in \mathsf{dom}(l_1)$ and each $\neg C \in \mathsf{dom}(l_1)$ satisfies one of the following conditions:*

(GDWFi) $D \in I$ *and there is a clause* $H \leftarrow A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m$ *in* $\mathsf{ground}(\Pi)$ *with* $H \subseteq D$ *such that* $\neg B_j \in I$ *and* $l_1(D) >_1 l_t(\neg B_j)$, $t \in \{1,2\}$, *for all* $j = 1, \ldots, m$, *and, for all* $i = 1, \ldots, n$, *there is* $D_i \subseteq D$ *with* $(D_i \vee A_i) \in I$ *where* $l_1(D) > l_1(D_i \vee A_i)$ *or* $l_1(D) >_1 l_2(D_i \vee A_i)$.

(GDWFii) $\neg C \in I$ *with atom* $A \in C$ *and for each clause* $H \leftarrow A_1, \ldots, A_n$, $\neg B_1, \ldots, \neg B_m$ *in* $\mathsf{ground}(\Pi)$ *with* $A \in H$ *(at least) one of the following conditions holds:*

- *(GDWFiia)* $\neg(A_1 \wedge \ldots \wedge A_n) \in I$ *and* $l_1(\neg C) \geq l_1(\neg(A_1 \wedge \ldots \wedge A_n))$.
- *(GDWFiia')* $\neg(A_1 \wedge \ldots \wedge A_n) \in I$ *and* $l_1(\neg C) >_1 l_2(\neg(A_1 \wedge \ldots \wedge A_n))$.
- *(GDWFiib)* $(B_1 \vee \ldots \vee B_m) \in I$ *and* $l_1(\neg C) >_1 l_t(B_1 \vee \ldots \vee B_m)$ *for* $t \in \{1,2\}$.

and each $D, \neg C \in \mathsf{dom}(l_2)$ *satisfies one of the following conditions:*

(GDWFi') $D \in I$ *and there is a clause* $H_1 \vee \cdots \vee H_l \leftarrow A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m$ *in* $\mathsf{ground}(\Pi)$ *such that* $\emptyset \neq ((H \cup B) \setminus D') \subseteq D$, $H_k \in D'$ *for each* $\neg H_k \in I$ *with* $l_2(D) >_1 l_t(\neg H_k)$, $t \in \{1,2\}$, $B_j \in D'$ *for each* $\neg B_j \in I$ *with* $l_2(D) >_1 l_t(\neg B_j)$, $t \in \{1,2\}$, *for all* $k = 1, \ldots, l$ *and all* $j = 1, \ldots, m$, *and, for all* $i = 1, \ldots, n$, *there is* $D_i \subseteq D$ *with* $(D_i \vee A_i) \in I$ *where* $l_2(D) >_2 l_2(D_i \vee A_i)$ *or* $A_i \in I$ *where* $l_2(D) >_1 l_s(A_i)$, $s \in \{1,2\}$.

(GDWFii') $\neg C \in I$ *and* $C \in EGCWA(Dis(\Pi,S) \cup S)$, $C \notin S$ *and* $l_2(\neg C) >_1 l_t(L)$, $t \in \{1,2\}$, *if and only if* $L \in S$.

The reason for introducing two mappings is to extrapolate exactly the simultaneous iteration of the two operators dealing with positive, negative respectively, information. The very same argument necessitates the different orderings. From a more general perspective, e.g. (GDWFiia) and (GDWFiia') employ basically the same kind of dependency, just the proof of the following theorem stating the equivalence enforces the diverse conditions.

Theorem 4.1. *Let* Π *be a disjunctive program with generalized disjunctive well-founded model* M*. Then, in the disjunctive knowledge ordering,* M *is the greatest model amongst all models* I *for which there exist disjunctive* I*-partial level mappings* l_1 *and* l_2 *for* Π *such that* Π *satisfies (GDWF) w.r.t.* I, l_1, *and* l_2.

Example 4.2. (Example 4.1 continued) From [11] we know that if $D \in \mathcal{T}^D_{M_\alpha}$ then β is the least ordinal such that $D \in \mathcal{T}^D_{M_\alpha} \uparrow (\beta+1)$ and $l_1(D) = (\alpha, \beta)$, if $D \in T^{ED}_{M_\alpha}$ then β is the least ordinal such that $D \in T^D_{Dis(\Pi, M_\alpha)} \uparrow (\beta+1)$ and $l_2(D) = (\alpha, \beta)$. For negative conjunctions it holds that $l_1(\neg C) = (\alpha, 0)$ if $C \in \mathcal{F}^D_{M_\alpha}$ and $l_2(\neg C) = (\alpha, 0)$ if $C \in F^{ED}_{M_\alpha}$. Thus, we obtain e.g. $l_1(l \vee r) = l_2(l \vee r) = (0,0)$ by (GDWFi) and (GDWFi'), $l_1(f) = (2,0)$ by (GDWFi), $l_2(\neg p) = (0,0)$ by (GDWFii'), $l_1(\neg e) = (1,0)$ by (GDWFiia') and $l_1(\neg g) = (1,0)$ by (GDWFiia).

The condition (GDWFii') directly refers to EGCWA due to problems with minimal models in the level mapping framework (see Section 6).

5 Disjunctive Well-founded Semantics

The third approach we study is the disjunctive well-founded semantics presented by Brass and Dix in [9]. We use again disjunctive interpretations for representing information even though in [9] the syntactically different pure disjunctions are applied. D-WFS is only defined for (disjunctive) DATALOG programs which are programs whose corresponding language does not have any function symbols apart from (nullary) constants. Thus they correspond to propositional programs and we use the notation Φ from [9] for DATALOG programs.

We recall the operators defining D-WFS. Both map sets of *conditional facts* which are disjunctive clauses without any positive atoms in the body and we start with T_Φ. Given Φ and a set of conditional facts Γ, we have that $T_\Phi(\Gamma) = \{(H \cup \bigcup_i (H_i \setminus \{A_i\})) \leftarrow \neg(B \cup \bigcup_i B_i) \mid$ there is $H \leftarrow A_1, \ldots, A_n, \neg B$ in $\mathsf{ground}(\Phi)$ and conditional facts $H_i \leftarrow \neg B_i \in \Gamma$ with $A_i \in H_i$ for all $i = 1, \ldots, n.\}$ The iteration of T_Φ is given as $T_\Phi \uparrow 0 = \emptyset$, $T_\Phi \uparrow (n+1) = T_\Phi(T_\Phi \uparrow n)$, and $T_\Phi = \bigcup_{n<\omega} T_\Phi \uparrow n$ and yields a fixed point.

The next operator is top-down starting with the previous fixed point also applying the notion of $\mathsf{heads}(S)$ which is the set of all atoms occurring in some head of a clause contained in a given set of ground clauses S: given a set of conditional facts Γ we define $R(\Gamma) = \{H \leftarrow \neg(B \cap \mathsf{heads}(\Gamma)) \mid H \leftarrow \neg B \in \Gamma$, and there is no $H' \leftarrow$ in Γ with $H' \subseteq B$ or there is no $H' \leftarrow \neg B'$ in Γ with $H' \subseteq H$ and $B' \subseteq B$ where at least one $\subseteq$ is proper.$\}$ Note that the second condition forcing one $\subseteq$ to be proper is necessary since otherwise we could remove each conditional fact by means of itself. The iteration of this operator is defined as $R \uparrow 0 = T_\Phi$, $R \uparrow (n+1) = R(R \uparrow n)$ and the fixed point of this operator is called the *residual program* of Φ.

Given the residual program $\mathsf{res}(\Phi)$, the disjunctive well-founded model M_Φ is $M_\Phi = \{D \in EB_\Phi \mid$ there is $H \leftarrow$ in $\mathsf{res}(\Phi)$ with $H \subseteq D\} \cup \{\neg D \mid D \in EB_\Phi$ and $\forall D' \in D : D' \notin \mathsf{heads}(\mathsf{res}(\Phi))\}$. Though T_Φ is monotonic, R is not and we cannot generalize the following results to all disjunctive logic programs. We should note that in [13] the approach was extended to disjunctive logic programs by combining the transformation rules with constraint logic programming. But the operators are not extended as well and we remain with that restriction.

Example 5.1. Recall Π from Example 3.1. It is obvious that Π is also a DATALOG program and we obtain $M_\Pi = \{l \vee r, f, \neg p, \neg c, \neg e, \neg g\}$. Note that M_Π is closed by definition of the model.

In the following, we present the alternative characterization of D-WFS.

Definition 5.1. *Let Φ be a DATALOG program, let I be a model for Φ, and let l be a disjunctive I-partial level mapping for Φ. We say that Φ satisfies (DWF) with respect to I and l if each $D \in \mathsf{dom}(l)$ satisfies one of the following conditions:*

(DWFi) $D \in I$ and there is a clause $H \leftarrow A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m$ in $\mathsf{ground}(\Phi)$ with $H \subseteq D$ such that there is $D_i \subseteq D$ with $(D_i \vee A_i) \in I$, $l(D) > l(D_i \vee A_i)$, and $l(D) \succ l(D_i \vee A_i)$ if $l(D) >_1 l(D_i \vee A_i)$, for all $i = 1, \ldots, n$, and $\neg B_j \in I$ and $l(D) >_1 l(B_j)$ for all $j = 1, \ldots, m$.

(DWFii) $\neg D \in I$ *and for each clause* $H \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ *in* $\mathsf{ground}(\Phi)$ *with* $A \in H$ *and* $A \in D$ *(at least) one of the following conditions holds:*

(DWFiia) $\neg A_i \in I$ *and* $l(D) \geq l(A_i)$.

(DWFiib) $D' \in I$ *with* $D' \subseteq B$ *and* $l(D) >_1 l(D')$.

(DWFii') $\neg D \in I$ *and for each conditional fact* $H \leftarrow \neg B$ *in* T_Φ *with* $A \in H$ *and* $A \in D$ *(at least) one of the following conditions holds:*

(DWFiia') *there is* $H' \leftarrow \neg B'$ *in* $R \uparrow \alpha$ *with* $H' \subset H$ *and* $B' \subseteq (B \setminus D')$ *where* $A \notin H'$, $B_j \in B$, $\neg B_j \in I$, *and* $l(D) >_1 (l(B_j) + 1)$ *for all* $B_j \in D'$, *and* $l(D) >_1 (\alpha, \beta)$ *for some* β.

(DWFiib') $D' \in I$ *with* $D' \subseteq B$ *and* $l(D) >_1 l(D')$.

It is evident that (DWFiib) and (DWFiib') apply the same kind of dependency only that the former does this wrt. to one clause while the latter may employ several, i.e. (DWFiib) can be considered a special case of (DWFiib') which appears basically for easier comparison.

Theorem 5.1. *Let* Φ *be a (disjunctive) DATALOG program with disjunctive well-founded model* M. *Then, in the disjunctive knowledge ordering,* M *is the greatest model amongst all models* I *for which there exists a disjunctive* I*-partial level mapping* l *for* Φ *such that* Φ *satisfies (DWF) with respect to* I *and* l.

Example 5.2. (Example 5.1 continued) From [11] we know that if $D \in M$ then $l(D) = (\alpha, \beta)$ where α is the least ordinal such that $H \leftarrow$ in $R \uparrow \alpha$ with $H \subseteq D$ and β is the least ordinal such that the corresponding conditional fact $H \leftarrow \neg B$ in $T_\Phi \uparrow (\beta + 1)$. Furthermore, if $\neg D \in M$ then $l(D) = (\alpha, 0)$ where α is the least ordinal such that for each $A \in D$ there is no conditional fact $H \leftarrow \neg B$ in $R \uparrow \alpha$ with $A \in H$. Thus, we obtain e.g. $l(f) = (2, 0)$ by (DWFi), $l(p) = (1, 0)$ by (DWFiia'), $l(c) = (1, 0)$ by (DWFiib) and $l(e) = (1, 0)$ by (DWFiia).

