Rough Description Logic Programs

Tingting Zou
College of Computer Science and Technology, Jilin University, ChangChun, China
Email: zoutt10@mails.jlu.edu.cn

Shuai Lv
College of Computer Science and Technology, Jilin University, ChangChun, China
Email: lus@jlu.edu.cn

Lei Liu*
College of Computer Science and Technology, Jilin University, ChangChun, China
Email: liulei@jlu.edu.cn

***Abstract*—The Semantic Web is an extension of the current World Wide Web, and aims to help computers to understand and process web information automatically. In recent years, the integration ontologies and rules has become a central topic in the Semantic Web. Therefore, many researchers have focused their study on investigating the combination of answer set programming with description logics for the semantic web. However, these can not deal with uncertainty and inexactness. To address this problem, we propose tightly coupled rough description logic programs (or simply rough dl-programs) under the answer set semantics, which can model uncertain, inexact information, and can deal with non monotonic reasoning at the same time. To our knowledge, this is the first such approach. First of all, we define the syntax and semantics of rough dl-program *KB*=(*L*,*P*), which is a tight integration of disjunctive logic program under the answer set semantics, rough set theory and rough description logic. Then, we present some reasoning problems of rough dl-program. Finally, we show some semantic properties of rough dl-program under the answer set semantics.**

***Index Terms*—Description logics, Rough description logics, Description logic programs, Answer set semantics, Semantic web**

I. INTRODUCTION

The Semantic Web is an extension of the current World Wide Web, and aims to help computers to understand and process web information automatically[1,2] . The process of the Semantic Web can be described as follows: firstly a machine-readable meaning is added to web pages; secondly share terms in web resources can be precisely represented by ontologie；finally knowledge representation technologies are utilized for automated reasoning from Web resources[3,4].

At present, the highest layer of the semantic web, which has reached a sufficient maturity, is the ontology layer in form of the OWL Web Ontology Language [5].

The next and ongoing step aims at sophisticated representation and reasoning capabilities of the Rules, Logic, and Proof layers of the Semantic Web [6,7].

As we have seen, the integration ontologies and rules has become a central topic in the Semantic Web. In fact, standard ontology language is based on Description Logics(DLs), and the existing proposals for a rule language for use in the Semantic Web originate from Logic Programmings. Recently, significant research efforts have focused on integration description logics and logic programmings. Eiter et al proposed description logic programs, which combined disjunctive logic programmings under answer set semantics with description logics in loose integration [8,9]. Subsequently, Lukasiewicz presented a new method for description logic programs under the answer set semantics, which was a tight integration of disjunctive logic programs under the answer set semantics with description logics[10,11]. Moreover, Lukasiewicz introduced vagueness into description logic program, and proposed description logic program that combined fuzzy description logics and fuzzy disjunctive logic programs [12,13]. Subsequently, he presented tightly coupled fuzzy description logic programs under the answer set semantic, which extended tightly disjunctive description logic program by fuzzy vagueness in both the description logic and the logic program component [14,15]. Furthermore, Lukasiewicz proposed the notion of probabilistic description logic programs, and described the syntax and semantics of probabilistic description logic programs [16,17]. Moreover, Andrea Calì present tightly coupled probabilistic dl-programs under the answer set semantics, which were a tight integration of disjunctive logic programs under the answer set semantics and Bayesian probabilities [18,19]. Furthermore, Lukasiewicz and Straccia presented probabilistic fuzzy description logic programs, which combined fuzzy description logics, fuzzy logic programs, and probabilistic uncertainty in a uniform framework for the semantic web [20].

Moreover, there are some works to explore formalisms for dealing with uncertainty and inexactness. In particular,

*Corresponding author.

doi:10.4304/jcp.7.11.2719-2725

rough DL, which combined DL with rough set theory[21,22], can represent and reason on uncertain or inexact information. Schlobach et al introduced lower approximation and upper approximation concepts for the first time, and then advanced a rough DL RDL. However, only one approximation concept cannot accurately express the concept [23,24]. Afterwards, Jiang et al defined an approximation concept that consists of lower and upper approximations, and then proposed rough DL RDLAC, moreover, they introduced approximation concept satisfiability and approximation concepts rough subsumption reasoning problems [25]. Furthermore, we propose a rough description logic with concrete domain RSHOIQ(D), which combines DL RSHOIQ with rough set theory and concrete domain [26].

In this paper, we continue this line of research. We propose tightly coupled rough description logic programs (or simply rough dl-program) under the answer set semantics, which are a tight integration of disjunctive logic programs under the answer set semantics, rough set theory and rough description logics. To our knowledge, this is the first such approach. Firstly, we define the syntax and semantics of rough dl-program *KB*=(*L*,*P*), which consists of a rough description logic knowledge base L and a rough disjunctive logic programs P. More concretely, the concepts and roles from L can be regarded as unary resp. binary predicate of rough rules in P. Furthermore, we present some reasoning problems, definitely satisfiable and possibly satisfiable of an approximate atom, brave consequence and cautious consequence of KB. Finally, we show some semantic properties of rough dl-program under the answer set semantics. In a word, rough dl-program can model uncertain, inexact information, and can deal with non monotonic reasoning at the same time.

The rest of this paper is organized as follows. In section II, we recall rough set theory and rough description logics. Section Ⅲ defines rough dl-programs under the answer set semantics. In section Ⅳ, we present some semantic properties of rough dl-program. Section Ⅴ summarizes our main results.

II. Preliminaries

In this section, we first recall some work related to rough set theory. Then we introduce the syntax and semantics of rough description logic *RSHOIQ*(D).

