THE SEMANTICS AND QUERY EVALUATION METHOD FOR ALC-DESCRIPTION DATALOG PROGRAM

Truong Cong Tuan

University of Sciences, Hue University

Email: tctuan@husc.edu.vn

Received: 30/11/2022; Received in revised form: 20/12/2022; Accepted: 26/12/2022

ABSTRACT

The Datalog program is an important class of deductive database and has been extended in many different directions. This paper focuses on extending the Datalog program by combining Datalog program with description logic ALC. We suggest the semantics and propose a query evaluation method for this program class.

Keywords: Deductive database, Datalog Program, description logic.

1. INTRODUCTION

In the past decades, Datalog program and description logic are two areas of research that have gained a lot of interest and have had many useful applications in different fields.

Description logic is a family of formal languages, which are effectively used to describe ontology knowledge and which play an important role in building the Semantic Web, with the aim of increasing the ability to link between resources and ability to understand information in computers [3]. Description logic is well suited for representing structured knowledge in terms of classes and relationships between classes, but not for representing complex queries.

Besides the research on descriptive logic, the studies of deductive database have also obtained many important achievements in both theory and practical application. Deductive database is an extension of relational database and it has the ability to process inference as well as incomplete information. Deductive database has useful applications in fields such as artificial intelligence, decision support systems, financial analysis, etc.

In deductive database, Datalog is an important class of programs. It is a declarative language where programs are created neither from statements nor from functions but mainly based on a set of predicates, suitable for representing knowledge by rules of inference. However, the Datalog program uses an EDB data model similar to the relational data model, which has many limitations in representing knowledge.

Currently, a research direction to integrate description logic with Datalog program has been of great interest. The advantage of this combination is that it has taken advantage of the research results in both the above fields to support the processing of deductive reasonings in the Semantic Web as well as to provide an effective tool in representing knowledge. In this paper, we suggest the semantics and propose a query evaluation method for this new program class.

The rest of the paper is organized as follows.

In Section 2, 3 we summarize the description logic ALC and the Datalog program, Section 4 presents the syntax and semantics of the ALC description Datalog program, and Section 5 focuses on the query evaluation method for this class. Finally, the conclusion is given in Section 6.

2. DESCRIPTION LOGIC ALC

Descriptive logic is a family of formal languages and is well suited for representing and inferring knowledge in a particular domain of interest. In description logic, the domain of interest is described in terms of individuals, concepts, roles, and constructs. Each individual represents an object, each concept represents a set of individuals with common properties, and each role represents a binary relationship between objects or between individuals and data values. Complex concepts are built from concepts, role names, and individual names by combining constructors. A descriptive logic system allows the description of related concepts and implicit knowledge. This implicit knowledge can be inferred from already represented knowledge through inference services or inference sets. Descriptive logic is built on three basic components including the set of individuals, the set of atomic concepts (which can be understood as classes) and the set of atomic roles.

In the next section, description logic ALC will be introduced [3].

Description Logic ALC is a descriptive logic language with simple syntac rules, ALC allows the use of constructors: $\neg$ (negation), $\sqcap$ (intersection), $\sqcup$ (union), $\exists$ (existential quantifier), $\forall$(universal quantifier).

Definition 2.1 (Concept) Let Σ_C be the set of atomic concepts, Σ_R the set of atomic roles, $\Sigma_C \cap \Sigma_R = \emptyset$. The concepts of description logic ALC are recursively defined as follows:

(i) If $A \in \Sigma_C$ then A is a concept of ALC,

(ii) If C, D are concepts and $R \in \Sigma_R$ is a role, then $\top$, $\bot$, $\neg$ C, C $\sqcap$ D, C $\sqcup$ D, $\forall$R.C, $\exists$R.C are ALC concepts.

Example 2.1 We suppose that Person and Female are atomic concepts.

Person ⊓ Female and Person ⊓ ¬Female is a concept describing people that are female, and those that are not female.

Person ⊓ ¬Female is a concept describing people that are not female.

