Front. Comput. Sci., 2013, 7(6): 812–821
DOI 10.1007/s11704-013-2284-2

RESEARCH ARTICLE

Integrity constraints in OWL ontologies based on grounded circumscription

Dantong OUYANG[1,2], Xianji CUI[1,2], Yuxin YE (✉)[1,2,3]

1 College of Computer Science and Technology, Jilin University, Changchun 130012, China
2 Key Laboratory of Symbolic Computation and Knowledge Engineering Ministry of Education, Changchun 130012, China
3 National Engineering Research Center of Geophysics Exploration Instruments, Changchun 130026, China

Abstract The extensions for logic-based knowledge bases with integrity constraints are rather popular. We put forward an alternative criteria for analysis of integrity constraints in Web ontology language (OWL) ontology under the closed world assumption. According to this criteria, grounded circumscription is applied to define integrity constraints in OWL ontology and the satisfaction of the integrity constraints by minimizing extensions of the predicates in integrity constraints. According to the semantics of integrity constraints, we provide a modified tableau algorithm which is sound and complete for deciding the consistency of an extended ontology. Finally, the integrity constraint validation is converted into the corresponding consistency of the extended ontology. Comparing our approach with existing integrity constraint validation approaches, we show that the results of our approach are more in accordance with user requirements than other approaches in certain cases.

Keywords semantic Web, description logic, ontology, integrity constraints, grounded circumscription

1 Introduction

Web ontology language (OWL) [1] is the endorsed standard ontology definition language for the world wide Web consortium (W3C) and used for representing Web data in the semantic Web. As the logical foundation of OWL, description logics (DLs) [2] provide sound and complete reasoning algorithms for standard reasoning tasks, such as concept satisfaction, consistency, and instance checking. However, beside standard reasoning tasks, ontology data need to satisfy integrity constraints in data-centric applications.

Integrity constraints were originally proposed in database and artificial intelligence knowledge representation languages to guarantee the legal states that are considered acceptable by knowledge bases. In a database, integrity constraints are captured by dependencies. For example, an integrity constraint that states that "every student must have a student ID". It is captured by an inclusion dependency, and the dependency may be interpreted as a check during database updates. Whenever a student is added into the database, a check may be performed to check whether the student ID for that student is provided, if not, the integrity constraint is violated and the update is rejected.

Similarly integrity constraints are needed in ontologies to guarantee their integrity. For example, in the development of information systems, integrity constraints are required to represent the application requirements in the ontology [3]. Furthermore, in the translation from a database model to an OWL model, they are also required to identify the ontology elements and restrictions on these elements [4].

Along this view, we aim to add the integrity constraints into the ontology to check whether ontology data satisfy certain constraints. However, it is not a simple issue to mimic

Received September 5, 2012; accepted July 3, 2013

E-mail: yeyx@jlu.edu.cn

these integrity constraints in the ontology and we should consider the following three issues. The main challenge of modeling integrity constraints in an ontology is the difference in semantics: databases adhere to the closed world assumption (CWA), that is, statements which are not logical consequences of a given knowledge base are considered false (use the "negation as failure" idea). Whereas the standard semantics of OWL follows the open world assumption (OWA), that is, statements which are not logical consequences of a given knowledge base are not necessarily considered false; this allows for uncertain information. The standard OWL semantics makes data in the semantic Web faithfully capture the information in the real world. However, due to the appearance of uncertain information, it is difficult to model such constraints under the OWA. Therefore, we are required to model the integrity constraints under the CWA. Furthermore, according to the user requirements, the integrity constraints are used for checking whether the information which is absolutely required has been specified explicitly. In other words, when we define the satisfaction of integrity constraints, we only need to check the individuals which are explicitly stated in the ontology. Last but not least, since the ontology has the ability of inferring implicit information, the standard reasoning is also essential and required to be considered during the integrity constraint validation process. Thus, the modeling of integrity constraints presented in this paper is a contribution toward finding an appropriate semantics of integrity constraints which meets the above requirements.

The main contributions of the paper are listed as follows:

- We extend the ontology with integrity constraints based on the grounded circumscription and define the semantics of integrity constraints under the closed world assumption.
- To meet the user requirements, we apply the idea of circumscription to minimize the models of the extended ontology w.r.t. integrity constraints and further define the satisfaction of the integrity constraints in ontologies.
- To consider standard reasoning in the integrity constraint validation process, we provide a full and complete description of the consistency algorithm for extended ontologies, which contributes to a feasible implementation of integrity constraint validation in ontologies.

In Section 2, we discuss the existing approaches for modeling integrity constraints in DL knowledge bases. In Section 3 we extend the traditional DL knowledge base with integrity constraints based on the minimal idea of grounded circumscription, and introduce the correspondence between the extended DL knowledge base and OWL ontologies. We then propose an algorithm to reduce the integrity constraint validation into the consistency of the extended ontology, and further prove the feasibility of the algorithm in Section 4. We conclude in Section 5.

2 Related work

Integrity constraints describe the admissible states of a knowledge base. To mimic integrity constraints, a long research tradition has been to extend logic-based knowledge representations using integrity constraints.

