On Integrating Description Logics and Rules under Minimal Hypotheses

Technical Report 3028
AIFB, Karlsruhe Institute of Technology

Anees Mehdi[1⋆], Alexandre Miguel Pinto[2⋆] and Sebastian Rudolph[1]

[1] Institute AIFB—Karlsruhe Institute of Technology,
{mehdi, rudolph}@kit.edu
[2] CISUC-DEI-FCT—Universidade de Coimbra,
ampinto@uc.pt

Abstract. A central and much debated topic in the Knowledge Representation and Reasoning community is how to combine open-world with closed-world formalisms, such as Description Logics (DLs) with Logic Programming. We propose a new approach to defining the semantics of hybrid theories, composed of a DL and a Normal Logic Program (NLP) parts, which employs Pinto and Pereira's Minimal Hypotheses semantics (MHs) for the latter. Because this semantics is more general than the currently employed semantics for hybrid DL-NLP KBs based on Stable Model (SM) semantics, and because MH semantics guarantees model existence for every NLP, our hybrid semantics also guarantees the existence of models for any hybrid DL-NLP theory with consistent DL fragment and consistent DL-NLP ensemble. Finally, due to the MHs featuring beneficial theoretical properties, like relevance and cumulativity, existential query answering tasks may not need to consider the whole hybrid KB, as it is necessarily the case with current state-of-the-art approaches based on the SM semantics.

1 Introduction

1.1 Background

Description Logics (DLs) are a family of knowledge representation formalisms that are decidable fragments of first-order logic [2], where decidability is ensured via several syntactic restrictions. These restrictions lead to problems when expressing some non-tree like relationships. Such relations can easily be expressed using logic programming rules. Nevertheless, rule-based formalisms have their own shortcomings because typically they do not allow reasoning with unbounded infinite domains and hence cannot be used in many scenarios where modeling incomplete information is required. Unlike DLs, rules, for example, cannot reason about implicit objects of a domain. For instance, in DLs one can easily define the concept *Father* as "a male person with a child person" (in DLs syntax $\mathit{Father} \equiv \mathit{Person} \sqcap \mathit{Male} \sqcap \exists \mathit{hasChild}.\mathit{Person}$) without providing further information about the children themselves. This, of course, is not possible in rule-based

⋆ Equal contribution

formalisms unless all the children are explicitly stated. Yet another difference between DLs and logic-based formalism is that DLs, as decidable fragments of first-order logic, are inherently monotonic whereas many ruled-based formalisms allow for default negation and are thus capable of defeasible inferencing.

The benefits of the increased expressiveness of DLs and rule-based formalisms coupled in a unified manner has motivated a significant body of research. Much work has been done in this direction where some deal with integrating DLs with first-order rules (for example, [12]) while others focus on achieving a unified framework for DLs and non-monotonic rules ([8, 15, 16, 14, 20], etc). In such formalisms, a Knowledge Base (KB) has two components: a DL-KB[3] and a Logic Program (LP) and is called hybrid KB. In this work, we focus on the latter direction of research and present a new approach of integrating DLs with Normal Logic Programs (NLPs).

1.2 Motivation

NLPs are the simplest class of LPs allowing for default negation in the bodies of rules. Since the conclusions (heads) of rules of NLPs are always positive atoms, and no explicit negation is allowed, no contradictory sets of conclusions may be derived from NLPs, i.e., NLPs are always consistent — when viewed as SAT problems, NLPs are always satisfiable. DLs, however, although not allowing for default negation, do allow for explicit negation, therefore a DL-KB may be inconsistent. Moreover, even when a DL-KB is consistent the resulting combination of it with an NLP may be inconsistent if there is a sub-alphabet shared amongst these two components, e.g., if the conclusions of the DL part explicitly contradict (via explicit negation) the conclusions of the NLP part. Thus, the whole hybrid DL-NLP KB consistency must be dependent only on the DL's consistency and on the DL-NLP combination consistency, never on the isolated NLP's consistency which is guaranteed.

The main difference between the approaches in [15, 16] and [14] is the underlying semantics considered for the NLP part. The former are based on the Stable Models (SMs [10]) semantics and the later considers the Well-Founded Semantics (WFS)[9] for the NLP. While the WFS guarantees model existence for the NLP it does so at the cost of resorting to the third *undefined* truth-value, and sometimes this might not be acceptable for the purposes the user intends. The SM-based approaches do offer 2-valued-complete models, but they do not guarantee their existence for every NLP. Besides, since SM semantics lacks other useful properties such as relevance and cumulativity[4], the whole NLP must be considered when answering an existential query and no tabling (lemma storing) techniques can be employed to speed up computations. This has a severe impact on query-answering computation times.

In order to have a **2-valued semantics** for hybrid DL-NLP KBs, where the whole KB consistency is ensured as far as the isolated NLP component is concerned, a new

[3] DL-KBs are usually called Ontologies in the Semantic Web community. In this paper, we use these two terms interchangeably.

[4] If a semantics enjoys relevance that means one needs only to consider the rules in the syntactic dependencies call-graph of a query in order to answer it. If a semantics enjoys cumulativity that means one can safely store previously proven true query results in order to speed up later computations.

semantics (other than the WFS and SM) for the NLP part must be considered, one that enjoys all these properties (guarantee of 2-valued-model existence, relevance, cumulativity). Our work's goal is the pursuit of such a hybrid semantics for DL-NLP KBs resorting to a new 2-valued semantics for NLPs enjoying those properties.

1.3 Approach and Results

The knowledge represented by a hybrid DL-NLP KB can only be more complex than the "sum" of the individual DL and NLP components if these share a common sub-alphabet. It is via this shared set of bridging symbols that conclusions from one of the components may be used by the other to derive extra conclusions not deducible otherwise. Therefore, a sensible semantics for a hybrid KB must symmetrically allow each component to inform the other with its local conclusions about shared symbols.

Following the motivational guidelines above we define a hybrid semantics for DL-NLP KBs which differs from the ones in [15, 16] and [14] (a comparison is presented in Section 5) in two ways: 1) we resort to a *crossed fixed-point* approach as a means to characterize the models of the hybrid semantics; and 2) we do so by using the classical semantics for the DL part, and the new 2-valued Minimal Hypotheses (MH) semantics [19] for the NLP part.