In addition, we suppose that hasChild is an atomic role, we can form the concepts Person ⊓ ∃hasChild.⊤ and Person ⊓ ∀hasChild. Female denote those people that have a child, and those people whose children are all female. Using the bottom concept, we can also describe those people without a child by the concept Person ⊓ ∀hasChild.⊥.

Definition 2.2 (Interpretations) An interpretations $\mathcal{I}$ of description logic ALC consists of a non-empty set $\Delta^{\mathcal{I}}$ (the domain of the interpretation) and an interpretation function $.^{\mathcal{I}}$ which assigns to every individual with an element of $\Delta^{\mathcal{I}}$, every atomic concept A with a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every atomic role R with a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is extended to concept descriptions by the following inductive definitions:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$$
$$\bot^{\mathcal{I}} = \emptyset$$
$$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$$
$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$
$$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$
$$(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b \in \Delta^{\mathcal{I}}, \text{ nếu } (a, b) \in R^{\mathcal{I}} \text{ thì } b \in C^{\mathcal{I}}\}$$
$$(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b \in \Delta^{\mathcal{I}}, (a, b) \in R^{\mathcal{I}} \text{ và } b \in C^{\mathcal{I}}\}$$

Definition 2.3

(i) The concept C is said to be included in the concept D, denoted C ⊑ D if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all interpretations $\mathcal{I}$, and C equivalent to D, denoted C ≡ D, if C ⊑ D and D ⊑ C.

(ii) An inclusive clause is of the form C ⊑ D, where C and D are two concepts.

Definition 2.4

(i) A finite set of inclusion clauses $\mathcal{T}$ is called a TBox.

(ii) An interpretation $\mathcal{I}$ is a model of the inclusion clause C ⊑ D if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ and $\mathcal{I}$ is a model of TBox $\mathcal{T}$ if it satisfies all the inclusion clauses in $\mathcal{T}$.

Definition 2.5

(i) An individual assertion is of the form a:C and a role assertion is of the form aRb, where C is a concept and R is a role, a, b are individual names. A membership assertion is either an individual or a role assertion.

(ii) An ABox $\mathcal{A}$ is a finite set of member assertions, that is, assertions of the form a:C or aRb.

(iii) An interpretation $\mathcal{I}$ satisfies the statement a:C if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and satisfies aRb if $(a^{\mathcal{I}},b^{\mathcal{I}}) \in R^{\mathcal{I}}$. I is said to be a model of Abox $\mathcal{A}$ if $\mathcal{I}$ satisfies all the assertions in $\mathcal{A}$.

Definition 2.6 (Knowledge Base) A knowledge base L of description logic ALC is a pair of TBox $\mathcal{T}$ and ABox $\mathcal{A}$. An interpretation $\mathcal{T}$ is a model of L if $\mathcal{T}$ is a model of both $\mathcal{T}$ and $\mathcal{A}$.

3. DATALOG PROGRAM

This section only presents some basic concepts of the Datalog program. Full details of the Datalog program can be found in [1], [2].

Definition 3.1 (Term)

(i) Constant or variable is the term.

(ii) A atom is of the form $p(t_1,...,t_n)$, where p is the n-arity predicate, $t_1,...,t_n$ are the terms.

Definition 3.2 (Datalog Program) A Datalog program consists of a finite set of rules of the form:

$$A_0 \leftarrow A_1,\dots,A_n \qquad \text{(where } n \geq 0\text{)} \qquad (1)$$

where $A_0,\dots,A_n$ are atoms whose argument is a constant or variable and all variables occurring in a rule are (implicitly) universally quantified over the whole rule. The atomic formula A_0 is called the head of the rule whereas $A_1,\dots,A_n$ is called its body. Comma (,) in (1) substitutes for conjunction operator (∧).

When n = 0, (1) becomes p ← and is called a unit rule, the symbol "←" can be omitted. The semantics of the unit rule p ← is that any substitution of the variables in p by constants will make p true. Unit rules with constant arguments are also known as facts.