The conventional perspective of integrity constraints is to directly represent them with TBox axioms. Then the satisfaction of the set of integrity constraints IC over a knowledge base K is defined by checking whether $K \cup \mathrm{IC}$ is consistent or K entails IC [5]. However, this is not a suitable basis for integrity constraint validation. Consider IC contains an integrity constraint states that "every student must have a student ID". It is expressed with the following axiom.

$$\text{Stduent} \sqsubseteq \exists \text{hasID.studID}. \tag{1}$$

Suppose that K contains the following assertion states that "Peter is a student".

$$\text{Student(Peter)}, \tag{2}$$

then $K \cup \mathrm{IC}$ is consistent and Eq. (1) is satisfied. But we would expect it to be violated by K, since K states that Peter is a student without specifying a student ID for him. For the entailment of K, suppose $K = \{\ \}$. Intuitively, it should satisfy Eq. (1), but Eq. (1) is violated since $K \not\models \text{Stduent} \sqsubseteq \exists\text{hasID.studID}$. This is because in the standard DLs, TBox axioms are used for inferring new information rather than checking the satisfaction of integrity constraints. Thus, we cannot directly use the standard axioms as integrity constraints. From the perspective of integrity constraints in a relational database, we see that integrity constraint validation should be under the CWA. Along this view, some significant proposals involve hybrid knowledge bases which consist of ontology axioms and rules [6,7]. Integrity constraints are directly represented by rules and satisfactions of integrity constraints are reduced to the consistencies of hybrid knowledge bases. However, since most of them are of a hybrid nature, combining description logic and rules may complicate the reasoning tasks.

In addition, Reiter argues that integrity constraints are epistemic in nature and about "what the knowledge base knows" [8]. Levesque [9] has proposed that integrity constraints can use the semantics of auto-epistemic logic. Along this view, the extended DLs with epistemic operator **K** [10] and "negation as failure" **A** [11] are proposed to model the integrity constraints. Furthermore, the integrity constraints are also modeled based on modal logic and integrity constraint validation is converted into SPARQL queries [12]: an integrity constraint is violated if and only if the answer to corresponding SPARQL queries is not null. It is an effective approach to check the satisfaction. However, when K also includes the standard axioms that are used for inferring new information, it may obtain incorrect results because of missing the implicit information. Consider the integrity constraint set IC contains an integrity constraint stating that "every student must take a course".

$$\text{Student} \sqsubseteq \exists \text{takesCourse.Course}. \tag{3}$$

Suppose $K = (S, A)$, where the standard axiom in S (4) states that "every graduate course is also a course", and ABox assertions A contains the assertions (Eqs. (5)–(7)) state that "John is a student who takes a course of English".

$$\text{GraduateCourse} \sqsubseteq \text{Course}. \tag{4}$$

$$\text{Student(John)}. \tag{5}$$

$$\text{takesCourse(John, English)}. \tag{6}$$

$$\text{GraduateCourse(English)}. \tag{7}$$

In this way, the SPARQL queries only focus on the assertions (Eqs. (5)–(7)), and the answer to the queries is "John". Thus Eq. (3) is violated. Intuitively, according to standard axiom Eq. (4) and assertion Eq. (7), we can infer the assertion Course(English) and Eq. (3) should be satisfied. Thus, the standard axioms need to be considered in the integrity constraint validation process.

Moreover, by comparing the difference between OWL and relational databases, Motik et al. [13] have defined the satisfaction of integrity constraints using minimal Herbrand models: an integrity constraint axiom α in IC is satisfied by K if for all minimal Herbrand models I of K, $I \models \alpha$. Finally, they translated all the TBox axioms in K and IC into first-order formulas and used logic programming to complete integrity constraint validation. While this approach solves the outlined problem to a certain extent, there are some problems. Consider IC contains an integrity constraint states that "every professor must teach a course".

$$\text{Professor} \sqsubseteq \exists \text{teacherOf.Course}. \tag{8}$$

Suppose $K = (S, A)$, where the standard axiom in S (9) states that "every faculty member is either a professor or a lecturer", and ABox assertions A contains the assertion (10) states that "Mary is a faculty member".

$$\text{FacultyMember} \sqsubseteq \text{Professor} \sqcup \text{Lecturer}. \tag{9}$$

$$\text{FacultyMember(Mary)}. \tag{10}$$

Then according to the definition referred to the approach of Motik et al., we know that Eq. (8) is violated since there is a minimal Herbrand model such that individual Mary is an instance of Professor, but does not have a specified course for Mary. In fact, according to Eq. (9) we cannot determine whether the individual Mary is an instance of Professor or not, thus it is rational to assume that the constraint will not apply to the individual Mary, and Eq. (8) is satisfied. Thus, we know that not all the minimal Herbrand models of K are suitable to check the satisfaction of K. Therefore, it is necessary to find a more appropriate minimal model for K w.r.t. IC.

According to the related work discussed above, we find that these models of integrity constraints do not satisfy the condition in certain aspects. To deal with the above issues, our proposed model of integrity constraints is as follows: (1) from the perspective of integrity constraints in relational databases, the integrity constraint validation task should be performed under the closed world assumption; (2) to meet user requirements in practice, the definition of integrity constraints satisfaction should only apply to the instances related to the integrity constraints. That is, in model-theoretic semantics, we should use the minimal model w.r.t. integrity constraints; (3) to maintain distinguished features for DL knowledge bases, standard reasoning should also be considered during integrity constraint validation process. In the following section, we will introduce grounded circumscription to define the semantics of integrity constraints and obtain an alternative minimal model of K to define the satisfaction of integrity constraints in extended ontologies.

In our previous work [14], the integrity constraints in OWL ontology are represented by TBox axioms called IC-axiom, we consider the integrity constraint validation for continually changing ontologies based on the integrity constraint validation method of Tao et al. By analyzing the difference between OWL and relational database, we have implemented the storage of OWL ontology data with integrity constraints in relational databases [15] and provide corresponding query translations [16]. In this paper, we define an alternative algorithm to check the satisfaction of integrity constraints in ontologies, which is more appropriate to the data-centric applications, that is, our algorithm obtains more intuitive results.

