ICIC Express Letters
Part B: Applications
Volume **7**, Number **1**, January **2016**

ICIC International ©2016 ISSN 2185-2766
pp. 117–122

POSSIBILISTIC DESCRIPTION LOGIC PROGRAMS

Tingting Zou* and Ansheng Deng

Information Science and Technology College
Dalian Maritime University
No. 1, Linghai Road, Dalian 116026, P. R. China
*Corresponding author: zoutt@dlmu.edu.cn

Received July 2015; accepted October 2015

Abstract. *Description logics (DLs) are a class of knowledge representation languages. However, DLs cannot well model a great deal of real-world problems because of the restriction of represented formalism. To address this problem, we further extend DLs such that they can deal with uncertain, incomplete, inconsistent information and non monotonic reasoning at the same time. We propose tightly coupled possibilistic description logic programs (or simply possibilistic dl-program) under the possibilistic answer set semantics, which are a tight integration of disjunctive logic programs, possibilistic logics and possibilistic description logics. To our knowledge, this is the first such approach. First of all, we define the syntax and semantics of possibilistic dl-program. Then, we present some possibilistic inferences. Moreover, we show some semantic properties of possibilistic dl-program. Finally, we provide two algorithms to compute the consistency degree.*

Keywords: Description logics, Possibilistic logics, Possibilistic description logics, Answer set semantic, Disjucntive logic programs

1. **Introduction.** Description logics are a class of knowledge representation languages, and can model an application domain of interest by a structured and formally well-understood way [1,2]. Unfortunately, a great deal of real-world problems cannot be modelled by DLs, because crisp DLs cannot model uncertain, incomplete and inconsistent information, and cannot deal with non monotonic reasoning. Possibilistic logic can represent and reason on uncertain, incomplete and inconsistent knowledge [3]. Moreover, answer set programming is an efficient method to deal with non monotonic reasoning [4].

On the one hand, many researchers have focused their study on possibilistic description logics in recent years [5]. Hollunder introduced uncertainty in terminological logics using possibilistic logics, which is the first method for possibilistic description logics [6]. Subsequently, Dubois et al. proposed a new possibilistic description logic [7]. Moreover, Qi et al. proposed possibilistic description logics, which are an extension of description logics with possibilistic logics [8]. Moreover, Lesot et al. proposed a new algorithm to compute the inconsistency degree of a possibilistic DL knowledge base [9].

On the other hand, the integration DLs and answer set programming has become a central topic in recent years. Eiter et al. proposed description logic programs [10]. Subsequently, Shen et al. proposed well-justified FLP answer set for description logic programs [11]. Motik and Rosati presented a hybrid formalism of MKNF knowledge bases [12]. Moreover, Lukasiewicz proposed description logic program that combined fuzzy description logics and fuzzy disjunctive logic programs [13]. Zou et al. proposed rough description logic programs under the answer set semantics, which can model vagueness information, and can deal with non monotonic reasoning at the same time [14].

However, the above possibilistic description logics and description logic programs cannot deal with uncertain, incomplete, inconsistent information and non monotonic reasoning at the same time. Therefore, this paper aims to further extend description logic programs such that they can model uncertain, incomplete and inconsistent information.

In this paper, we propose tightly coupled possibilistic description logic programs under the possibilistic answer set semantics. To our knowledge, this is the first such approach. In this paper, we firstly define the syntax, semantics and possibilistic inferences. Then, we show some semantic properties. Finally, we give two algorithms to compute the consistency degree.

2. **Possibilistic Description Logic Programs.** Let Φ be a function-free first-order vocabulary with nonempty finite sets of constant symbols Φ_C and predicate symbols Φ_P, $\Phi_c \subseteq \mathbf{I_A} \cup \mathbf{I_D}$. Let X be a set of variables. A term is either a variable from X or a constant symbol from Φ_C. An atom is an expression of the form $h(t_1, \ldots, t_n)$, where h is a predicate symbol of arity $n \geq 0$ from Φ_P, and $t_1, \ldots, t_n$ are terms. We use M to denote a set of atoms. A literal l is an atom a or a negated atom not a, where $a \in M$.