Definition 3.2 (Herbrand universe/Herbrand base/Herbrand Interpretation) Let P be the Datalog program. Herbrand universe of P, denoted U_P is the set of all constants of P. Herbrand base of P, denoted B_P is the set of all ground atoms of P. The Herbrand interpretation of P is a subset of the Herbrand base B_P.

Definition 3.3 (Herbrand model). A Herbrand interpretation I of P is said to be a Herbrand model of P iif every rules of P is true in I.

Definition 3.4 (Semantics of Datalog program) Semantics of Datalog program P is the least Herbrand model of P.

The least Herbrand model of P that can be calculated by repeating the operator T_P, T_P is defined as follows:

Definition 3.5 (Immediate consequence operator) Let P be a Datalog program. Let ground(P) be the set of ground instances of the rules in P. Notation 2^{B_P} is a superset of

B_P. The immediate consequence operator of P is the mapping T_P: $2^{B_P} \rightarrow 2^{B_P}$ is defined as follows:

For each $I \in 2^{B_P}$, $T_P(I) = \{ a \in B_P \mid \exists\ a \leftarrow b_1,\ldots,b_n \in ground(P) \text{ and } \{b_1,\ldots,b_n\} \subseteq I\}$.

Theorem 3.1 [1] Let P be a Datalog program. Then the operator T_P is monotonous and the least fixed point of T_P is the least Herbrand model of P.

Proposition 3.1 [1] Let P be a Datalog program. The least Herbrand model M_P of P is the limit of the sequence $T_P\uparrow n$, $n \in N$, where $T_P\uparrow 0 = \varnothing$, $T_P\uparrow(i+1) = T_P(TP\uparrow i)$.

Example 3.1 See the following Datalog P program:

edge(1,2) edge(2,3) edge(3,4) edge(4,5)

path(X,Y) ← edge(X,Y)

path(X,Z) ← edge(X,Y) , path(Y,Z)

Using the T_P operator, the least Herbrand model of P is calculated as follows:

M_P = {edge(1,2), edge(2,3), edge(3,4), edge(4,5), path(1,2), path(2,3), path(3,4), path(4,5), path(1,3), path(2,4), path(3,5), path(1,4), path(2,5), path(1,5)}.

4. ALC DESCRIPTION DATALOG PROGRAM

This section will present the syntax and semantics of the ALC description Datalog program KB = (L,P). Informally, the KB consists of an ALC knowledge base L and an extended Datalog program P. For the extended Datalog program P, each concept, the role of the ALC description logic in turn can be viewed as the 1-arity predicate and the 2-arity predicate in P, called as the description logic predicate. Predicates that appear only in the extended Datalog program and not in L are called Datalog predicates. In the extended Datalog program P, the predicates can be either description logic predicates or Datalog predicates. A description logic atom is an atom whose predicate is a description logic predicate.

A. Syntax

Definition 4.1 (Extended Datalog rule) The extended Datalog rule is a formula of the form:

$$A_0 \leftarrow A_1,\ldots,A_n \qquad \text{(where } n \geq 0) \qquad (1)$$

where A_0 is a atom, Ai are description logic atoms whose argument are constant or variable.

Example 4.1 Consider a knowledge base with the concepts Student, Course, Topic, respectively, which refer to the objects of students, the subject, and the topic of the subject, respectively. See the following extended Datalog rule:

complete(X,Z) ← pass(X,Y), subject(Y,Z), X:Student, Y:Course, Z:Topic

In this example, atom pass(X,Y) means that student X has passed the subject Y and subject(Y,Z) means that Z is the topic of subject Y. Description logic atoms X: Student, Y:Course, Z:Topic in this rule are description logic assertions. This extended Datalog rule defines the predicate complete, the atom complete(X,Z) means that student X has completed topic Z if student X has passed the exam of subject Y and Z is the topic of subject Y.

Definition 4.2 (ALC description Datalog Program) An ALC description Datalog program KB = (L, P) consists of an ALC knowledge base L and a finite set P of extended Datalog rules.