3 Integrity constraint ontology with grounded circumscription

In order to add the integrity constraints into the ontology, we define the extended ontology with integrity constraints.

Definition 1 Call an extended knowledge base (K, IC) the integrity constraint knowledge base (IC-KB), where K is a knowledge base which contains a finite set of ABox assertions A and a finite set of standard TBox axioms S, and IC is a finite set of integrity constraint axioms.

The TBox axioms distinguish the standard axioms that are only used for inferring new information from the integrity constraint axioms that are used for checking integrity. As discussed in the above section, it may be preferable to validate the integrity constraints under the CWA, and the standard OWL is under the OWA. Therefore, for standard axioms, we still use standard reasoning under the standard OWA, whereas validation of integrity constraints is performed under the CWA with integrity constraint axioms.

In this paper, we only focus on how to check the satisfaction of integrity constraints. For the standard reasoning tasks, you can refer to [2]. Along this view, we aim to extend traditional DL attribute concept language with complement (ALC) with grounded circumscription to define the satisfaction of integrity constraints. Some results in this paper also apply to many other description logics besides ALC, and we will point this out in each case.

3.1 Semantics of integrity constraints

Since the TBox axioms in DL knowledge base are similar to the schemas in databases, we use TBox axioms to represent the integrity constraints in the ontology. However, the meaning of a dependency in database is different from the meaning of an inclusion axiom in DL knowledge base: while the inclusion axiom $C_1 \sqsubseteq C_2$ represents conceptual knowledge and says that, no matter what is known about individuals, concept C_2 subsumes C_1. Whereas, the dependency $\forall x.(C_1(x) \rightarrow C_2(x))$ represents the incidental fact that "if an individual is known to be an instance of C_1, then one can conclude that it should also be an instance of C_2". Consider the following example, suppose that a knowledge base K contains the standard axiom Eq. (11) that states that "every professor is also a faculty member", and an assertion Eq. (10) that states that "a is not a faculty member".

$$\text{Professor} \sqsubseteq \text{FacultyMember}. \tag{11}$$

$$\neg\text{FacultyMember}(a). \tag{12}$$

Obviously we can derive $\neg$Professor(a) from K. In this representation we assume a conceptual relation between the terms Professor and FacultyMember. The inclusion axioms are used for inferring new information but not as a constraint. Then let K' be the knowledge base that results from K when Eq. (11) is replaced with the dependency

$$\forall x.(\text{Professor}(x) \rightarrow \text{FacultyMember}(x)). \tag{13}$$

The assertion $\neg$Professor(a) turns out to be not derivable from K'. The explanation of dependency is that in knowledge base K' we do not find an individual that is an instance of the concept Professor. Therefore rule Eq. (13) does not "fire" and the constraint (dependency) is satisfied. Therefore, it would be desirable to have a precise model-theory explanation of the behavior of dependencies.

From the perspective of the application, only users want to determine whether the information, which is required, has been specified correctly in knowledge base. That is, we only need to consider the individuals that explicitly appear in knowledge base K. Along this view, we can state that predicates in integrity constraint axioms are closed. Semantically, their extensions only contain known individuals in K. The set of individuals that explicitly appears in knowledge base K is denoted by Ind(K). It turns out that this semantics provides us with the required models. Restricting the extensions of predicates in integrity constraint axioms will allow us to rephrase the dependency Eq. (13) by the inclusion axiom Professor $\sqsubseteq$ FacultyMember, which says that all objects that are known to be professors are also faculty members.

At the base of this, integrity constraints can be modeled by TBox axioms whose predicates are all closed. We call each predicate in integrity constraint axioms G-predicate, denoted with constructor M, and DL ALCM is the extension of DL ALC with G-predicates. Let N_C, N_R and N_I be the countable infinite disjoint sets of atomic concept names, atomic role names and individual names. The syntax is defined iteratively as follows:

$$C, D \ ::= \ \top \mid \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C \mid MC,$$

$$R \ ::= \ R_\mathrm{a} \mid MR_\mathrm{a},$$

where A represents an atomic concept, R_a atomic role, C, D are concepts and M is the G-constructor.

The semantics of DL ALCM is obtained by interpreting concepts and roles in DL ALCM. An ALCM interpretation

$I_M = (\triangle, \cdot^{I_M})$, where $\triangle$ is the domain of interpretation and the mapping function $\cdot^{I_M}$ maps $A \in N_C$ to the subset of $\triangle$ and $R_a \in N_R$ to the subset of $\triangle \times \triangle$. It is obvious that for the standard predicates, the interpretation of the concept(/role) is the same with standard DL ALC. Whereas, for the G-predicates, their interpretations only contain the individuals in Ind(K).

$$(MC)^{I_M} = \{a \in C^{I_M} \mid a \in \mathrm{Ind}(K)\},$$

$$(MR)^{I_M} = \{(a, b) \in R^{I_M} \mid a, b \in \mathrm{Ind}(K)\}$$

For the negation of concepts appears in integrity constraint axioms, the interpretation is as follows: $(\neg MA)^{I_M} = N_k^{I_M} \backslash A^{I_M}$, where $(N_k)^{I_M} = \{x^{I_M} \mid x \in \mathrm{Ind}(K)\}$. We can see that this is different from the standard interpretation, the interpretation of $\neg MA$ is the set of objects in $\triangle$ that are mappings of known individual names in K that are not instance of A. It is similar to the negation as failure used in the CWA. Thus, this interpretation of predicates in integrity constraint axioms is appropriate to the CWA.

We next need to identify certain axioms as integrity constraints. In fact, all the TBox axioms can be identified as integrity constraints, and they mainly depend on the practical applications. In the following, we discuss axioms that are likely to be identified as integrity constraints where all the predicates are all closed.

