

RDL: Enhancing Description Logic with Rules

Yi Zhou and Yan Zhang

Artificial Intelligence Research Group
School of Computing, Engineering and Mathematics
University of Western Sydney, Australia

Abstract. In this paper, we propose Rule Description Logic (RDL) for enhancing Description Logic (DL) with nonmonotonic recursive rules, like those in Answer Set Programming (ASP). We define the world view semantics for RDL and show that it is faithful with respect to both DL and ASP. More importantly, we show that the full language of RDL is decidable.

1 Introduction

In the last two decades, the integration of Description Logics (DL) and nonmonotonic formalisms and/or rule-based formalisms has been one of the major challenging tasks in the area of knowledge representation [1–11]. Recently, research on this topic has become even more attractive due to the success of applying description logics to Semantic Web [12] and the urgent task in it of adding rules onto the ontology layer.

Meanwhile, Answer Set Programming (ASP) has emerged as a promising paradigm for nonmonotonic rule-based reasoning due to its declarative semantics and computational advantages [13–15]. In fact, ASP is exactly a representative approach that combines nonmonotonic reasoning and recursive rule-based reasoning in a simple and natural way.

Under this backdrop, we consider the task of enhancing Description Logics with nonmonotonic recursive rules, like those in Answer Set Programming. By nonmonotonic, we mean, syntactically, the negation-as-failure operator might be used in rules, and semantically, some consequences might not be preserved when adding new knowledge. By recursive, we mean iterative reasoning procedure will be applied on rules.

As argued in [7], work in this aspect should satisfy the following criteria:

- *Faithfulness.* The integration of DL and ASP should preserve the semantics of both formalisms.
- *Tightness.* Both the DL and the ASP component should be able to contribute to the consequences of the other.
- *Flexibility.* The integration should be flexible to the Closed/Open World Assumption.
- *Decidability.* The integration should be decidable.

In addition, we believe that a *direct* semantics is essential for the integration (of DL and ASP) in order to capture its underlying intuitions.

Although it looks natural, these criteria are hard to be achieved at the same time. As far as we know, none of the existing work satisfies all the above criteria. For instance, some of them are not tight as one component is building upon another. More importantly, many of the existing approaches are not decidable for their full language. Usually, a strong syntactic condition called safety needs to be assumed in order to obtain decidability. This, indeed, loses the most favorable feature of Description Logic.

In this paper, we propose a novel formalism, called Rule Description Logic (RDL for short), whose syntax is extended from description logic with the rules of the form

$$C_0 \leftarrow C_1, \dots, C_m, \text{not } C_{m+1}, \dots, \text{not } C_n,$$

where $0 \leq m \leq n$, and all $C_i (0 \leq i \leq n)$ are $\perp$ or concepts in description logic. We define a world view semantics for RDL based on a fixed-point definition, similar to Reiter's default logic [16]. Roughly speaking, a world view of an RDL rule base is a collection of interpretations, representing a possible set of models that an agent believes by the given rule base.

We show that RDL satisfies all the criteria mentioned previously. We prove that the world view semantics is faithful in the sense that restricting an RDL rule base to a knowledge base in DL (a logic program in ASP), the world views of the rule base coincide with the models of the knowledge base (the answer sets of the program, resp.). More importantly, we prove the bounded model property for RDL. As a consequence, the full language of RDL (i.e., no further syntactic restriction required) is decidable.

The rest of the paper is organized as follows. The following section briefly introduces the basic description logic $\mathcal{ALC}$. Next, section 3 presents the syntax as well as the semantics of RDL and addresses some basic related issues. Section 4 proves that RDL is faithful with respect to both DL and ASP, while section 5 proves that RDL is decidable. Then, section 6 discusses some related works. Finally, section 7 concludes the paper with some remarks.

2 The Description Logic $\mathcal{ALC}$

In this paper, we focus on the basic description logic $\mathcal{ALC}$, which includes and only includes the basic concept constructors of conjunction, disjunction, negation and existential and universal quantifications [17]. The syntax of $\mathcal{ALC}$ is defined over a set N_C of *concept names*, a set N_R of *role names*, a set N_I of *individual names* and the connectives $\top$ (tautology), $\perp$ (falsity), $\neg$ (negation), $\sqcap$ (conjunction), $\sqcup$ (disjunction), $\exists$ (existential quantification) and $\forall$ (universal quantification). In $\mathcal{ALC}$, concepts are defined recursively as follows:

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

where $A \in N_C$ and $R \in N_R$.