Finally, we mention that $\succ$ was introduced for technical reasons in the proof to match the precise behavior of the operators [11].

6 Discussions

6.1 Comparison of the Characterizations

It was already shown in [9] that D-WFS and GDWFS satisfy five program transformation principles while SWFS does not, and that GDWFS always derives more or equal knowledge than D-WFS [14]. However, there is no similar result for D-WFS and SWFS since they are incomparable with respect to the derived knowledge (cf. our main example: SWFS derives $\neg b$ while D-WFS concludes $\neg p$).

We will now further compare the semantics on the basis of our characterizations. We will in particular attempt to obtain some insights into good general criteria for a well-founded semantics for disjunctive programs.

Level-mapping characterizations separate positive and negative information. One key insight which can be drawn from our investigations is that any characterization basically states that a true disjunction D satisfies the following scheme with respect to the model I and the program Π.

$D \in I$ and there is a clause $H \leftarrow A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m$ in $\mathsf{ground}(\Pi)$ with $H \subseteq D$ such that there is $D_i \subseteq D$ with $(D_i \vee A_i) \in I$, $l(D) > l(D_i \vee A_i)$, for all $i = 1, \ldots, n$, and $\neg B_j \in I$ and $l(D) > l(B_j)$ for all $j = 1, \ldots, m$.

We can see that this corresponds in general to (SWFia) from Definition 3.2, to (GDWFi) from Definition 4.2, and to (DWFi) from Definition 5.1. We only have to consider that the relation $>$ is technically not sufficient and that we sometimes apply a more precise order. Nevertheless, in all cases we obtain levels such that $l(D)$ is greater with respect to the specific ordering. There are further differing details. For (SWF), we have to abstract additionally from the notion of derivation sequences and their children, and there is also (SWFib) which arises from proof-theoretical treatments. In case of (GDWF) we have additionally a condition (GDWFi') but that is the part (corresponding to T_Π^D) which derives more knowledge than the well-founded semantics and should thus not be an intended result for a well-founded semantics for disjunctive programs. We claim that the condition given above is the 'disjunctive' version of (WFi) from Definition 2.1 and we propose it to be a condition for any semantics aiming to extend the well-founded semantics to disjunctive programs.

If we look for adequate extensions of (WFii) to disjunctive programs then we see that the conditions for negative information differ more. However, we still obtain straightforward extensions of (WFii) for each of the semantics only abstracting a little from the technical details. We generalize to the following scheme:

$\neg D \in I$ and for each clause $H \leftarrow A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m$ in $\mathsf{ground}(\Pi)$ with $A \in H$ and $A \in D$ (at least) one of the following conditions holds:
(iia)$\neg\alpha \in I$ and $l(D) \geq l(\alpha)$.
(iib) $D' \in I$ with $D' \subseteq B$ and $l(D) >_1 l(D')$.

For SWFS, we have (SWFiia') with $\alpha = D \vee A_i$ and (SWFiia") with $\alpha = A_i$ corresponding to (iia) depending on whether $H \subseteq D$ or $H \not\subseteq D$ but $H \cap D \neq \emptyset$[7], and (SWFiib') as equivalent to (iib). In case of GDWFS, we have (GDWFiia) and (GDWFiia') both with $\alpha = \neg(A_1 \wedge \ldots \wedge A_n)$ for (iia) and (GDWFiib) for (iib). We only have to take care that $l(D)$ has to be $l(\neg D)$ in this case since we are dealing with negated conjunctions and thus indefinite information. Finally, (DWFiia) with $\alpha = A_i$ and (DWFiib) correspond to (iia) and (iib), respectively. We claim as well that the scheme above should be part of any well-founded semantics for disjunctive programs ignoring some minor differences depending e.g. on whether we represent negative information by conjunctions or disjunctions.

Unfortunately, since positive information may be indefinite, it is also possible to obtain a correspondence to (iib) which results from several clauses (consider the program $\Pi = \{p \vee q \leftarrow; r \leftarrow s, \neg p; s \leftarrow \neg q\}$ where $\neg r$ is derivable). This is covered by (SWFiic), (GDWFii'), and (DWFiib'). Still, this is not the whole characterization for any of the semantics. (SWFiib") extends (SWFiib') to include particular atoms from the head. (GDWFii') is in fact much more powerful by

[7] Note that in both cases there is an A common to H and D.

means of the EGCWA and allows for deriving more knowledge difficult to characterize in a clause-based approach. In case of D-WFS we also have (DWFiia') which resolves the elimination of non-minimal clauses, a feature not contained in SWFS and also covered by (GDWFii') for GDWFS.

Summarising, it is obvious (and certainly expected) that it is in the derivation of negative information where the semantics differ wildly. All characterizations contain extensions of (WFii), but contain also additional non-trivial conditions some of which are difficult to capture within level mapping characterizations. The obtained uniform characterizations thus display in a very explicit manner the very different natures of the different well-founded semantics – there is simply not enough resemblance between the approaches to obtain a coherent picture. We can thus, basically, only confirm in a more formal way what has been known beforehand, namely that the issue of a good definition of well-founded semantics for disjunctive logic programs remains widely open. We believe, though, that our approach delivers structural insights which can guide the quest.

6.2 Minimal Models

As mentioned when dealing with the EGCWA appearing in GDWFS it is difficult within the level mapping framework to characterize minimal models which are the main evaluation principle for EGCWA. In the appendix of [11] it was concluded that the best possible characterization obtained for minimal models is the following:

Corollary 6.1. *([11]) Let Π be a definite disjunctive program and M be a model of Π. If there exists a total level mapping $l : B_\Pi \to \alpha$ such that for each $A \in M$ exists a clause $A \vee H_1 \vee \cdots \vee H_l \leftarrow A_1, \ldots, A_n$ in $\mathsf{ground}(\Pi)$ with $A_i \in M$, $H_k \notin M$ or $l(H_k) > l(A)$, and $l(A) > l(A_i)$, for all $i = 1, \ldots, n$ and all $k = 1, \ldots, l$, then M is a minimal model of Π.*

This is of course not a characterization but just saying that a model satisfying the given level mapping characterization is in fact minimal. Unfortunately, it is not possible to state this the other way around since there are minimal models which do not satisfy this condition[8].

Example 6.1.

$$\begin{aligned} a \vee b &\leftarrow \\ a &\leftarrow b \\ b &\leftarrow a \end{aligned}$$

This program has only one minimal model $\{a, b\}$, so according to the condition above, the first clause cannot be used since both atoms in the head are true. With the remaining two clauses we cannot have a level mapping satisfying the given condition since we must have $l(a) > l(b)$ and $l(b) > l(a)$ which is not possible.

[8] Note though that in [15] a similar result was obtained working in both directions restricted to head-cycle free programs.

Apparently, the condition imposed is too strong but all attempts (cf. [11]) to correct the problem end up with a condition too weak being satisfied also by models which are not minimal.

We can thus not apply a more precise condition for EGCWA. More generally, any semantics based on minimal models seems to fail being characterized in the framework (excluding cases like GDWFS where we simply do not treat the details of EGCWA). So surprisingly, even though disjunctive stable models are a straightforward extension of stable models, the corresponding characterization does not extend easily if at all.

It remains to be said that in opposite to that there exist characterizations for various semantic extensions of the well-founded semantics, though being rather complicated and diverse, which might allow the conclusion that (almost) any of the approaches has a better structural foundation than minimal models.

7 Conclusions

We have characterized three of the extensions of the well-founded semantics to disjunctive logic programs. It has been revealed that these characterizations are non-trivial and we have seen that they share a common derivability for true disjunctions. The conditions for deriving negative information however vary a lot. Some parts of the characterizations are common extensions of conditions used for the well-founded semantics while others cover specific deduction mechanisms occurring only in one semantics. We have obtained some structural insights into the differences and similarities of proposals for disjunctive well-founded semantics, but the main conclusion we have to draw is a negative one: Even under our formal approach which provides uniform characterizations of different semantics, the different proposals turn out to be too diverse for a meaningful comparison. The quest for disjunctive well-founded semantics thus remains widely open. Our uniform characterizations provide, however, arguments for approaching the quest in a more systematic way.

In this paper, we covered only those of the well-founded semantics which a priory appeared to be the most important and promising ones. Obviously, further insights could be obtained from considering also the remaining proposals reported e.g. in [16–20, 14].

References

1. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In Kowalski, R.A., Bowen, K.A., eds.: Logic Programming. Proceedings of the 5th International Conference and Symposium on Logic Programming, MIT Press (1988) 1070–1080
2. van Gelder, A., Ross, K.A., Schlipf, J.S.: The well-founded semantics for general logic programs. Journal of the ACM **38** (1991) 620–650
3. Lobo, J., Minker, J., Rajasekar, A.: Foundations of Disjunctive Logic Programming. MIT Press (1992)

4. Minker, J., Seipel, D.: Disjunctive logic programming: A survey and assessment. In: Essays in Honour of Robert A. Kowalski, Part I, LNAI 2407, Springer (2002)
5. Przymusinski, T.C.: Stable semantics for disjunctive programs. New Generation Computing **9** (1991) 401–424
6. Ross, K.A.: The well founded semantics for disjunctive logic programs. In Kim, W., Nicolas, J.M., Nishio, S., eds.: First International Conference on Deductive and Object Oriented Databases, Elsevier Science Publishers B.V. (North-Holland) (1990) 385–402
7. Baral, C., Lobo, J., Minker, J.: Generalized disjunctive well-founded semantics for logic programs. In Z. W. Ras, M.Z., Emrich, M., eds.: Proceedings of the Fifth International Symposium on Methodologies for Intelligent Systems. ACM SIGMOD-SIGACT-SIGART, North Holland (1989) 465–473
8. Yahya, A., Henschen, L.: Deduction in non-Horn databases. Journal Automated Reasoning **1** (1985) 141–160
9. Brass, S., Dix, J.: Disjunctive semantics based upon partial and bottom-up evaluation. In: Proceedings of the 12th International Logic Programming Conference, Tokyo, MIT Press (1995) 199–213
10. Hitzler, P., Wendt, M.: A uniform approach to logic programming semantics. Theory and Practice of Logic Programming **5** (2005) 123–159
11. Knorr, M., Hitzler, P.: A comparison of disjunctive well-founded semantics. Technical report, AIFB, University of Karlsruhe, Germany (2006) http://www.aifb.uni-karlsruhe.de/Publikationen/showPublikation?publ_id=1383.
12. Minker, J.: On indefinite databases and the closed world assumption. In: Lecture Notes in Computer Science 138, Springer, Berlin (1982) 292–308
13. Dix, J., Stolzenburg, F.: Computation of non-ground disjunctive well-founded semantics with constraint logic programming. In Dix, J., Pereira, L.M., Przymusinski, T.C., eds.: Selected Papers of the Workshop on Non-Monotonic Extensions of Logic Programming in Conjunction with Joint International Conference and Symposium on Logic Programming 1996, Springer, Berlin, Heidelberg, New York (1997) 202–224
14. Dix, J., Müller, M.: An axiomatic approach to semantics of disjunctive programs. In van Hentenryck, P., ed.: Logic Programming, Proc. of the 11th International Conference, MIT Press (1994) 303–320
15. Ben-Eliyahu, R., Dechter, R.: Propositional semantics for disjunctive logic programs. Ann. Math. Artif. Intell. **12** (1994) 53–87
16. Wang, K., Zhou, L.: Comparisons and computation of well-founded semantics for disjunctive logic programs. ACM Trans. Comput. Logic **6** (2005) 295–327
17. Alcântara, J., Damásio, C.V., Pereira, L.M.: A well-founded semantics with disjunction. In M. Gabbrielli, G.G., ed.: Proceedings of the 21st Intl.Conf. on Logic Programming (ICLP '05). LNCS 3668, Springer (2005) 341–355
18. Przymusinski, T.: Stationary Semantics for Normal and Disjunctive Logic Programs. In Delobel, C., Kifer, M., Masunaga, Y., eds.: DOOD '91, Proceedings of the 2nd International Conference, Springer, LNCS 566 (1991)
19. Przymusinski, T.C.: Static semantics for normal and disjunctive logic programs. Annals of Mathematics and Artificial Intelligence **14** (1995) 323–357
20. Baral, C., Lobo, J., Minker, J.: Wf 3 : A semantics for negation in normal disjunctive logic programs. In Ras, Z.W., Zemankova, M., eds.: Methodologies for Intelligent Systems, LNAI 542, Springer (1991) 459–468

Imagining Contexts*

Ander Altuna

School of Computing
University of Leeds, Leeds, LS2 9JT, UK
ander@comp.leeds.ac.uk

Abstract. The aim of this paper is to present a formal semantics inspired by the notion of Mental Imagery, largely researched in Cognitive Science and Experimental Psychology, that grasps the full significance of the concept of context. The outcomes presented here are considered important for both the Knowledge Representation and Philosophy of Language communities for two reasons. Firstly, the semantics that will be introduced allows to overcome some unjustified constraints imposed by previous quantificational languages of context, like flatness or the use of constant domains among others, and increases notably their expressive power. Secondly, it attempts to throw some light on the debate about the relation between meaning and truth by formally separating the conditions for a sentence to be meaningful from those that turn it true within a context.