- Existential constraints: $C \sqsubseteq \exists R.D$ involve two concepts C and D and a relation R between them. They state that every known instance of C in K must participate in one or more-relationships with known instances of D in K. They are closely related to inclusion dependencies in relational databases.
- Value constraints: $C \sqsubseteq \forall R.D$ also involve two concepts C and D and a relation R between them. They state that every known instance of C and every pair of instances of R in K must have known instances of D in K.
- Typing constraints can be used to check whether objects are correctly typed. Typical examples of such statements are domain and range constraints: when interpreted as integrity constraints, they state that R-links can only point from or to objects that are explicitly typed as C for a role R and a concept C. the domain and range constraints are generally of the form $\top \sqsubseteq \forall R.C$ and $\exists R.C \sqsubseteq \top$.

3.2 GC-DL with integrity constraint

According to the semantics of integrity constraints in DL knowledge bases, we will define satisfaction of the integrity constraints over a DL knowledge base in the following. Since the integrity constraint satisfaction can be detected by the minimal models of knowledge bases [13], we aim to find appropriate models of the extended knowledge base w.r.t. integrity constraints to define satisfaction.

Intuitively, the integrity constraints can be captured by minimal models. However, according to the example shown in Motik's work [13], we find that the minimal Herbrand model is not sufficient to define the satisfaction of integrity constraints. Since we focus on the satisfaction of integrity constraints, the models of extended knowledge bases should be minimal w.r.t. the integrity constraint axioms. Its idea is similar to circumscription [17], which is a non-monotonic logic and employs the minimization idea by restricting the extensions of some predicates to be minimal. In this paper, we use the minimization idea of grounded circumscription [18] to define the satisfaction of integrity constraints in ontologies.

First, we give definition of the models of extended DL knowledge base. The ALCM knowledge base K consists of TBox and ABox. TBox is the sets of terminology axioms of the form $C_1 \sqsubseteq C_2$ where C_1, C_2 are G-concepts or standard concept. ABox is the set of assertions of the form $C(a)$, $R(a, b)$, where C is a G-concept or standard concept, R is a G-role or standard role, and a, b are names of individuals. We call that the interpretation I_M satisfy the TBox axiom $C_1 \sqsubseteq C_2$ if $C_1^{I_M} \subseteq C_2^{I_M}$, the concept assertion $C(a)$ if $a^{I_M} \in C^{I_M}$, the role assertion $R(a, b)$ if $(a^{I_M}, b^{I_M}) \in R^{I_M}$. Call I_M the grounded model of a knowledge base K w.r.t. IC, if I_M satisfies all the axioms and assertions of K. The set of grounded model of K w.r.t. IC is denoted with Mod(K).

Then, we define the partial order w.r.t. integrity constraint axioms to obtain the minimal model of extended DL knowledge base w.r.t. integrity constraints.

Definition 2 Let I, J be two grounded models of K, $\mathrm{Sig}_{\mathrm{IC}}$ be the set of predicate names appear in IC, we call that the I is preferable than J w.r.t. IC, denoted with $J \prec_{\mathrm{IC}} I$, if and only if all the following conditions hold:

- $\triangle^I = \triangle^J$
- $a^I = a^J$, for every $a \in N_I$
- $e^I \subseteq e^J$, for every $e \in \mathrm{Sig}_{\mathrm{IC}}$
- There exists one $e \in \mathrm{Sig}_{\mathrm{IC}}$, such that $e^I \subset e^J$

Definition 3 Let (K, IC) be an IC-KB, An interpretation I is a GC-model of K w.r.t. IC, if it is the grounded model of K w.r.t. IC and minimal w.r.t. IC. An IC-KB is said to be GC-satisfiable (GC-consistent) if it has a GC-model w.r.t. IC.

The set of GC-models of K w.r.t. IC is denoted with $\mathrm{GCM}(K) = \{I \in \mathrm{Mod}(K) \mid \nexists J, I \in \mathrm{Mod}(K), J \prec_{\mathrm{IC}} I\}$. Now, according to the minimal model w.r.t. integrity constraints, we present the definition of satisfaction of integrity constraints.

Definition 4 Let (K, IC) be an IC-KB, for every IC-axiom $\alpha \in \mathrm{IC}$, we say that K satisfies α, denoted with $K \models_{\mathrm{IC}} \alpha$ if and only if for every GC-model I of K w.r.t. IC such that $I \models \alpha$.

In the following section, we will give the algorithm for the satisfaction of integrity constraints which is accordance with above definition.

3.3 OWL and DL ALCM

Description logic is the logical foundation of OWL and the logical counterpart of OWL Lite, OWL DL, and OWL2 are the DLs SHIF(D), SHOIN(D), and SROIQ(D) [19], respectively. Because of that, we refer to OWL and description logics interchangeably throughout this paper.

In this paper, we only consider DL ALCM; and the corresponding OWL descriptions are shown in Table 1. In this table the first column gives the OWL abstract syntax for the constructions, while the second column shows the DL syntax. Each predicate of the form MC and MR represents the concept or role in the integrity constraint axioms, respectively. Note that all the integrity constraint axioms are individually placed in a file.