Definition 2.1. *A possibilistic atom is $p = (a, \delta)$, $a \in M$, $\delta \in (0, 1]$. Moreover, we define the classical projection * and the necessity degree of p as follows: $p^* = a$, $p^d = \delta$.*

Definition 2.2. *A possibilistic disjunctive rule (or simply possibilistic rule) r is*

$$r = (a_1 \vee \cdots \vee a_k \leftarrow b_1 \wedge \cdots \wedge b_l \wedge not\ b_{l+1} \wedge \cdots \wedge not\ b_n, \delta), \quad (1)$$

*where $k \geq 0$, $l \geq 0$, $k + l > 0$, $\{a_1, \cdots, a_k, b_1, \cdots, b_l, b_{l+1}, \cdots, b_n\} \subseteq M$, and $\delta \in (0, 1]$. Moreover, the classical projection * and the necessity degree of r is defined as follows:*

$$r^* = a_1 \vee \cdots \vee a_k \leftarrow b_1 \wedge \cdots \wedge b_l \wedge not\ b_{l+1} \wedge \cdots \wedge not\ b_n, \quad r^d = \delta. \quad (2)$$

Let PM be a set of possibilistic atoms in which every atom a occurs at most one time in PM, that is to say, $\forall a \in M$, $|\{(a, \delta) \in PM\}| \leq 1$, $\delta \in (0, 1]$. In general, we can write r as a form of $(A \leftarrow B^+ \wedge not\ B^-, \delta)$, where $A = a_1 \vee \cdots \vee a_k$, $B^+ = b_1 \wedge \cdots \wedge b_l$, and $B^- = b_{l+1} \wedge \cdots \wedge b_n$.

Definition 2.3. *A possibilistic disjunctive logic program (possibilistic program) P is a finite set of possibilistic rules. Let $P^* = \{r^* | r \in P\}$, and then P is normal possibilistic program iff $k = 1$ for all rules in P^*; P is a positive possibilistic program iff $n = l$ for all rules in P^*.*

Definition 2.4. *A possibilistic description logic program (for short, possibilistic dl-program) $KB = (L, P)$ includes a possibilistic description logic knowledge base L and a possibilistic program P. Moreover, KB is a normal possibilistic dl-program iff P is a normal possibilistic program. KB is a positive possibilistic dl-program iff P is a positive possibilistic program.*

Now, we define the semantics of possibilistic dl-program under the possibilistic answer set semantics. A term is ground iff it includes only constant symbols from Φ_C. An atom $h(t_1, \ldots, t_n)$ is ground iff all terms $t_1, \ldots, t_n$ are ground. We use $PossG(P)$ to denote all possibilistic ground programs of a possibilistic program P.

Definition 2.5. *A possibilistic atom $p = (a, \delta)$ is a possibilistic ground atom iff the atom a is ground. r is possibilistic ground rule iff all atoms of r are possibilistic ground atoms.*

Definition 2.6. *A possibilistic ground instance of r is a possibilistic ground rule $r' = \left(a_1' \vee \cdots \vee a_k' \leftarrow b_1' \wedge \cdots \wedge b_l' \wedge not\ b_{l+1}' \wedge \cdots \wedge not\ b_n', \delta\right)$, where, $a_1', \cdots, a_k'$, $b_1', \cdots, b_l'$, b_{l+1}', $\cdots, b_n'$ are obtained by substituting constant symbol from Φ_C for every variable appearing in $a_1, \cdots, a_k$, $b_1, \cdots, b_l$, $b_{l+1}, \cdots, b_n$ respectively. A possibilistic ground program of a possibilistic program P is a set of all possibilistic ground instances of possibilistic rules in P.*

Definition 2.7. *The possibilistic Herbrand base relative to Φ, written as PHB_Φ, is a set of all possibilistic ground atom $\{p_1, p_2, \ldots, p_u\}$, for every possibilistic ground atom $p_i = (h_i(t_{1i}, \ldots, t_{ni}), \delta_i)$, $i = 1, 2, \ldots, u$, h_i is a predicate symbol of arity $n \geq 0$ from Φ_P, and $t_{1i}, \ldots, t_{ni}$ are constant symbols from Φ_C, and $\delta \in (0, 1]$.*

Definition 2.8. *A possibilistic interpretation* $I = \{p_1', p_2', \ldots, p_v'\}$ *relative to possibilistic dl-program* $KB = (L, P)$ *is a subset of* PHB_Φ.

