Blending Rules and Ontology in Argumentation

Anca Marginean
Department of Computer Science, Technical University of Cluj-Napoca
Baritiu 28, RO-400391 Cluj-Napoca, Romania
E-mail: anca@cs-gw.utcluj.ro

Abstract. It is good to know, but to know how to mix what you know is even better. We propose an argumentative manner for dealing with knowledge from different sources using different formalisms as ontologies and rules. Each of the involved formalisms are better as they are, without translating one to another, therefore our aim is to mix formalisms’ reasoning abilities without significant translations between them. We define two schemes for building arguments from both rules and ontology. The resulting argumentation network is visualized in an application meant not to make a decision, but to facilitate the human user in evaluating and assessing the interdependences between available knowledge.

Keywords. Argumentation. AIF. Ontology reasoning. Rule reasoning

1 Introduction

When we become capable of critical thinking, we understand that although some things can never be known with certainty, some judgments are more valid than others because of their coherence, their fit with the evidence, and their usefulness. When reasoning dialectically, we are willing to consider evidence from a variety of sources and to consider alternative interpretations, to justify our conclusions as representing the most complete, plausible or compelling understanding of an issue, based on currently available evidence - this means abandoning ignorant certainty in favor of intelligent confusion. Attempting to reach the human mind argumentative reasoning abilities, more logics were developed as non monotonic reasoning, defeasible reasoning, possibilistic reasoning, abduction based or assumption based reasoning.

Another important issue for knowledge management area is given by the semantics. Behind each statement (whatever its form is - human communication or parameters settings in computer application) there is a semantic. At a first glance identification of the semantics seems a simple process, its complexity being hidden behind its large use and revealed only when trying to formalize it. In the argumentation field, the most recent approach for capturing the semantic is given by the Argumentation Interchange Format, meant to integrate argumentation schemes and arguments in a semantic world.

Starting from these ideas and from two common models for knowledge representation - rules as Horn Clauses and ontologies following Description Logic semantics, we propose a method for extending the argumentation based on rule derivation with ontological knowledge, while keeping independent those two models and their reasoning mechanisms. The argumentative approach is appropriate for blending knowledge from two sources due to its abilities to clearly represent the interdependencies between existing information. The objective of the current work is restricted to identification and representation of the knowledge relevant for a case considering two different representational models for the involved knowledge.

2 Knowledge models. Rules. Ontologies

Rules as a model for representing knowledge were used for quite a long time, starting from propositional logic with Modus Ponens to variants of First Order Logic, disjunctive logics, fuzzy logics or RuleML for the Semantic Web. Rules structure the information, having a prescriptive nature. Whatever the
logic model on which the rule resides, it is commonly accepted that a rule includes at least two major components: the premises and the conclusion. The semantic is straightforward: if the premises are true, than the conclusion also is true.

But the rule based models lack the ability to deal with semantics, that should endorse a thorough understanding of the knowledge expressed through the rules. The ontologies bring the descriptive abilities. Therefore, a combination of these two models seems appropriate, as it was already proved by the SWRL initiative (Semantic Web Rule Language) that combines two standards one for ontologies and one for rules: OWL (Ontology Web Language) respectively RuleML.

2.1 Ontology

An ontology represents the relation between the concepts in a domain. Description Logics support description of concepts and properties through different constructors and axioms, while the reasoning process is based on tableau based methods.

[5] identified the subset of Descriptive Logic semantics that can be translated into Horn clauses. Based on these, a $\rho$ relation for translating literals between subsets of FOL and DL expressions can be established [4]. Due to the space limits, we do not include this relation here.

Reasoning on ontology means to find facts that are implicit in the ontology given explicitly stated facts. In other words, find what you know, but you don’t know you know it yet. Automated reasoning on ontologies is complex, with non-trivial logical background. On small examples it is not obvious why this reasoning is useful, but the benefit becomes clearer when used for bigger ontologies.

When the conclusions are also explained, the automated reasoning becomes really useful. Explanations make ontology reasoning more transparent to the user, and this is possible due to axiom pinpointing [7]. Axiom pinpointing means identifying debugging-relevant axioms, where an axiom is relevant if a contradictory TBox becomes coherent once the axiom is removed or if, at least, a particular, previously unsatisfiable concept turns satisfiable. Due to the relation satisfiability - entailment proved in [6], the axiom pinpointing that was previously possible only for unsatisfiability becomes a general model for entailment explanation. The Pellet reasoner gives such explanations for (i) unsatisfiable concepts, (ii) for entailment of subsumption, (iii) for realization (Abox reasoning), and (iv) for inconsistent ontology.