A. Rough Set Theory

Pawlak advanced rough set theory for the first time, and provided formal description of rough set theory. Let U be a universe which is a finite and non-empty set, and let $R^{\sim}$ be an equivalent relation over U. Then an approximation space is defined by $apr=(U,R^{\sim})$. For any set $A\subseteq U$, it may not represented in a crisp way, but it can be characterized by using a pair of lower and upper approximations

$$\underline{apr}(A)=\bigcup_{[x]_{R^{\sim}}\subseteq A}[x]_{R^{\sim}}=\{x\,|\,[x]_{R^{\sim}}\subseteq A\},$$

$$\overline{apr}(A)=\bigcup_{[x]_{R^{\sim}}\cap A\neq\varnothing}[x]_{R^{\sim}}=\{x\,|\,[x]_{R^{\sim}}\cap A\neq\varnothing\}.$$

Where $[x]_{R^{\sim}}=\{y\,|\,(x,y)\in R^{\sim}\}$ is the equivalent class containing x. The rough set is commonly denoted as a tuple $A=<\underline{A},\overline{A}>$, where $\underline{A}$ is called lower approximations and $\overline{A}$ is called upper approximations with respect to A.

Let A, B be any subsets of U, then the lower approximation and upper approximation have the following properties:

1) $\underline{A}=\overline{\neg A}$, $\overline{A}=\underline{\neg A}$.

2) $\underline{A\cap B}=\underline{A}\cap\underline{B}$, $\overline{A\cap B}\subseteq\overline{A}\cap\overline{B}$.

3) $\underline{A\cup B}\supseteq\underline{A}\cup\underline{B}$, $\overline{A\cup B}=\overline{A}\cup\overline{B}$

B. Rough Description Logic

Let **A**, $\mathbf{R_A}$, $\mathbf{R_D}$, $\mathbf{I_A}$, $\mathbf{I_D}$, and **D** be pairwise disjoint sets of atomic concepts, abstract role names, concrete role names, abstract individuals, concrete individuals and concrete datatypes. The set $\mathbf{R_A}\cup\{R^{-}\,|\,R\in\mathbf{R_A}\}\cup\mathbf{R_D}$ is called R*SHOIQ*(D) roles, where R^{-} is the inverse role of R. The set of R*SHOIQ*(D) concepts are defined inductively according to the following abstract syntax:

$$C,D\,\text{訾}\,A|\;\;|\mathrm{T}|\;\;C|C\,\text{篁}D|C\quad D|\{o_1,\mathrm{K}\,,o_n\}|$$

$$\$R.C\,|\,"R.C\,|\,\text{常}n\,S.C\,|\quad n\,S.C\,|$$

$$\{u_1,\mathrm{K}\,,u_n\}|T.\mathbf{d}\,|\,"T.\mathbf{d}\,|\quad n\,T.\mathbf{d}\,|$$

$$\$?\;\;n\,T.\mathbf{d}\,|\underline{C}|\overline{C}|\quad\underline{C},\overline{C}>$$

where A denotes atomic concept, C and D denotes concepts, $o_1,\ldots,o_n\in\mathbf{I_A}$, $R,S\in\mathbf{R_A}$, S is a simple role, $n\in\mathbf{N}$, $u_1,\ldots,u_n\in\mathbf{I_D}$, $T\in\mathbf{R_D}$, $E\in\mathbf{D}$, d is a concrete predicate.

For any R*SHOIQ*(D) concept C, the approximate concept of C is defined by the pair $<\underline{C},\overline{C}>$, where $\underline{C}$ is called lower approximations of C , and $\overline{C}$ is called upper approximations of C. Furthermore, If concept C is a crisp concept, then $\underline{C}=C=\overline{C}$. In addition, the approximation concept of C can be denoted by <C,C>.

Therefore, a rough interpretation is defined by a 5-tuple $I=(\Delta^I,\Delta^D,R^{\sim},\bullet^I,\bullet^D)$, where the abstract domain Δ^I denotes a nonempty set of objects, the datatype domain Δ^D denotes the interpretation domain of all datatypes (disjoint from Δ^I) with data values, $R^{\sim}$ is an equivalence relation over Δ^I, and two interpretation functions $\bullet^I$ and $\bullet^D$ that assign each atomic concept

$A \in \mathbf{A}$ to a subset $A^I \subseteq \Delta^I$, each abstract role name $R \in \mathbf{R_A}$ to a relation $R^I \subseteq \Delta^I \times \Delta^I$, each concrete role name $T \in \mathbf{R_D}$ to a relation $T^I \subseteq \Delta^I \times \Delta^D$, each concrete datatype $E \in \mathbf{D}$ to a subset $E^D \subseteq \Delta^D$, each abstract individual $o \in \mathbf{I_A}$ to an element $o^I \in \Delta^I$, each concrete individual $u \in \mathbf{I_D}$ to an element $u^I \in \Delta^D$. The mapping $\bullet^I$ and $\bullet^D$ can be extended to all roles and concepts as follows:

$\bot^I = \varnothing$;
$\mathbf{T}^I = \Delta^I$;
$(\neg C)^I = \Delta^I \setminus C^I$;
$(C \text{ ó } D)^I = C^I \cap D^I$;
$(C \text{ ò } D)^I = C^I \cup D^I$;
$\{o_1, \ldots, o_n\}^I = \{o_1^I, \ldots, o_n^I\}$;
$(\exists R.C)^I = \{x \in \Delta^I \mid \exists y \in \Delta^I, < x, y > \in R^I \wedge y \in C^I\}$;
$(\forall R.C)^I = \{x \in \Delta^I \mid \forall y \in \Delta^I, < x, y > \in R^I \to y \in C^I\}$;
$(? \ \ n\, S.C)^I \quad \{x \mid \#\{y \mid < x, y > \text{掟} S^I \mathit{and}\ y \quad C^I\} \quad n\}$;
$(? \ \ n\, S.C)^I \quad \{x \mid \#\{y \mid < x, y > \text{掟} S^I \mathit{and}\ y \quad C^I\} \quad n\}$;
$\{u_1, \ldots, u_n\}^I = \{u_1^D, \ldots, u_n^D\}$;
$(\exists T.E)^I = \{x \in \Delta^I \mid \exists t \in \Delta^D, < x, t > \in T^I \wedge t \in E^D\}$;
$(\forall T.E)^I = \{x \in \Delta^I \mid \forall t \in \Delta^D, < x, t > \in T^I \to t \in E^D\}$;
$(? \ \ n\, T.\mathbf{d})^I \quad \{x \mid \#\{t \mid < x, t > \text{掟} T^I \mathit{and}\ t \quad \mathbf{d^D}\} \quad n\}$;
$(? \ \ n\, T.\mathbf{d})^I \quad \{x \mid \#\{t \mid < x, t > \text{掟} T^I \mathit{and}\ t \quad \mathbf{d^D}\} \quad n\}$;

$(\underline{C})^I = \{x \in \Delta^I \mid \forall y \in \Delta^I, < x, y > \in R^{\sim} \to y \in C^I\}$;

$(\overline{C})^I = \{x \in \Delta^I \mid \exists y \in \Delta^I, < x, y > \in R^{\sim} \wedge y \in C^I\}$.

$(< \underline{C}, \overline{C} >)^I = < (\underline{C})^I, (\overline{C})^I >$.

A rough TBox of R*SHOIQ*(D) is a finite set of rough concept axioms. Let C and D be R*SHOIQ*(D) concepts, $AC = < \underline{C}, \overline{C} >$ and $AD = < \underline{D}, \overline{D} >$ be approximation concepts of C and D respectively. Rough concept axioms consist of rough concept inclusion axioms of the form $AC \text{ ô } AD$ or $< \underline{C}, \overline{C} > \text{ô} < \underline{D}, \overline{D} >$ and rough equivalence axioms of the form $AC \equiv AD$ or $< \underline{C}, \overline{C} > \equiv < \underline{D}, \overline{D} >$. A rough interpretation I satisfies rough concept inclusion axioms $AC \text{ ô } AD$ iff $(\underline{C})^I \subseteq (\underline{D})^I$ and $(\overline{C})^I \subseteq (\overline{D})^I$. A rough interpretation I satisfies rough equivalence axioms $AC \equiv AD$ iff $(\underline{C})^I = (\underline{D})^I$ and $(\overline{C})^I = (\overline{D})^I$. Finally, a rough interpretation I is called a model of R*SHOIQ*(D) TBox Γ if it satisfies all rough concept axioms in Γ.

A rough RBox is a finite set of rough role axioms. Let $R, S \in \mathbf{R_A}$ and $T, U \in \mathbf{R_D}$, then the rough role axioms consist of rough transitive role axioms of the form $\mathbf{Trans}(R)$, and rough role inclusion axioms of the form $R \text{ ô } S$ or $T \text{ ô } U$. A rough interpretation I satisfies rough transitive role axioms $\mathbf{Trans}(R)$ if $\forall x, y, z \in \Delta^I$, $\{< x, y >, < y, z >\} \in R^I \to < x, z > \in R^I$. A rough interpretation I satisfies $R \text{ ô } S$ if $R^I \subseteq S^I$, it satisfies $T \text{ ô } U$ if $T^I \subseteq U^I$. Finally, a rough interpretation I is called a model of R*SHOIQ*(D) RBox $\Re$ if it satisfies all rough role axioms in $\Re$.

A rough ABox is a finite set of rough assertions. Let C be R*SHOIQ*(D) concepts, $R \in \mathbf{R_A}$, $T \in \mathbf{R_D}$, $a, b \in \mathbf{I_A}$, $u \in \mathbf{I_D}$, $E \in \mathbf{D}$, then rough assertions of the form $a : C$ are called rough concept assertions, rough assertions of the form $(a, b) : R$ or $(a, u) : T$ are called rough role assertions, and rough assertions of the form $a \not\approx b$ (or $a \approx b$) are called rough inequality(or equality) assertions. For a rough interpretation I, I satisfies $a : C$ iff $a^I \in C^I$; I satisfies $(a, b) : R$ iff $(a^I, b^I) \in R^I$; I satisfies $(a, u) : T$ iff $(a^I, u^D) \in T^I$; I satisfies $a \not\approx b$ iff $a^I \neq b^I$; I satisfies $a \approx b$ iff $a^I = b^I$. Finally, a rough interpretation I is called a model of R*SHOIQ*(D) ABox Λ if it satisfies all rough role axioms in Λ.

A R*SHOIQ*(D) knowledge base is Σ a triple $< \Gamma, \Re, \Lambda >$, where Γ denotes rough TBox, $\Re$ denotes rough RBox and Λ denotes rough ABox. A rough interpretation I is called a model of Σ if it satisfies all rough axioms in Σ.

Ⅲ. Rough Description Logic Programs Under The Answer Set Semantics

In this section, we propose rough description logic programs. Firstly, we define the syntax and semantics of rough description logic programs. Finally, we present some reasoning problems for rough description logic programs.

A. Sntax

Let Φ be a function-free first-order vocabulary with nonempty finite sets of constant symbols F_C and predicate symbols F_P, and the sets **C**

Let Φ be a function-free first-order vocabulary with nonempty finite sets of constant symbols F_C and predicate symbols F_P, and the sets $\mathbf{C_A}$, $\mathbf{R_A}$, $\mathbf{R_D}$, $\mathbf{I_A}$, $\mathbf{I_D}$, and **D** is defined as sectionⅡ. Suppose $\mathrm{F}_c \text{ 腿} \mathbf{I_A} \quad \mathbf{I_D}$, thus every ground atom made from $\mathbf{C_A}$, $\mathbf{R_A}$, $\mathbf{R_D}$, and F_C can be interpreted in the description logic component.