1 Introduction

In human communication every sentence is uttered in a context and interpreted in a context. These contexts are regarded as the set of facts that hold true at the time of utterance and interpretation respectively. If a sentence refers unambiguously to a fact that is universally accepted, it will be considered true regardless of the differences between the context of the agent who uttered it and the agent who interprets it. This is the case of mathematics, which is based on an unambiguous formal language that expresses facts derived from a set of universally accepted axioms, such as the Zermelo-Fraenkel set theory. Quine [1] referred to these sentences as “eternal” and looked for a language whose sentences were all of this kind. However, in contrast with the language of mathematics, human language, and consequently that of any form of artificial intelligence, is highly dependent on context. And as a consequence of this, Quine’s project turned to be a difficult enterprise, that could result even impossible, if the notion of context is not included in the characterization of the truth function.

Although the interest in a formal theory of context within the AI community had already been present years before, there was no official research programme in this direction until in 1993 McCarthy [2] presented it as a candidate solution

* This work has been funded by Tecnalia-Labein (Building 700, Bizkaia Technology Park, 48160 Derio, Spain) under the FCT-Iñaki Goenaga doctoral scholarship programme.

for the problem of generality [3]. Since then, many logics [4–7] have emerged with the aim of capturing all of the common-sense intuitions about contextual reasoning that were introduced in [2]. However, most of these languages only deal with the propositional case and are therefore unable to treat contexts as first-class citizens included in the domain of discourse, what is one of the main desiderata behind the formalization of contexts. Only the quantificational logic of context presented in [6] is capable of formulating statements that predicate on contexts. Nevertheless, its semantics is too restrictive and imposes counter-intuitive constraints like flatness [5] or the use of constant domains [8] among others. Due to the lack of an adequate solution to the challenges posed in [2], Guha and McCarthy [9] restated the initial motivations by providing a classification on the different kinds of contexts that a satisfactory logic of context should be able to represent.

In parallel with the research in contextual reasoning developed in AI, the theory of a mental representation of ideas in the form of mental images [1] has been largely researched by cognitive scientists, experimental psychologists and philosophers [10]. Although there exist a number of controversies on how these images are formed or if after all they are images or not, the common thesis is that mind can recreate quasi-perceptual experiences similar to those that are presumed to be caused by external factors. According to the *analog* or *quasi-pictorial* theory of imagery [11], the human ability for the interpretation of symbols is equivalent to the recreation of quasi-perceptual experiences by mind. The memory of past perceptual experiences and their possible recombination are the basis of the imagery that an agent uses when interpreting a sentence.

We endorse the quasi-pictorial theory of imagery and argue that by taking it as an inspiration we can develop a logic of context that meets the challenges introduced by [2]. This inspiration is mainly realized in two features of our semantics. Firstly, in contrast with the truth-conditional theory of meaning, in our logic the meaning of a sentence will be regarded as a set of quasi-perceptual experiences instead of as a set of worlds. Secondly, a sentence will be considered to be supported by a context if its meaning is part of the image produced by the interpretation of that context. We claim that this separation between meaning and truth is necessary to grasp the concept of context in its full extent.

In this paper we present a semantics that formalizes a conceptualization of a quasi-pictorial theory of Mental Imagery by which it notably increases the expressiveness of previous logics of context and overcomes some difficulties posed by them. The paper is structured as follows. First, we introduce informally the main features of the logic and compare it with previous logics of context and other formalisms. Second, the language of our logic and its formation rules are described. Third, we define a model of interpretation inspired by a conceptualization of a theory of imagery and subsequently explain the associated theory of

[1] It must be noted that all along this document we do not use the term “image” with a static connotation but we refer to both instantaneous and durative quasi-perceptual recreations. Besides, it is not limited to visual experiences but to all the kinds of experiences an agent can perceive through its senses.

meaning, together with the characterization of the truth function and the way we resolve traditional problems like existence and denotation. We end the paper by extracting the conclusions and envisaging some future work that we plan to undertake in this line of research.

2 The Logic

Our logic cannot simply be defined as an extension to predicate calculus, because there are some fundamental aspects in the semantics that turn it very different from the classical model theoretic semantics of first order logic. However, we can compare the expressive power of both logics and say that the logic presented here increases the expressiveness of predicate calculus with identity by adding the following capabilities:

1. Like in previous logics of context [4–7], formulas can be stated in a context. Therefore, there is no contradiction in asserting a formula and its negation while they are in different contexts. However, in contrast with those logics of context based on the *ist* predicate [4–6], we do not force every sentence to be preceded by a context. Instead, if a sentence is not preceded by a context, it will be assumed to be a description of actuality. The reason for this is that, unlike [2], we do not judge intuitively correct to state that actuality can be transcended and therefore we consider it to be the outermost context. Nevertheless, this does not make our logic differ on the transcendence capabilities described in [4–6], because what it is claimed as unlimited transcendence by these approaches is actually limited in each context tree by its respective initial context k_0.
2. Formulas in our language can refer to contexts and quantify over them like in [4, 6]. In our logic it is not allowed, however, to predicate on any context, but only on those that are accessible from the context under which the formula in question is being asserted. References to non-accessible contexts will therefore fail to denote.
3. A given contextualized formula can be quoted or not depending on the context in terms of which that formula is being expressed. In order to express a formula in terms of the context in which is being contextualized it will need to be quoted. Otherwise, it will be assumed that the meaning of the terms used on that formula correspond to the outer context. The use of quotation marks in a formula will therefore allow for the abstraction from the meaning of its terms and the disambiguation of the indexicals it may contain.
4. Like suggested in [12], we differentiate between internal and external negations. While the external negation of a formula can be satisfied even if its terms fail to denote or the formula is meaningless, the internal negation of a formula requires that it is meaningful and its terms succeed to denote in the context in which it is being asserted.
5. Formulas can express a parthood relation between the references of two terms. This relation will result particularly useful when formalizing normalcy assumptions between contexts [9]. If a context is said to be part of other

context and the latter supports a set of normalcy conditions expressed in the form of universal quantifications, all these conditions will consequently become normalcy assumptions in the former context.

In addition to the mentioned expressive capabilities, the semantics we present overcomes some counter-intuitive restrictions imposed by [6] and adds some novel ways of dealing with meaningless sentences, existence and designation. Below are roughly introduced the fundamental aspects that characterize this semantics:

1. An image is a mereologically structured object.
2. The meaning of the non-logical symbols of our logic ranges over a set that contains the imagery an agent possesses. This set is partially ordered according to two mereological parthood relations that will be introduced in the next section. In terms of possible worlds semantics, the imagery set is equivalent to a kind of *possibilia*. The meaning function assigns to each constant symbol a subset of the imagery containing all the possible counterparts [13] that it can denote.
3. Like [6] we differentiate between individuals of the *discourse* sort and the *context* sort. Like the rest of individuals contexts are interpreted as images mereologically structured. The domain of discourse of a context consists of its set of grounded parts. Therefore, each context defines in a natural and flexible way its own domain of discourse over which the denotation of terms of the discourse sort ranges. This domain is equivalent to *actuality*. In a given context the denotation of terms of the context sort ranges over the set of contexts that are accessible from it, which is equivalent to its set of figured parts.
4. In contrast with Intensional Logic [14–16], the denotation of constant symbols is not a function from contexts or states to members of the domain of discourse. Instead, the object denoted by a constant is the unique member of the intersection of the meaning of that constant and the grounded part expansion of the context in question, if it is of the discourse sort, or the set of contexts that are accessible from it, if it is of the context sort. This will help to determine whether a symbol succeeds to denote under a certain context.
5. Unlike in [6], there is no flatness restriction among contexts. In other words, the set of axioms holding at a particular context depends on the context from which it is accessed.

We will make use of classical Extensional Mereology [17] for the elaboration of the semantics. The binary relation *"is part of"* will be represented by the symbol $\preccurlyeq$ in our model. Therefore, if an object x is said to be part of an object y, we will write $x \preccurlyeq y$. In addition to the classical operators of mereology we will make use of the part-expansion of an object. This operation is defined below.

Definition 1. *Given an object Γ, its part-expansion $\downarrow \Gamma$ is the set containing every part of Γ.*

$$\downarrow \Gamma = \{x : x \preccurlyeq \Gamma\} \tag{1}$$

3 Formal System

3.1 Syntax

A language $\mathcal{L}$ of our logic is any language of classical two-sorted predicate calculus with identity and a infinite set of non-logical symbols, together with a parthood relation and a set of symbols to express the contextualization, the quotation, and the internal negation of a formula. For simplicity we will make no use of functions. Below is the list of logical symbols of our language and the notational convention we will use for the non-logical ones:

1. n-ary predicate symbols: P^n , P_1^n , P_2^n , ...
2. Constants of the discourse sort: a , a_1 , a_2 , ...
3. Constants of the context sort: k , k_1 , k_2 , ...
4. Variables of both sorts: x , x_1 , x_2 , ...
5. External and internal negation: $\neg$, $^-$
6. Connectives: $\vee$, $\wedge$, $\supset$
7. Quantifiers: $\forall$, $\exists$
8. Identity: $=$
9. Parthood: $\leq_g$, $\leq_f$
10. Quotation marks: " , "
11. Auxiliary symbols: : , [,]

Given a language $\mathcal{L}$, we will use $\mathbb{C}$ to refer to the set of constants of the discourse sort, and $\mathbb{K}$ to refer to the set of constants of the context sort. The set of variables of both sorts will be given by $\mathbb{V}$, while $\mathbb{P}$ will denote the set of predicates.