3 Arguments from ontology

Argumentation schemes are stereotypical patterns of human reasoning, especially defeasible ones. We will consider schemes and arguments based on them in the way they are described in Argumentation Interchange Format, that is through information nodes, schema application nodes, and form nodes.

3.1 Ontology Reasoning Schema

In the presence of evidence $Ev$, based on the relations described in the ontology $O$, different conclusion can be drawn. In order to make this process more transparent, the schema for ontology reasoning will include three kinds of nodes beside the node for the schema descriptor itself: justification descriptor, entailment descriptor and evidence descriptor.

An Ontology reasoning scheme description is a tuple $(EvD, \alpha, cd, \gamma)$, where:

- $EvD \subseteq N^{Prem}_F$ is the set of additional knowledge description; the reasoning is applied on the ontology joined with this knowledge. Generally it represents ground knowledge about certain instances.
- $\alpha$ is the ontology reasoning schema. According to the reasoning type, there are schemes for axiom entailment, realization, unsatisfiability and inconsistency.
- $cd \in N^{Conc}_F$ represents the description of the information supported by the reasoning. The conclusion’s description depends on the type of the used ontology reasoning.
• γ stands for the explanation description of the entailment: it will be fulfilled by the ontology statements included in the explanation of the reasoning.

Figure 1. Ontology reasoning scheme description

According to the type of the reasoning involved, the ontology schema can be one of: (1) Axiom entailment: (i) concept subsumption \( O \vdash C \sqsubseteq D \), (ii) concept equality \( O \vdash C \equiv D \), (iii) role subsumption \( O \vdash R \sqsubseteq S \), (iv) Role equality \( O \vdash R \equiv S \); (2) Realization: \( O \vdash i \in C \); (3) Concept unsatisfiability: \( O \vdash C \equiv \bot \). In this case, there are two types of justification nodes: (i) the clash information \( \gamma_{\text{clash}} \), representing the root cause of the contradiction - for eg. a concept \( C \) is related to a concept \( Y \) that is forced to belong to a concept and its complement, (ii) and the minimal set of support \( \gamma_{\text{supp}} \). The latter one corresponds to the typical justification of the entailment: the axioms involved in the entailment, while the former is more an explanation of the conclusion, the cause. (4) Inconsistency checking: for this scheme, the justification types described for unsatisfiability hold also. The clash information description can have more subtypes: (i) an instance \( X \) is related to an instance \( Y \), and \( Y \) is forced to belong to a concept and its complement, and (ii) an instance \( X \) belongs to a concept \( C \) and its complement, (iii) cardinality: an individual has a max cardinality restriction but is related to more distinct individuals (iv) datatype conflict.

An Ontology argument based on the the ontology reasoning schema description \((Ev, \alpha, cd, \gamma)\) is a tuple \(\langle P_{Ev}, P_{Expl}, \tau, c \rangle\), where:

• \( P_{Ev} \subseteq N_I \) is a set of information nodes containing information added to the ontology about the current case

• \( P_{Expl} \subseteq N_I \) is a set of nodes representing the statements from the ontology involved in the entailment - the minimum support returned by the explanation module of Pellet.

• \( \tau \in N^{RA}, \tau \xrightarrow{\text{fulfillsSchema}} \alpha. \)

• \( \tau \xrightarrow{\text{edge}} c, c \in N_I, c \xrightarrow{\text{fulfillsConclusionDescr}} cd \) the conclusion.

3.2 Rules and ontology schema

Before defining the schema for blending the ontological knowledge with knowledge inferred from rules, we make some observations. At first, we underline that the rules are applied for certain instances, while the concepts from ontology describe sets of instances.