Table 1 Syntax of ALCM and corresponding OWL

OWL	DL ALCM
Class	**Concept**
A(URI reference)	A
owl:Thing	$\top$
owl:Nothing	$\bot$
intersectionOf(C_1, C_2)	$C_1 \sqcap C_2$
unionOf(C_1, C_2)	$C_1 \sqcup C_2$
complementOf(C)	$\neg C$
Restriction(R someValuesFrom(C))	$\exists R.C$
Restriction(R allValuesFrom(C))	$\forall R.C$
groundCircumOf(C)	MC
Property	**Role**
R (URI reference)	R
groundCircumOf(R)	MR
Object	**Individual**
a	a
Axiom	
Class(A partial $C_1 ... C_n$)	$A \sqsubseteq C_1 \sqcap \cdots \sqcap C_n$
Class(A complete $C_1 ... C_n$)	$A \equiv C_1 \sqcap \cdots \sqcap C_n$
subClassOf(C_1 C_n)	$C_1 \sqsubseteq C_2$
Fact	
Individual(a type(C_1)...type(C_n)	$a \in C$
Value(R_1, a_1) ... Value(R_n, a_n))	$(a, b) \in R$

4 Integrity constraint validation of OWL ontology with grounded circumscription

In the following, we will give a GC-consistency algorithm for the extended DL knowledge bases and further convert the integrity constraint validation task into the corresponding GC-consistency task. For simplicity of presentation, we only consider GC-satisfiability for DL ALC, but the procedure should be adaptable to other DLs.

4.1 GC-consistency algorithm of IC-KB

GC-consistency of an IC-KB (K, B) is similar to the tableau procedure for the traditional description logics [2]. The GC-consistency of (K, IC) is also constructed through the directed graph G. The graph consists of nodes, edges, and labels. Individuals are represented as nodes along with their labels that represent the concepts that contain them in K. Additionally, pairs of individuals are represented as edges along with labels that represent roles which link the individuals as role assertions in K. Unlike the traditional description logics, since the semantics of the extended knowledge base is under the CWA, we use the meaning of negation as failure. Therefore, for each concept in integrity constraint axioms and nodes in the graph, if there is no corresponding concept in the label, we add the negation of the concept into the label set.

The algorithm begins with the initialized graph G, and

Table 2 Tableau expansion rules for (K, IC)

Expansion rule	Operation
$\rightarrow_{\mathrm{GC}}$Rule	**if** $(C_1 \sqsubseteq C_2) \in \mathrm{IC}$ and $\mathrm{NNF}(\mathrm{DM}(C_1 \sqcap \neg C_2)) \notin L(x)$ **then** set $L(x) := L(x) \cup \{\mathrm{NNF}(\mathrm{DM}(C_1 \sqcap \neg C_2))\}$
$\rightarrow_{\sqsubseteq}$Rule	**if** $(C_1 \sqsubseteq C_2) \in \mathrm{IC}$ and $\mathrm{NNF}(\neg C_1 \sqcup C_2) \notin L(x)$ **then** set $L(x) := L(x) \cup \{\mathrm{NNF}(\neg C_1 \sqcup C_2)\}$
$\rightarrow_{\sqcap}$Rule	**if** $C_1 \sqcap C_2 \in L(x)$ and $\{C_1, C_2\} \not\subset L(x)$ **then** set $L(x) := L(x) \cup \{C_1, C_2\}$
$\rightarrow_{\sqcup\mathrm{GC}}$Rule	**if** $C_1 \sqcup C_2 \in L(x)$ and $\{C_1, C_2\} \cap L(x) = \oslash$ **then** set $L(x) := L(x) \cup \{C_i\}$ for some $C_i \notin \mathrm{Sig(IC)}$
$\rightarrow_{\exists\mathrm{GC}}$Rule	**if** $\exists R.C \in L(x)$ and x has no R-successory y with $C \in L(y)$ **then** for every $b \in \mathrm{Ind}(K)$ has predecessor $x \in \mathrm{Ind}(K)$ do set $L(b) := L(b) \cup \{C\}$ set $L(x, b) := L(x, b) \cup \{R\}$
$\rightarrow_{\forall}$Rule	**if** $\forall R.C \in L(x)$ and x has an R-successor y with $C \notin L(y)$ **then** set $L(y) := L(y) \cup \{C\}$

Note: NNF(C) is the normal negation form of C that negation occurs only in front of concept names; DM(C) is the corresponding standard-concept of GC-concept C that constructor M has been eliminated.

proceeds by non-deterministically applying the rules defined in Table 2. We have modified ∃-rule and ⊔-rule, such that the candidate models adhere to the semantics of extended knowledge bases. These rules are applied iteratively until a clash is detected or none of the rules is applicable. A graph G contains clash if the node labels contain both C and $\neg C$, or $\perp$. The directed graph is so-called completion graph. The process can be understood as creating a candidate grounded model for the knowledge base K. We may determine that the IC-KB is GC-consistent, if we can get a clash-free completion graph, otherwise, GC-inconsistent.

In the following, we discuss the characteristics of the IC-KB.

Theorem 1 (termination). Given any IC-KB (K, IC), the algorithm procedure for (K, IC) terminates.

Proof First, note that node labels can only consist of concepts and its sub-concepts in K. Thus, there is only a finite set of possible node labels and there is a global bound $m \in N$, which is the cardinality of node labels. Additionally, the number of times any rule can be applied to a node is finite, since the labels trigger the rules and the size of the labels is bounded by m. On the other hand, since no new node can be created, the algorithm may be terminated when no rules are in use.

Theorem 2 (soundness). The expansion rules are applied to an IC-KB (K, IC) and result in a clash-free completion graph G, then K has a grounded model I w.r.t. IC.