Definition 2. *The set of terms $\mathbb{T}$ and well-formed formulas (wffs) $\mathbb{W}$ are inductively defined on their construction by using the following formation rules:*

1. *Each variable or constant of any sort is a term.*
2. *If $t_1, \ldots, t_n$ are terms and P^n is an n-ary predicate, then $P^n(t_1, \ldots, t_n)$ and $\overline{P^n}(t_1, \ldots, t_n)$ are wffs.*
3. *If t_1 and t_2 are terms, then $t_1 = t_2$, $t_1 \leq_g t_2$ and $t_1 \leq_f t_2$ are wffs.*
4. *If A is a wff, then $[\neg A]$ is a wff.*
5. *If A and B are wffs, then $[A \vee B]$, $[A \wedge B]$ and $[A \supset B]$ are wffs.*
6. *If A is a wff and x is a variable of any sort, then $(\forall x)[A]$ and $(\exists x)[A]$ are wffs.*
7. *If A is a wff and k is a constant of the context sort, then $[k : A]$ and $[k : \text{“}A\text{”}]$ are wffs.*

It must be noted that the treatment of the parthood relation as a logical symbol of our logic entails that its axiomatization as a transitive, reflexive and antisymmetric relation will be included in the set of axioms of the logic itself. We will refer to the axioms of Extensional Mereology [17] for this.

3.2 A Model of Interpretation

In our attempt to elaborate a formal semantics inspired by a quasi-pictorial theory of mental imagery, we consider that an image is a mereologically structured object and therefore it is a whole composed of parts. An image will be said to model the set of facts that its parts support and consequently the truth value assigned to a sentence will be relativized to the context under which is being considered. However, we will differentiate between two kinds of parts of which images may consist, namely *grounded* and *figured* parts. It is easy to understand the intuition behind this differentiation if we consider an example in which an agent is situated in an augmented-reality scenario. In this situation the agent will perceive some objects as genuine parts of reality and others as artificial objects recreated by some kind of device. We will say that the former objects are part of the actuality constructed by this agent in a *grounded* sense while the latter objects are part of the actuality constructed by this agent in a *figured* sense. Our intuition is that contexts, like those artificially recreated objects of the example, exist and are part of reality in a figured sense.

In order to capture these two different senses of parthood, the model structure will include a grounded parthood relation and a figured parthood relation. While the former will determine the domain of those objects of the discourse sort, the latter will determine the domain of those objects of the context sort and their accessibility. A formal definition of the model structure is given below.

Definition 3. *In this system a model, $\mathfrak{M}$, is a structure $\mathfrak{M} = \langle \mathcal{I}, \preccurlyeq_g, \preccurlyeq_f, \Omega, \mathcal{M} \rangle$ whose components are defined as follows:*

1. *$\mathcal{I}$ is a non-empty set. It consists of all the imagery an agent can recreate at the moment she is performing the interpretation.*
2. *$\preccurlyeq_g$ is a partial ordering on $\mathcal{I}$. It is therefore a transitive, reflexive, and antisymmetric relation on $\mathcal{I}$. It denotes the mereological parthood relation on the members of $\mathcal{I}$ in a grounded sense.*
3. *$\preccurlyeq_f$ is a partial ordering on $\mathcal{I}$. It is therefore a transitive, reflexive, and antisymmetric relation on $\mathcal{I}$. It denotes the mereological parthood relation on the members of $\mathcal{I}$ in a figured sense.*
4. *Ω is a distinguished member of $\mathcal{I}$. It represents the image of actuality constructed by the agent performing the interpretation.*
5. *$\mathcal{M}$ is a function from the non-logical symbols of $\mathcal{L}$ to a mapping from members of $\mathcal{I}$ to subsets of $\mathcal{I}$. $\mathcal{M}$ denotes the meaning function that assigns each constant of $\mathcal{L}$ a mapping from contexts to its set of possible denotations and each predicate of $\mathcal{L}$ a mapping from contexts to its extension over $\mathcal{I}$. In the definition of $\mathcal{M}$ given below we use the standard mathematical notation $\mathcal{P}(X)$ to refer to the powerset of X.*

$$\mathcal{M} : \begin{cases} \mathbb{C} \rightarrow [\mathcal{I} \rightarrow \mathcal{P}(\mathcal{I})] \\ \mathbb{K} \rightarrow [\mathcal{I} \rightarrow \mathcal{P}(\mathcal{I})] \\ \mathbb{P} \rightarrow [\mathcal{I} \rightarrow \mathcal{P}(\mathcal{I}^n)] \end{cases} \tag{2}$$

3.3 Meaning

As introduced in the previous section, the meaning of a non-logical symbol does not depend on the context under which the truth of the statement in which it occurs is being considered. We consider that the meaning of a term is a subset of $\mathcal{I}$ that contains all the images that term can possibly denote. This is the set of possible counterparts [13] that term stands for. In the same way, the meaning of a predicate symbol is defined as a subset of $\mathcal{I}^n$. The meaning of those non-logical symbols that are not included in the vocabulary of an agent will be equivalent to the empty set. We will refer to these symbols as meaningless.

On the other hand, the meaning assigned to a constant or a predicate will vary depending on whether the sentence is being asserted using the terms of one context or the terms of other. This is the reason why the introduction of quotation marks is important. A context will have to quote the report of a sentence in order to dissociate itself from the meaning given to the non-logical symbols in that sentence. This resolves the problems with statements containing ambiguous references [9].

Assignment. We proceed to define the assignment function in our logic.

Definition 4. *If x is a variable of any sort, an assignment into $\mathfrak{M}$ is a function φ such that $\varphi(x)$ is a subset of $\mathcal{I}$.*

$$\varphi : \mathbb{V} \to [\mathcal{I} \to \mathcal{P}(\mathcal{I})] \tag{3}$$

It will be useful to introduce the concept of x-variant assignment for the characterization of the truth function that will be presented in the next section.

Definition 5. *An assignment ψ is an x-variant of an assignment φ if φ and ψ agree on all variables except possibly x.*

Valuation. As usual we will define the valuation of the non-logical symbols of our logic in terms of the assignment and meaning functions. However, as we have mentioned before, the valuation of the terms of the logic will not yet result in the denotation of these, because the latter is relativized to the context under which the truth of a particular formula is being considered. We will define this notion more formally in the next section.

Definition 6. *Given a model $\mathfrak{M} = \langle \mathcal{I}, \preccurlyeq_g, \preccurlyeq_f, \Omega, \mathcal{M} \rangle$, an assignment φ and a context Δ member of $\mathcal{I}$, a valuation $\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}$ of the non-logical symbols of our language into $\mathfrak{M}$ under φ and in terms of Δ is defined as follows:*

1. $\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(t) = \varphi(t)(\Delta)$ *if t is a variable.*
2. $\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(t) = \mathcal{M}(t)(\Delta)$ *if t is a constant of any sort.*
3. $\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(P^n) = \mathcal{M}(P^n)(\Delta)$ *if P^n is an n-ary predicate.*

Meaningful Formula. We do not need to check the truth of a formula with regard to a context in order to know whether it is meaningful or not. This will only depend on the valuation of the terms and predicates it contains. Informally, we will say that a sentence is meaningful with regard to a model constructed by an agent if this agent can recreate some image for each of the terms in the sentence and at least one of these images is included in the set to which she would attribute the predicate in question. For example, let us take the sentence "the smell of your jacket is red". If "the smell of your jacket" and "red" are interpreted as they are usually in English, an English speaking agent will not be able to recreate an image of the smell of someone's jacket that is included in the set of images to which she would attribute the red colour. Therefore we will say that this sentence is meaningless for that agent. Below is presented a more formal definition of meaningful formula:

Definition 7. *An atomic wff expressing the P^n-ness of a sequence of terms $t_1, \ldots, t_n$ in terms of a context Δ is said to be meaningful in a model $\mathfrak{M}$ if and only if there exists some assignment φ such that the cardinality of the intersection of the cartesian product of the valuations of $t_1, \ldots, t_n$ under φ in terms of k and the valuation of P^n under φ in terms of Δ is equal or greater than one.*

$$\begin{aligned} &P^n(t_1, \ldots, t_n) \text{ is a meaningful formula iff} \\ &\quad \left|\left[\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(t_1) \times \cdots \times \mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(t_n)\right] \cap \mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(P^n)\right| \geq 1 \end{aligned} \tag{4}$$

Note that the condition that the valuation of each of the non-logical symbols included in the sentence must be different from the empty set is implicit in this definition. Therefore, if a sentence is to be meaningful in a model, all of its non-logical symbols must be meaningful in that model as well.

This definition can be extended to sentences expressing the parthood or identity relation between two terms. The set of meaningful formulas will be trivially defined by induction on their construction.

3.4 Truth

The meaning of the non-logical symbols of a formula cannot determine by itself its truth value. In our logic the truth value of a formula is relativized to the context in which it is asserted. As we have said in the definition of a model in our logic, the image of a context supports the set of facts that are supported by its parts. Therefore, the first requisite for a formula to be supported by a context is that at least one counterpart of each of the terms of that sentence is part of the image of that context. On the other hand, one and only one counterpart can be part of the image of the same context or otherwise the term will be an ambiguous designator in that context. In this section, we define in what conditions a term succeeds to denote when it is used in a particular context and how the truth function is characterized according to this definition of denotation.

Denotation. Below we define formally the conditions under which a term succeeds to denote when considered under a certain context.

Definition 8. *A term t succeeds to denote into a model $\mathfrak{M}$ under an assignment φ in terms of a context Δ when considered under a context Γ if and only if the cardinality of the intersection of the valuation of t under φ in terms of Δ with the grounded part-expansion of Γ, if t is of the discourse sort, or the figured part-expansion of Γ, if t is of the context sort, is a singleton.*

$$\begin{cases} \left|\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(t) \cap \downarrow_g \Gamma\right| = 1 & \textit{if } t \textit{ is of the discourse sort} \\ \left|\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(t) \cap \downarrow_f \Gamma\right| = 1 & \textit{if } t \textit{ is of the context sort} \end{cases} \tag{5}$$

Definition 9. *A term t fails to denote into a model $\mathfrak{M}$ under an assignment φ in terms of a context Δ when considered under a context Γ if and only if the cardinality of the intersection of the valuation of t under φ in terms of Δ with the grounded part-expansion of Γ, if t is of the discourse sort, or the figured part-expansion of Γ, if t is of the context sort, is zero.*

$$\begin{cases} \left|\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(t) \cap \downarrow_g \Gamma\right| = 0 & \textit{if } t \textit{ is of the discourse sort} \\ \left|\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(t) \cap \downarrow_f \Gamma\right| = 0 & \textit{if } t \textit{ is of the context sort} \end{cases} \tag{6}$$

Definition 10. *A term t is an ambiguous designator into a model $\mathfrak{M}$ under an assignment φ in terms of a context Δ when considered under a context Γ if and only if the cardinality of the intersection of the valuation of t under φ in terms of Δ with the grounded part-expansion of Γ, if t is of the discourse sort, or the figured part-expansion of Γ, if t is of the context sort, is greater that one.*

$$\begin{cases} \left|\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(t) \cap \downarrow_g \Gamma\right| > 1 & \textit{if } t \textit{ is of the discourse sort} \\ \left|\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(t) \cap \downarrow_f \Gamma\right| > 1 & \textit{if } t \textit{ is of the context sort} \end{cases} \tag{7}$$

If a term succeeds to denote then we will say that it denotes that image that, at the same time, is in its meaning and is part of the image of the context in consideration. This is formally defined below.

Definition 11. *If a term t succeeds to denote into a model $\mathfrak{M}$ under an assignment φ in terms of a context Δ when considered under a context Γ, its denotation $\mathcal{V}^{\mathfrak{M},\Gamma}_{\varphi,k}(t)$ into $\mathfrak{M}$ under φ in terms of Δ when considered under Γ is that unique element that is member of the intersection of the valuation of t under φ in terms of Δ with the grounded part-expansion of Γ, if t is of the discourse sort, or the figured part-expansion of Γ, if t is of the context sort.*

$$\mathcal{V}^{\mathfrak{M},\Gamma}_{\varphi,\Delta}(t) =_{def} (\iota x) \begin{cases} x \in \left[\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(t) \cap \downarrow_g \Gamma\right] & \textit{if } t \textit{ is of the discourse sort,} \\ x \in \left[\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(t) \cap \downarrow_f \Gamma\right] & \textit{if } t \textit{ is of the context sort.} \end{cases} \tag{8}$$

Truth Function. Once the conditions under which a term succeeds to denote and the value that takes its denotation have been defined, we can proceed to characterize the truth function on a model $\mathfrak{M}$ by induction on the construction of the wffs of our logic.