The semantics of $\mathcal{ALC}$ is defined in terms of an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is the *domain* that contains a non-empty set of elements, and $\cdot^{\mathcal{I}}$ is the *interpretation function* that maps each concept name $A \in N_C$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each role name $R \in N_R$ to a binary relation $R^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, and each individual name $a \in N_I$ to an element $a^{\mathcal{I}}$ in $\Delta^{\mathcal{I}}$. The interpretation function is extended to all concepts recursively as follows:

- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$.
- $\perp^{\mathcal{I}} = \emptyset$.
- $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$.
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$.
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$.
- $(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b. \text{ if } (a, b) \in R^{\mathcal{I}}, \text{ then } b \in C^{\mathcal{I}}\}$.
- $(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b. (a, b) \in R^{\mathcal{I}} \text{ and } b \in C^{\mathcal{I}}\}$.

We say that an interpretation $\mathcal{I}$ *satisfies* a concept C , or $\mathcal{I}$ is a *model* of C , iff $C^{\mathcal{I}} \neq \emptyset$.

A *TBox* is a finite set of *general concept implications* of the form $C \sqsubseteq D$, where C and D are concepts. In addition, we write $C \equiv D$ as a shorthand of $C \sqsubseteq D$ and $D \sqsubseteq C$. An *ABox* is a finite set of *concept assertions* of the form $C(a)$ and *role assertions* of the form $R(a, b)$, where a and b are individual names, R is a role name and C is a concept. A *(DL) knowledge base (KB)* is a pair $(\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox.

An interpretation $\mathcal{I}$ *satisfies* a general concept implication $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; $\mathcal{I}$ *satisfies* a concept assertion $C(a)$ if $a \in C^{\mathcal{I}}$; $\mathcal{I}$ *satisfies* a role assertion $R(a, b)$ if $(a, b) \in R^{\mathcal{I}}$. Then, $\mathcal{I}$ *satisfies*, or $\mathcal{I}$ is a *model* of, a TBox $\mathcal{T}$ if it satisfies all general concept implications in $\mathcal{T}$, an ABox $\mathcal{A}$ if it satisfies all (concept and role) assertions in $\mathcal{A}$ and a DL knowledge base $(\mathcal{T}, \mathcal{A})$ if it satisfies both $\mathcal{T}$ and $\mathcal{A}$.

3 RDL: Syntax and Semantics

This section presents the syntax as well as the semantics of RDL, which can be regarded as an integration of Description Logic and Answer Set Programming. However, although Description Logic (i.e. $\mathcal{ALC}$) is fully embraced, RDL only contains a fragment of Answer Set Programming. Nevertheless, as we will argue later, this fragment is indeed adequate for the task of enhancing DL with rules.

The syntax of RDL is extended from $\mathcal{ALC}$ by adding rules of the form

$$C_0 \leftarrow C_1, \dots, C_m, \text{not } C_{m+1}, \dots, \text{not } C_n, \quad (1)$$

where $0 \leq m \leq n$, and $C_i (0 \leq i \leq n)$ is a concept or $\perp$. Roughly speaking, this represents a nonmonotonic recursive rule that applies on all individuals.

An *RBox*¹ is a finite set of rules of form (1). An *(RDL) rule base (RB)* is a triple $(\mathcal{R}, \mathcal{T}, \mathcal{A})$, where $\mathcal{R}$ is an RBox and $(\mathcal{T}, \mathcal{A})$ is a standard DL knowledge base. In particular, a knowledge base can be considered as a rule base with empty RBox.

The semantics of RDL is defined in terms of collections of interpretations instead of interpretations by fixing a domain and the interpretations on individuals. Let Δ be a non-empty set of elements, called a *domain*. A *collection of interpretations* on Δ (*collection* for short if clear from the context) is a non-empty set of interpretations that share the same domain Δ and map each individual $a \in \mathbb{N}_{\perp}$ to the same element in Δ . In other words, each collection $\mathcal{W}$ is associated with a domain Δ and a mapping M from all individuals to elements in Δ , which are recognized by all interpretations in the

¹ Although this term appeared elsewhere in the literature, we use it in RDL for convenience. The syntactic and semantic meanings are different from other approaches.

collection. For convenience, we use $\Delta^{\mathcal{W}}$ to denote the domain and $M^{\mathcal{W}}$ to denote the *individual mapping*.

Let $\Gamma = (\mathcal{R}, \mathcal{T}, \mathcal{A})$ be a rule base and $\mathcal{W}$ a collection. We define an operator $\Gamma(\mathcal{W})$ by the maximal collection (in the sense of set inclusion) satisfying the following conditions.