Proof From the clash-free completion graph G, we create an interpretation I, the interpretation domain contains all the non-blocked nodes in the completion graph. Further, for every atomic concept C, set C^I to be the set of all non-blocked nodes x for which $C \in L(x)$; for every role name R, set R^I to be the set of pairs (x, y) which $R \in L(x, y)$ and y is not blocked. Since the completion graph is free of inconsistency clashes and the expansion rules from Table 2 follow the model definition in the above section. Then the resulting interpretation is indeed a model of K w.r.t. IC. Furthermore, since the existential rule ensures that the extensions of closed predicates contain only known individuals. Hence, it is a grounded model w.r.t. IC.

Theorem 3 (completeness). For an IC-KB (K, IC), if K has a grounded model w.r.t. IC, then the expansion rules can be applied to the initial graph G of (K, IC), and may lead to an completion graph G without inconsistency clash.

Proof Given a grounded model I of K w.r.t. IC, we can obtain a completion graph G by applying the completion rules to G: for every node x and y in the graph, $L(x) \subseteq \{ C \mid \rho(x) \in C^I \}$ and $L(x, y) \subseteq \{ R \mid (\rho(x), \rho(y)) \in R^I \}$ are satisfied, where ρ is mapping from nodes to $\triangle^I$. Since I is the grounded model of (K, IC) the completion graph does not contain inconsistency clash.

Theorem 4 Let (K, IC) be an IC-KB, then K has a grounded model w.r.t. IC if and only if it is GC-satisfiable.

Proof The "if" part of the proof is trivial. We prove the "only if" part. For any grounded model I, Let $|N_I|$ denote the sum of the cardinalities of all extensions of all the minimized predicates in IC. Note that, for any two grounded models I and J of K w.r.t. IC, we have $|N_J| < |N_I|$, whenever $J \prec_{\mathrm{IC}} I$. Hence for any grounded model I which is not a GC-model, there is a grounded model I_1 with $I_1 \prec_{\mathrm{IC}} I$. If I_1 is the GC-model, then we find the GC-model $W = I_1$, otherwise, we continue the process. After a finite number of steps the process may terminate and we find n grounded models I_i $(1 \leqslant i \leqslant n)$, such that $I_n \prec_{\mathrm{IC}} I_{n-1} \prec_{\mathrm{IC}} \ldots \prec_{\mathrm{IC}} I_1 \prec_{\mathrm{IC}} I$, but there is no other model I_{n+1} such that $I_{n+1} \prec_{\mathrm{IC}} I_n$. Since all the concepts and roles in K are closed, the number of individual names are finite, and there are at most n grounded models I_i that satisfy the $\prec_{\mathrm{IC}}$ relation. If one of the n grounded models belongs to the GC-models, then we find the model W, otherwise none of the n grounded models belongs to the GC-models. From the definition of GC-model of K w.r.t. IC, there exists a grounded model I_{n+1}, such that $I_{n+1} \prec_{\mathrm{IC}} I_n$, which is a contradiction.

4.2 Integrity constraint validation

In the following, we consider the satisfaction of integrity constraints for the extended knowledge base. According to the Definition 4, it is necessary to check whether, for every GC-model I of K w.r.t. IC, there is $C_1^I \subseteq C_2^I$ for an IC-axiom of the form $C_1 \sqsubseteq C_2$. That is, for every individual a, check whether exists $C_2(a)$ in K when the assertion $C_1(a)$ is in K at the same time. Thereafter, we only consider the individuals of concepts in the left side of IC-axioms.

In the standard reasoning, we can see that the concept C_1 is subsumed by another concept C_2 if and only if the concept $C_1 \sqcap \neg C_2$ is unsatisfiable; a knowledge base K entails an assertion $C(a)$, denoted by $K \models C(a)$ if and only if $K \cup \{\neg C(a)\}$ is not consistent. In this way, the IC validation can be reduced to GC-consistent checking of the integrity constraint knowledge bases. In the following, we prove the feasibility.

Theorem 5 Given an extended IC-KB (K, IC), K is an ALC knowledge base and IC is the set of IC-axioms in the form of $C_1 \sqsubseteq C_2$ where the predicates in C_1 and C_2 are G-predicates. $K \models_{\mathrm{IC}} C_1 \sqsubseteq C_2$ if and only if for every individual a in Ink, $K \cup \{(C_1 \sqcap \neg C_2)(a)\}$ is not GC-consistent.

Proof. We prove the "only if" part.
($\Rightarrow$): $K \models_{\mathrm{IC}} C_1 \sqsubseteq C_2$
$\Rightarrow$ for every $I \in \mathrm{GCM}(K), I \models C_1 \sqsubseteq C_2$
$\Rightarrow$ for every $I \in \mathrm{GCM}(K)$ and every individual a, if $a^I \in C_1^I$, then $a^I \in C_2^I$
$\Rightarrow I \models K$ and if $I \models C_1(a)$ then $I \models C_2(a)$
$\Rightarrow$ if $I \models K \cup \{C_1(a)\}$ then $I \models K \cup \{C_2(a)\}$
Assume $K \cup \{(C_1 \sqcap \neg C_2)(a)\}$ is GC-consistent, i.e., it has a GC-model.
$\Rightarrow$ There exists a GC-model I, such that $I \models K \cup \{(C_1 \sqcap \neg C_2)(a)\}$
$\Rightarrow I \models K \cup \{C_1(a)\}$ and $I \models K \cup \{\neg C_2(a)\}$, which yields a contradition.