Definition 12. *Truth ($\Vdash$), with respect to an assignment φ into a model $\mathfrak{M} = \langle \mathcal{I}, \preccurlyeq_g, \preccurlyeq_f, \Omega, \mathcal{M} \rangle$, is characterized as follows:*

1. *A context Γ included in the imagery $\mathcal{I}$ of a model $\mathfrak{M}$ supports the assertion [internal negation] of the P^n-ness of a sequence of terms $t_1, \ldots, t_n$ under an assignment φ in terms of a context Δ included in $\mathcal{I}$ if and only if every term $t_1, \ldots, t_n$ succeeds to denote under φ in terms of Δ when considered under Γ and the tuple formed by the denotations of $t_1, \ldots, t_n$ under φ in terms of Δ when considered under Γ belongs to [the complement of] the valuation of P^n under φ in terms of Δ.*

$$\begin{aligned} &\mathfrak{M}, \Gamma \Vdash_{\varphi,\Delta} P^n(t_1, \ldots, t_n) \textit{ iff} \\ &\quad t_1, \ldots, t_n \textit{ succeed to denote under } \varphi \textit{ in terms of } \Delta \textit{ in } \Gamma \\ &\quad \textit{and } \langle \mathcal{V}^{\mathfrak{M},\Gamma}_{\varphi,\Delta}(t_1), \ \ldots \ , \mathcal{V}^{\mathfrak{M},\Gamma}_{\varphi,\Delta}(t_n) \rangle \in \mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(P^n) \end{aligned} \tag{9}$$

$$\begin{aligned} &\mathfrak{M}, \Gamma \Vdash_{\varphi,\Delta} \overline{P^n}(t_1, \ldots, t_n) \textit{ iff} \\ &\quad t_1, \ldots, t_n \textit{ succeed to denote under } \varphi \textit{ in terms of } \Delta \textit{ in } \Gamma \\ &\quad \textit{and } \langle \mathcal{V}^{\mathfrak{M},\Gamma}_{\varphi,\Delta}(t_1), \ \ldots \ , \mathcal{V}^{\mathfrak{M},\Gamma}_{\varphi,\Delta}(t_n) \rangle \in \left[\mathcal{V}^{\mathfrak{M}}_{\varphi,\Delta}(P^n)\right]^C \end{aligned} \tag{10}$$

2. *A context Γ included in the imagery $\mathcal{I}$ of a model $\mathfrak{M}$ supports the assertion of the identity relation between two terms t_1 and t_2 under an assignment φ in terms of a context Δ included in $\mathcal{I}$ if and only if t_1 and t_2 succeed to denote under φ in terms of Δ when considered under Γ and the denotations of t_1 and t_2 under φ in terms of Δ when considered under Γ are equal.*

$$\begin{aligned} &\mathfrak{M}, \Gamma \Vdash_{\varphi,\Delta} t_1 = t_2 \textit{ iff} \\ &\quad t_1 \textit{ and } t_2 \textit{ are proper descriptions under } \varphi \textit{ in terms of } \Delta \textit{ in } \Gamma \\ &\quad \textit{and } \mathcal{V}^{\mathfrak{M},\Gamma}_{\varphi,\Delta}(t_1) = \mathcal{V}^{\mathfrak{M},\Gamma}_{\varphi,\Delta}(t_2) \end{aligned} \tag{11}$$

3. *A context Γ included in the imagery $\mathcal{I}$ of a model $\mathfrak{M}$ supports the assertion of the grounded[figured] parthood relation of a term t_1 into a term t_2 under an assignment φ in terms of a context Δ included in $\mathcal{I}$ if and only if t_1 and t_2 succeed to denote under φ in terms of Δ when considered under Γ and the denotation of t_1 under φ in terms of Δ when considered under Γ is a grounded[figured] part of the denotation of t_2 under φ in terms of Δ when*

considered under Γ.

$$\mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} t_1 \leq_g t_2 \textit{ iff } t_1 \textit{ and } t_2 \textit{ succeed to denote under } \varphi \textit{ in terms of } \Delta \textit{ in } \Gamma \textit{ and } \mathcal{V}^{\mathfrak{M},\Gamma}_{\varphi,\Delta}(t_1) \preccurlyeq_g \mathcal{V}^{\mathfrak{M},\Gamma}_{\varphi,\Delta}(t_2) \quad (12)$$

$$\mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} t_1 \leq_f t_2 \textit{ iff } t_1 \textit{ and } t_2 \textit{ succeed to denote under } \varphi \textit{ in terms of } \Delta \textit{ in } \Gamma \textit{ and } \mathcal{V}^{\mathfrak{M},\Gamma}_{\varphi,\Delta}(t_1) \preccurlyeq_f \mathcal{V}^{\mathfrak{M},\Gamma}_{\varphi,\Delta}(t_2) \quad (13)$$

4. *A context* Γ *included in the imagery* $\mathcal{I}$ *of a model* $\mathfrak{M}$ *supports the external negation of a formula* A *under an assignment* φ *in terms of a context* Δ *included in* $\mathcal{I}$ *if and only if it does not support* A *under* φ *in terms of* Δ.

$$\mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} \neg A \textit{ iff not } \mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} A \quad (14)$$

5. *The following clauses are defined as usual.*

$$\mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} A \wedge B \textit{ iff } \mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} A \textit{ and } \mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} B \quad (15)$$

$$\mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} A \vee B \textit{ iff } \mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} A \textit{ or } \mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} B \quad (16)$$

$$\mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} A \supset B \textit{ iff not } \mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} A \textit{ or } \mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} B \quad (17)$$

6. *A context* Γ *included in the imagery* $\mathcal{I}$ *of a model* $\mathfrak{M}$ *supports the universal quantification of a variable* x *in a formula* A *under an assignment* φ *in terms of a context* Δ *included in* $\mathcal{I}$ *if and only if* Γ *supports* A *for every* x*-variant assignment* ψ *under which* x *succeeds to denote in terms of* Δ *when considered under* Γ.

$$\mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} (\forall x)[A] \textit{ iff for every } x\textit{-variant assignment } \psi\textit{, if } x \textit{ succeeds to denote under } \psi \textit{ in terms of } \Delta \textit{ in } \Gamma \textit{ then } \mathfrak{M},\Gamma \Vdash_{\psi,\Delta} A \quad (18)$$

7. *A context* Γ *included in the imagery* $\mathcal{I}$ *of a model* $\mathfrak{M}$ *supports the existential quantification of a variable* x *in a formula* A *under an assignment* φ *in terms of a context* Δ *included in* $\mathcal{I}$ *if and only if* Γ *supports* A *under some* x*-variant assignment* ψ *under which* x *succeeds to denote in terms of* Δ *when considered under* Γ.

$$\mathfrak{M},\Gamma \Vdash_{\varphi,\Delta} (\exists x)[A] \textit{ iff for some } x\textit{-variant assignment } \psi\textit{, } x \textit{ succeeds to denote under } \psi \textit{ in terms of } \Delta \textit{ in } \Gamma \textit{ and } \mathfrak{M},\Gamma \Vdash_{\psi,\Delta} A \quad (19)$$

8. *A model* $\mathfrak{M}$ *supports a formula* A *under an assignment* φ *if and only if the image of actuality* Ω *in* $\mathfrak{M}$ *supports* A *under an assignment* φ *in terms of* Ω.

$$\mathfrak{M} \Vdash_{\varphi} A \textit{ iff } \mathfrak{M},\Omega \Vdash_{\varphi,\Omega} A \quad (20)$$

9. *A context $\mathit{\Gamma}$ included in the imagery $\mathcal{I}$ of a model $\mathfrak{M}$ supports a formula A contextualized in a context k under an assignment φ in terms of a context $\mathit{\Delta}$ included in $\mathcal{I}$ if and only if k succeeds to denote under φ in terms of $\mathit{\Delta}$ when considered under $\mathit{\Gamma}$ and its denotation under φ in terms of $\mathit{\Delta}$ when considered under $\mathit{\Gamma}$ supports A under φ in terms of $\mathit{\Delta}$.*

$$\begin{aligned}&\mathfrak{M},\mathit{\Gamma} \Vdash_{\varphi,\mathit{\Delta}} k : A \textit{ iff} \\ &\quad k \textit{ succeeds to denote under } \psi \textit{ in terms of } \mathit{\Delta} \textit{ in } \mathit{\Gamma} \\ &\quad \textit{and } \mathfrak{M}, \mathcal{V}^{\mathfrak{M},\mathit{\Gamma}}_{\varphi,\mathit{\Delta}}(k) \Vdash_{\varphi,\mathit{\Delta}} A\end{aligned} \tag{21}$$

10. *A context $\mathit{\Gamma}$ included in the imagery $\mathcal{I}$ of a model $\mathfrak{M}$ supports the quotation a formula A contextualized in a context $\mathit{\Delta}$ under an assignment φ in terms of a context $\mathit{\Delta}$ included in $\mathcal{I}$ if and only if $\mathit{\Delta}$ succeeds to denote under φ in terms of $\mathit{\Delta}$ when considered under $\mathit{\Gamma}$ and its denotation under φ in terms of $\mathit{\Delta}$ when considered under $\mathit{\Gamma}$ supports A under φ in terms of its denotation under φ in terms of $\mathit{\Delta}$ when considered under $\mathit{\Gamma}$.*

$$\begin{aligned}&\mathfrak{M},\mathit{\Gamma} \Vdash_{\varphi,\mathit{\Delta}} k : \text{“}A\text{”} \textit{ iff} \\ &\quad k \textit{ succeeds to denote under } \psi \textit{ in terms of } \mathit{\Delta} \textit{ in } \mathit{\Gamma} \\ &\quad \textit{and } \mathfrak{M}, \mathcal{V}^{\mathfrak{M},\mathit{\Gamma}}_{\varphi,\mathit{\Delta}}(k) \Vdash_{\varphi,\mathcal{V}^{\mathfrak{M},\mathit{\Gamma}}_{\varphi,\mathit{\Delta}}(k)} A\end{aligned} \tag{22}$$

Definition 13. *A formula A is said to be valid if and only if it is supported by every image $\mathit{\Gamma}$ of every model $\mathfrak{M}$ under every assignment φ and in terms of every context $\mathit{\Delta}$.*

$$\Vdash A \textit{ iff } (\forall \mathfrak{M})\,(\forall \mathit{\Gamma})\,(\forall \varphi)\,(\forall \mathit{\Delta})\,[\mathfrak{M},\mathit{\Gamma} \Vdash_{\varphi,\mathit{\Delta}} A] \tag{23}$$

As can be appreciated from the definition of the truth function, the principle of bivalence holds with regard to both external and internal negations. However, the bivalence with respect to the internal negation of a formula under a certain context holds if and only if all its terms succeed to denote when considered under that context and the sentence is meaningful, while the bivalence with regard to the external negation of a formula holds regardless of these conditions. Therefore the validity of the principle of bivalence can only be determined locally in the case of internal negations. This differentiation resolves how to deal with foreign languages [7] in the quantificational case. Concretely, this means that if a sentence is expressed in a language different from the one an agent knows, then the actuality constructed by this agent will not support that sentence neither its internal negation. Below are formally expressed the principles of bivalence with regard to both kinds of negation.

$$\begin{aligned}&\mathfrak{M},\mathit{\Gamma} \Vdash_{\varphi,\mathit{\Delta}} P^n(t_1,\ldots,t_n) \vee \overline{P^n}(t_1,\ldots,t_n) \text{ iff} \\ &\quad t_1,\ldots,t_n \text{ are proper descriptions under } \varphi \text{ in terms of } \mathit{\Delta} \text{ in } \mathit{\Gamma} \\ &\quad \text{and } P^n(t_1,\ldots,t_n) \text{ is a meaningful sentence.}\end{aligned} \tag{24}$$

$$\Vdash A \vee \neg A \quad \text{in any case.} \tag{25}$$

On the other hand, in line with Modal Realism [13], we treat the universal quantifier as implicitly ranging over actuality. Therefore, as can be seen in the equation (16), only those assignments under which x succeeds to denote in the context in consideration are required to validate the quantified formula.