Definition 2.9. *Let* I *be a possibilistic interpretation relative to* $KB = (L, P)$*. Then* I *is a possibilistic model of a possibilistic ground atom* $p = (a, \delta)$*, denoted* $I \mid\!- (a, \delta)$*, if and only if* $(a, \delta) \in I$ *or there exists a possibilistic ground atom* $(a, \delta') \in I$ *such that* $\delta \leq \delta'$*.* I *is a possibilistic model of a possibilistic ground rule* r*, denoted by* $I \mid\!- r$*, if and only if,* $I \mid\!- (a, \min\{\delta, \beta_1, \ldots, \beta_l\})$ *for some* $a \in H(r^*)$*, if* $I \mid\!- (b_i, \beta_i)$*,* $b_i \in B^+(r^*)$*,* $\beta_i \in (0, 1]$*,* $i = 1, 2, \ldots, l$*, and* $I \nvdash (b_j, \beta_j)$*,* $b_j \in B^-(r^*)$*,* $\beta_j \in (0, 1]$*,* $j = l+1, l+2, \ldots, n$*.* I *is a possibilistic model of a possibilistic program* P*, denoted by* $I \vDash P$*, if and only if* $I \mid\!- r$ *for all* $r \in PossG(P)$*.*

Definition 2.10. *Let* I *be a possibilistic interpretation relative to* $KB = (L, P)$*. Then* I *is a possibilistic model of* L*, denoted* $I \vDash L$*, if and only if there exists a possibilistic distribution* π *for knowledge base* $L \cup I$ *such that* $\pi \vDash L \cup I$*.*

Definition 2.11. *Let* I *be a possibilistic interpretation relative to* $KB = (L, P)$*. Then* I *is a possibilistic model of KB, denoted* $I \vDash KB$*, if and only if* $I \vDash L$ *and* $I \vDash P$*.*

There may be many possibilistic models for $KB = (L, P)$. Let I_1, I_2 be two possibilistic models of KB, then $I_1 \Cap I_2 = \{(a, \min\{\delta, \delta'\}) | (a, \delta) \in I_1, (a, \delta') \in I_2\}$, and $I_1 \Subset I_2$ if and only if $I_1^* \subset I_2^*$, or $I_1^* = I_2^*$ and for any $a \in I_1^*$, if $(a, \delta) \in I_1$ and $(a, \delta') \in I_2$, then $\delta \leq \delta'$.

Definition 2.12. *Let* I *be a possibilistic model of KB. Then* I *is a least possibilistic model of KB, if and only if there does not exist a possibilistic model* I' *of KB, such that* $I' \Subset I$*.*

Definition 2.13. *A possibilistic reduction for* P *is defined as follows:*

$$P_{PM^*} = \{(A \leftarrow B^+,\ \delta) | r = (A \leftarrow B^+ \wedge not\ B^-,\ \delta) \in P, B^- \cap PM^* = \emptyset, B^+ \subseteq PM^*\}. \tag{3}$$

Moreover, a possibilistic reduction for $KB = (L, P)$ *is* $KB_{PM^*} = (L, P_{PM^*})$*.*

Definition 2.14. *Let* I *be a possibilistic interpretation relative to* $KB = (L, P)$*. Then* I *is a possibilistic answer set of* KB *if and only if* I *is a least possibilistic model of* $KB_{I^*} = (L, P_{I^*})$*.*

Let $KB^* = (L^*, P^*)$ be the classical dl-program associated with $KB = (L, P)$, where $L^* = \{\phi | (\phi, \beta) \in L\}$, $P^* = \{r^* | r \in P\}$. KB is consistent if and only if KB^* is consistent.

Definition 2.15. *The strict* α*-cut of* L *is* $L_{>\alpha} = \{(\phi, \beta) \in L | \beta > \alpha\}$*, and the strict* α*-cut of* P *is* $P_{>\alpha} = \{r \in P | n(r) > \alpha\}$*. The strict* α*-cut of* KB *is* $KB_{>\alpha} = (L_{>\alpha}, P_{>\alpha})$*,* $\alpha \in (0, 1]$*.*

Definition 2.16. *The consistency degree of a possibilistic dl-program* KB *is*

$$Consdegree(KB) = \begin{cases} 0, & \text{if } KB^* \text{ is consistent} \\ \min\{\alpha | KB_{>\alpha} \text{ is consistent}\}, & \text{otherwise} \end{cases} \tag{4}$$