Then we prove the "if" part.
($\Leftarrow$): Assume to the contrary, $K \not\models_{IC} C_1 \sqsubseteq C_2$
$\Rightarrow$ There exists a GC-model I, such that $I \not\models C_1 \sqsubseteq C_2$
$\Rightarrow$ There exist a GC-model I, and an individual a such that if $a^I \in C_1^I$, then $a^I \notin C_2^I$
$\Rightarrow$ There exist a GC-model I, and an individual a such that if $I \models C_1(a)$ then $I \not\models C_2(a)$
But according to the condition, $K \cup \{(C_1 \sqcap \neg C_2)(a)\}$ is not GC-consistent.
$\Rightarrow$ There exists no interpretation I, such that $I \models K \cup \{(C_1 \sqcap \neg C_2)(a)\}$
$\Rightarrow$ There exists no interpretation I, such that $I \models K \cup \{C_1(a)\}$ and $I \models K \cup \{(\neg C_2)(a)\}$, which yields a contradiction.

We give the IC-satisfaction algorithm in the following. The set of the individuals corresponding to nodes used in the following algorithm is denoted with Ink.

Algorithm IC-satisfaction algorithm

Input: a knowledge base K, an IC-axiom $\alpha \in$ IC

Output: return temp, where temp is true when K satisfies α, otherwise false

Initialize temporal variable temp = true.

Step 1. For every IC-axiom in the form of $C_1 \sqsubseteq C_2$ in IC, initialize the graph G by K(ABox) of the given Integrity constraint knowledge base (K, IC) as follows:

Step1.1 for every individual in K, if it appears at least once in the form of $C(a)$ or $R(a, b)$ in K, then create a node a, for every individual a, where C and R are predicates appears in α;

Step1.2 Add C to $L(a)$, for every assertion in the form of $C(a)$;

Step1.3 Add R to $L(a, b)$, for every assertion in the form of $R(a, b)$;

Step 2. For every IC-axiom α with the form of $C_1 \sqsubseteq C_2$ in IC, Check the satisfaction of α by GC-consistency algorithm.

Step2.1 Traverse all the nodes of G, find one node a which has not be traversed in G, if the concept in the left side of α is contained in $L(a)$, then go to Step 2.2, else to Step 2.3.

Step2.2 Expanding G for the node a by applying the extended rules in Table 2. If it does not contain clash, set temp=false and the individual a do not satisfy the constraint.

Step2.3 if the traversing does not end, then go to Step 2.1, else return temp.

4.3 Discussion

In this paper, we have restricted the extension of predicates in integrity constraint axioms and consider the standard reasoning in integrity constraint validation process. This process is under the local closed world assumption(LCWA): knowledge representation languages which have both OWA and CWA modeling features are said to adhere the LCWA [18]. There are also several work for extending DL knowledge bases to model this semantics. Such as non-monotonic rules from logic programming [20] or restrictions for the extensions of predicates in DL knowledge base [17,18]. However, these are different from our work discussed above. Most focus on the combination of CWA and OWA, and the standard reasoning tasks in the extended knowledge bases. In our work, we apply grounded circumscription to define the semantics and satisfaction of the integrity constraints and focus on integrity constraint validation tasks under CWA.

Now, we analyze the integrity constraint validation in three aspects and compare our approach with existing work by the examples presented in Section 2. It is easy to verify that our modeling of integrity constraints provides expected results for these examples.

• Semantics of integrity constraints

The integrity constraint validation should be under the closed world assumption. In this paper, we restrict the extensions of the predicates in integrity constraint axioms that only contain the individuals appear in DL knowledge base. In this way, the integrity constraints are under the CWA. Taking the integrity constraint Eq. (1) for example, the GC-model of K is {Student(Peter)} and does not satisfy Eq. (1), then Eq. (1) is violated.

• Definition of integrity constraint satisfaction

From the analysis in the above section, we know that the definition of integrity constraint satisfaction should be dependent on the integrity constraints. In this paper, we apply grounded circumscription to get the GC-model and define satisfaction of the integrity constraints. For the integrity constraint Eq. (8), since there are no assertions specify that the individual Mary is the instance of Professor in GC-model of K, the GC-consistency algorithm does not apply to the individual Mary, and therefore it is satisfied.

• Standard reasoning in integrity constraint validation

In the integrity constraint validation, we should consider standard reasoning results, which is distinguished feature of DLs. That is, the implicit information in the knowledge base should be inferred by standard reasoning and used in integrity constraint validation. In this paper, we convert integrity constraint validation into GC-consistency checking by modifying the expansion rules. In this way, the implicit information is also considered in integrity constraint validation. For integrity constraint Eq. (3), it is converted into Student $\sqcap$ $\forall$takesCourse.$\neg$Course. Then for the individual John, we check the GC-consistency of extended knowledge base $K \cup$ {(Student $\sqcap$ $\forall$takesCourse.$\neg$Course)(John)}. We can infer L(John) = {Student, $\forall$takesCourse.$\neg$Course} and L(English) = {GraduateCourse, Course, $\neg$Course}, which both have clashes, then Eq. (3) is satisfied.

Our work is similar to the Motik's work [13] in that we also use the minimal models of knowledge bases to define the satisfaction of integrity constraints. However, the approach in [13] is more complex than ours. Using Motik's technique there is a need to translate all the axioms (standard axioms and IC-axioms) in the DL knowledge base into the first-order formulas in logic programming. Then use the reasoning tools for the logic programming, which may complicate the calculus. Since we only modify certain expansion rules in the existing tableau algorithm and use the existing reasoning tools in this paper, it is simpler than converting the DL knowledge base into other logics. For the traditional axioms with existential quantifiers in [13] under positive polarity and universal quantifiers under negative polarity, the skolemization may contain functions and the corresponding minimal model may be exponential in size. When the standard TBox axioms with existential qualification are referred to the integrity constraints, the complexity of integrity constraint validation is determined by the complexity of SkS which is determined by the number of quantified alternations. Whereas, the grounded circumscription is added to restrict that the individuals must occur in K and prohibit the generation of new nodes when applying the expansion rules in the tableau algorithm. Since there is no new node generated, the modified tableau algorithm is simpler than the traditional tableau algorithm. Thus, the upper complexity of the algorithm used in above section is no more than the standard reasoning for DL ALC.