Let X be a set of variables. A term is either a variable from X or a constant symbol from F_C. Let Q denotes unary predicate symbol, Con(Q) denotes concept set expressed by Q and $R^{\sim}$ denotes equivalence relation on Con(Q). We define approximate predicate symbols in the following.

Definition 3.1 (approximate predicate symbols). For unary predicate symbol Q, approximate predicate symbol is of the form $Q = (\underline{Q}, \overline{Q})$, where $\underline{Q}$ is lower approximate predicate symbol and $\overline{Q}$ is upper approximate predicate symbol. Moreover,

$$\mathrm{Con}(\underline{Q}) = \{x \in \mathrm{Con}(Q) \mid R^{\sim}(x) \subseteq \mathrm{Con}(Q)\},$$

$$\mathrm{Con}(\overline{Q}) = \{x \in \mathrm{Con}(Q) \mid R^{\sim}(x) \cap \mathrm{Con}(Q) \neq \varnothing\}.$$

Obviously, we can obtain the following properties.

- $\mathrm{Con}(\underline{Q}) \subseteq \mathrm{Con}(Q) \subseteq \mathrm{Con}(\overline{Q})$;
- $\forall x \in \mathrm{Con}(\underline{Q})$, $\underline{Q}(x) = 1$, otherwise $\underline{Q}(x) = 0$;
- $\forall x \in \mathrm{Con}(\overline{Q})$, $\overline{Q}(x) = 1$, otherwise $\overline{Q}(x) = 0$;

Definition 3.2. For any approximate predicate symbol $Q = (\underline{Q}, \overline{Q})$ and x, unary predicate $Q(x)$ is definitely true iff lower approximate predicate $\underline{Q}(x) = 1$; unary predicate $Q(x)$ is possibly true iff upper approximate predicate $\overline{Q}(x) = 1$.

Definition 3.3. An approximate atom is of the form $\alpha = (\underline{\alpha}, \overline{\alpha})$, where $\underline{\alpha}$ is of the form $\underline{Q}(t)$ and $\overline{\alpha}$ is of the form $\overline{Q}(t)$, $Q = (\underline{Q}, \overline{Q})$ is an approximate predicate symbol from Φ , t is term.

An atom is either an approximate atom or of the form $h(t_1, \mathrm{K}\ , t_n)$, where h is a predicate symbol of arity $n \ge 0$ form F_P , and $t_1, \ldots, t_n$ are terms. We use M to denote a set of atoms. A literal l is an atom α or a negated atom not α . If an atom α is not an approximate atom, then $\underline{\alpha} = \alpha$ and $\overline{\alpha} = \alpha$.

Definition 3.4. A disjunctive rough rule (or simply rough rule) r is of the form

$$\underline{\alpha}_1 \vee \cdots \vee \underline{\alpha}_k \leftarrow \underline{\beta}_1 \wedge \cdots \wedge \underline{\beta}_l \wedge not \underline{\beta}_{l+1} \wedge \cdots \wedge not\ \underline{\beta}_n \quad (3.1)$$

$$\overline{\alpha}_1 \vee \cdots \vee \overline{\alpha}_k \leftarrow \overline{\beta}_1 \wedge \cdots \wedge \overline{\beta}_l \wedge not\ \overline{\beta}_{l+1} \wedge \cdots \wedge not\ \overline{\beta}_n \quad (3.2)$$

where
$k \ge 1, n \ge l \ge 0, \{a_1, \mathrm{L}\ , a_k, b_1, \mathrm{L}\ , b_l, b_{l+1}, \mathrm{L}\ , b_n\}$ Í M .

Let r be a disjunctive rough rule, then the set $\{\alpha_1, \ldots, \alpha_k\}$ is called the head of r, i.e. $H(r) = \{\alpha_1, \ldots, \alpha_k\}$, and the set $\{\beta_1, \ldots, \beta_l, \beta_{l+1}, \ldots, \beta_n\}$ is called the body of r, i.e., $B(r) = B^+(r) \cup B^-(r)$, $B^+(r) = \{\beta_1, \ldots, \beta_l\}$, $B^-(r) = \{\beta_{l+1}, \ldots, \beta_n\}$.

Definition 3.5. A rough disjunctive program (or simply rough program) P is a finite set of disjunctive rough rules of the form (3.1) and (3.2). Moreover, P is normal rough program if and only if $k = 1$ for all rough rules in P; P is a positive rough program if and only if $n = l$ for all rough rules in P.

Definition 3.6. A rough description logic program (for short, rough dl-program) *KB=(L,P)* includes a rough description logic knowledge base *L* and a rough program *P*. KB is a normal rough dl-program if and only if *P* is normal rough program. KB is a positive rough dl-program if and only if *P* is positive rough program.

B. Semantics

Now, we define the answer set semantics of rough dl-program based on Herbrand interpretation. More formally, a term is ground iff it includes only constant symbols from F_C . An atom α is ground iff all terms of α are ground.