Definition 2.17. *A possibilistic atom* (a, δ) *is called credutions possibilistic consequence of a possibilistic positive dl-program KB, denoted by* $KB \vDash_c (a, \delta)$*, if and only if, there exists a possibilistic answer set* I *of KB such that* $I \mid\!- (a, \delta)$*.*

Definition 2.18. *A possibilistic atom* (a, δ) *is called skeptical possibilistic consequence of a possibilistic positive dl-program KB, denoted by* $KB \vDash_s (a, \delta)$*, if and only if, for all possibilistic answer set* $I_1, I_2, \ldots, I_m$ *of KB such that* $I_1 \Cap I_2 \Cap \ldots \Cap I_m \mid\!- (a, \delta)$*.*

3. **Semantic Properties.** In this section, we present some semantic properties.

Theorem 3.1. *Let $KB = (L, P)$ be a possibilistic dl-program, and I be any possibilistic answer set of KB. Then I is a least possibilistic model of KB.*

Proof: I is a least possibilistic model of $KB_{I^*} = (L, P_{I^*})$. So, $I \vDash L$ and $I \vDash P_{I^*}$. Thus, $I \vDash L$ and $I \,|\!-\, r$ for all $r \in PossG(P_{I^*})$. So $I \,|\!-\, r$ for all $r \in PossG(P)$, $I \vDash L$ and $I \vDash P$. Thus, I is a possibilistic model of KB. Suppose that there exists a possibilistic model J of KB such that $J \Subset I$ and $J \neq I$. Then $J \vDash L$ and $I \,|\!-\, r$ for all $r \in PossG(P)$. So $J \,|\!-\, r$ for all $r \in PossG(P_{I^*})$. Thus, J is also a possibilistic model of KB_{I^*}. However, this is a contradiction that I is a least possibilistic model of KB_{I^*}. As a result, I is a least possibilistic model of KB.

Theorem 3.2. *Let $KB = (L, P)$ be a positive possibilistic dl-program. I is a possibilistic answer set of KB if and only if I is a least possibilistic model of KB.*

Theorem 3.3. *Let $KB = (L, P)$ be a possibilistic dl-program, and $L = \emptyset$. Then I is a possibilistic answer set of KB if and only if I is a possibilistic answer set of P.*

Proof: It is known that I is a possibilistic answer set of KB iff I is a least possibilistic model of $KB_{I^*} = (L, P_{I^*})$. So, I is a possibilistic model of KB_{I^*}, iff $I \vDash L$, and $I \,|\!-\, r$ for $r \in PossG(P_{I^*})$. Because $L = \emptyset$, then I is a possibilistic model of KB_{I^*}, iff $I \,|\!-\, r$ for $r \in PossG(P_{I^*})$ iff I is a possibilistic model of P_{I^*}. Thus, I is a least possibilistic model of KB_{I^*} iff I is a least possibilistic model of P_{I^*}.

Theorem 3.4. *Let (a, δ) be a possibilistic ground atom of PHB_Φ. Then for all possibilistic answer set I of a positive possibilistic dl-program $KB = (L, P)$, $I \,|\!-\, (a, \delta)$ if and only if for all possibilistic distribution π such that $\pi \vDash L \cup PossG(P), \pi \vDash (a, \delta)$.*