In conclusion, we believe that the modeling of integrity constraints based on the grounded circumscription is more appropriate to meet user requirements. Furthermore, compared with the existing method, the integrity constraint validation method presented in this paper is much simpler than others which are also based on the model-theory definitions.

5 Conclusion

Motivated by analyzing existing work for extending the DL knowledge base with integrity constraints, we apply grounded circumscription to define the semantics and satisfaction of the integrity constraints which can capture the meaning of integrity constraints in OWL ontologies. Then, according to the semantics of integrity constraints, we have modified the expansion rules in tableau algorithm to check the satisfaction of integrity constraints in OWL ontologies. Finally, we have compared our work with existing work. We find that our modeling of integrity constraints not only obtains more intuitive results but also is appropriate to the simpler integrity constraint validation methods in certain cases. Further work needs to be done in the IC-satisfaction Algorithm to accommodate description logics which are more expressive than ALC.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 61133011, 41172294, 61003101, 61170092); Jilin Province Science and Technology Development Plan (20101501, 20100185, 201101039); Doctoral Fund of Ministry of Education of China (20100061110031); Opening Fund of Top Key Discipline of Computer Software and Theory in Zhejiang Provincial Colleges at Zhejiang Normal University (ZSDZZZZXK12); Jilin University Doctoral Interdisciplinary Research Funded Project (2012JC012).

References

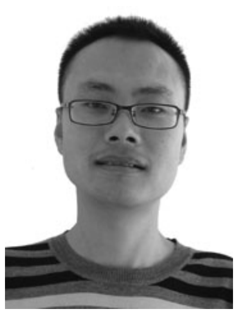
1. Antoniou G, Van Harmelen F. In: Web ontology language: OWL. Springer, 2004, 67–92
2. Baader F, McGuinness D L, Nardi D. The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2007
3. Kremen P, Kouba Z. Ontology-driven information system design. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications

and Reviews, 2012, 42(3): 334–344

4. Khalid M A, Edgar H S. Translating relational & object-relational database models into OWL models. In: Proceedings of the 2009 IEEE International Conference on Information Reuse & Integration. 2009, 336–341
5. Kowalski R. Logic for Data Description. Springer, 1978
6. Eiter T, Lukasiewicz T, Schindlauer R, Tompits H. Combining answer set programming with description logics for the semantic web. Artificial Intelligence, 2008, 172(12): 1495–1539
7. Motik B, Rosati R. Reconciling description logics and rules. Journal of the ACM, 2010, 57(5): 1–62
8. Reiter R. On integrity constraints. In: Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge. 1988, 97–111
9. Levesque H. All I know: a study in autoepistemic logic. Artificial Intelligence, 1990, 42(2): 263–310
10. Donini F, Lenzerini M, Nardi D, Schaerf A, Nutt W. An epistemic operator for description logics. Artificial Intelligence, 1998, 100(1): 225–274
11. Donini F, Bari P, Nardi D. Description logics of minimal knowledge and negation as failure. ACM Transactions on Computational Logic (TOCL), 2002, 3(2): 177–225
12. Tao J, Sirin E, Bao J, McGuinness D. Integrity constraints in OWL. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence. 2010
13. Motik B, Horroks I, Sattler U I. Bridging the gap between owl and relational databases. Web Semantics: Science, Services and Agents on the World Wide Web, 2009, 7(2): 74–89
14. Cui X J, Ouyang D T, Ye Y X, Wang H Y. Integrity maintenance of continually changed OWL ontology. Journal of Computational Information Systems, 2012, 8(7): 2931–2939
15. Ouyang D T, Cui X J, Ye Y X. Mapping integrity constraint ontology to relational databases. The Journal of China Universities of Posts and Telecommunications, 2010, 17(6): 113–121
16. Cui X J, Ouyang D T, Ye Y X, Wang X L. Translation of Sparql to SQL based on integrity constraint. Journal of Computational Information Systems, 2011, 7(2): 394–402
17. McCarthy J. Circumscriptionała form of non-monotonic reasoning. Artificial Intelligence, 1980, 13(1): 27–39
18. Sengupta K, Krisnadhi A, Hitzler P. Local closed world semantics: grounded circumscription for OWL. In: Proceedings of the 10th International Conference on the Semantic Web. 2011(1): 617–632
19. Horrocks I, Patel-Schneider P F, Harmelen F. From SHIQ and RDF to OWL: the making of a web ontology language. Web Semantics: Science, Services and Agents on the World Wide Web, 2003, 1(1): 7–26
20. Lifschitz V. Foundations of logic programming. Principles of Knowledge Representation, 1996, 3: 69–127

Dantong Ouyang is a PhD supervisor. She is a member of the Subject Appraisement Group of the Degree-Assignment Committee of the National State Department and the member of Theoretical Computer Science Committee of China Computer Federation (CCF), the vice dean of the CCF Artificial Intelligence and Pattern Recognition Committee. Her research interests are in the areas of logic in the semantic web and automated reasoning.

Xianji Cui is currently a PhD student in the College of Computer Science and Technology in Jilin University, China, where she received her BE in Computer Software and Theory in 2011. Her research interests are in the areas of ontology engineering and semantic web technologies.



Yuxin Ye is currently a lecturer with the College of Computer Science and Technology in Jilin University, China. He received his PhD in Computer Software and Theory in 2010. His research interests are in the area of reasoning in the semantic web.