Why the Minimal Hypotheses semantics? In [19] Pinto and Pereira proposed the MH semantics, which, besides complying with all the requirements identified in 1.2, also takes all SMs as MH models as well, which ensures the "SM-backward-compatibility" of MH semantics. The original NLP-exclusive motivations for the MH are manifold and are presented in [18, 19]; these lie outside the scope of our paper, and thus we do not repeat that discussion here, only very briefly outlining some of the intuitions behind it. However, the goals pursued with this current work demand that the semantics for the NLP part must enjoy the properties listed at the end of 1.2, thus adding yet another set of motivation arguments for a semantics such as the MH, instead of the SM, that pile upon the original ones in [18, 19].

Since the DL part is a fragment of first-order logic it is monotonic and thus there is no room for different possible semantics for it. The NLP part, however, since it may contain default negation, allows for multiple possible ways of interpreting the meaning of rules, e.g., one may consider a 2-valued semantics or a 3-valued semantics for it. Using a 3-valued semantics, e.g., the WFS, has certain advantages over common 2-valued semantics (like the SMs and MHs), such as tractable complexity (the reasoning tasks with the WFS are known to be polynomial). However, in settings where one wants to identify the multiple alternative scenarios compatible with the NLP part and do some computations with each of them, a 2-valued semantics allowing for multiple 2-valued models is necessary. In this paper we take this last direction.

Because the NLP part is always consistent (by not allowing rules to derive negative conclusions), the 2-valued semantics for this component of the DL-NLP hybrid KB must ensure model existence in order to guarantee the whole hybrid DL-NLP consistency as far as the isolated NLP part is concerned. This is not the case with the SM semantics, but it is with the MH one. Also, because the NLP itself may be an aggregation of independent sets of rules, it makes no sense to have to consider the whole NLP when answering an existential query which concerns only a subset of the NLP which is syntactically independent from the rest. The SM semantics does not allow to take advantage of such independence because it lacks the relevance property; whereas the MH semantics does allow it by enjoying this property.

Besides guaranteeing model existence for NLPs, and enjoying relevance, the MH semantics also enjoys cumulativity (these properties of non-monotonic formalisms are detailed in [5, 6]). This property means that the whole semantics of the program remains unchanged if atoms known to be true are added as facts. Both relevance and cumulativity potentially improve the time performance of some reasoning tasks by performing some pre-processing and considering only the relevant sub-part of the program. This allows us for black-box approach of performing reasoning tasks in hybrid KBs by using a standard DL reasoner and an MH model calculator.

We define a hybrid semantics for hybrid DL-NLP KBs based on the classical semantics for the DL part, and the MH semantics for the NLP part by resorting to a *crossed fixed-point*. We contrast this MH-based choice against a SM-based alternative like the one in [16] and conclude that the former has advantages over the latter, namely by being more general than the ones based on SMs in the sense that all SMs are MH models as well, and, further, MH assigns a semantics to *all* NLPs.

The rest of the paper is organized as follows. We first provide some preliminaries in Section 2. The notion of hybrid (DL-NLP) KBs, along with an example, is presented in Section 3 where we also provide our crossed fixed-point semantics for hybrid DL-NLP KBs. The notion of reasoning in our formalism, along with the complexity issues, are discussed in Section 4. In Section 5 we compare our approach to some of the existing ones and some concluding remarks along with directions for future work in Section 6.

2 Preliminaries

We now review some notions we will be using henceforth.

2.1 Description Logic $\mathcal{SROIQ}$

The DL $\mathcal{SROIQ}$ is one of the most expressive DLs which provide the logical foundation of OWL 2[17]. For the syntax of $\mathcal{SROIQ}$, let N_I, N_C, and N_R be finite, disjoint sets called *individual names*, *concept names* and *role names* respectively. Further we assume that N_R is the union of disjoint sets **Rs** (simple roles) and $\mathbf{R_n}$ (non-simple roles). These atomic entities can be used to form complex ones in the usual way (see Table 1).

A *$\mathcal{SROIQ}$-knowledge base* is a tuple $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ where $\mathcal{T}$ is a $\mathcal{SROIQ}$-TBox, $\mathcal{R}$ is a regular $\mathcal{SROIQ}$-role hierarchy[5] and $\mathcal{A}$ is a $\mathcal{SROIQ}$-ABox each being a finite set of corresponding axioms as presented in Table 2. The semantics of $\mathcal{SROIQ}$ is defined via interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ composed of a non-empty set $\Delta^{\mathcal{I}}$ called the *domain of* $\mathcal{I}$ and a function $\cdot^{\mathcal{I}}$ mapping individuals to elements of $\Delta^{\mathcal{I}}$, concepts to subsets of $\Delta^{\mathcal{I}}$ and roles to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. This mapping is extended to complex roles and concepts as in Table 1 and finally used to evaluate axioms (see Table 2). We say $\mathcal{I}$

[5] We assume the usual regularity assumption for $\mathcal{SROIQ}$, but omit it for space reasons.

Name	Syntax	Semantics
inverse role	R^-	$\{(x,y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (y,x) \in R^{\mathcal{I}}\}$
universal role	U	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
nominals	$\{a_1, \dots, a_n\}$	$\{a_1^{\mathcal{I}}, \dots, a_n^{\mathcal{I}}\}$
univ. restriction	$\forall R.C$	$\{x \mid \forall y.(x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
exist. restriction	$\exists R.C$	$\{x \mid \exists y.(x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
Self concept	$\exists S.\mathsf{Self}$	$\{x \mid (x,x) \in S^{\mathcal{I}}\}$
qualified number	$\leqslant n\, S.C$	$\{x \mid \#\{y \in C^{\mathcal{I}} \mid (x,y) \in S^{\mathcal{I}}\} \leq n\}$
restriction	$\geqslant n\, S.C$	$\{x \mid \#\{y \in C^{\mathcal{I}} \mid (x,y) \in S^{\mathcal{I}}\} \geq n\}$

Table 1. Syntax and semantics of role and concept constructors in $\mathcal{SROIQ}$.Herein x and y denote individuals names, R an arbitrary role name and S a simple role name. C and D denote concept expressions.