Definition 3.7. A ground instance of a rough rule r the form (3.1) and (3.2) is defined as follows:

$$\underline{\alpha'}_1 \vee \cdots \vee \underline{\alpha'}_k \leftarrow \underline{\beta'}_1 \wedge \cdots \wedge \underline{\beta'}_l \wedge not \underline{\beta'}_{l+1} \wedge \cdots \wedge not\ \underline{\beta'}_n\ ,$$

$$\overline{\alpha'}_1 \vee \cdots \vee \overline{\alpha'}_k \leftarrow \overline{\beta'}_1 \wedge \cdots \wedge \overline{\beta'}_l \wedge not\ \overline{\beta'}_{l+1} \wedge \cdots \wedge not\ \overline{\beta'}_n ,$$

where, $\underline{\alpha'}_1, \cdots, \underline{\alpha'}_k, \underline{\beta'}_1, \cdots, \underline{\beta'}_l, \underline{\beta'}_{l+1}, \cdots, \underline{\beta'}_n$, $\overline{\alpha'}_1, \cdots, \overline{\alpha'}_k, \overline{\beta'}_1, \cdots, \overline{\beta'}_l, \overline{\beta'}_{l+1}, \cdots, \overline{\beta'}_n$ are obtained by substituting constant symbol from F_C for every variable appearing in $\underline{\alpha}_1, \cdots, \underline{\alpha}_k, \underline{\beta}_1, \cdots, \underline{\beta}_l, \underline{\beta}_{l+1}, \cdots, \underline{\beta}_n$, $\overline{\alpha}_1, \vee \cdots, \overline{\alpha}_k, \overline{\beta}_1, \cdots, \overline{\beta}_l, \overline{\beta}_{l+1}, \cdots, \overline{\beta}_n$ respectively. A ground program of a rough program P is a set of all ground instances of rough rules in P. Let Ground(P) to denote all ground programs of a rough program P.

Let Φ be a function-free first-order vocabulary with nonempty finite sets of constant symbols F_C and predicate symbols F_P . Then the Herbrand base relative to Φ , written as HB_F , denotes the set of all ground atoms that can be made from the predicate symbols from F_P , and the constant symbols from F_C

Definition 3.8. Let *KB*=(*L*,*P*) be a rough dl-program, Φ be a function-free first-order vocabulary, HB_F be a Herbrand base relative to Φ . Then a rough interpretation I relative to KB is a subset of HB_F .

Definition 3.9. Let *KB*=(*L*,*P*) be a rough dl-program, I be a rough interpretation relative to KB. Then a rough interpretation I is a model of a ground atom α , denoted $I \models a$, if and only if $\alpha \in I$. A rough interpretation I is a model of a ground rough rule r of the form (3.1) and (3.2), denoted $I \models r$, if and only if

(1) $I \models \underline{\alpha}$ for some a Î $H(r)$, if $I \models \underline{\beta}_i$, $\beta_i \in B^+(r)$, $i = 1, 2, ..., l$, and $I \not\models \underline{\beta}_j$, $\beta_j \in B^-(r)$, $j = l+1, l+2, ..., n$;

(2) $I \models \overline{\alpha}$ for some a Î $H(r)$, if $I \models \overline{\beta}_i$, $\beta_i \in B^+(r)$, $i = 1, 2, ..., l$, and $I \not\models \overline{\beta}_j$, $\beta_j \in B^-(r)$, $j = l+1, l+2, ..., n$;

Definition 3.10. Let *KB*=(*L*,*P*) be a rough dl-program, I be a rough interpretation relative to KB. Then a rough interpretation I is a model of a rough program P, denoted by $I \models P$, if and only if $I \models r$ for all $r \in \mathrm{Ground}(P)$.

Definition 3.11. Let *KB*=(*L*,*P*) be a rough dl-program, I be a rough interpretation relative to KB. Then a rough interpretation I is a model of a rough description logic knowledge base *L*, denoted $I \models L$, if and only if $L \cup I \cup \{\neg\alpha \mid \alpha \in HB_{\Phi} - I\}$ is satisfiable.

Definition 3.12. Let *KB*=(*L*,*P*) be a rough dl-program, I be a rough interpretation relative to KB. Then a rough interpretation I is a model of KB, denoted $I \models KB$, if and only if $I \models L$ and $I \models P$. *KB* is satisfiable iff it has a model.

Definition 3.13. Let *KB*=(*L*,*P*) be a rough dl-program, M be a set of atoms. Then a rough reduction for P is defined as follows:

$$\begin{aligned} P^M = \{ & \underline{\alpha}_1 \vee \cdots \vee \underline{\alpha}_k \leftarrow \underline{\beta}_1 \wedge \cdots \wedge \underline{\beta}_l, \\ & \overline{\alpha}_1 \vee \cdots \vee \overline{\alpha}_k \leftarrow \overline{\beta}_1 \wedge \cdots \wedge \overline{\beta}_l \mid \\ & \underline{\alpha}_1 \vee \cdots \vee \underline{\alpha}_k \leftarrow \underline{\beta}_1 \wedge \cdots \wedge \underline{\beta}_l \wedge \\ & not\, \underline{\beta}_{l+1} \wedge \cdots \wedge not\ \underline{\beta}_n \in P, \qquad . \\ & \overline{\alpha}_1 \vee \cdots \vee \overline{\alpha}_k \leftarrow \overline{\beta}_1 \wedge \cdots \wedge \overline{\beta}_l \wedge \\ & not\ \overline{\beta}_{l+1} \wedge \cdots \wedge not\ \overline{\beta}_n \in P, \\ & B^{-}(r) \cap M = \varnothing \} \end{aligned}$$

Moreover, a rough reduction for KB is $KB^M = (L, P^M)$.

Definition 3.14. Let *KB*=(*L*,*P*) be a rough dl-program, I be a rough interpretation relative to KB. Then I is an answer set of KB if and only if I is a minimal model of $KB^I = (L, P^I)$. *KB* is consistent iff *KB* has an answer set.

C. Reasoning Problems

We define some reasoning problems for rough dl-programs.