B. Semantics

The concepts of the Herbrand universe/interpretation/model of the ALC description Datalog program KB = (L,P) are extended similarly in the case of the Datalog program. An interpretation I of KB is said to be a model of KB if it satisfies the rules of P and the knowledge base L. The semantics of KB are extended from the semantics of the Datalog program.

Definition 4.3 (Semantics) Semantics of the ALC description Datalog program KB = (L,P) as the least Herbrand model M_{KB} of KB.

Similar to the Datalog program, the M_{KB} can be calculated using the T_{KB} operator defined as follows:

Definition 4.4 (Immediate consequence operator for KB) Let KB = (L,P) be a ALC description Datalog program. The Immediate consequence operator for KB is the mapping: $T_{KB} : 2^{B_{KB}} \rightarrow 2^{B_{KB}}$ is defined as follows: For each $I \in 2^{B_{KB}}$,

$T_{KB}(I) = \{A \in B_P \mid \exists\, A \leftarrow A_1,\ldots,A_n \in ground(P)$ such that $A_i \in I$ if A_i is atom and A_i satisfy L if A_i is description logic atoms, $i \in \{1,\ldots,n\}\}$

Theorem 4.1 Let KB = (L,P) be a ALC description Datalog program. Then the T_{KB} operator is monotonous and the least fixed point of T_{KB} is the least Herbrand model M_{KB} of KB.

Proof: T_{KB} is a direct extension of T_P, so the correctness of the theorem is directly inferred from Theorem 3.1.

Example 4.2 Consider a ALC description Datalog program ALC KB = (L, P), where:

The ALC knowledge base L includes the following concepts: GV (lecturer), GVQL (lecturer doing management work), CBGD (teacher), DAY (teaching), SV (student), CHUDE (topic), HP (course), HPNC (advanced course), HPCB (basic course). Suppose:

1) Lecturers are teaching staff.

2) Lecturer doing the management work is the lecturers and do not teach any courses.

3) The set of courses is divided into basic courses and advanced courses.

4) Hung is the lecturer and Hung teaches the ttnt course, Mai is the lecturer and teaches the advanced courses, Hoa is a student, ttnt is the advanced course, cstt and ltlg are the topics.

Then the ALC knowledge base L is described as follows:

GV ⊑ CBGD

GVQL = GV ⊓ ¬∀DAY.HP

HPNC ⊔ HPCB = HP

HPNC ⊓ HPCB ⊑ ⊥

Hung:GV, Hung DAY ttnt,

Mai:GV ⊓∀DAY.HPNC,

Hoa: SV, ttnt: HPNC,

cstt: CHUDE, ltlg: CHUDE

The extended Datalog program P includes the following rules:

complete(X,Z) ← pass(X,Y), subject(Y,Z), X:SV,Y:HP(Y), Z:CHUDE

guide(X,Y) ← complete(X,Z), expert(Y,Z), X:SV, Z:CHUDE, Y:CBGD ⊓ ∃DAY.HPNC

guide(X,Y) ← X:SV, Y:GVQL

pass(Hoa,ttnt), subject(ttnt,cstt), subject(ttnt,ltlg),

expert(Hung,cstt), expert(Mai,ltlg)

The atoms in P have the following meanings:

guide(X,Y): Student X can be guided by lecturer Y,

complete(X,Y): Student X has studied course Y in the curriculum,

expert(X,Y): Lecturer X is an expert on topic Y,

pass(X,Y): Student X has passed the exam of course Y,

subject(X,Y): Course X has topic Y.

The semantics of the ALC Description Datalog program KB found by the T_{KB} operator is:

M_{KB} = { pass(Hoa,ttnt), subject(ttnt,cstt), subject(ttnt,ltlg), expert (Hung,cstt), expert(Mai,ltlg), complete(Hoa,cstt), guide(Hoa,Hung) }

5. QUERY EVALUATION FOR ALC DESCRIPTION DATALOG PROGRAM

This section will present a query valuation method for ALC description Datalog program.