Axiom α	$\mathcal{I} \models \alpha$, if	
$R_1 \circ \dots \circ R_n \sqsubseteq R$	$R_1^{\mathcal{I}} \circ \dots \circ R_n^{\mathcal{I}} \subseteq R^{\mathcal{I}}$	RBox $\mathcal{R}$
$\mathsf{Dis}(S,T)$	$S^{\mathcal{I}} \cap T^{\mathcal{I}} = \emptyset$	
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$	TBox $\mathcal{T}$
$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$	ABox $\mathcal{A}$
$R(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$	
$a \doteq b$	$a^{\mathcal{I}} = a^{\mathcal{I}}$	
$a \not\doteq b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$	

Table 2. Syntax and semantics of $\mathcal{SROIQ}$ axioms

satisfies a knowledge base $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ (or $\mathcal{I}$ is a model of $\mathcal{O}$, written $\mathcal{I} \models \mathcal{O}$) if it satisfies all axioms of $\mathcal{T}$, $\mathcal{R}$, and $\mathcal{A}$. We say that a knowledge base $\mathcal{O}$ *entails* an axiom α (written $\mathcal{O} \models \alpha$) if all models of $\mathcal{O}$ are models of α.

2.2 Logic Programs

A Normal Logic Program (NLP) is a set of rules of the form

$$h \leftarrow b_1, \dots, b_n, not\ c_1, \dots, not\ c_m \tag{1}$$

with $m, n \geq 0$ and finite, where h, the b_i and the c_j are function-free first-order atoms. *Literals* are atoms or their negations. Default negated literals (DNLs) are those of the form $not\ c$. For a rule of form (1), we use $head(r)$ to represent the atom h and $body(r)$ to represent the set $\{b_1, \dots, b_n, not\ c_1, \dots, not\ c_m\}$. A rule r with $body(r) = \emptyset$ is called a *fact*. An atom a occurring in a program P is called *rule-less* (or we say that it has no rules) if there is no rule r in P such that $head(r) = a$.

Given a program P and a rule $r \in P$, we write $\overline{body(r)}$ to denote the subset of $body(r)$ whose literals' atoms do not have rules that depend on r. Intuitively, $\overline{body(r)}$ represents the part of the body of the rule r not in loop.

A NLP naturally induces a directed graph where the rules are the nodes and there is an arc from r_1 to r_2 iff r_2 syntactically depends on r_1, i.e., if the head of r_1 appears, possibly default negated, in the body of r_2. When two rules depend on each other we say there is a *loop* in this rule graph. When such a loop is formed through DNLs, i.e., when at least one of the heads of rules in the loop appears as a DNL in the body of a rule of the loop, we say there is a *Loop Over Negation* (LON). Moreover, when there is an Even (Odd) number of DNLs through which the LON is formed we say there is an Even (Odd) Loop Over Negation (ELON/OLON).

An Integrity Constraint (IC) is a special kind of logic rule where the head is $\perp$. ICs are not part of NLPs, but (non-Normal) LPs are unions of "normal" rules with ICs. This way, a problem can be modeled by a LP using the normal rules as *generators* of candidate solutions (the models), and using the ICs as *filters* to discard unsatisfying candidate solutions.

2.3 Semantics for Normal Logic Programs

One of the cornerstone semantic principles for NLPs is the Closed World Assumption (CWA) which reifies the *default* character of the not operator in logic programs. Intuitively, the CWA principle says the truth-value of any atom in a program should be $assumed\ false$ unless there is support for its truth via a rule with that atom as head; that is why a program like $P_1 = \{a \leftarrow not\ b\}$ has only the model $\{a, not\ b\}$ — b is $assumed\ false$ by CWA. However, when we have LONs (e.g., $a \leftarrow not\ b \quad b \leftarrow not\ a$) we no longer can apply the CWA directly. Instead, *first* we *assume* some truth-values for the atoms of DNLs, *and then* we apply the CWA under the *assumed* context, i.e., the LONs give us a degree of choice by allowing several alternative models based on freely chosen *assumption* for the truth-values of the atoms of DNLs. The way the SM semantics is defined allows it to assign models to ELONs, but not to OLONs[6].

In order for the intended use of LPs described in 2.2 above to be effective, a semantics for the "normal" part must always guarantee model existence, otherwise there would be some way of using normal rules to play the role of ICs, which would compromise the declarative meaning of both normal rules and ICs. For the sake of declarativity, rules with $\perp$ head should be the only way to write ICs in a LP: no normal rule, or combination of normal rules, should possibly act as IC(s) under any given semantics. An NLP should always have at least one model according to any given 2-valued semantics.

Since the SM semantics fails to assign models to OLONs (which are patterns formed by normal rules alone), it should not be used for the intended purpose of LPs described above. For this reason we turn to the MH semantics which guarantees model existence for NLPs. Our formalism can thus handle hybrid KBs where the NLP contains OLONs.

2.4 Minimal Hypotheses semantics for NLPs

In general, 2-valued semantics for NLPs allow for several possible alternative models. The interpretations accepted as models can be seen as sets of beliefs the semantics

[6] By failing to assign a semantics to OLONs, the SM semantics treats them as modeling errors, although this claim has been refuted time and again — e.g., [7] says "Let P be a knowledge base represented either as a logic program, or as a nonmonotonic theory or as an argumentation framework. Then there is not necessarily a bug in P if P has no stable semantics."

consider plausible. In this regard, the atoms $true$ in a model can be envisaged as either assumed hypotheses or their respective consequences via the rules. The MH semantics considers the atoms of some DNLs in an NLP as the assumable hypotheses (a more detailed explanation is provided below). MH semantics aims at minimizing the sets of assumed hypotheses in each model necessary to fully determine the 2-valued truth-value of all literals in an NLP, and thus takes an approach akin to that of minimal abduction.

MH semantics: Hypotheses set of a NLP Not all atoms of DNLs in a program are eligible as assumable hypotheses (whenever possible the CWA must be enforced). In [18, 19] the authors defined a syntactic transformation applicable to a NLP P in order to find which DNLs are its assumable hypotheses, the set $Hyps(P)$. In an informal and intuitive manner, and resorting to a simpler notation, this is the transformation:

Definition 1. Program Transformation — Layered Remainder of P.

Given an NLP P, the *layered remainder* of P is the program $\mathring{P}$ obtained by repeatedly applying the following reduction rules to P until a fixed-point is reached:

1. for any rule $r \in P$ with $not\ b \in body(r)$ s.t. b has no rules, remove $not\ b$ from $body(r)$.
2. for a rule r and a fact b in P, if $not\ b \in \overline{body(r)}$ [7] then remove r from P.
3. remove every fact a from the body of every rule in P.
4. remove every rule r from P s.t. there is a $b \in body(r)$ and b has no rules.
5. remove all subsets of rules that depend on each other via only positive literals (atoms) in their bodies, but only if there are no other rules with the same heads.