Definition 3.15. Let *KB*=(*L*,*P*) be a rough dl-program, and $\alpha = (\underline{\alpha}, \overline{\alpha})$ be an approximate atom. Then α with respect to KB is definitely satisfiable if and only if there exists a model I of KB, such that $\underline{\alpha} \in I$, otherwise α is called definitely unsatisfiable.

Definition 3.16. Let *KB*=(*L*,*P*) be a rough dl-program, and $\alpha = (\underline{\alpha}, \overline{\alpha})$ be an approximate atom. Then α with respect to KB is possibly satisfiable if and only if there exists a model I of KB, such that $\overline{\alpha} \in I$, otherwise α is called possibly unsatisfiable.

Definition 3.17. Let *KB*=(*L*,*P*) be a rough dl-program, $\alpha \in HB_{\Phi}$ be a ground atom. Then α is a called brave consequence of KB, denoted by $KB \models_b \alpha$, if and only if, there exists a answer set I of KB such that $I \models a$.

Definition 3.18. Let *KB*=(*L*,*P*) be a rough dl-program, $\alpha \in HB_{\Phi}$ be a ground atom. Then α is a called cautious consequence of KB, denoted by $KB \models_c \alpha$, if and only if, for every answer set I of KB such that $I \models a$.

Ⅳ. Semantic Properties

In this section, we present some semantic properties of rough dl-program under anwer set semantics. Firstly, we show the relation between answer set and minimal model of a rough dl-program.

Theorem 4.1. Let *KB*=(*L*,*P*) be a rough dl-program, I be any answer set of KB. Then I is a minimal model of KB.
Proof. According to Definition 3.14, I is a minimal model of $KB^I = (L, P^I)$. So, $I \models L$ and $I \models P^I$. Thus, $I \models L$ and $I \models r$ for all $r \in \text{Ground}(P^I)$. This is equivalent to $I \models r$ for all $r \in \text{Ground}(P)$. So, $I \models L$ and $I \models P$. Therefore, I is a model of KB.

Now, we show that I is also a minimal model of KB. Suppose that there exists a model J of KB such that $J \subset I$. Then $J \models L$ and $J \models r$ for all $r \in \text{Ground}(P)$. This is equivalent to $J \models r$ for all $r \in \text{Ground}(P^I)$. Thus, J is also a model of KB^I . However, this is a contradiction that I is a minimal model of KB^I . As a result, I is a minimal model of KB.

Theorem 4.2. Let *KB*=(*L*,*P*) be a positive rough dl-program. I is a answer set of KB if and only if I is a minimal model of KB.
Proof. According to Theorem 4.1, if I is a answer set of KB, then I is a minimal model of KB. Now we need to prove that if I is a minimal model of KB, then I is a answer set of KB.

Let I be a minimal model of KB. Then $I \models L$, and $I \models r$ for all $r \in \text{Ground}(P)$. This is equivalent to $I \models r$ for all $r \in \text{Ground}(P^I)$. So, I is a model of $KB^I = (L, P^I)$. We now show that I is also a minimal model of KB^I . Suppose that there exists a model J of KB^I such that $J \subset I$. Then $J \models L$, and $J \models r$ for all $r \in \text{Ground}(P^I)$. This is equivalent to $J \models L$ and $J \models r$ for all $r \in \text{Ground}(P)$. Thus, J is also a model of KB. However, this is a contradiction that I is a minimal model of KB. So, I is also a minimal model of KB^I . Therefore, I is a answer set of KB.

In summary, I is a answer set of KB iff I is a minimal model of KB.

Now, we show that the answer set semantics of a rough dl-program $KB = (\varnothing, P)$ is in accord with the answer set semantics of P.

Theorem 4.3. Let *KB*=(*L*,*P*) be a rough dl-program, and $L = \varnothing$. Then I is an answer set of KB if and only if I is an answer set of P.

Proof. It is known that I is an answer set of KB iff I is a minimal model of $KB^I = (L, P^I)$. So, I is a model of KB^I, iff $I \models L$ and $I \models r$ for all $r \in \text{Ground}(P^I)$. Because $L = \varnothing$, then I is a model of KB^I, iff $I \models r$ for all $r \in \text{Ground}(P^I)$ iff I is a model of P^I. Thus, I is a minimal model of KB^I iff I is a minimal model of P^I. Therefore, I is an answer set of KB iff I is an answer set of P.

Theorem 4.4. Let *KB*=(*L*,*P*) be a positive rough dl-program, and a be a ground atom of HB_F. Then a is definitely satisfied by all answer sets of KB if and only if $\underline{\alpha}$ is true in all first-order models of $L \cup \text{Ground}(P)$.

Proof. Because *KB* is a positive rough dl-program, then the set of all answer set of KB is equivalent to the set of all minimal model of KB. Moreover, a is definitely satisfied by all minimal model of KB if and only if a is definitely satisfied by all model of KB. So, a is definitely satisfied by all answer set of KB iff a is definitely satisfied by all model of KB. Now, we need to prove that a is definitely satisfied by all model of KB iff $\underline{\alpha}$ is true in all first-order models of $L \cup \text{Ground}(P)$.

($\Rightarrow$) Suppose that for a is definitely satisfied by all model of KB. Let J be any first-order models of $L \cup \text{Ground}(P)$. Now, we define a rough interpretation $I \subseteq \text{HB}_\Phi$ such that $\alpha \in I$ iff $J \models \alpha$. Let $L' = L \cup I \cup \{\neg\alpha \mid \alpha \in HB_\Phi - I\}$, then I is a model of L. Because J is a first-order models of $\text{Ground}(P)$, then $J \models r$ for $r \in \text{Ground}(P)$. Thus, $I \models r$ for $r \in \text{Ground}(P)$. So, I is also a model of P. Therefore, I is a model of KB. According to the known condition, a is definitely satisfied by all model of KB, so $\underline{\alpha} \in I$. Thus, $\underline{\alpha}$ is true in J. Therefore, $\underline{\alpha}$ is true in all first-order models of $L \cup \text{Ground}(P)$.