Definition 5.1 (Goal, Answer)

1. A goal (or query) G for ALC description Datalog program KB = (L,P) is a formula of the form:

$$\text{false} \leftarrow q_1,\ldots,q_m \qquad (1)$$

where $m \geq 0$, each qi is a atom or an individual assertion constraint of the description logic. (1) can be simply written:

$$\leftarrow q_1,\ldots,q_m$$

2. An answer for KB $\cup$ G is a substitution θ for variables of G. We say θ is the correct answer for KB if $G\theta$ is a logical consequence of KB.

Definition 5.2 (Resolution) Let G be the goal of the form:

$$\leftarrow q_1,\ldots,q_i,\ldots,q_m$$

and C is an extended Datalog rule of the form:

$$p \leftarrow b_1,\ldots,b_n$$

Then, the unification of goal G with rule C will get goal G' derived from G and C using substitution θ if the following conditions are satisfied:

(i) $q_i \in \{q_1,\ldots,q_m\}$ is an atom, called a selected atom of goal G,

(ii) θ is the most general substitution such that $p\theta = q_i\theta$,

(iii) G' is the goal $\leftarrow (q_1,\ldots,q_{i-1}, b_1,\ldots,b_n, q_{i+1},\ldots,q_m)\theta$.

Definition 5.3 (Derivation) Let (L,P) be a ALC description Datalog program and G_0 is a goal. A derivation for (L,P) $\cup$ G includes:

1. A sequence of goals (which can be finite or infinite): $G_0,\ldots,G_n,\ldots$

2. A sequence of extended Datalog rules of P: $C_0,\ldots,C_n,\ldots$

3. A sequence of substitutions $\theta_1,\ldots,\theta_n,\ldots$ such that for each G_{i+1} is the unification of G_i and C_i using θ_{i+1}.

A derivation can end up with a final goal of the form $c_1,\ldots,c_k$ where c_i is the individuan assertion constraint of the descriptor logic, which is called an empty rule with constrain or simply an empty rule.

Definition 5.4 (Successful derivation, Failed derivation)

(i) A derivation succeeds when the goal ends with an empty rule.

(ii) A derivation fails when the goal does not end with an empty rule, the selected atom in this goal cannot unify with the head of every rule in the program.

Definition 5.5 (Computed answer substitution) Let (L,P) be a ALC description Datalog program and G as a goal. A substitution to compute answer of the successful derivation to (L,P)∪G is a substitution θ obtained by restricting to the variables of G for $\theta_1\theta_2\ldots\theta_n$, where $\theta_1,\theta_2,\ldots,\theta_n$ is the sequence of substitutions used in the successful derivation for (L,P)∪G.

Example 5.1 Consider the ALC description Datalog program KB = (L,P) in Example 4.2 and the goal G_0: ← guide(Hoa,Hung) with the meaning "Hoa was not guided by professor Hung".

For the goal G_0, find a rule in P whose head predicate is guide:

guide(X,Y) ← complete(X,Z), expert(Y,Z), X: SV, Z: CHUDE,Y: CBGD ⊓ ∃DAY.HPNC

Unify these two atoms guide(Hoa,Hung) and guide(X,Y) we get the substitution θ_1 = {X/ Hoa, Y/Hung }. The first derivation step generates the goal G_1:

G_1: ← complete(Hoa,Z), expert(Hung,Z), Hoa: SV, Z: CHUDE, Hung: CBGD ⊓ ∃DAY.HPNC

With the goal G_1, the selected atom is complete(Hoa,Z), now we find the rule in P:

complete(X,Z) ← pass(X,Y), subject(Y,Z), X: SV, Y: HP, Z: CHUDE.