The set $Hyps(P)$ of assumable hypotheses of P is the set of atoms of DNLs of $\mathring{P}$. $\Diamond$

The transformation defined above is similar to the one presented in [3] except that in the condition of the rule 2, $body(r)$ is considered rather than $\overline{body(r)}$. The program obtained from P in that way is called the *remainder* instead and is used to calculate the Well-Founded Model (WFM) of P. For a detail discussion and comparison of both transformations we refer to [18].

Example 2. MH semantics: Layered Remainder.

Let $P = a \leftarrow not\ b \quad b \leftarrow not\ c \quad c \leftarrow not\ a, d \quad d \leftarrow not\ e \quad x \leftarrow x \quad a$
Applying rule 1 from Definition 1 we delete the $not\ e$ from the body of rule $d \leftarrow not\ e$. Because d is now a fact we can apply the rule 3 from Definition 1 deleting d from the body of rule $c \leftarrow not\ a, d$. Applying rule 5 from Definition 1 we delete the rule $x \leftarrow x$ from the program which is now

$P = a \leftarrow not\ b \quad b \leftarrow not\ c \quad c \leftarrow not\ a \quad d \quad a$

Whereas the calculus of the Remainder of P would allow the fact a to be used to delete the rule $c \leftarrow not\ a$ in order to further simplify the program, that is not the case with the Layered Remainder: since the rule $c \leftarrow not\ a$ is involved in a loop with a rule for a, this $c \leftarrow not\ a$ rule is not allowed to be deleted, because $not\ a \notin \overline{body(c \leftarrow not\ a)}$.

[7] The literals in $\overline{body(r)}$ are those whose rules for the corresponding atoms are all in layers strictly below that of r.

No more rules from Definition 1 are applicable and hence we have reached the Layered Remainder of P. The set of assumable hypotheses are the atoms of the DNLs still remaining: $Hyps(P) = \{a, b, c\}$. ◊

Definition 3. Minimal Hypotheses semantics — (simplified from [18, 19]).

For a NLP P with set of assumable hypotheses $Hyps(P)$, a set of literals M is a MH model of P iff the WFM of $P \cup H$ is 2-valued complete and it coincides with M, where $H \subseteq Hyps(P)$ is empty, or non-empty set-inclusion minimal, i.e., H is minimal but sufficient to determine the 2-valued truth-values of all atoms of P[8]. ◊

In the Example 2 above, since $Hyps(P) = \{a, b, c\}$ we have four different subsets of hypotheses that yield MH models: $H_1 = \emptyset$, which yields the MH model M_1 of P which is $M_1 = WFM(P \cup H_1) = WFM(P \cup \emptyset) = WFM(P) = \{a, b, not\ c, d\}$; $H_2 = \{a\}$, which yields the same MH model M_1 of P which is $M_1 = WFM(P \cup H_2) = WFM(P \cup \{a\}) = WFM(P) = \{a, b, not\ c, d\}$; $H_3 = \{b\}$, which yields yet again the same MH model M_1 of P which is $M_1 = WFM(P \cup H_3) = WFM(P \cup \{b\}) = WFM(P) = \{a, b, not\ c, d\}$; and finally $H_4 = \{c\}$, which yields the other MH model M_2 of P which is $M_2 = WFM(P \cup H_4) = WFM(P \cup \{c\}) = WFM(P) = \{a, not\ b, c, d\}$. The MH models of the program in Example 2 are thus $M_1 = \{a, b, not\ c, d\}$, and $M_2 = \{a, not\ b, c, d\}$.

3 Hybrid Knowledge Bases: *Syntax and Semantics*

A hybrid DL-NLP KB $\mathcal{K}$ is a pair $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ where $\mathcal{O}$ is a DL-KB and $\mathcal{P}$ is an NLP. Let $\Sigma_{\mathcal{O}}$ denote the signature of (the set of predicate symbols and constants occurring in) $\mathcal{O}$, and $\Sigma_{\mathcal{P}}$ denote the signature of $\mathcal{P}$, then $\Sigma_{\mathcal{K}}$ denotes common signature of $\mathcal{K}$, i.e., the set of *shared* predicate symbols and constants occurring in both $\mathcal{O}$ and $\mathcal{P}$ — $\Sigma_{\mathcal{K}} = \Sigma_{\mathcal{O}} \cap \Sigma_{\mathcal{P}}$. Let $\mathcal{AB}_{\Sigma}$ denote the set of all possible atoms over signature Σ.

Example 4. The affordable car problem.

Consider this example in which we present a hypothetical online recommendation system for selling vehicles. The background knowledge of the car sales company is formalized in an ontology that contains the following axioms:

$$\mathit{Vehicle} \equiv \mathit{Car} \sqcup \mathit{Van} \sqcup \mathit{Truck} \quad (1)$$
$$\mathit{Car} \equiv \mathit{ABS} \sqcup \mathit{Airbagged} \sqcup \mathit{Automatic} \quad (2)$$
$$\mathit{AffordableCar} \equiv \mathit{Car} \sqcap \neg(\mathit{ABS} \sqcap \mathit{Airbagged} \sqcap \mathit{Automatic}) \sqcap \mathit{StandardSeats} \quad (3)$$
$$\mathit{LuxuryCar} \equiv \mathit{Car} \sqcap \mathit{ABS} \sqcap \mathit{Airbagged} \sqcap \mathit{Automatic} \sqcap \mathit{LeatherSeats} \quad (4)$$

and in a logic program containing the non-monotonic rule:

$$\mathit{StandardSeats(C)} \leftarrow not\ \mathit{LeatherSeats(C)} \quad (5)$$

Besides the kind of vehicles for sale (Axiom (1)), the ontology says the cars this company sells always come with some different system (Axiom (2)). According to

[8] The reasons for allowing non-empty set-inclusion minimal Hs are explained in [18] and they are related to allowing cumulativity.

Axiom (3) an affordable car is a car missing at least one of the systems (ABS, Airbags or Automatic transmission) and has standard seats. Moreover, luxury cars have all three systems and special leather seats (Axiom (4)). In other words, each system in a car, and the special seats, add to the price of the car. Also, by default, a car is sold with standard fabric seats, unless it is explicitly demanded by the customer the car must have leather seats. This is expressed by Rule (5) in the program.