1. $\Gamma(\mathcal{W})$ has the same domain and the same individual mapping as $\mathcal{W}$.
2. Every interpretation in $\Gamma(\mathcal{W})$ is a model of the knowledge base $(\mathcal{T}, \mathcal{A})$.
3. For each rule r in R of the form (1) and each d in $\Delta^{\mathcal{W}}$, if
 - (a) for all $i (1 \leq i \leq m)$, and all $\mathcal{I} \in \Gamma(\mathcal{W})$, $d \in C_i^{\mathcal{I}}$ and
 - (b) for all $j, (m+1 \leq j \leq n)$, there exists $\mathcal{J} \in \mathcal{W}$ such that $d \notin C_j^{\mathcal{J}}$,
 then $d \in C_0^{\mathcal{I}}$ for all $\mathcal{I} \in \Gamma(\mathcal{W})$.

We say that a collection $\mathcal{W}$ is a *world view*² of a rule base Γ iff $\mathcal{W}$ is a fixed point of the operator Γ , i.e. $\Gamma(\mathcal{W}) = \mathcal{W}$. In this case, we say that $\mathcal{W}$ is a world view of Γ with respect to $\Delta^{\mathcal{W}}$ and $M^{\mathcal{W}}$.

The world view semantics for RDL follows some basic ideas of the well-known semantics for default logic [16]. First of all, both semantics define an operator Γ , and take the fixed point of Γ into account. Secondly, all the (nonmonotonic recursive) rules need to be satisfied by fixing their negation-as-failure part with the original candidate. However, in Reiter's semantics [16], Γ is defined on propositional deductive closures, while in the world view semantics, it is defined on collections of interpretations.

Similar to ASP, a rule base may have zero, one or more world views. Intuitively, each world view represents an agent's view of all the possible models by given the rule base. A world view of a rule base may consist of a single or multiple interpretations. Each interpretation represents a possible world that the agent believes in this view.

Example 1. (Originated from [2]) Consider the following example of reasoning about exceptions: "Usually, mammals live on lands. However, whales are mammals but not living on lands." Formalized in RDL, this can be represented as

$$\exists \text{live.Land} \leftarrow \text{Mammal}, \text{not } \neg \exists \text{live.Land}, \quad (2)$$

$$\text{Whale} \sqsubseteq \text{Mammal} \sqcap \neg \exists \text{live.Land}, \quad (3)$$

where (2) forms the RBox, and (3) forms the TBox of this rule base.

Suppose that the ABox is

$$\{\text{Mammal}(\text{snoopy})\}. \quad (4)$$

Fix a domain $\Delta = \{d_1\}$ and an individual mapping M such that $M(\text{snoopy}) = d_1$. Then, the rule base $\Gamma_1 = \{(2), (3), (4)\}$ has the following world view with respect to Δ and M :

$$\mathcal{W}_1 = \{\{Mammal(d_1), Land(d_1), live(d_1, d_1)\}\}^3$$

² This term is borrowed from [18].

³ Note that $Mammal(d_1)$ and $Land(d_1)$ appear at the same time since there is no common-sense restriction in the rule base that these two concepts are disjoint.

To verify this, firstly it is obvious that the interpretation in $\mathcal{W}_1$ satisfies the knowledge base $\{(3), (4)\}$. Secondly, $\mathcal{W}_1$ itself satisfies condition 3 in the definition of the operator Γ . Hence, $\mathcal{W}_1 \subseteq \Gamma(\mathcal{W}_1)$. To show $\mathcal{W}_1 = \Gamma(\mathcal{W}_1)$, it suffices to prove that for any $\mathcal{W}'_1$ such that $\mathcal{W}_1 \subset \mathcal{W}'_1$, if all the interpretations in it satisfy the knowledge base $\{(3), (4)\}$, then it does not satisfy condition 3. This is indeed the case since $\exists \text{live.Land}(d_1)$ must hold according to condition 3.

It can be checked that $\mathcal{W}_1$ is the unique world view of Γ_1 w.r.t Δ and M . For instance, consider the collection

$$\{\{Whale(d_1), Mammal(d_1)\}\}.$$

Although the interpretation in this collection satisfies the TBox, and itself satisfies condition 1-3 of the operator Γ , it is not a maximal one since

$$\begin{aligned} &\{\{Whale(d_1), Mammal(d_1)\}, \\ &\{Whale(d_1), Mammal(d_1), Land(d_1)\}\} \end{aligned}$$

also satisfies conditions 1-3 of the operator Γ .