In the second resolution step, we get the substitution θ_2 = {X/ Hoa}. The second derivation step generates the goal G_2:

G_2: ← pass(Hoa,Y), subject(Y,Z), expert(Hung,Z), Hoa:SV, Y:HP, Z:CHUDE, Hung:CBGD ⊓ ∃DAY.HPNC

With the goal G_2, the selected atom is pass(Hoa,Y), we find the fact pass(Hoa,ttnt). The substitution obtained in this step is θ_3 = {Y/ttnt}. The third derivation step generates the goal G_3:

G_3: ← subject(ttnt,Z), expert(Hung,Z), Hoa:SV, ttnt:HP, Z:CHUDE, Hung:CBGD ⊓ ∃DAY.HPNC

With goal G_3, the selected atom is subject(ttnt,Z), we find the fact subject(ttnt,cstt). The substitution obtained in this step is θ_4 = {Z/cstt}. The fourth derivation step generates the goal G_4:

G_4: ← expert(Hung,cstt), Hoa: SV, ttnt: HP, cstt: CD, Hung: CBGD ⊓ ∃DAY.HPNC

With the goal G_4, the selected atom is expert(Hung,cstt), we find the unit rule expert(Hung,cstt) of P to resolve, so we get the empty rule:

← Hoa: SV, ttnt: HP, cstt: CHUDE, Hung: CBGD ⊓ ∃DAY.HPNC

Thus, the derivation for (L,P) ∪G0 is successful and gets the correct answer for the goal:

G_0: ← guide(Hoa,Hung)

This means that Hoa is guided by professor Hung.

Theorem 5.1 The resolution to evaluate the goal G for the ALC description Datalog program KB = (L,P) is correct, which means every substitution to compute the answer for KB G is the correct answer.

Proof. Letting G be the goal $\leftarrow q_1,\ldots,q_m$ and $\theta_1,\ldots,\theta_n$ is the sequence of substitutions used in the resolution of KB $\cup$ {G}. We prove that $(q_1,\ldots, q_m)\theta_1\ldots\theta_n$ is a logical consequence of KB by using induction on the length of the resolution.

For n = 1, the goal G has the form $\leftarrow q_1$, the program P has a unit rule of the form $A \leftarrow$ and $q_1\theta_1 \equiv A\theta_1$. Since $A\theta_1 \leftarrow$ is an instance of the unit rule of P, this implies that $q_1\theta_1$ is a logical consequence of KB.

Next, we assume the theorem's conclusion is true for the substitution to compute the answer of the resolution for KB $\cup$ {G} of length n-1. Now the successful derivation is:

$$G_0 \xrightarrow{C_0} G_1\ldots G_{n-1} \xrightarrow{C_{n-1}} \text{empty rule}$$

where G_0 is the first goal: $\leftarrow q_1,\ldots,q_m$

Suppose q_j is a selectd atom in the first derivation step of the goal G_0 and C_0 is the rule in P: $B_0 \leftarrow B_1,\ldots,B_k$ $(k \geq 0)$.Then $q_j\theta_1 = B_0\theta_1$ and goal G_1 has the form:

$$\leftarrow (q_1,\ldots,q_{j-1}, B_1,\ldots,B_k,q_{j+1},\ldots,q_m)\theta_1 \quad (1)$$

According to the induction hypothesis, the formula:

$$(q_1,\ldots,q_{j1},B_1,\ldots,B_k,q_{j+1},\ldots,q_m)\theta_1\ldots\theta_n$$

is the logical consequence of KB. Therefore:

$$(B_1,\ldots, B_k)\theta_1\ldots\theta_n \quad (2)$$

is also a logical consequence of KB. From (1) we also have:

$$(q_1,\ldots,q_{j-1},q_{j+1},\ldots,q_m)\theta_1\ldots\theta_n \quad (3)$$

is the logical consequence of KB. Then from (2) and from:

$$(B_0 \leftarrow B_1,\ldots,B_k)\theta_1\ldots\theta_n \quad (4)$$

is the logical consequence of the program, we obtain the logical consequence of KB is $B_0\theta_1\ldots\theta_n$. From (3) and (4), we have $(q_1,\ldots, q_{j-1}, B_0, q_{j+1},\ldots, q_m)\theta_1\ldots\theta_n$ is the logical consequence of KB. Since θ_1 is the substitution of B_0 and q_j, B_0 can be replaced by q_j in (4). Therefore $(q_1,\ldots,q_m)\theta_1\ldots\theta_n$ is the logical consequence of KB.