($\Leftarrow$) Suppose that $\underline{\alpha}$ is true in all first-order models of $L \cup \text{Ground}(P)$. Let $I \subseteq \text{HB}_\Phi$ be any model of KB. So, $I \models L$. Then $L' = L \cup I \cup \{\neg\alpha \mid \alpha \in HB_\Phi - I\}$ is satisfiable. Let J be a first-order model of L'. Then J is also a first-order models of L. Moreover, Because $I \models r$ for $r \in \text{Ground}(P)$, then J is also a model of Ground(P). Thus, J is a first-order model of $L \cup \text{Ground}(P)$. According to known condition, $\underline{\alpha}$ is true in J. Thus, a is definitely satisfied by I. Therefore, a is definitely satisfied by all models of KB

In summary, a is definitely satisfied by all answer sets of KB if and only if $\underline{\alpha}$ is true in all first-order models of $L \cup \text{Ground}(P)$.

Theorem 4.5. Let *KB*=(*L*,*P*) be a positive rough dl-program, and a be a ground atom of HB_F. Then a is possibly satisfied by all answer sets of KB if and only if $\overline{\alpha}$ is true in all first-order models of $L \cup \text{Ground}(P)$.

Proof. Because *KB* is a positive rough dl-program, then the set of all answer set of KB is equivalent to the set of all minimal model of KB. Moreover, a is possibly satisfied by all minimal model of KB if and only if a is possibly satisfied by all model of KB. So, a is possibly satisfied by all answer set of KB iff a is possibly satisfied by all model of KB. Now, we need to prove that a is possibly y satisfied by all model of KB iff $\overline{\alpha}$ is true in all first-order models of $L \cup \text{Ground}(P)$.

($\Rightarrow$) Suppose that for a is possibly satisfied by all model of KB. Let J be any first-order models of $L \cup \text{Ground}(P)$. Now, we define a rough interpretation $I \subseteq \text{HB}_\Phi$ such that $\alpha \in I$ iff $J \models \alpha$. Let $L' = L \cup I \cup \{\neg\alpha \mid \alpha \in HB_\Phi - I\}$, then I is a model of L. Because J is a first-order models of $\text{Ground}(P)$, then $J \models r$ for $r \in \text{Ground}(P)$. Thus, $I \models r$ for $r \in \text{Ground}(P)$. So, I is also a model of P. Therefore, I is a model of KB. According to the known condition, a is possibly satisfied by all model of KB, so $\overline{\alpha} \in I$. Thus, $\overline{\alpha}$ is true in J. Therefore, $\overline{\alpha}$ is true in all first-order models of $L \cup \text{Ground}(P)$.

($\Leftarrow$) Suppose that $\overline{\alpha}$ is true in all first-order models of $L \cup \text{Ground}(P)$. Let $I \subseteq \text{HB}_\Phi$ be any model of KB. So, $I \models L$. Then $L' = L \cup I \cup \{\neg\alpha \mid \alpha \in HB_\Phi - I\}$ is satisfiable. Let J be a first-order model of L'. Then J is also a first-order models of L. Moreover, Because $I \models r$ for $r \in \text{Ground}(P)$, then J is also a model of Ground(P). Thus, J is a first-order model of $L \cup \text{Ground}(P)$. According to known condition, $\overline{\alpha}$ is true in J. Thus, a is possibly satisfied by I. Therefore, a is possibly satisfied by all models of KB

In summary, a is possibly satisfied by all answer sets of KB if and only if $\overline{\alpha}$ is true in all first-order models of $L \cup \text{Ground}(P)$.

Theorem 4.6. Let *KB*=(*L*,*P*) be a positive rough dl-program, and a be a ground atom of HB_F, and $P = \varnothing$. Then a is definitely satisfied by all answer sets of KB if and only if $\underline{\alpha}$ is true in all first-order models of L.

Proof. It is easy to prove according to Theorem 4.4.

Theorem 4.7. Let *KB*=(*L*,*P*) be a positive rough dl-program, and a be a ground atom of HB_F, and $P = \varnothing$. Then a is possibly satisfied by all answer sets of KB if and only if $\overline{\alpha}$ is true in all first-order models of L.

Proof. It is easy to prove according to Theorem 4.5.

The above theorems show that the answer set semantics of a rough dl-program is also a faithful extension of the semantics of a rough description logic knowledge base.

Ⅴ. CONCLUSION

We have proposed tightly coupled rough description logic programs (rough dl-programs) under the answer set semantics, which generalize the tightly coupled description logic programs by rough set theory in both the logic program and the description logic component. In this paper, we first provide the syntax and semantics of rough dl-program, then we present some reasoning problems of rough dl-program, finally we show that the answer set of rough dl-program has a close relation with the minimal model, and the rough dl-program faithfully extends both rough disjunctive logic program and rough description logic. In a word, rough dl-program can well represent and reason a great deal of real-word problems.

An interesting topic of future research is to implement of the presented approach. Another interesting issue is to extend rough dl-programs by a new semantics.

ACKNOWLEDGMENT

This work is partially supported by the National Natural Science Foundation of China under Grant Nos. 60873044, 61070084; the China Postdoctoral Science Foundation under Grant No. 2011M500612; the Opening Fund of Top Key Discipline of Computer Software and Theory in Zhejiang Provincial Colleges at Zhejiang Normal University of China under Grant No. ZSDZZZZXK11; the Fundamental Research Funds for the Central Universities under Grant Nos. 201103124, 201103133; the Science and Technology Development Program of Jilin Province of China under Grant No. 20100186; and the "985 Project" of Jilin University of China.