The equations (18) and (19) show how this semantics facilitates entering into an inner context from a relative actuality and reversely transcending back from it. As can be seen in the equation (19), when entering a context the use of quotation marks entails the change of the context in terms of which the non-logical terms are valuated by the one in which we are entering.

4 Conclusions

In this paper we have presented a formal semantics for a logic of context that is inspired by a quasi-pictorial theory of Mental Imagery [11], which is a very active research area in the disciplines of Cognitive Science and Experimental Psychology. The semantics we have elaborated not only addresses how to interpret the reasoning between contexts but also increases the expressivity of previous logics of context by adding some new constructors to the set of logical symbols. Among these are the quotation marks that, like in natural language, enable an agent to use the terms in which another agent expresses herself and the parthood relation, which results very useful when formalizing normalcy assumptions between contexts. On the other hand, we have shown a characterization of the truth function that allows the differentiation between external and internal negations [12], what is necessary in order to adjust the principle of bivalence to the case of meaningless sentences or foreign languages [7].

Besides, this semantics has proved to overcome some unjustified restrictions that were imposed by previous quantificational logics of context [6], like flatness or the use of constant domains among others. This makes our logic more intuitively appropriate for accommodating the concept of context that Guha and McCarthy restated in [9].

From the point of view of the Philosophy of Language, we have elaborated a theory of meaning that provides a novel solution to the classical problems of meaningless sentences, designation and existence. The separation between meaning and truth that we have formalized allows to identify these cases and to deal with them adequately when it comes to evaluate the truth value of the sentences of our language.

At the moment of writing this paper, we are looking into a complete and sound axiomatization that allows to give a definition of derivability adequate for our logic. We also plan to research into how to accommodate temporal concepts, like events or actions, in this formalism.

References

1. Quine, W.V.O.: Propositional objects. In: Ontological Relativity and Other Essays (J.Dewey Essays in Philosophy). Columbia University Press, New York (1969) 139–160

2. McCarthy, J.: Notes on formalizing contexts. In Bajcsy, R., ed.: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, San Mateo, California, Morgan Kaufmann (1993) 555–560
3. McCarthy, J.: Generality in artificial intelligence. In Lifschitz, V., ed.: Formalizing Common Sense: Papers by John McCarthy. Ablex Publishing Corporation, Norwood, New Jersey (1990) 226–236
4. Guha, R.V.: Contexts: a formalization and some applications. PhD thesis, Stanford University, California, USA (1992)
5. Buvac, S., Buvac, V., Mason, I.A.: Metamathematics of contexts. Fundamenta Informaticae **23**(2/3/4) (1995) 263–301
6. Buvac, S.: Quantificational logic of context. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence. (1996) 600–606
7. Ghidini, C., Giunchiglia, F.: Local models semantics, or contextual reasoning = locality + compatibility. Artificial Intelligence **127**(2) (2001) 221–259
8. Fitting, M., Mendelshon, R.L.: First Order Modal Logic. Kluwer Academic Publishers, Dordrecht (1998)
9. Guha, R., McCarthy, J.: Varieties of context. In Blackburn, P., Ghidini, C., Turner, R.M., Giunchiglia, F., eds.: Modeling and Using Context: Proceedings of the Fourth International and Interdisciplinary Conference, Context 2003, Berlin, Springer-Verlag (2003) 164–177
10. Thomas, N.J.T.: Mental Imagery, Philosophical Issues About. In: Encyclopedia of Cognitive Science. Volume 2. Nature Publishing/Macmillan (2003) 1147–1153
11. Kosslyn, S.M.: Mental images and the brain. Cognitive Neuropsychology **22** (2005) 333–347
12. Slater, B.H.: Internal and external negations. Mind **88** (1979) 588–591
13. Lewis, D.: On the Plurality of Worlds. Blackwell Publishing (1986)
14. Montague, R.: Formal Philosophy: Selected Papers of Richard Montague. Yale University Press, New Haven (1974)
15. Gallin, D.: Intensional and Higher-Order Logic. North-Holland Publishing Company, Amsterdam (1975)
16. Thomason, R.H.: Type theoretic foundations for context, part 1: Contexts as complex type-theoretic objects. In Bouquet, P., Serafini, L., Benerecetti, P.B.M., Castellani, F., eds.: Modeling and Using Contexts: Proceedings of the Second International and Interdisciplinary Conference, Context 1999, Berlin, Springer-Verlag (1999) 352–374
17. Simons, P.: Parts: A Study in Ontology. Oxford University Press (2003)

Description Logic Programs: Normal Forms*

Pascal Hitzler[1] and Andreas Eberhart[2]

[1] AIFB, University of Karlsruhe
[2] Hewlett-Packard Germany

Abstract. The relationship and possible interplay between different knowledge representation and reasoning paradigms is a fundamental topic in artificial intelligence. For expressive knowledge representation for the Semantic Web, two different paradigms – namely Description Logics (DLs) and Logic Programming – are the two most successful approaches. A study of their exact relationships is thus paramount.
An intersection of OWL with (function-free non-disjunctive) Datalog, called DLP (for *Description Logic Programs*), has been described in [1, 2]. We provide normal forms for DLP in Description Logic syntax and in Datalog syntax, thus providing a bridge for the researcher and user who is familiar with either of these paradigms. We argue that our normal forms are the most convenient way to *define* DLP for teaching and dissemination purposes.

1 Introduction

The Web Ontology Language OWL[3] [3] has been recommended by the W3C consortium as a standard for the Semantic Web. Based on Description Logics [4], it provides a sound foundation for the development of sophisticated Semantic Web technology. It is however understood that the expressivity of OWL lacks certain features which can most naturally be covered by rule-based approaches akin to Logic Programming [5], like F-Logic and its variants [6]. At the same time, pure Logic Programming based approaches to ontology modelling are also being used in practice, in particular in the form of F-Logic. Providing interoperability between the two paradigms is thus of practical importance.

In [1] the intersection between OWL-DL and function-free disjunctive Datalog has been described, and called *Description Logic Programs* (DLP). Since then, this paradigm has been extended considerably. Most notably, it has been developed into an efficient and flexible reasoning system using techniques from disjunctive Datalog for OWL-DL reasoning [7–10] — including the translation of a major fragment of OWL-DL to disjunctive Datalog. But it has also been used in different contexts, e.g. for defining OWL Light$^-$ [11].

---

* The authors acknowledge support by the German Federal Ministry for Education and Research under the SmartWeb project (01 IMD01 B), by the Deutsche Forschungsgemeinschaft (DFG) under the ReaSem project, and by the EU in the IST project NeOn (IST-2006-027595, http://www.neon-project.org/)).

[3] http://www.w3.org/2004/OWL/

At the same time, DLP has been a focus of discord in the scientific dispute about the use of open-world versus closed-world knowledge representation and reasoning in the semantic web [12]. We believe, however, that DLP can serve as a basic interoperability layer between these paradigms, at least for scientific investigations, as spelled out in [12]. It may even find more practical uses if considered as a tractable fragment of OWL in the sense of the W3C member submission on OWL 1.1[4], or as a basis for the W3C Rule Interchange Format RIF[5], as it provides a bridge e.g. between OWL and the Web Rule Language WRL[6].

This short technical note has been written with the sole purpose of describing normal forms for DLP, both in Description Logic and in Datalog syntax. We see this as a helpful step for dissemination into adjacent fields of research and possibly also into practice. At the same time, our normal forms can be used as definitions for DLP which – in our opinion - are much more concise and more transparent than others.

For clarification, we note that we do *not* consider Datalog to come with a specific semantics (like the minimal model semantics) which is different from its first-order logic semantics. We simply consider it to be a syntactic fragment of first order logic which thus inherits its semantics. Some people prefer the notion *OWL-Horn* in this case, instead of *DLP*, but it does not really matter in our context.

The paper is structured as follows. In Section 2 we provide normal forms for DLP in both DL and Datalog form, and formally prove that they are indeed normal forms. In Section 3 we give an extended example for DLP using our syntax, and in Section 4 we conclude.

2 Normal Forms

We assume that the reader is familiar with basic Description Logics [4], with OWL [3] and basic notions from logic programming [5]. For detailed background on DLP we recommend [2], and for a much shorter overview [1].

We need to fix terminology first. We call DLP the (semantic) fragment common to OWL Lite and Datalog, i.e. we abstract (for the time being) from a concrete syntax: Every OWL Lite statement which is semantically equivalent — in the sense of first order logic — to a (finite) set of function-free Horn clauses (i.e. Datalog rules) constitutes a valid DLP statement. Likewise, every function-free Horn clause which is semantically equivalent to some set of OWL Lite statements constitutes a valid DLP statement[7]. Allowing integrity constraints, we call the

[4] http://www.w3.org/Submission/2006/10/

[5] http://www.w3.org/2005/rules/

[6] http://www.w3.org/Submission/2005/08/

[7] In our terminology, the set of OWL Lite statements $\{C \sqsubseteq D \sqcup E, D \equiv E\}$ would not qualify as a set of DLP statements, although it is semantically equivalent to $\{C \sqsubseteq D, D \equiv E\}$, which is expressible in DLP. We are well aware of this restriction, but will not be concerned with it in the moment, because this more general notion

resulting fragment DLP IC (or just IC). Allowing integrity constraints and equality, we call the resulting fragment DLP ICE (or ICE). We write DLP$^+$ for the (semantic) fragment common to OWL DL and (function-free non-disjunctive) Datalog. Analogously, we write DLP$^+$ IC, IC$^+$, etc.

In the following, we will give normal forms, both on the Description Logic side and on the Datalog side. I.e. we provide syntactic fragments which allow expressing (semantically) everything in DLP.

2.1 Normal Form for Description Logic Syntax

Allowed are the following, where a, b, a_i stand for individuals, C stands for a concept name and $R, Q, R_i, Q_{i,j}$ stand for role names.