Now, suppose that the ABox is

$$\{Whale(mobydick)\}. \quad (5)$$

Fix a domain $\Delta' = \{d_2\}$ and an individual mapping M' such that $M'(mobydick) = d_2$. It can be checked that the unique extension of $\{(2), (3), (5)\}$ with respect to Δ' and M' is

$$\begin{aligned} &\{\{Whale(d_2), Mammal(d_2), Land(d_2)\}, \\ &\{Whale(d_2), Mammal(d_2)\}, \\ &\{Whale(d_2), Mammal(d_2), \text{live}(d_2, d_2)\}\}. \end{aligned}$$

Semantically, a major difference between Description Logic (DL) and Rule Description Logic (RDL) is that the semantics for former is defined in terms of interpretations, while the semantics for the latter is defined in terms of collections (of interpretations). It is necessary to consider collections instead of interpretations for RDL. Consider a simple rule base Γ , whose RBox consists of the following two rules:

$$\begin{aligned} A_1 \sqcup A_2 &\leftarrow \text{not } B \\ B &\leftarrow \text{not } A_1 \sqcup A_2. \end{aligned}$$

Given a domain $\Delta = \{d\}$, Γ has two world views with respect to Δ :

$$\begin{aligned} &\{\{A_1(d)\}, \{A_2(d)\}, \{A_1(d), A_2(d)\}\}, \\ &\{\{B(d)\}\}, \end{aligned}$$

where the first world view contains three interpretations and the second one contains only a unique interpretation. One can observe that the relationship between $\{A_1(d)\}$

and $\{A_2(d)\}$ is different from the relationship between $\{A_1(d)\}$ and $\{B(d)\}$. To conclude, $\{A_1(d)\}$ and $\{A_2(d)\}$ are complementary in the sense that these two interpretations complement each other according to the rule base, while $\{A_1(d)\}$ and $\{B(d)\}$ are contradictory in the sense that one contradicts to another according to the rule base.

The reason why collections are needed instead of interpretations can also be evidenced from propositional default logic [16], which can be regarded as a natural combination of classical logic and answer set programming in the propositional case. Notice that the extensions in default logic are propositional theories, which indeed correspond to a collection of propositional interpretations (assignments) but not merely interpretations (assignments). Lifting this idea for integrating DL and ASP, it is natural to consider world views in our task as collections rather than interpretations.

4 RDL vs DL/ASP

Since RDL claims to be an approach to enhance Description Logics with rules in Answer Set Programming, it is important to investigate the relationship between RDL and DL as well as the relationship between RDL and ASP. One important measure is so-called faithfulness [7]. That is, when restricting the syntax of RDL with DL (ASP resp.), does the semantics of RDL coincide with the semantics of DL (ASP resp.)? Another measure is expressiveness, i.e., does RDL have the full expressive power of DL (ASP resp.)?

This section answers the first question positively. We show that RDL is indeed faithful with respect to both DL and ASP. For the second question, it can be observed that RDL embraces the full expressive power of DL, but only contains a conservative fragment of ASP.

The relationship between RDL and DL is illustrated by the following theorem.

Theorem 1 (RDL vs DL). *Let Γ be a rule base with empty RBox, i.e. $\Gamma = (\mathcal{R}, \mathcal{T}, \mathcal{A})$, where $\mathcal{R} = \emptyset$. Let Δ be a domain and M an individual mapping on Δ . Then, Γ has a unique world view with respect to Δ and M , which is the collection of all models of the knowledge base $(\mathcal{T}, \mathcal{A})$ sharing the domain Δ and the individual mapping M .*

Theorem 1 shows that, fixing a domain and an individual mapping, the unique world view of a knowledge base contains and only contains all models of it. Hence, RDL is faithful with respect to DL. On the other hand, from Theorem 1, it can be observed that RDL embraces the full expressive power of DL in the sense that, given any knowledge base, its models (under the context of DL) are exactly captured by its unique world view (under the context of RDL).

To compare RDL and ASP,⁴ we need to figure out the syntactic intersection of these two approaches. A simple idea is to eliminate all the compositional connectives $\neg$, $\sqcap$, $\sqcup$, $\exists$ and $\forall$ and to eliminate TBox in RDL. In fact, this fragment of RDL only allows atomic concepts; it allows neither roles nor compounded concepts. Moreover, TBox assertions are forbidden. This fragment can be straightforwardly transformed to a fragment of first-order ASP, which only allows unary predicates and rules of the following form.