REFERENCES

[1] T. Berners-Lee, *Weaving the Web*, Harper, San Francisco, CA, 1999.

[2] T. Berners-Lee, J. Hendler, O. Lassila, “The semantic Web”, *Scientific American*, vol. 284, no. 5, pp. 34-43, 2001.

[3] D. Fensel, W. Wshlster, H. Lieberman, J. Hendler (Eds.), *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, MIT Press, 2002.

[4] Haibin Wang, Yan-Qing Zhang and Rajshekhar Sunderraman, “Extensible soft semantic web services agent”, *Soft Computing*, vol. 10, no. 11, pp. 1021-1029, 2006.

[5] I. Horrocks, P.F Patel-Schneider, F. van Harmelen, “From SHIQ and RDF to OWL: The making of a Web ontology language”, *J. Web Sem*., vol. 1, no. 1, pp. 7-26, 2003.

[6] H. Boley, S. Tabet, G. Wagner, “Design rationale of RuleML: A amrkup language for semantic web rules”, in *Procedings First Semantic Web Working Symposium (SWWS-2001)*, Stanford University, pp. 381-401, 2001.

[7] I. Horrocks, P.F. Patel-Schneider, “A proposal for an OWL rules languages”, in: *Proceedings WWW-2004*, ACM Press, pp. 723-731, 2004.

[8] T. Eiter, T. Lukasiewicz, R. Schindlauer, H. Tompits, “Combining Answer Set Programming with Description Logics for the Semantic Web”, in: *Prodeedings of the 9th International Conference on Principles of knowledge Representation and Reasoing(KR’04)*, pp. 141-151, 2004.

[9] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, “Tompits H. Combining answer set programming with description logics for the Semantic Web”, *Artificial Intelligence*, vol. 172, no. 12-13, pp. 1495-1539, 2008.

[10] T. Lukasiewicz, “A Novel Combination of Answer Set Programming with Description Logics for the Semantic Web”, in *ESWC*, pp. 384-398, 2007.

[11] T. Lukasiewicz, “A Novel Combination of Answer Set Programming with Description Logics for the Semantic Web”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 11, pp. 1577-1592, 2010.

[12] T. Lukasiewicz, “Fuzzy Description Logic Programs under the Answer Set Semantics for the Semantic Web”, in *RuleML*, pp. 89-96, 2006.

[13] T. Lukasiewicz, “Fuzzy Description Logic Programs under the Answer Set Semantics for the Semantic Web”, *Fundamenta Informaticae*, vol. 82, no. 3, pp. 289-310, 2008.

[14] T. Lukasiewicz, U.Straccia, “Tightly Integrated Fuzzy Description Logic Programs Under the Answer Set Semantics for the Semantic Web”, in *RR*, pp. 289-298, 2007.

[15] T. Lukasiewicz, U.Straccia, “Tightly Coupled Fuzzy Description Logic Programs under the Answer Set Semantics for the Semantic Web”, *International Journal on Semantic Web and Information System*, vol. 4, no.3, pp. 68-89, 2008.

[16] T. Lukasiewicz, “Probabilistic Description Logic Programs”, in *ECSQARU*, pp. 737-749, 2005.

[17] T. Lukasiewicz, “Probabilistic description logic programs”, *International Journal of Approximate Reasoning*, vol. 45, no. 2, pp. 288-307, 2007.

[18] A. Calì, T. Lukasiewicz, “Tightly Integrated Probabilistic Description Logic Programs for the Semantic Web”, in *ICLP*, pp. 428-429, 2007.

[19] A. Calì, T. Lukasiewicz, L. Predoiu, H. Stuckenschmidt, “Tightly Coupled Probabilistic Description Logic Programs for the Semantic Web”, *Journal on Data Semantics*, vol. 12, pp. 95-130, 2009.

[20] T. Lukasiewicz, U. Straccia, “Description logic programs under probabilistic uncertainty and fuzzy vagueness”, *International Journal of Approximate Reasoning*, vol. 50, no. 6, pp. 837-853, 2009.

[21] Z. Pawlak and A. Skowron, “Rough sets: some extensions”, *Information Sciences*, vol.177, no.1, pp.29-40, 2007.

[22] C.Y. Huang, G.H. Tzeng, C.C. Chan and H.C. Wu, “Semiconductor Market Fluctuation Indicators and Rule Derivations Using the Rough Set Theory”, *International Journal of Innovative Computing, Information and Control*, vol.5, no.6, pp. 1485-1503, 2009.

[23] S. Schlobach, M. Klein and L. Peelen, “Description logics with approximate definitions: precise modeing of vague concepts”, in *Proc. 20th Internat. Joint Conf. on Artificial Inteligence*, AAAI Press, New York, pp. 557-562, 2007.

[24] M. C.A. Klein, P. Mika and S. Schlobach, “Rough Description Logics for Modeling Uncertainty in Instance Unification”, in *Proceedings of the Third ISWC Workshop on Uncertainty Reasoning for the Semantic Web*, Busan, Korea, 2007.

[25] Y. Jiang, J. Wang, S. Tang and B. Xiao, “Reasoning with rough description logic: an approximate concepts approach”, *Information Sciences*, vol.179, pp. 600-612, 2009.

[26] Lei Liu, Tingting Zou, Shuai Lv, “Rough description logics with concrete domain”, *ICIC Express Letters, Part B: Applications*, vol. 2, no. 5, pp. 1143-1149, 2011.