- ABox:
  $C(a)$ (individual assertion)
  $R(a, b)$ (property assertion)
  $a = b$ (ICE) (individual equivalence)
- Property Characteristics:
  $R \equiv Q$ (equivalence)
  $R \sqsubseteq Q$ (subproperty)
  $\top \sqsubseteq \forall R.C$ $(C \neq \bot)$ (domain)
  $\top \sqsubseteq \forall R^-.C$ $(C \neq \bot)$ (range)
  $R \equiv Q^-$ (inverse)
  $R \equiv R^-$ (symmetry)
  $\top \sqsubseteq \leq 1R$ (ICE) (functionality)
  $\top \sqsubseteq \leq 1R^-$ (ICE) (inverseFunctionality)
- TBox: We allow expressions of the form

$$\exists Q_{1,1}^{(-)} \ldots \exists Q_{1,m_1}^{(-)} \texttt{Left}_1 \sqcap \cdots \sqcap \exists Q_{k,1}^{(-)} \ldots \exists Q_{k,m_k}^{(-)} \texttt{Left}_k \sqsubseteq \forall R_1^{(-)} \ldots \forall R_n^{(-)}.\texttt{Right}$$

  where the following apply.
  - For DLP we allow $\texttt{Left}_j$ to be of the forms C, $\{o_1, \ldots, o_n\}$, $\bot$ or $\top$, and $\texttt{Right}$ to be of the forms C or $\top$.
  - For DLP IC we allow $\texttt{Left}_j$ to be of the forms C, $\{o_1, \ldots, o_n\}$, $\bot$, or $\top$, and $\texttt{Right}$ to be of the form C, $\top$, or $\bot$.
  - For DLP ICE we allow $\texttt{Left}_j$ to be of the forms C, $\{o_1, \ldots, o_n\}$, $\bot$, or $\top$, and $\texttt{Right}$ to be of the form C, $\top$, $\bot$, or $\{o\}$.
  - For the DLP$^+$ versions we furthermore allow $\texttt{Right}$ to be of the form $\exists R^{(-)}.\{a\}$.

  The superscript $^{(-)}$ shall indicate, that an inverse symbol may occur in these places. Note that (by a common abuse of notation) we allow any of k, m_i, n to be zero. For $k = 0$ the left hand side becomes $\top$. Note also that we could have disallowed $\bot$ on the left and $\top$ on the right, since in either

of semantic equivalence is not readily accessible by syntactic means. Note, however, that $C \sqsubseteq D \sqcup D$ qualifies as a DLP statement, since it is semantically equivalent to $C \sqsubseteq D$.

case the statement becomes void. Likewise, it would suffice to require $n = 0$ in all cases, since universal quantifiers on the right are expressable using existentials on the left. Disallowing the existential quantifiers on the left (while keeping universals on the right) is also possible, but at the expense of the introduction of an abundance of new concept names. As an example, note that $\exists R.C \sqcap \exists Q.D \sqsubseteq E$ would have to be translated into the set of statements $\{C_1 \sqcap D_1 \sqsubseteq E, C \sqsubseteq \forall R^-.C_1, D \sqsubseteq \forall Q^-.D_1\}$, where C_1 and D_1 are new concept names. Our representation is more compact.

2.2 Normal Form for Datalog Syntax

Allowed are the following, where $x, y, z, y_i, x_{i,j}$ are variables, a, b, c, a_j are constant symbols, C, D are unary predicate symbols, and $Q, R, R_{i,j}$ are binary predicate symbols.

- Corresponding to ABox:
  $C(a) \leftarrow$ (individual assertion)
  $R(a, b) \leftarrow$ (property assertion)
  $a = b \leftarrow$ (individual equivalence)
- Corresponding to Property Characteristics:
  $Q(x, y) \leftarrow R(x, y)$ (subproperty)
  $C(y) \leftarrow R(x, y)$ (domain)
  $C(y) \leftarrow R(y, x)$ (range)
  $R(x, y) \leftarrow Q(y, x)$ (inverse subproperty)
  $R(x, y) \leftarrow R(y, x)$ (symmetry)
  $y = z \leftarrow R(x, y) \wedge R(x, z)$ (ICE) (functionality)
  $y = z \leftarrow R(y, x) \wedge R(z, x)$ (ICE) (inverseFunctionality)
- Corresponding to TBox: We allow rules of the form

$$\begin{aligned}\texttt{Left}(y) \leftarrow\ & Q_{1,1}^{(-)}(x_{1,1}, x_{1,2}) \wedge \cdots \wedge Q_{1,m_1}^{(-)}(x_{1,m_1}, x) \wedge \texttt{Right}_1(x) \\ & \wedge \ldots \\ & \wedge Q_{k,1}^{(-)}(x_{k,1}, x_{k,2}) \wedge \cdots \wedge Q_{k,m_1}^{(-)}(x_{k,m_k}, x) \wedge \texttt{Right}_k(x) \\ & \wedge R_1^{(-)}(x, y_1) \wedge \cdots \wedge R_n^{(-)}(y_{n-1}, y),\end{aligned}$$

  where $\texttt{Right}_j(x)$ is of the form $C(x)$ or $R^{(-)}(x, a)$, and $\texttt{Left}(y)$ is of the form $D(y)$, or (for DLP IC) $\bot$, or (for DLP ICE) $y = b$, or (for DLP$^+$ versions) $Q(y, c)$. Furthermore, we require all variables $x, y, y_i, x_{i,j}$ to be mutually distinct.
  The meaning of the inverse symbol here is as follows: For a binary predicate symbol R we let $R^-(x, y)$ stand for $R(y, x)$. A bracketed inverse symbol in the superscript $^{(-)}$ hence means that the order of the arguments of the corresponding predicate symbol is not relevant.
  By slight abuse of notation we allow any of k, n, m_j to be zero, which may cause the body of the rule to be empty. For $m_j = 0$ the form of

$$Q_{j,1}^{(-)}(x_{j,1}, x_{j,2}) \wedge \cdots \wedge Q_{j,m_1}^{(-)}(x_{j,m_j}, x) \wedge \texttt{Right}_j(x)$$

reduces to $\texttt{Right}_j(x)$, with $\texttt{Right}_j(x)$ as indicated. For $n = 0$ we require y to be x.

Concerning the terminology just introduced, we can show the following theorem.

Theorem 1. *Every DLP$^{(+)}$ (DLP$^{(+)}$ IC, DLP$^{(+)}$ ICE) statement made in normal form for Description logic syntax is semantically equivalent to a set of DLP$^{(+)}$ (DLP$^{(+)}$ IC, DLP$^{(+)}$ ICE) statements made in normal form for Datalog syntax. Conversely, every DLP$^{(+)}$ (DLP$^{(+)}$ IC, DLP$^{(+)}$ ICE) statement made in normal form for Datalog syntax is semantically equivalent to a set of DLP$^{(+)}$ (DLP$^{(+)}$ IC, DLP$^{(+)}$ ICE) statements made in normal form for Description Logic syntax.*

Proof. We use the translations between Description Logic and Datalog as provided in [1, 2], and summarized in Table 1. How to obtain the semantically equivalent statements for the ABox and the Property Characteristics parts is evident from this summary.

Now consider a rule

$$\begin{aligned}\texttt{Left}(y) \leftarrow\ & Q_{1,1}^{(-)}(x_{1,1}, x_{1,2}) \wedge \cdots \wedge Q_{1,m_1}^{(-)}(x_{1,m_1}, x) \wedge \texttt{Right}_1(x)\\ & \wedge \ldots\\ & \wedge Q_{k,1}^{(-)}(x_{k,1}, x_{k,2}) \wedge \cdots \wedge Q_{k,m_1}^{(-)}(x_{k,m_k}, x) \wedge \texttt{Right}_k(x)\\ & \wedge R_1^{(-)}(x, y_1) \wedge \cdots \wedge R_n^{(-)}(y_{n-1}, y),\end{aligned}$$

where $\texttt{Left}(y)$ and $\texttt{Right}_j(x)$ are as indicated above. This translates to the statement

$$\exists Q_{1,1}^{(-)} \ldots \exists Q_{1,m_1}^{(-)} \texttt{Ri}_1 \sqcap \cdots \sqcap \exists Q_{k,1}^{(-)} \ldots \exists Q_{k,m_k}^{(-)} \texttt{Ri}_k \sqsubseteq \forall R_1^{(-)} \ldots \forall R_n^{(-)}.\texttt{Le},$$

where $\texttt{Le}$ is

- D if $\texttt{Left}(x)$ is $D(x)$,
- $\bot$ if $\texttt{Left}(x)$ is $\bot$,
- $\{b\}$ if $\texttt{Left}(x)$ is $x = b$, and
- $\exists Q^{(-)}.\{c\}$ if $\texttt{Left}(x)$ is $Q^{-}(x, c)$

and $\texttt{Re}_j$ is

- C if $\texttt{Right}_j(x_{j,1})$ is $C(x_{j,1})$, and
- $\exists R^{(-)}.\{a_j\}$ if $\texttt{Right}_j(x_{j,1})$ is $R^{(-)}(x_{j,q}, a_j)$.

We need to justify our translation by showing that the resulting Datalog rule is semantically equivalent to the Description Logic statement from which it was obtained. It boils down to somewhat tedious equivalence transformations in first order logic following the exhibitions in [1, 2], and we will not be bothered with

OWL DL	DL statement		DLP rule
ABox			
indiv. assertion	$C(a)$		$C(a) \leftarrow$
property assertion	$R(a,b)$		$R(a,b) \leftarrow$
indiv. equiv.	$a = b$	ICE	$a = b \leftarrow$
indiv. inequiv.	$\neg(a = b)$	not expressible in general	
TBox			
equivalence	$C \equiv D$	not expressible in general	
GCI	$C \sqsubseteq D$		$D(x) \leftarrow C(x)$
top	$\top$		expressible
bottom	$C \sqsubseteq \bot$	IC (ri)	$\bot \leftarrow C(x)$
conjunction	$C \sqcap D \sqsubseteq E$		$E(x) \leftarrow C(x) \wedge D(x)$
	$C \sqsubseteq E \sqcap F$		$E(x) \leftarrow C(x)$
			$F(x) \leftarrow C(x)$
disjunction	$C \sqcup D \sqsubseteq E$	(le)	$E(x) \leftarrow C(x)$
			$E(x) \leftarrow D(x)$
atomic negation	$\neg A$	not expressible in general	
univ. restriction	$D \sqsubseteq \forall R.C$ $(C \neq \bot)$	(ri)	$C(y) \leftarrow D(x) \wedge R(x,y)$
	$D \sqsubseteq \forall R.\bot$	IC (ri)	$\bot \leftarrow D(x) \wedge R(x,y)$
exist. restriction	$\exists R.C \sqsubseteq D$ $(C \neq \bot)$	(le)	$D(x) \leftarrow R(x,y) \wedge C(y)$
	$\exists R.\bot \sqsubseteq D$	IC (le)	$\bot \leftarrow R(x,y) \wedge C(y)$
one-of 1	$C \sqsubseteq \{a\}$	ICE	$a = x \leftarrow C(x)$
	$\{a\} \sqsubseteq C$		$C(a) \leftarrow$
hasValue	$\exists R.\{a\} \sqsubseteq C$		$C(x) \leftarrow R(x,a)$
	$C \sqsubseteq \exists R.\{a\}$	DLP$^+$	$R(x,a) \leftarrow C(x)$
one-of	$\{o_1, \ldots, o_n\} \sqsubseteq C$	(le)	$C(o_i) \leftarrow$ (for $i = 1, \ldots, n$)
card. restrictions	...	not expressible in general	
Property Characteristics			
equivalence	$R \equiv Q$		$R(x,y) \leftarrow Q(x,y)$
			$Q(x,y) \leftarrow R(x,y)$
subproperty	$R \sqsubseteq Q$		$Q(x,y) \leftarrow R(x,y)$
domain	$\top \sqsubseteq \forall R.C$ $(C \neq \bot)$		$C(y) \leftarrow R(x,y)$
range	$\top \sqsubseteq \forall R^-.C$ $(C \neq \bot)$		$C(y) \leftarrow R(y,x)$
inverse	$R \equiv Q^-$		$R(x,y) \leftarrow Q(y,x)$
			$Q(x,y) \leftarrow R(y,x)$
symmetry	$R \equiv R^-$		$R(x,y) \leftarrow R(y,x)$
transitivity			$R(x,y) \leftarrow R(x,z) \wedge R(z,y)$
functionality	$\top \sqsubseteq \leq 1R$	ICE	$y = z \leftarrow R(x,y) \wedge R(x,z)$
inverseFunctionality	$\top \sqsubseteq \leq 1R^-$	ICE	$y = z \leftarrow R(y,x) \wedge R(z,x)$