⁴ In this paper, we consider the recent generalization of the first-order answer set semantics [14, 15], which is defined over arbitrary structures rather than the Herbrand structure [13].

$$A_0(x) \leftarrow A_1(x), \dots, A_m(x), \text{not } A_{m+1}(x), \dots, \text{not } A_n(x), \quad (6)$$

where $0 \leq m \leq n$, x is a variable and A_i ($0 \leq i \leq n$) is a unary predicate or $\perp$. This fragment of ASP is indeed a very small fragment of ASP since it only allows unary predicates and all atoms in a rule must share the same variable.

Theorem 2 (RDL vs. ASP). *Let Γ be a rule base that only allows atomic concepts and has no TBox. Let Π^Γ be the answer set program obtained from Γ by translating each rule r in Γ of form (1) to a rule of form (6). Then, every world view of Π has and only has a unique interpretation, which is an answer set of Π^Γ . Conversely, an answer set of Π^Γ itself composes a world view of Γ .*

Theorem 2 shows that, by restricting to the common syntax of RDL and ASP, the world views (under the context of RDL) exactly correspond to the answer sets (under the context of ASP) in the sense that every world view contains and only contains a unique interpretation, which corresponds to an answer set. Hence, RDL is faithful with respect to ASP.

It can be observed that RDL does not fully embrace the expressive power of ASP. A simple observation is that roles (i.e. binary predicates) cannot be minimized in RDL, i.e., the head of all rules cannot be roles. Also, RDL does not allow predicates with arity more than 2.

However, we argue that rules of the form (1) in RDL are adequate enough from a Description Logic point of view. This is due to the fact that DLs only allow assertions like $C \sqsubseteq D$ in TBox. More expressive forms of assertions, e.g. $R_1 \sqsubseteq R_2$, are not allowed in DLs either. In fact, syntactically, the rule form in RDL (i.e. form (1)) is very similar to a general concept implication in standard DL, except that the negation-as-failure operator is introduced. Nevertheless, their semantic meanings are essentially different.

Also, it can be observed from Theorems 1 and 2 that, semantically, the DL and the ASP components are actually corresponding to two dimensions of RDL. More precisely, the DL component is corresponding to the interrelationships among interpretations in a particular world view, while the ASP component is corresponding to the interrelationships among different world views.

5 Decidability of RDL

Different from standard DLs, there are more typical decision problems in RDL as models are generalized to world views. For instance, the problem of checking whether a concept is satisfiable by all interpretations in some world views of an RDL rule base is different from the problem of checking whether it is satisfiable by some interpretations in all world views.

The decidability of all kinds of decision problems follows from the bounded model property of RDL.

Theorem 3 (Bounded model property). *If a rule base has a world view, then it has a world view whose domain size is bounded.*

Proof. In particular, we show that there is a world view whose domain size is less than $O(2^{2^n})$, where n is the length of the rule base. As the full proof is rather tedious, here we only outline the basic ideas due to a space limit.

We can assume that the individual set is empty and all the rules in RBox only contain atomic concepts. Suppose that $\Gamma = (\mathcal{R}, \mathcal{T}, \emptyset)$, where $\mathcal{R}$ only contains atomic concepts. Let $cl(\Gamma)$ be the smallest set of concepts that contains all sub-concepts occurring in Γ , and is closed under single negation. Given a subset $S \subseteq cl(\Gamma)$, we use i_S to denote the index of S , and Ind to denote the set of all indices (with respect to Γ).

Suppose that $\mathcal{I}$ is an interpretation whose domain is Δ . Construct a mapping $f_{\mathcal{I}}$ from Δ to Ind (with respect to $\mathcal{I}$) such that for all $d \in \Delta$, $f_{\mathcal{I}}(d) = i_S$, where $S = \{C \mid C \in cl(\Gamma), d \in C^{\mathcal{I}}\}$. By $f(\mathcal{I})$, we denote the set of indices mapped from all elements in Δ , i.e., $f(\mathcal{I}) = \{f_{\mathcal{I}}(d) \mid d \in \Delta\}$.

Let $\mathcal{W}$ be a collection with the domain Δ . Let F be a set of indices and i an index. By $\mathcal{P}(\mathcal{W})$, we denote the set $\{(F, i) \mid \exists \mathcal{I} \in \mathcal{W}, d \in \Delta \text{ such that } f(\mathcal{I}) = F \text{ and } f_{\mathcal{I}}(d) = i\}$. Then, $\mathcal{P}(\mathcal{W})$ is a finite set of pairs of the form (F, i) . By $Ind(\mathcal{W})$, we denote the power set of $\mathcal{P}(\mathcal{W})$. We construct a mapping f (with respect to $\mathcal{W}$) from Δ to $Ind(\mathcal{W})$ that for all $d \in \Delta$, $f(d) = \{(F, i) \mid \mathcal{I} \in \mathcal{W}, f(\mathcal{I}) = F, f_{\mathcal{I}}(d) = i\}$.