6. CONCLUSION

The paper has focused on presenting the syntax and semantics of the ALC description Datalog program - an integration of the ALC description logic and the Datalog program. The ALC description Datalog program semantics is a natural extension of the Datalog program semantics. In addition, the paper also provides an inference mechanism to evaluate the query for the ALC description Datalog program. This method is a natural extension of SLD resolution for query evaluation for a Datalog program. For the case where the ALC description Datalog program contains negation in the rules, the semantics and method of query valuation will be further considered. In the framework of the paper, we have not discussed this issue.

REFERENCES

[1]. Abiteboul S. ,Hull R. ,Vianu V. , Foundation of Databases, Addision Wesley Publishing, MA, 1995.

[2]. Apt K. R. , Logic Programming, Elsevier Science Publishers, 1990.

[3]. Baader F. , D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.

[4]. Cuenca Grau, B., Motik, B., Stoilos, G., Horrocks, I.: "Computing Datalog rewritings beyond Horn ontologies". In: IJCAI. pp. 832-838, 2013.

[5]. 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.

[6]. Heymans S. , T. Eiter, G. Xiao, "Tractable reasoning with DL-programs over Datalog-rewritable description logics", in: H. Coelho, R. Studer, M. Wooldridge (Eds.), ECAI 2010, Proc. 19th European Conference on Artificial Intelligence, vol. 215 of Frontiers in Artificial Intelligence and Applications, IOS Press, 35–40, 2010.

[7]. Kaminski, M., Nenov, Y., Cuenca Grau, B.: Datalog rewritability of disjunctive Datalog programs and its applications to ontology reasoning. in: C. E. Brodley, P. Stone (Eds.), Proc. 28th AAAI Conference on Artificial Intelligence, AAAI, 1077–1083, 2014.

[8]. Mei, J., Lin, Z., Boley, H., Li, J., & Bhavsar, V. C, The DatalogDL: Combination of deduction rules and description logics, Computational Intelligence, 23(3), pp 356-372, 2007.

[9]. Shen, Y. D., "Well-supported semantics for description logic programs". In Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11), 2011.

[10]. Thomas Lukasiewicz, "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.

[11]. Xiao G., Heymans S., and Eiter T., "DReW: a reasoner for Datalog-rewritable description logics and dl-programs". In Informal Proc. 1st Int'l Workshop on Business Models, Business Rules and Ontologies, 2010.

NGỮ NGHĨA VÀ PHƯƠNG PHÁP ĐÁNH GIÁ TRUY VẤN ĐỐI VỚI CHƯƠNG TRÌNH DATALOG MÔ TẢ ALC

Trương Công Tuấn

Khoa Công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế

Email: tctuan@husc.edu.vn

TÓM TẮT

Chương trình Datalog là một lớp quan trọng của cơ sở dữ liệu suy diễn và được mở rộng theo nhiều hướng khác nhau. Bài báo tập trung vào việc mở rộng chương trình Datalog bằng cách kết hợp chương trình Datalog với logic mô tả ALC. Chúng tôi đề xuất ngữ nghĩa và phương pháp đánh giá truy vấn cho lớp chương trình này.

Từ khoá: Cơ sở dữ liệu suy diễn, Chương trình Datalog, logic mô tả.

Trương Công Tuấn sinh ngày 28/4/1960 tại Hội An. Ông tốt nghiệp đại học Tổng hợp Huế năm 1982, chuyên ngành toán học tối ưu; tốt nghiệp Tiến sỹ tại Viện Công nghệ thông tin Hà Nội năm 2003 chuyên ngành Khoa học máy tính; được phong học hàm PGS năm 2012. Hiện đang công tác tại Khoa Công nghệ thông tin, trường Đại học Khoa học, Đại học Huế.

Lĩnh vực nghiên cứu: Hệ cơ sở tri thức, Lập trình logic và Cơ sở dữ liệu suy diễn.