Proof: It is known that the set of all possibilistic answer set of KB is equivalent to the set of all least possibilistic model of KB. Thus, for $(a, \delta) \in PHB_\Phi$, for all least possibilistic model of KB, $I \,|\!-\, (a, \delta)$ iff for all possibilistic model of KB, $J \,|\!-\, (a, \delta)$. ($\Rightarrow$) Suppose that for all possibilistic model J of KB, $J \,|\!-\, (a, \delta)$. Let π be any possibilistic distribution such that $\pi \vDash L \cup PossG(P)$. Now, we define a possibilistic interpretation $I' \subseteq PHB_\Phi$ such that $I' \,|\!-\, (b, \beta)$ iff $\pi \vDash (b, \beta)$. Let $L' = L \cup I$, then I' is a possibilistic model of L. Because $\pi \vDash PossG(P)$, then $\pi \vDash r$ for $r \in PossG(P)$. Thus, $I' \,|\!-\, r$ for $r \in PossG(P)$. So, I' is also a possibilistic model of P. Therefore, I' is a possibilistic model of KB. According to $I' \,|\!-\, (a, \delta)$. So, $\pi \vDash (a, \delta)$. ($\Leftarrow$) Suppose that for all possibilistic distribution π such that $\pi \vDash L \cup PossG(P)$, $\pi \vDash (a, \delta)$. Let $I \subseteq PHB_\Phi$ be any possibilistic model of KB. So, $I \vDash L$, and $I \,|\!-\, r$ for all $r \in PossG(P)$. Thus, there exists a possibilistic π' such that $\pi' \vDash L \cup I$. So, $\pi' \vDash L$, $\pi' \vDash I$. Moreover, $\pi' \vDash r$ for all $r \in PossG(P)$, thus $\pi' \vDash PossG(P)$. So, $\pi' \vDash L \cup PossG(P)$. According to $\pi' \vDash (a, \delta)$, thus, $(a, \delta) \in I$, or there exists a possibilistic ground atom $(a, \delta') \in I$, such that $\delta' \leq \delta$. So, $I \,|\!-\, (a, \delta)$.

Theorem 3.5. *Let $KB = (L, P)$ be a positive possibilistic dl-program, and (a, δ) be a possibilistic ground atom of PHB_Φ, and $P = \emptyset$. Then for all possibilistic answer set I of KB, $I \,|\!-\, (a, \delta)$ iff for all possibilistic distribution π such that $\pi \vDash L$, $\pi \vDash (a, \delta)$.*

4. **Computing Consistency Degree.** In this section, we first propose an algorithm to compute the consistency degree of a possibilistic dl-program in Figure 1. Let $KB = (L, P)$ be a possibilistic dl-program, where $L = \{(\phi_i, \beta_i) | \beta_i \in (0, 1],\ i = 1, 2, \ldots, n\}$, $P = \{r_1, r_2, \ldots, r_m\}$, $\alpha > Consdegree(KB)$. Then, $L_{>\alpha}$ is consistent, but it does not ensure that $P_{>\alpha}$ is necessarily consistent. Therefore, we improve the Algorithm 1, and present a new Algorithm 2 in Figure 2.

Theorem 4.1. *Let $KB = (L, P)$ be a possibilistic dl-program, Con is the result computed by Algorithm 1. Then Con is the consistency degree of KB.*

```
Algorithm 1
Input: A possibilistic dl-program KB=(L,P), where L = {(φ_i, β_i) | β_i ∈ (0,1], i = 1,2,...,n} ,
P = {r_1, r_2, ..., r_m} .
Output: The consistency degree of KB.
Begin
        If L is consistent and P is consistent, then return 0.0;
        Q = Asc(β_1,...,β_n, n(r_1),...,n(r_m));     u=1;     v=|Q|;
        While (u < v) do
                Con=Q(u);
                If L_{>Con} is consistent and P_{>Con} is consistent, then return Con;
                Else     u=u+1;
        Con= Q(u);     return Con;
End
```

FIGURE 1. Algorithm 1

```
Algorithm 2
Input: A possibilistic dl-program KB=(L,P), where L = {(φ_i, β_i) | β_i ∈ (0,1], i = 1,2,...,n} ,
P = {r_1, r_2, ..., r_m} .
Output: The consistency degree of KB.
Begin
        If L is consistent and P is consistent, then return 0.0;
        Q = Asc(β_1,...,β_n);  u=1;     v=|Q|;  Con=0;
         While (u < v) do
                t = ⌈(u+v)/2⌉;                α = Q(t);
                If L_{>α} is consistent, then
                        If u-v=1, then     Con=Q(v), skip;   Else     v=t;
                Else     u=t;
         If (Con==0), then    Con= Q(v);
        Q' = Asc(β_1,...,β_n);  u=1;     v =|Q'|;
         While (Q'(u) < Con), do     u=u+1;
         While (u < v) do
                Con=Q(u);
                If P_{>Con} is consistent, then          return Con;   Else          u=u+1;
         Con= Q(u);   return Con;
End
```

FIGURE 2. Algorithm 2

Proof: According to Algorithm 1, Con is returned at two cases, in the while loop and outside of the while loop. With respect to the first case, $L_{>Con}$ is consistent and $P_{>Con}$ is consistent. It is easy to know that Con is just the consistency degree of KB, i.e., $Con = Consdegree(KB)$. Considering the second case, $u = v$, $Con = Q(u) = Q(v)$, $L_{>Con} = \emptyset$, $P_{>Con} = \emptyset$. So, $L_{>Con}$ is consistent and $P_{>Con}$ is consistent, and thus Con is the consistency degree of KB. In a word, Con is the consistency degree of KB.