Let $\Delta' = \{f(d) \mid d \in \Delta\}$. Clearly, for every $d' \in \Delta'$, there exists $d \in \Delta$ such that $f(d) = d'$. Given an interpretation $\mathcal{I} \in \mathcal{W}$, construct a new interpretation $\mathcal{I}'$ (with respect to $\mathcal{W}$) as follows:

- the domain of $\mathcal{I}'$ is Δ' ;
- for each atomic concept A and each $d \in \Delta$, if $d \in A^{\mathcal{I}}$, then $f(d) \in A^{\mathcal{I}'}$;
- for each atomic role R and each two $d_1, d_2 \in \Delta$, if $(d_1, d_2) \in R^{\mathcal{I}}$, then $(f(d_1), f(d_2)) \in R^{\mathcal{I}'}$.

By induction on the structure of C , it can be proved that:

Claim. For any $d \in \Delta$, $\mathcal{I} \in \mathcal{W}$ and $C \in cl(\Gamma)$, $d \in C^{\mathcal{I}}$ iff $f(d) \in C^{\mathcal{I}'}$.

Now suppose that a collection $\mathcal{W}$ is a world view of Γ . Then, we show that the collection $\mathcal{W}' = \{\mathcal{I}' \mid \mathcal{I} \in \mathcal{W}\}$ is a world view of Γ as well. Firstly, $\mathcal{W}' \subseteq \Gamma(\mathcal{W}')$ since $\mathcal{W}'$ itself satisfies conditions 1-3 in the definition of the operator Γ according to Claim 1.

Hence, $\Gamma(\mathcal{W}') = \mathcal{W}'$. Otherwise, $\mathcal{W}' \subset \Gamma(\mathcal{W}')$. Suppose that $\mathcal{J}'$ is in $\Gamma(\mathcal{W}')$ but not in $\mathcal{W}'$. We construct a new interpretation $\mathcal{J}$ as follows:

- the domain of $\mathcal{J}$ is Δ ,
- for each atomic concept A and each $d \in \Delta$, if $f(d) \in A^{\mathcal{J}'}$, then $d \in A^{\mathcal{J}}$.
- for each atomic role R and each pair $(d_1, d_2) \in \Delta \times \Delta$, if $(f(d_1), f(d_2)) \in R^{\mathcal{J}'}$, then $(d_1, d_2) \in R^{\mathcal{J}}$.

Again, by induction, it can be proved that:

Claim. For any $d \in \Delta$, $\mathcal{J} \in \mathcal{W}^* \setminus \mathcal{W}$ and $C \in cl(\Gamma)$, $d \in C^{\mathcal{J}}$ iff $f(d) \in C^{\mathcal{J}'}$.

Let $\mathcal{W}^* = \mathcal{W} \cup \{\mathcal{J} \mid \mathcal{J}' \in \Gamma(\mathcal{W}') \setminus \mathcal{W}'\}$. Then, $\mathcal{W} \subset \mathcal{W}^*$. Clearly, $\mathcal{W}^*$ satisfies conditions 1-2 in the definition of the operator Γ with respect to $\mathcal{W}$. We now show that it satisfies condition 3 as well. Let $d \in \Delta$ and r a rule in Γ of form (1). Suppose that $d \in C_i^{\mathcal{I}}$ for all i ($1 \leq i \leq m$) and all $\mathcal{I} \in \mathcal{W}^*$, and $d \notin C_j^{\mathcal{J}}$ for all j ($m+1 \leq j \leq n$)

and all $\mathcal{J} \in \mathcal{W}$. Then, by Claims 1 and 2, $f(d) \in C_i^{\mathcal{I}'}$ for all $i(1 \leq i \leq m)$ and all $\mathcal{I}' \in \Gamma(\mathcal{W}')$, and $d \notin C_j^{\mathcal{J}'}$ for all $j(m+1 \leq j \leq n)$ and all $\mathcal{J}' \in \mathcal{W}'$. Hence, $f(d) \in C_0^{\mathcal{I}'}$ for all $\mathcal{I}' \in \Gamma(\mathcal{W}')$. Again, by Claims 1 and 2, $d \in C_0^{\mathcal{I}}$ for all $\mathcal{I} \in \mathcal{W}^*$. This shows that $\mathcal{W}^* \subseteq \Gamma(\mathcal{W})$. Therefore, $\mathcal{W} \subset \mathcal{W}^* \subseteq \Gamma(\mathcal{W})$, a contradiction.