Modus Ponens allows to infer the conclusion of a rule in case that all the literals from the rule’s body are true. What happens if according to the ontology (without any further case specific information), the intersection of translation \( \rho \) of two literals from the body of the rule is an unsatisfiable concept? The correctness of the rule is under question in this case - according to the ontology it is not possible to have all the premises of the rule true in the same time - therefore application of an inference based on this rule is attacked (undermined). The same thing happens if the intersection between translation of the rule’s head and translation of one of the literals from the body give an unsatisfiable concept. In this case, the ontology states that there is no model for the intersection between the conclusion of the rule and one of its premises. Both these cases of attack use only the ontology and a rule, without further knowledge (case specific information). The ontology contradicts some presumptions related to the rule: that it is possible to have
all the premises true, respectively it is possible to have in the same time the head of the rule and all its premises. It is not a question of what it is true or not about a certain instance, it is the case of impossible situation, no matter what the case is. Of course in an argumentation framework, this impossibility is more or less important, according to the preferences between arguments. We do not deal in this paper with this aspect, considering that the human user viewing the argumentation network decides this preference relation, according to more or less objective criteria.

There is another case of attack, directly related to the instances involved in the current context. It is based on the inconsistency raised by joining the ontology and the ground information represented as premises of the rule and conclusion of the rule. In case the first two types of attack are not present (so all the intersections give satisfiable concepts), but the inconsistency is still present, then the inconsistency says that the application of the rule is generally possible, but not in this case.

In order to define an inference schema for reasoning both from ontology and rules, we define two types of conflict schema nodes $S^C$:

- ontology unsatisfiability conflict: $S^C_{Unsat} \subseteq S^C$ that is linked to a premise description node $N_{FUUnsat} \in N^P_{Prem}$ that will be fulfilled by the unsatisfiable concept. This unsatisfiable concept is in the same time the conclusion of an ontology reasoning argument.

- ontology inconsistency conflict: $S^C_{Inconsist} \subseteq S^C$ that is linked to a premise description node $N_{FUInconsist} \in N^P_{Prem}$ that will be fulfilled by an inconsistency node of the ontology joined with the added knowledge.

The Rules and ontology schema has as its central part the classic Modus Ponens inference style, but it adds different kinds of exceptions to the rules application, either general, either specific to the current case. A Rules and ontology schema description is a tuple $(EvD, RD, \alpha, cd, \Gamma, \Delta)$, where:

- $EvD \subseteq N^P_{Prem}$ is the set of premises description for the rule fulfilling the rule description
- $RD \in N^P_{Prem}$ is the rule description
- $\alpha$ is the scheme
- $cd \subseteq N^C_{conc}$ is the conclusion description
- $\Gamma \subseteq S^C = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$ the set of all the conflicting nodes giving the exceptions: $\forall \zeta \in \Gamma$ there is $\alpha \xrightarrow{hasException} \zeta$. $\Gamma_1 \subseteq S^C_{Unsat}$ is the set of all ontological unsatisfiability conflicts determined by the unsatisfiability of the concept representing the intersection of two rule’s premises described trough associated $N_{FUUnsat}$. $\Gamma_2 \subseteq S^C_{Unsat}$ is the set of all ontological unsatisfiability conflicts having as premise description $N_{FUUnsat}$ the unsatisfiability of translation $\rho$ of one of the rule’s premises and the rule’s head; $\Gamma_3 \subseteq S^C_{Inconsist}$ is the set of all ontological inconsistency conflicts having as premise description $N_{FUInconsist}$ the ground information that raised the inconsistency.

- $\Delta \subseteq N_F$ representing the set of all the premise descriptions associated to the conflict schemes from $\Gamma$: $\forall \zeta \in \Gamma$ there is a relation of type $\zeta \xrightarrow{hasCauseDescr} \delta$, saying that the conflict described through $\zeta$ has a cause described by $\delta$

An argument based on the Rules and ontology schema description $(EvD, RD, \alpha, cd, \Gamma, \Delta)$ is a tuple $(P_{EvD}, P_{RD}, \tau, c, P_\Gamma, P_\Delta)$, where:

- $P_{RD} \in N_I$ is an information node containing the rule $r ~hd(r) \leftarrow bd(r)$ and $P_{RD} \xrightarrow{fulfillsRuleDescr} RD$

- $P_{EvD} \subseteq N_I$ is the set of all the premises of the rule application, in other words, the literals from the body of the rules $bd(r)$ and $\forall p \in P_{EvDP} \xrightarrow{fulfillsPremiseDescr} EvD$