Table 1. Translation from DL to Datalog, taken from [1, 2]. The abbreviation *ri* (*le*) means *right* (*left*) *of GCI only*.

the details. We can, however, make our transformation transparent by means of the transformations listed in Table 1. The statement

$$\exists Q_{1,1}^{(-)} \dots \exists Q_{1,m_1}^{(-)} \mathtt{Ri}_1 \sqcap \dots \sqcap \exists Q_{k,1}^{(-)} \dots \exists Q_{k,m_k}^{(-)} \mathtt{Ri}_k \sqsubseteq \forall R_1^{(-)} \dots \forall R_n^{(-)} .\mathtt{Le}$$

can be written as the pair of statements

$$\exists Q_{1,1}^{(-)} \dots \exists Q_{1,m_1}^{(-)} \mathtt{Ri}_1 \sqcap \dots \sqcap \exists Q_{k,1}^{(-)} \dots \exists Q_{k,m_k}^{(-)} \mathtt{Ri}_k \sqsubseteq D$$

$$D \sqsubseteq \forall R_1^{(-)} \dots \forall R_n^{(-)} .\mathtt{Le},$$

where D is a new concept name. These statements can be translated separately into

$$\begin{aligned} \mathtt{D}(x) \leftarrow\ & Q_{1,1}^{(-)}(x_{1,1}, x_{1,2}) \wedge \dots \wedge Q_{1,m_1}^{(-)}(x_{1,m_1}, x) \wedge \mathtt{Right}_1(x) \\ & \wedge \dots \\ & \wedge Q_{k,1}^{(-)}(x_{k,1}, x_{k,2}) \wedge \dots \wedge Q_{k,m_1}^{(-)}(x_{k,m_k}, x) \wedge \mathtt{Right}_k(x) \end{aligned}$$

and

$$\mathtt{Left}(y) \leftarrow D(x) \wedge R_1^{(-)}(x, y_1) \wedge \dots \wedge R_n^{(-)}(y_{n-1}, y).$$

By unfolding over $D(x)$ we obtain the desired combined rule.

The translation can obviously be performed in both directions, so there is nothing more to show.

It is possible to strengthen Theorem 1 by providing a translation between single Description Logic statements and single Datalog rules (in normal form). In this case we would have to disallow the property characteristics *inverse* on the OWL side, which can be done since $R \equiv Q^-$ is expressible e.g. by the set of statements $\{R \sqsubseteq Q^-, Q^- \sqsubseteq R\}$, each member of which is in turn translatable into a single Datalog statement. Similarly, property equivalence would have to be disallowed. We think that the form we have chosen is more concise.

Theorem 2. *All description logic programs following [1, 2] can be written in normal form.*

Proof. All statements belonging to DLP as described in [1, 2] are listed in Table 1. It is easy to check that all possibly resulting Datalog statements listed in the last column are already in normal form, which suffices to show the statement.

3 Examples

A rule of thumb for the creation of DLP ontologies is: *Avoid concrete domains and number restrictions, and be careful with quantifiers, disjunction, and nominals.* We give a small example ontology which includes the safe usage of the latter constructs. It shall display the modelling expressivity of DLP.

For the TBox, we model the following sentences.

(1) Every man or woman is an adult.
(2) A grown-up is a human who is an adult.
(3) A woman who has somebody as a child, is a mother.
(4) An orphan is the child of humans who are dead.
(5) A lonely child has no siblings.
(6) AIFB researchers are employed by the University of Karlsruhe.

They can be written in DLP as follows.

$$\mathsf{Man} \sqcup \mathsf{Woman} \sqsubseteq \mathsf{Adult} \tag{1}$$
$$\mathsf{GrownUp} \sqsubseteq \mathsf{Human} \sqcap \mathsf{Adult} \tag{2}$$
$$\mathsf{Woman} \sqcap \exists \mathsf{childOf}^{-}.\top \sqsubseteq \mathsf{Mother} \tag{3}$$
$$\mathsf{Orphan} \sqsubseteq \forall \mathsf{childOf}.(\mathsf{Dead} \sqcap \mathsf{Human}) \tag{4}$$
$$\mathsf{LonelyChild} \sqsubseteq \neg \exists \mathsf{siblingOf}.\top \tag{5}$$
$$\mathsf{AIFBResearcher} \sqsubseteq \exists \mathsf{employedBy}.\{\mathsf{UKARL}\} \tag{6}$$

In normal form in Description Logic syntax these are as follows.

$$\mathsf{Man} \sqsubseteq \mathsf{Adult} \tag{1}$$
$$\mathsf{Woman} \sqsubseteq \mathsf{Adult} \tag{1}$$
$$\mathsf{GrownUp} \sqsubseteq \mathsf{Human} \tag{2}$$
$$\mathsf{GrownUp} \sqsubseteq \mathsf{Adult} \tag{2}$$
$$\mathsf{Woman} \sqcap \exists \mathsf{childOf}^{-}.\top \sqsubseteq \mathsf{Mother} \tag{3}$$
$$\mathsf{Orphan} \sqsubseteq \forall \mathsf{childOf}.\mathsf{Dead} \tag{4}$$
$$\mathsf{Orphan} \sqsubseteq \forall \mathsf{childOf}.\mathsf{Human} \tag{4}$$
$$\mathsf{LonelyChild} \sqsubseteq \forall \mathsf{siblingOf}.\bot \tag{5}$$
$$\mathsf{AIFBResearcher} \sqsubseteq \exists \mathsf{employedBy}.\{\mathsf{UKARL}\} \tag{6}$$

We note that for (5) we require DLP IC, while for (6) we require DLP^{+}. For the RBox, we use the following.

$\mathsf{parentOf} \equiv \mathsf{childOf}^{-}$	parentOf and childOf are inverse roles.
$\mathsf{parentOf} \sqsubseteq \mathsf{ancestorOf}$	parentOf is a subrole of ancestorOf.
$\mathsf{fatherOf} \sqsubseteq \mathsf{parentOf}$	fatherOf is a subrole of parentOf.
$\top \sqsubseteq \forall \mathsf{ancestorOf}.\mathsf{Human}$	Human is the domain of ancestorOf.
$\top \sqsubseteq \leq 1 \mathsf{fatherOf}^{-}$	fatherOf is inverse functional.

We can populate the classes and roles by means of an ABox in the following way.

$$\{\mathsf{Bernhard}, \mathsf{Benedikt}, \mathsf{Rainer}, \mathsf{Ganter}\} \sqsubseteq \mathsf{Man}$$
$$\{\mathsf{Ruth}, \mathsf{Ulrike}\} \sqsubseteq \mathsf{Woman}$$
$$\mathsf{Bernhard} = \mathsf{Ganter}$$
$$\mathsf{employedBy}(\mathsf{Bernhard}, \mathsf{TUD})$$
$$\ldots$$

Note that an ABox statement such as

$$\{\text{Ruth}, \text{Ulrike}\} \sqsubseteq \text{Woman}$$

is simply syntactic sugar for the two statements

$$\text{Woman}(\text{Ruth}) \qquad \text{Woman}(\text{Ulrike})$$

We therefore consider it to be part of the ABox. To be precise, the original statement is (syntactically) not in OWL Lite, but the equivalent set of three ABox statements is. The statement Bernhard = Ganter requires DLP ICE.

Note also that class inclusions cannot in general be replaced by equivalences. For example, the statement

$$\text{Adult} \sqsubseteq \text{Man} \sqcup \text{Woman}$$

is not in DLP.

For illustration, we give the knowledge base in Datalog normal form. The TBox is as follows.

$$\begin{aligned}
\text{Adult}(y) &\leftarrow \text{Man}(y) && (1)\\
\text{Adult}(y) &\leftarrow \text{Woman}(y) && (1)\\
\text{Human}(y) &\leftarrow \text{GrownUp}(y) && (2)\\
\text{Adult}(y) &\leftarrow \text{GrownUp}(y) && (2)\\
\text{Mother}(y) &\leftarrow \text{childOf}(x, y) \wedge \text{Woman}(y) && (3)\\
\text{Dead}(y) &\leftarrow \text{Orphan}(x) \wedge \text{childOf}(x, y) && (4)\\
\text{Human}(y) &\leftarrow \text{Orphan}(x) \wedge \text{childOf}(x, y) && (4)\\
&\leftarrow \text{LonelyChild}(x) \wedge \text{siblingOf}(x, y) && (5)\\
y = \text{UKARL} &\leftarrow \text{AIFBResearcher}(x) \wedge \text{employedBy}(x, y) && (6)
\end{aligned}$$

Translating the RBox yields the following statements.

$$\begin{aligned}
\text{parentOf}(x, y) &\leftarrow \text{childOf}(y, x)\\
\text{childOf}(x, y) &\leftarrow \text{childOf}(y, x)\\
\text{ancestorOf}(x, y) &\leftarrow \text{parentOf}(x, y)\\
\text{parentOf}(x, y) &\leftarrow \text{fatherOf}(x, y)\\
\text{Human}(y) &\leftarrow \text{ancestorOf}(x, y)\\
y = z &\leftarrow \text{fatherOf}(y, x) \wedge \text{fatherOf}(z, x)
\end{aligned}$$

4 Conclusions

We have presented normal forms for Description Logic Programs, both in Description Logic syntax and in Logic Programming syntax. We have formally shown that these are indeed normal forms.

We believe that these normal forms can and should be used for *defining* Description Logic Programs. We have found that some of the definitions used in the literature remain somewhat ambiguous, so that the language is not entirely specified. This brief note rectifies this problem in providing a frame of reference.

References

1. Grosof, B., Horrocks, I., Volz, R., Decker, S.: Description logic programs: Combining logic programs with description logics. In: Proc. of WWW 2003, Budapest, Hungary, May 2003, ACM (2003) 48–57
2. Volz, R.: Web Ontology Reasoning with Logic Databases. PhD thesis, University of Manchester (2004)
3. Antoniou, G., van Harmelen, F.: Web Ontology Language: OWL. In Staab, S., Studer, R., eds.: Handbook on Ontologies. Springer (2004) 67–92
4. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
5. Lloyd, J.W.: Foundations of Logic Programming. Springer, Berlin (1988)
6. Angele, J., Lausen, G.: Ontologies in F-logic. In Staab, S., Studer, R., eds.: Handbook on Ontologies. Springer (2004) 29–50
7. Hustadt, U., Motik, B., Sattler, U.: Reducing SHIQ-description logic to disjunctive datalog programs. In Dubois, D., Welty, C.A., Williams, M.A., eds.: Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004), Whistler, Canada, June 2-5, 2004, AAAI Press (2004) 152–162
8. Hustadt, U., Motik, B., Sattler, U.: Reasoning in description logics with a concrete domain in the framework of resolution. In de Mántaras, R.L., Saitta, L., eds.: Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004, IOS Press (2004) 353–357
9. Motik, B.: Reasoning in description logics using resolution and deductive databases. PhD theis, University Karlsruhe, Germany (2006)
10. Motik, B., Sattler, U., Studer, R.: Query answering for OWL-DL with rules. Journal of Web Semantics: Science, Services and Agents on the World Wide Web **3** (2005) 41–60
11. de Bruijn, J., Pollerres, A., Lara, R., Fensel, D.: OWL$^-$. WSML Working Draft, DERI (2004)
12. Hitzler, P., Haase, P., Krötzsch, M., Sure, Y., Studer, R.: DLP isn't so bad after all. In Grau, B.C., Horrocks, I., Parsia, B., , Patel-Schneider, P., eds.: Proceedings of the Workshop OWL – Experiences and Directions, Galway, Ireland. Volume 188 of CEUR Workshop Proceedings. (2005) Online http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-188/sub2.pdf.