Suppose now there is a customer who will be happy if she gets an affordable car c, and her preferences regarding car systems are given as in the following rules:

$$\textit{Automatic(c)} \leftarrow not\ \textit{ABS(c)} \qquad (6)$$
$$\textit{ABS(c)} \leftarrow not\ \textit{Airbagged(c)} \qquad (7)$$
$$\textit{Airbagged(c)} \leftarrow not\ \textit{Automatic(c)} \qquad (8)$$
$$\textit{Happy} \leftarrow \textit{AffordableCar(c)} \qquad (9)$$

The task at hand here is to find an affordable car while still satisfying her preferences. Note that using the stable models as the semantic basis for this hybrid KB leads to no solution for the problem because the SM semantics is unable to assign models to the OLON formed by the rules $(6), (7)$ and (8). We will see later that such a system is easily realizable in our approach. ◊

The semantics of a hybrid KB $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ must take into account the semantics of both of its components $\mathcal{O}$ and $\mathcal{P}$ — we consider the MH semantics for the NLP part. The intuitive definition of our semantics is as follows. Since MHs allows for the existence of several alternative models and the ontology have several models, the joint KB $\mathcal{K}$ must also allow for the existence of several hybrid models. Also, the literals of a model of each one of the two components must be used by the other to allow for the possible entailment of further consequences. Hence, the truth values of atoms in a non-deterministically chosen MH model of the NLP part, which are part of the shared symbols in $\Sigma_{\mathcal{K}}$, may allow for the further entailment of new conclusions in the DL part. In turn, these new DL entailments, which are also part of the same $\Sigma_{\mathcal{K}}$, may allow further conclusions to be drawn in the NLP part in the form of a new MH model with more $true$ atoms. Also, coherence must be enforced: explicitly negated literals entailed from the DL part must imply their default negated shared $\Sigma_{\mathcal{K}}$ counterparts in the NLP part. This mutual, or *crossed*, enrichment of each DL and NLP component with the truth-values of shared $\Sigma_{\mathcal{K}}$ atoms coming from the other component must be repeated until a joint *crossed* fixed-point is reached. There may be several crossed fixed-point hybrid interpretations, only the consistent ones are our hybrid models.

Definition 5. MH-based semantics of hybrid KB.

Let $\mathcal{O}$ be a consistent DL theory and $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a hybrid DL-NLP KB. A pair (I, M) is an MH-based hybrid model of $\mathcal{K}$ iff

- M is an MH model of $\mathcal{P} \cup (I^+ \cap \mathcal{AB}_{\Sigma_{\mathcal{K}}})$ with
- $\{not\ B : \neg B \in I^- \wedge B \in \mathcal{AB}_{\Sigma_{\mathcal{K}}}\} \subseteq M^-$ (coherence) and
- $\big(\mathcal{O} \cup (M^+ \cap \mathcal{AB}_{\Sigma_{\mathcal{K}}}) \cup (\{\neg B : not\ B \in M^- \wedge B \in \mathcal{AB}_{\Sigma_{\mathcal{K}}}\})\big) \cup I$ is consistent.

where $M = M^+ \cup M^-$, $M^+ \subseteq \mathcal{AB}_{\Sigma_{\mathcal{P}}}$, $M^- = \{not\ B : B \in \mathcal{AB}_{\Sigma_{\mathcal{P}}} \setminus M^+\}$; and $I = I^+ \cup I^-$, $I^+ \subseteq \mathcal{AB}_{\Sigma_{\mathcal{O}}}$, $I^- = \{\neg B : B \in \mathcal{AB}_{\Sigma_{\mathcal{O}}} \setminus I^+\}$. We use the term hybrid model instead of MH-based hybrid model whenever it is obvious from the context. ◊

Consider again the Example 4 about the affordable car problem with the unique constant c. The atom bases are

$$\mathcal{AB}_{\Sigma_\mathcal{P}} = \{\textit{StandardSeats(c)}, \textit{LeatherSeats(c)}, \textit{Automatic(c)}, \textit{ABS(c)}, \textit{Airbagged(c)}, \textit{Happy}, \textit{AffordableCar(c)}\},$$
$$\mathcal{AB}_{\Sigma_\mathcal{O}} = \{\textit{StandardSeats(c)}, \textit{LeatherSeats(c)}, \textit{Automatic(c)}, \textit{ABS(c)}, \textit{Airbagged(c)}, \textit{Vehicle(c)}, \textit{AffordableCar(c)}, \textit{LuxuryCar(c)}, \textit{Car(c)}, \textit{Van(c)}, \textit{Truck(c)}\}, and$$
$$\mathcal{AB}_{\Sigma_\mathcal{K}} = \{\textit{StandardSeats(c)}, \textit{LeatherSeats(c)}, \textit{Automatic(c)}, \textit{ABS(c)}, \textit{Airbagged(c)}, \textit{AffordableCar(c)}\}.$$

If we take the NLP part alone, resulting from joining together the rule of the car sales company with the rules of the buyer, we get

$$\textit{StandardSeats(c)} \leftarrow not\ \textit{LeatherSeats(c)}$$
$$\textit{Automatic(c)} \leftarrow not\ \textit{ABS(c)}$$
$$\textit{ABS(c)} \leftarrow not\ \textit{Airbagged(c)}$$
$$\textit{Airbagged(c)} \leftarrow not\ \textit{Automatic(c)}$$
$$\textit{Happy} \leftarrow \textit{AffordableCar(c)}$$

This program has three MH models:

$$M_1 = \{\textit{StandardSeats(c)}, not\ \textit{LeatherSeats(c)}, \textit{Airbagged(c)}, \textit{ABS(c)}, not\ \textit{Automatic(c)}, not\ \textit{Happy}, not\ \textit{AffordableCar(c)}\},$$
$$M_2 = \{\textit{StandardSeats(c)}, not\ \textit{LeatherSeats(c)}, \textit{Airbagged(c)}, not\ \textit{ABS(c)}, \textit{Automatic(c)}, not\ \textit{Happy}, not\ \textit{AffordableCar(c)}\}, and$$
$$M_3 = \{\textit{StandardSeats(c)}, not\ \textit{LeatherSeats(c)}, not\ \textit{Airbagged(c)}, \textit{ABS(c)}, \textit{Automatic(c)}, not\ \textit{Happy}, not\ \textit{AffordableCar(c)}\}.$$