We end up this proof by noticing that Δ' is bounded since the sizes of $cl(\Gamma)$, $P(\mathcal{W})$ and $Ind(\mathcal{W})$ are no more than $O(n)$, $O(2^n)$ and $O(2^{2^n})$, respectively.

As a consequence, the decidability of the existence checking problem follows directly from Theorem 3.

Corollary 1. *Checking whether a rule base has a world view is decidable.*

6 Discussions and Related Work

Discussions. The criteria mentioned in the introduction section are regarded to be important for the task of integrating DL and ASP. RDL indeed satisfies all these criteria. First of all, the world view semantics for RDL is a direct semantics, which is very similar to Reiter's semantics for default logic. Also, RDL is highly tight since concepts (predicates) in the DL component and those in the ASP component are not distinguished. Thirdly, RDL is faithful with respect to both DL and ASP, as shown in Section 4. Also, as proved in Section 5, the full language of RDL is decidable.

Finally, it can be observed that RDL is flexible to the Closed/Open World Assumption. The world view semantics for RDL does not force the Closed World Assumption. For instance, only from the rule base with a single concept assertion $Person(Mike)$, one cannot derive that $Mike$ is innocent. For employing Closed world Assumption for a particular concept C , one can simply add a rule $C \leftarrow \text{not } \neg C$ to the RBox. For instance, together with the Closed World Assumption on the concept $Innocent$ (i.e. $Innocent \leftarrow \text{not } \neg Innocent$), from the concept assertion $Person(Mike)$, one can actually derive that $Mike$ is innocent since the unique world view (w.r.t. a domain d and an individual mapping from $Mike$ to d) of the rule base is $\{Person(d), Innocent(d)\}$.

Related Work. As mentioned in the introduction section, the tasks of enhancing description logics with nonmonotonic formalisms and/or with rules have attracted many attentions in the literature. In this section, we briefly review some of the related work. Due to a space limit, we are not able to discuss all in very detail, and we leave this task to our full version.

Perhaps the closest work to RDL is other approaches of integrating DL and ASP. As far as we are concerned, there are five representative approaches in this category, the tight hybrid approach DL+log [8], the loose hybrid approach DL-programs [9], the embedding approach to MKNF⁺ [7], the embedding approach to QEL [10], and the grounding approach disjunctive DL-programs [11].

Rosati [8] proposed a hybrid approach, called DL+log, which is extended from DL by adding disjunctive rules of the following form

$$p_1; \dots; p_n \leftarrow p_{n+1}, \dots, p_m, \text{not } p_{m+1}, \dots, \text{not } p_l,$$

where $p_i(1 \leq i \leq l)$ is a first-order atom. Predicates in DL+log are distinguished by DL-predicates and non-DL-predicates, and the former cannot occur in the scope of not.

DL+log is a tight and faithful integration. However, it is not flexible since the Closed World Assumption cannot be applied on DL-predicates. In addition, the full language of DL+log is not decidable. In order to achieve decidability, a notion called weak-safety, similar to safety, is required.

Different from the above, Eiter et al. [9] proposed another hybrid approach, called dl-programs. The key idea is to allow the query results of DL knowledge bases as so-called *dl-atoms* in the bodies of a standard rule in ASP. The semantics is then defined based on grounding on the Herbrand base. This approach is faithful, flexible and decidable. However, it is not tight since the ASP component is building upon the DL component. This approach is significantly different from RDL. Firstly, RDL does not distinguish DL predicates and ASP predicates, which are indeed highly interactive in RDL. Secondly, the semantics of RDL is defined not only on Herbrand structures but, in general, on arbitrary ones. Last but not least, the elements in an RDL rule are regular concepts in DL but not atoms/literals.

An alternative direction of combining DLs and ASP is to define a “indirect” semantics of the integration by embedding it into another host language. One predominant work in this category is to use MKNF⁺ as the host language [7]. The rules are of the form

$$Kh_1 \vee \dots \vee Kh_l \leftarrow Kb_1, \dots, Kb_m, \text{not } b_{m+1}, \dots, \text{not } b_n,$$

where all h_i, b_j are first-order atoms. The semantics is defined via an embedding to the logic of MKNF [19]. As shown in [7], this approach is faithful, tight and flexible. However, its full language is not decidable. Again, a similar notion of safety is required to obtain decidability.

Another embedding approach is to use Quantified Equilibrium Logic (QEL) as the host language [10], which is demonstrated to be powerful enough to capture some other integrations, such as DL+log [8]. However, more detailed analysis, e.g. the decidability issue, are missing.