Theorem 4.2. *Let $KB = (L, P)$ be a possibilistic dl-program, Con is the result computed by Algorithm 2. Then Con is the consistency degree of KB.*

Proof: The proof is similar to proof of Theorem 4.1.

5. **Conclusions.** We have proposed tightly coupled possibilistic description logic programs under the possibilistic answer set semantics. Firstly, we provide the syntax, semantics and possibilistic inferences of possibilistic dl-program. Secondly, we show that the possibilistic answer set of possibilistic dl-program has a close relation with the least model, and the possibilistic dl-program faithfully extends both possibilistic disjunctive

logic program and possibilistic description logic. Finally, we present two algorithms computing the consistency degree of a possibilistic dl-program, and prove the correctness of the algorithms. In a word, possibilistic dl-program can well represent and reason a great deal of real-word problems. Two interesting topics of future research are implementation of the presented approach and extension of possibilistic dl-programs by a new semantics.

Acknowledgment. This work is partially supported by Liaoning Province Natural Science Fund Project of China (No.2015020034), and the National Natural Science Foundation of China (No. 61272171), and the Fundamental Research Funds for the Central Universities (Nos. 3132015044, 3132013325). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] F. Baader, D. Calvanese et al., *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, 2007.
[2] F. Simancik, B. Motik and I. Horrocks, Consequence-based and fixed-parameter tractable reasoning in description logics, *Artificial Intelligence*, vol.215, pp.29-77, 2014.
[3] D. Dubois and H. Prade, Possibilistic logic: A retrospective and prospective view, *Fuzzy Sets and Systems*, vol.144, pp.3-23, 2004.
[4] W. Faber, G. Pfeifer and N. Leone, Semantics and complexity of recursive aggregates in answer set programming, *Articial Intelligence*, vol.175, no.1, pp.278-298, 2011.
[5] T. Lukasiewicz and U. Straccia, Managing uncertainty and vagueness in description logics for the semantic web, *Web Semantics: Science, Services and Agents on the World Wide Web*, vol.6, pp.291-308, 2008.
[6] B. Hollunder, An alternative proof method for possibilistic logic and its application to terminological logics, *International Journal of Approximate Reasoning*, vol.12, no.2, pp.85-109, 1995.
[7] D. Dubois, J. Mengin and H. Prade, Possibilistic uncertainty and fuzzy features in description logic: A preliminary discussion, *Fuzzy Logic and the Semantic Web, Capturing Intelligence*, pp.101-114, 2006.
[8] G. Qi, Q. Ji, J. Z. Pan and J. Du, Extending description logics with uncertainty reasoning in possibilistic logic, *International Journal of Intelligent Systems*, vol.26, pp.353-381, 2011.
[9] M. J. Lesot, O. Couchariere, B. Bouchon-Meunier and J. L. Rogier, Inconsistency degree computation for possibilistic description logic: An extension of the tableau algorithm, *Annual Meeting of the North American on Fuzzy Information Processing Society*, pp.1-6, 2008.
[10] T. Eiter, G. Ianni, T. Lukasiewicz et al., Combining answer set programming with description logics for the semantic web, *Artificial Intelligence*, vol.172, nos.12-13, pp.1495-1539, 2008.
[11] Y.-D. Shen, K. Wang, T. Eiter et al., FLP answer set semantics without circular justifications for general logic programs, *Articial Intelligence*, vol.213, pp.1-41, 2014.
[12] B. Motik and R. Rosati, Reconciling description logics and rules, *Journal of ACM*, vol.57, pp.93-154, 2010.
[13] T. Lukasiewicz, Fuzzy description logic programs under the answer set semantics for the semantic web, *Fundam. Inform.*, vol.82, no.3, pp.289-310, 2008.
[14] T. Zou, S. Lv and L. Liu, Rough description logic programs, *Journal of Computers*, vol.7, no.11, pp.2719-2725, 2012.