Notice that *StandardSeats(c)* is $true$ and *LeatherSeats(c)*, *Happy*, and *AffordableCar(c)* are $false$ in every MH model of this NLP part alone. Taking, e.g., M_1, and adding $M_1^+ \cap \mathcal{AB}_{\Sigma_\mathcal{K}}$ and $\{\neg B : not\ B \in M_1^- \wedge B \in \mathcal{AB}_{\Sigma_\mathcal{K}}\}$ to $\mathcal{O}$ we get, along with the Axiom (1), (2), (3) and (4) in the ontology, the following ABox axioms
StandardSeats(c), ¬*LeatherSeats(c)*, *Airbagged(c)ABS(c)*, ¬*Automatic(c)*, ¬*AffordableCar(c)*

This extended ontology is now inconsistent: because we have, e.g., *ABS(c)*, we conclude *Car(c)* via the second axiom; also, because we have *Car(c)*, *StandardSeats(c)* and ¬*Automatic(c)* we conclude *AffordableCar(c)* via the third axiom, which explicitly contradicts the
¬*AffordableCar(c)* in the ontology. This shows that we must not just simply take an MH model (or even Stable Model, for that matter) for the NLP part and expect it to be straightforwardly a part of a hybrid model of the whole KB; there might be consequences entailed by the DL part that contradict what is entailed by the NLP part alone — that is the case with *AffordableCar(c)* in this example. Of course one might take advantage of such contradictions when developing a concrete algorithm to construct hybrid models as a means to revise the initially guessed pair (I, M), but we do not explore that path in this paper, leaving it for future work. We must guess a pair (I, M) and check if if complies with the three conditions in Definition 5. In this case an MH-based hybrid model would be (I, M) where $I = \{$*StandardSeats(c)*, ¬*LeatherSeats(c)*, *Airbagged(c)*, *ABS(c)*, ¬*Automatic(c)*, *AffordableCar(c)*, *Car(c)*, ¬*LuxuryCar(c)*, *Vehicle(c)*$\}$ and

$M = \{$*StandardSeats(c)*$, not$ *LeatherSeats(c)*, *Airbagged(c)*, *ABS(c)*, not *Automatic(c)*, *Happy*, *AffordableCar(c)*$\}$.

Our formalism naturally supports non-monotonicity in the NLP part: suppose now the customer won a lottery and so we add the new fact *LeatherSeats(c)*. In this case all the MH models will include *LeatherSeats(c)* and not *StandardSeats(c)* (and the hybrid models will include $\neg$*StandardSeats(c)* from the ontology part as well).

Before discussing different reasoning tasks in our approach, we define the notion of satisfiability of a first-order atom in a hybrid model of a hybrid KB. Note that this depends on the atom base of the signature (of ontology or program) the atom belongs to.

Definition 6. MH-based hybrid model satisfies atom.

Let (I, M) be an MH-based hybrid model of a hybrid KB $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ and let A be a First-order atom A. We say A *is satisfied in* (I, M), written $(I, M) \models A$, iff

- $\mathcal{O} \cup (M^+ \cap \mathcal{AB}_{\Sigma_\mathcal{K}}) \cup (\{\neg B : not\ B \in M^- \wedge B \in \mathcal{AB}_{\Sigma_m knfK}\}) \cup I \models A$ whenever $A \in \mathcal{AB}_{\Sigma_\mathcal{O}}$, and
- $A \in M$ whenever $A \in \mathcal{AB}_{\Sigma_\mathcal{P}}$ $\Diamond$

The set I in Def. 5 can be understood as a belief set one possesses s.t. it is consistent with what is stated in the ontology $\mathcal{O}$ along with the information entailed by the program ($M^+ \cap \mathcal{AB}_{\Sigma_\mathcal{K}}$ and $\{\neg B : not\ B \in M \wedge B \in \mathcal{AB}_{\Sigma_\mathcal{K}}\}$). Thus, for an atom to be satisfied in a hybrid model (I, M), it is required to be a logical consequence of the ontology along with the added information from the program and the belief set I if $A \in \mathcal{AB}_{\Sigma_\mathcal{O}}$. And if $A \in \mathcal{AB}_{\Sigma_\mathcal{P}}$ then it is required to be $true$ in M which is an MH model of the program enlarged with the new facts entailed from the ontology that are shared.

4 Reasoning

We present some important reasoning tasks and also present computational complexity results related to these tasks. But first note that we never discussed the DL-safety restriction on the rules occurring in the logic program part. Indeed, the rules never violate the restriction as the only way for the ontology and program to communicate with each other is via a finite set of shared ground atoms. Hence DL-safety restriction is satisfied trivially for all the rules.

Now similar to knowledge bases in other formalisms, one of the most basic reasoning task is to check if a given hybrid KB is consistent.

Definition 7. Consistency of a Hybrid KB.

Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a hybrid KB. Then $\mathcal{K}$ is said to be *consistent* iff there is at least one MH-based model for $\mathcal{K}$. $\mathcal{K}$ is said to be *inconsistent* otherwise. $\Diamond$

The checking of whether a hybrid KB is consistent is termed as *the consistency problem.*

Another interesting reasoning task is to check if a given atom is entailed from a hybrid KB. Like in non-monotonic formalisms we distinguish between credulous and skeptical reasoning here.

Definition 8. Entailment.

Given a first order atom A and a hybrid KB $\mathcal{K}$, we say A is *credulously/skeptically entailed from* $\mathcal{K}$ (written as $\mathcal{K} \models_C A / \mathcal{K} \models A$) iff for some/every MH-based hybrid model (I, M) of $\mathcal{K}$ we have that A is satisfied in (I, M) i.e., $(I, M) \models A$. $\Diamond$

Then the *entailment problem* is to check if the atom A is entailed (credulously or sceptically) from the hybrid KB $\mathcal{K}$.