Lukasiewicz [11] proposed an alternative approach, called disjunctive dl-programs, by grounding on finite domains. However, as pointed out in [7], although this approach is faithful *w.r.t.* ASP, it seems not faithful *w.r.t.* the standard semantics of DL, and it seems only capable of reasoning about positive atoms. Moreover, the grounding-based semantics is defined only over Herbrand structures rather than arbitrary ones.

To conclude, comparing RDL with the above approaches for enhancing DLs with rules in ASP, there are several major issues worthy to be highlighted. Syntactically, RDL uses concepts instead of atoms/literals in the rules. Semantically, RDL employs collections instead of interpretations. Most importantly, the full language of RDL is decidable, and no strong syntactic restrictions (e.g. safety) are forced. To sum up, RDL satisfies all the desirable criteria as listed in the introduction section.

7 Conclusion

In this paper, we introduced a novel formalism, called Rule Description Logic (RDL), to enhance Description Logic with nonmonotonic recursive rules in Answer Set

Programming. RDL embraces the full expressive power of DL as well as a conservative but powerful fragment of ASP. We showed that RDL is faithful *w.r.t.* both DL and ASP (see Theorems 1 and 2). Moreover, we proved the bounded model property of RDL (see Theorem 3), which indicates that RDL, not only its subclasses but its full language, preserves the most important feature of Description Logic - decidability. To the best of our knowledge, RDL is the first approach that meets all the criteria mentioned in the introduction section for the promising task of integrating DLs and ASP.

This work is only a first step. For future work, one task is to consider alternative Description Logics, such as *SHIQ* and DL-lite. Another task worth pursuing is to further extend the syntax as well as the semantics of RDL, e.g. to disjunctive rules. Certainly, it is also important to work out the exact upper/lower bounds of the related decision problems in RDL. Last but not least, the implementation task of RDL is crucial for its applications.

References

1. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. *Journal of Automated Reasoning* 14(1), 149–180 (1995)
2. Bonatti, P.A., Lutz, C., Wolter, F.: The complexity of circumscription in DLs. *Journal of Artificial Intelligence Research* 35, 717–773 (2009)
3. de Bruijn, J., Eiter, T., Polleres, A., Tompits, H.: Embedding non-ground logic programs into autoepistemic logic for knowledge-base combination. In: *IJCAI*, pp. 304–309 (2007)
4. Horrocks, I., Patel-Schneider, P.F., Bechhofer, S., Tsarkov, D.: OWL rules: A proposal and prototype implementation. *Journal of Web Semantics* 3(1), 23–40 (2005)
5. Grosz, B.N., Horrocks, I., Volz, R., Decker, S.: Description logic programs: combining logic programs with description logic. In: *WWW*, pp. 48–57 (2003)
6. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: AL-log: Integrating datalog and description logics. *Journal of Intelligent Information Systems* 10(3), 227–252 (1998)
7. Motik, B., Rosati, R.: Reconciling description logics and rules. *Journal of the ACM* 57(5) (2010)
8. Rosati, R.: DL+log: Tight integration of description logics and disjunctive datalog. In: *KR*, pp. 68–78 (2006)
9. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. *Artificial Intelligence* 172(12–13), 1495–1539 (2008)
10. de Bruijn, J., Pearce, D., Polleres, A., Valverde, A.: Quantified Equilibrium Logic and Hybrid Rules. In: Marchiori, M., Pan, J.Z., de Marie, C.S. (eds.) *RR 2007. LNCS*, vol. 4524, pp. 58–72. Springer, Heidelberg (2007)
11. Lukasiewicz, T.: A novel combination of answer set programming with description logics for the semantic web. *IEEE Transactions on Knowledge and Data Engineering* 22(11), 1577–1592 (2010)
12. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: the making of a web ontology language. *J. Web Sem.* 1(1), 7–26 (2003)
13. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: *ICLP 1988*, pp. 1070–1080 (1988)
14. Lin, F., Zhou, Y.: From answer set logic programming to circumscription via logic of GK. *Artificial Intelligence* 175(1), 264–277 (2011)

15. Ferraris, P., Lee, J., Lifschitz, V.: Stable models and circumscription. *Artificial Intelligence* 175(1), 236–263 (2011)
16. Reiter, R.: A logic for default reasoning. *Artificial Intelligence* 13(1-2), 81–132 (1980)
17. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press (2003)
18. Gelfond, M.: Logic programming and reasoning with incomplete information. *Annals of Mathematics and Artificial Intelligence* 12(1-2), 89–116 (1994)
19. Lifschitz, V.: Minimal belief and negation as failure. *Artificial Intelligence* 70(1-2), 53–72 (1994)