- $c \in N_I$ is the conclusion of the argument, that is the head of the rule $hd(r)$ and $c \xrightarrow{fulfillsConclusionDescr} cd$
• $\tau \in N^{RA}$ is the Modus Ponens style for rule application and $\tau \xrightarrow{\text{fulfillsSchema}} \alpha$

• $P_{\text{Gamma}} \subseteq N^{CA}$ is the set of all conflicting nodes and $\forall p \in P_{\Gamma}, \exists \gamma \in \Gamma s.t. p \xrightarrow{\text{fulfillsConflictDescr}} \gamma$;

• $P_{\text{Delta}} \subseteq N_I$ represents the causes of conflicts and $\forall p \in P_{\Delta}, \exists \delta \in \Delta s.t. p \xrightarrow{\text{fulfillsConflictCauseDescr}} \gamma$;

and between these nodes the following relations stand:

• $\forall p_{\gamma} \in P_{\Gamma}, \exists p_{\delta} \in P_{\Delta} s.t. p_{\delta} \xrightarrow{\text{edge}} p_{\gamma}$,

• $\forall p_{\gamma} \in P_{\Gamma}, p_{\gamma} \xrightarrow{\text{edge}} \tau$,

• $\forall p \in P_{RD} \cup P_{EvD}, p \xrightarrow{\text{edge}} \tau$.

Following the defeasible derivation style, an argumentation network for $\theta$ is built in the following way:

1. build all the possible derivation $\text{deriv}(\theta)$ for $\theta$, where a derivation means an ordered set of rules applications. A rule is applicable if for all its premises there is a derivation or it is entailed from the ontology.

2. for each entailment from ontology included in $\text{deriv}(\theta)$, add the corresponding ontology argument

3. for each rule application in $\text{deriv}(\theta)$ include the corresponding Rules and ontology argument.

Beside attacks described in schemes and associated arguments, there can also be some other conflict application node: (i) logical negation: $N^{CA}_{\text{Log}}$ between two information nodes, one being the logical negation of the other, and (ii) inconsistency $N^{CA}_{\text{Inconsist}}$ given by the inconsistency of the ontology augmented with two information nodes that are not related through a rule application case. We can observe that inconsistency based conflict application node can relate two information nodes but also an information node and a rule application node.

4.1 Visualizing tool RuleOnto

The argumentation network built according to the above description is represented in a visualization tool based on JUNG package for graphs. The main aspect of the implementation is the use of a Decorator class for description of Argumentation Graphs as DirectedGraphs. In this way, all the functionality provided by the framework for DirectedGraph is maintained and the new functionality is easily included. For rule reasoning a HornClause theory is used, while for the ontology reasoning it is used the Pellet reasoner. Integration of these three tools is possible due to the Jasper package - for Sicstus-Java integration, respectively Pellet API and Jung API.

The main functionality of RuleOnto application supporting the user in evaluating the
information from the argumentation network are: (1) nodes filtering, (2) node expansions allowing different granularity viewing level, (3) associated schema visualizing.

5 Related work

The mix of rules and ontology is not a new idea, but for now, many approaches were based on the translation of one formalism to another. Avoiding the translation, [9] extends Defeasible programming Logic by ontology through redefinition of the conflict as unsatisfiability and ontology inconsistency, and expanding the rule through ontology entailment. Our approach is different to that due to a new kind of relation between rule application and ontology reasoning - ontology reasoning can undercut a rule application, not only attack a conclusion.

[4] proposes building of arguments by defeasible derivations on defeasible programs derived from ontology by translating the ontological assertions into defeasible rules. This way, the reasoning power of ontology reasoners is not at all used, the ontology being reduced to a defeasible program. A similar approach is taken in DR-Prolog [1] or in [8] where a combination of CLIPS reasoning power and object oriented paradigm is presented.

The present work does not address any criteria for evaluating the arguments; but for further work we consider inclusion of some specificity criteria as in inheritance networks [2], values associated to arguments as in Value Based Argumentation and also a selection function for arguments building as the ones used in the assumption based argumentation implemented in CASAPI [3].

6 Conclusion

Both ontology reasoning in Description Logic semantic and rule based inference can be considered pretty mature fields taken as individualities. We aimed to give a model for blending this two formalisms in an argumentative manner, keeping their individuality intact, and giving a real support for a human user in assessing the available knowledge, even if it comes from two different sources.

References