It follows from Definition 5 that checking the consistency of a hybrid DL-NLP KB requires guessing sets I and M such that the conditions imposed by the definition are satisfied. In the following lemma we prove that the consistency problem is decidable provided the underlying language for formalizing the ontology is decidable. Further we show that the problem is worst-case optimal in the sense that it is no worse than the reasoning problem in the DL part or in the NLP part under MH semantics. To see this, let $\mathcal{L}$ be the description logic for formulating ontologies and let $\mathcal{C}$ be the complexity of the consistency problem in $\mathcal{L}$. Then,

Lemma 9. *Complexity of the consistency problem.*

For a given hybrid DL-NLP KB $\mathcal{K} = (\mathcal{O}, \mathcal{P})$, the consistency problem can be checked in

- *$\mathcal{C}$ if $\mathcal{C}$ is computationally worse than Σ_2^P, and*
- *Σ_2^P otherwise.*

Proof By definition both $\mathcal{AB}_{\Sigma_\mathcal{O}}$ and $\mathcal{AB}_{\Sigma_\mathcal{P}}$ are finite, indeed, polynomial in the size of the knowledge base. We non-deterministically guess the sets of atoms $I^+ \subseteq \mathcal{AB}_{\Sigma_\mathcal{O}}$ and $M^+ \subseteq \mathcal{AB}_{\Sigma_\mathcal{P}}$, and set $I = I^+ \cup I^-$ and $M = M^+ \cup M^-$ such that $I^- = \{\neg B : B \in \mathcal{AB}_{\Sigma_\mathcal{O}} \setminus I^+\}$ and $M^- = \{not\ B : B \in \mathcal{AB}_{\Sigma_\mathcal{P}} \setminus M^+\}$.

Now checking if M is an MH model of $\mathcal{P} \cup (I^+ \cap \mathcal{AB}_{\Sigma_\mathcal{K}})$ can be performed in Σ_2^P [19]. Coherence (the second condition in Definition 5) can be checked in linear time in the size of I and M. Finally, checking if the ontology $\big(\mathcal{O} \cup (M^+ \cap \mathcal{AB}_{\Sigma_\mathcal{K}}) \cup (\{\neg B : not\ B \in M^- \wedge B \in \mathcal{AB}_{\Sigma_\mathcal{K}}\})\big) \cup I$ is a consistent $\mathcal{L}$ KB can be performed in $\mathcal{C}$. Now, if $\mathcal{C}$ is worse than Σ_2^P, the overall consistency problem can be decided in $\mathcal{C}$. Otherwise, it can be decided in Σ_2^P.

Hence, the overall complexity of the consistency problem is dependent on the complexity of reasoning task in the underlying ontology language. By fixing $\mathcal{SROIQ}$ [11] as the ontology language, we get the following results:

Theorem 10. *Complexity of the consistency problem with $\mathcal{SROIQ}$ DL.*

Given a hybrid DL-NLP KB $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ with $\mathcal{O}$ formulated in the description logic $\mathcal{SROIQ}$, the problem of checking the consistency of $\mathcal{K}$ is N2EXPTIME*-complete.*

Proof The consistency in $\mathcal{SROIQ}$ is known to be in N2EXPTIME [13]. Thus as a consequence of Lemma 9, the consistency problem of $\mathcal{K}$ is in N2EXPTIME as well. The lower bound immediately follows from the fact that a $\mathcal{SROIQ}$ ontology can be translated into a hybrid KB by considering the program part empty and that the reasoning problem in $\mathcal{SROIQ}$ is N2EXPTIME-complete [13]. $\square$

Another important consequence of Lemma 9 is that by considering some tractable description logic, e.g., $\mathcal{EL}^{++}$ [1], the consistency problem in our approach does not get tractable. Indeed, it is Σ_2^P-hard regardless of the language of the ontology.

We now discuss the complexity issues regarding the entailment problem in hybrid KBs. Note that unlike description logics, we cannot translate the entailment problem in the consistency problem. This is the case as our formalism is non-monotonic in general. To check if a given atom A is entailed credulously from a hybrid KB $\mathcal{K}$ is to check for a hybrid model of $\mathcal{K}$ such that A is satisfied in it. This can be performed as follows:

if there are $I = (I^+ \cup I^-)$ and $M = (M^+ \cup M^-)$ with
$I^+ \subseteq \mathcal{AB}_{\Sigma_\mathcal{O}}$ and $I^- = \{\neg B : B \in \mathcal{AB}_{\Sigma_\mathcal{O}} \setminus I^+\}$, and
$I \subseteq \mathcal{AB}_{\Sigma_\mathcal{P}} \setminus \{A\}$ and $M^- = \{not\ B : B \in \mathcal{AB}_{\Sigma_\mathcal{P}} \setminus M^+\}$
such that
(1) M is an MH-based hybrid model of $\mathcal{P} \cup (I^+ \cap \mathcal{AB}_{\Sigma_\mathcal{K}})$
(2) $\{not\ B : \neg B \in I^- \wedge B \in \mathcal{AB}_{\Sigma_\mathcal{K}}\} \subseteq M^-$
(3) $\big(\mathcal{O} \cup (M^+ \cap \mathcal{AB}_{\Sigma_\mathcal{K}}) \cup (\{\neg B : not\ B \in M^- \wedge B \in \mathcal{AB}_{\Sigma_\mathcal{K}}\})\big) \cup I$ is consistent.
(4) $A \in M$ whenever $A \in \mathcal{AB}_{\Sigma_\mathcal{P}}$
(5) $\mathcal{O} \cup (M^+ \cap \mathcal{AB}_{\Sigma_\mathcal{K}}) \cup (\{\neg B : not\ B \in M^- \wedge B \in \mathcal{AB}_{\Sigma_\mathcal{K}}\}) \cup I \models A$ whenever $A \in \mathcal{AB}_{\Sigma_\mathcal{O}}$
then $\mathcal{K} \models_C A$; otherwise $\mathcal{K} \not\models_C A$

The steps for checking credulous entailment can be modified in order to check the sceptical entailment of an atom from a hybrid KB. The idea is to check that no hybrid model of the knowledge base is such that the atom is false in it. This is achieved by replacing condition (4) and (5) with
$(4')$ $not\ A \in M$, and
$(5')$ $\big(\mathcal{O} \cup (M^+ \cap \mathcal{AB}_{\Sigma_\mathcal{K}}) \cup (\{\neg B : not\ B \in M^- \wedge B \in \mathcal{AB}_{\Sigma_\mathcal{K}}\})\big) \cup I \not\models A$
respectively. Now if all the conditions are satisfied for some hybrid model, then $\mathcal{K} \not\models A$.

Similar to Lemma 9, we can prove that the complexity of the credulous entailment problem is Σ_2^P if the entailment problem in the ontology is better than Σ_2^P and it is same as the complexity of the entailment problem in the ontology otherwise. Additionally we can show that the sceptical entailment problem is in $\Sigma_2^P \cap \Pi_2^P$ [19]. if the entailment problem in the ontology is computationally better than $\Sigma_2^P \cap \Pi_2^P$. It corresponds to the complexity of the entailment problem in the ontology otherwise.

Again by fixing $\mathcal{SROIQ}$ as the ontology language we get the following results:

Theorem 11. *Complexity of the entailment problem with $\mathcal{SROIQ}$ DL.*

Given a hybrid DL-NLP KB $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ and a first-order atom A, both, credulous and sceptical entailment problem is N2EXPTIME*-complete.*

Proof similar to Theorem 10. □

5 Related Work

Among several existing approaches towards unifying DL with rules, probably the most mature ones include the formalisms presented in ([15], [16]) and [14]. In both approaches MKNF theories are used for specifying DL KB and rules. In [16], standard MKNF semantics is used which corresponds to the SM semantics of LPs. Comparing to our work: first our approach is not based on MKNF; rather, hybrid KBs in our approach are just pairs consisting of an ontology and an NLP with an overlapping signature. Further, in the approach of [16], hybrid KBs with rules forming a OLONs are inherently inconsistent. Nevertheless, as already mentioned, OLON are required for further expressivity in a rule-based language. In contrast, our approach can deal with NLPs containing OLONs. Since every stable model of a program is also its MH-model, our approach is more general than that of [16] for the hybrid KBs restricted to a DL KB and NLPs. A variation of the [16] approach is presented in [14] where WFS for MKNF theories is defined. This approach can handle OLONs by using a third truth value namely "undefined". But such approaches are not suitable when a 2-valued semantics is required.

The work closely related to our approach is that of the so-called multi-context system (MCS), a framework that allows for combining arbitrary monotonic and non-monotonic logics [4]. Hybrid KBs in our approach can be taken as multi-context system with two contexts, an ontology context and a program context. But comparing to our approach, note that we do not have, *per se*, the notion of bridge rules, rather, the exchange of information is via the shared signature only. This is in contrast to multi-context system where bridge rules are part of the multi-context system explicitly specified and that the head of the the rules are the only information that can be added to the knowledge base of the context to which the rule belongs to. Another interesting point we mention here is that the notion of MH-based hybrid models corresponds to the notion of equilibrium in [4]. An equilibrium is a belief state which contains for each context an acceptable belief set, given the belief sets of the other contexts. In our approach, a tuple (I, M) is said to be a hybrid model such that M is a MH model of the logic program part given the set I and additionally I is consistent with the ontology given the set M. Nevertheless, one still has to see how to translate hybrid KBs into MCS and use the known results from there.

6 Conclusion

We presented a new approach to integrate DL with NLPs. The underlying semantics for the NLP considered is the MH semantics which guarantees the existence of models even for programs with OLONs. By defining a fixed-point hybrid KB, we presented a practical method to draw crossed entailments from the KB. Moreover, our MH-based approach, besides covering all the cases an SM-based one would, still keeps within optimal complexity results.

As the avenues for the future work we will 1) extend this framework to deal with other classes of LPs, such as Extended LPs (which allow for explicit negation) and Disjunctive LPs (which allow for disjunctions in the heads of rules where the disjuncts may be explicitly negated), and in doing so we will resort to an extension of the MH semantics to those classes of LPs; 2) develop algorithms for computing hybrid models, and for handling the reasoning tasks defined in Section 4 – using intelligent heuristics these should be as efficient as possible in most cases, while still lying within the known complexity results. In doing so, the fixed-point semantics of our formalism lets us to take a black-box approach by using a standard DL reasoner and a tool for computing MH models.

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence IJCAI-05. Morgan-Kaufmann Publishers, Edinburgh, UK (2005)
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
3. Brass, S., Dix, J., Freitag, B., Zukowski, U.: Transformation-based bottom-up computation of the well-founded model. TPLP 1(5), 497–538 (2001)
4. Brewka, G., Eiter, T.: Equilibria in heterogeneous nonmonotonic multi-context systems. In: AAAI. pp. 385–390 (2007)
5. Dix, J.: A Classification Theory of Semantics of Normal Logic Programs: I. Strong Properties. Fundam. Inform. 22(3), 227–255 (1995)
6. Dix, J.: A Classification Theory of Semantics of Normal Logic Programs: II. Weak Properties. Fundam. Inform. 22(3), 257–288 (1995)
7. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. AI 77(2), 321–358 (1995)
8. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. AI 172(12-13), 1495–1539 (2008)
9. Gelder, A.V., Ross, K.A., Schlipf, J.S.: The well-founded semantics for general logic programs. J. ACM 38(3), 620–650 (1991)
10. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R., Bowen, Kenneth (eds.) Procs. ICLP'88. pp. 1070–1080. MIT Press (1988)
11. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06). pp. 57–67. AAAI Press (2006)
12. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML (2004), http://www.w3.org/Submission/SWRL/
13. Kazakov, Y.: Riq and sroiq are harder than shoiq. In: In Proc. KR'08 (2008)
14. Knorr, M., Alferes, J.J., Hitzler, P.: A coherent well-founded model for hybrid mknf knowledge bases. In: Procs. ECAI 2008. pp. 99–103. IOS Press (2008)
15. Motik, B., Rosati, R.: A faithful integration of description logics with logic programming. In: Procs. IJCAI'07. pp. 477–482. Morgan Kaufmann Publishers Inc. (2007)
16. Motik, B., Rosati, R.: Reconciling Description Logics and Rules. Journal of the ACM 57(5), 1–62 (2010)
17. OWL Working Group, W.: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation (27 October 2009), available at http://www.w3.org/TR/owl2-overview/
18. Pinto, A.M.: Every normal logic program has a 2-valued semantics: theory, extensions, applications, implementations. Ph.D. thesis, Universidade Nova de Lisboa (July 2011)
19. Pinto, A.M., Pereira, L.M.: Each normal logic program has a 2-valued minimal hypotheses semantics. CoRR abs/1108.5766 (2011)
20. Rosati, R.: On the decidability and complexity of integrating ontologies and rules 3(1), 41–60 (2005)