

A Tableau Decision Algorithm for Description Logic CATSbqr

Haifeng Liu^{1, a}

¹ School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

^a810230118@qq.com

Keywords: description logic, formalization method, automatic reasoning.

Abstract. The graphical representation of the UML class diagram does not have a clear semantics, so that it is unable to auto detect the semantic errors in the model. To solve this problems, somebody provides a tools called description logic CATSbqr, which can transports the class diagram to CATSbqr knowledge base, so it can use the reason mechanism of knowledge base to detect the corrective of the model. But CATSbqr lacks reasoning algorithm for satisfiability and subsumption reasoning tasks at present. Based on this, in this paper a kind of description logic tableaux based satisfiability reasoning algorithm for CATSbqr is presented, after that, we used the RacerPro reasoned to reasoning the knowledge base which transformation form DLs CATSbqr. The theoretical foundation for the implementation of automatic reasoning of satisfiability, redundancy, and subsumption relation of UML composition association is presented through description logic CATSbqr tableaux reasoning algorithms.

Introduction

Description logic[1] is a formal representation of knowledge, which is also called terminology logic. It inherits the main idea of KL-ONE, which is a set of the first order predicate logic. Description logic is a formal language of knowledge representation, which is suitable for the representation of knowledge about concept and concept hierarchy. A description logic system usually consists of four basic components: a set of representation concepts and relations; TBOX (including assertions); TBOX (instance assertions) and reasoning mechanisms on TBOX and ABOX. The most important feature of the description logic is the ability to express and determine, it can ensure that the inference algorithm can always stop, and can return the correct results. Description logics have clear model theoretic mechanism, through the description of the logical classification of concept hierarchy to will interpret the of description logic knowledge base, thus the use of description logic decidable reasoning service in concept hierarchy, the reasoning of concept hierarchy.

The description logic in many fields has a good application, especially many scholars have conducted in-depth research on the mode of the database[2]. For example, Calvanes Diego put forward the importance of the formal UML class diagram, and the use of DLRifd and UML on the ALCQI class diagram of the knowledge representation. Maurizio Lenzerini et al compared framework said the system and description logic system in semantic data model and oriented object data model of knowledge representation differences, and focus on the how the two models into description logic knowledge base, and use of description logic reasoning mechanism for reasoning. In research on formalization of the UML model, the semantic of aggregation association is not clear, so that the case toll can not generate the correct data access layer code. In order to solve this problem, Huan et al proposed description logic CATSbqr, which transports the aggregation association model to description logic CATSbqr, which can use the automatic mechanism of knowledge base to reason the model, she proposed the description logic CATSbpr, but not proposed how to reason it. This paper starts with the description logic reasoning mechanism, puts forward a kind of tableau CATSbqr reasoning algorithm, and its application in the RacerPro reasoner, then realization of the description logic CATSbqr reasoning. So that it can solve the problems such as collision detection, semantic error detection, and so on.

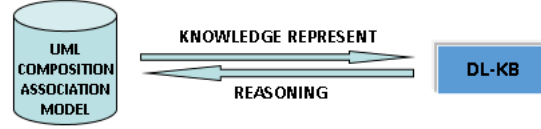


Fig. 1 The function of description logic systems

Description Logic CATSbqr

The syntax and semantics of CATSbqr. The name CATSbqr is a combination of the statics and actions represented [7] by the construction operator. The syntax and semantics of the CATSbqr language that represent the static structure of the data are similar to the CATS language[6]. In traditional description logic, the relation is binary association, but the CATS can describe multiple association. The basic elements of the description logic CATSbqr can represent the static structure like concepts, relations, and individuals, which are also called associations or attributes. The difference between CATSbqr and the traditional description logic lies in that the concept of CATSbqr is not only an abstraction of the individual set, but also an abstraction of a collection of instances of aggregated classes. There are two types of aggregate instances of the class: attribute aggregation and instance aggregation. Property aggregates abstract a complex object that is aggregated by different types of objects, for example, the date is a complex object that is gathered from the objects of the year, month, and day. An instance of a complex object, such as a basketball team, which is composed of a basketball team, is a complex object, which is abstracted from a particular class of objects.

The syntax and semantics of the description logic CATSbqr concept and the link constructors are shown in Table 1, where A is the atomic concept name, C and D as the basic concept names, and R and S are the basic relations. The atomic concepts and basic concepts are mapped to individual sets. The complex concept construction operator $\{E\}$ represents an instance of aggregation, $[r_1, \dots, r_k]$ and $\langle E|r_1, \dots, r_k \rangle$ represents the aggregation of attributes.

Table 1 The syntax and semantics of concepts and relations constructed operators

Name	Syntax	FOL-Semantics	Name	Syntax	FOL-Semantics
Atomic-Concept	A	$A^I \subseteq \Delta^I$	Key-tuple	$\langle C r_1, \dots, r_k \rangle$	$\subseteq [r_1, \dots, r_k]^I \cap C^I$
Top	\top	Δ^I	Atomic-link	r	$r^I \subseteq \Delta^I \times \Delta^I$
Bottom	\perp	\emptyset	Link negation	$r-$	$\{(o, o') (o, o') \in r^I\}$
Atomic negation	$\neg C$	$\Delta^I \setminus C^I$	Member	\exists	$\{(\{ \dots, o, \dots \}, o) \in \mathcal{O}^I \times \mathcal{O}^I\}$
Conjunction	$C \sqcap D$	$C^I \cap D^I$	Union	$r \sqcup s$	$r^I \cup s^I$
Universal quantif.	$\forall r.C$	$\{o \forall o' : (o, o') \in r^I \rightarrow o' \in C^I\}$	Concatenation	$r \circ s$	$\{(o, o'') \exists o' : (o, o') \in r^I \wedge (o', o'') \in s^I\}$
Existential quantif.	$\exists r.C$	$\{o \exists o' : (o, o') \in r^I \wedge o' \in C^I\}$	Difference	$r \setminus s$	$r^I \setminus s^I$
Qualified number restrictions	$\geq nr.C$ $\leq nr.C$	$\{o \#\{o' (o, o') \in r^I \wedge o' \in C^I\} \geq n\}$ $\{o \#\{o' (o, o') \in r^I \wedge o' \in C^I\} \leq n\}$	Transitive closure	r^*	$(r^I)^*$
Set	$\{C\}$	$\{\{o_1, \dots, o_k\} \in \mathcal{O}^I o_1, \dots, o_k \in C^I\}$	Identity	$\text{id}(C)$	$\{(o, o) o \in C^I\}$
Tuple	$[r_1, \dots, r_k]$	$\{[r_1 : o_1, \dots, r_k : o_k] \in \mathcal{O}^I o_1, \dots, o_k \in \mathcal{O}^I\}$			

Decision procedure for CATSbqr. The description logic CATSbqr can reduce to CQI by reify tuples, sets and key tuples, so that all the objects are elements objects.

Let K be a CATSbqr TBOX, We call the counterpart TBOX $T(K)$ is the TBOX transformed, t is the mapping from CATSbqr TBOX to CQI TBOX. We define the TBOX $T(K)$ in three parts, $\text{set}(K)$, $\text{tuple}(K)$ and $\text{key-tuple}(K)$.

$$\begin{aligned}\text{set}(K) &= \{t(C) \cap (\leq 1 \text{memb}.T) \subseteq \forall \text{memb}.(\leq 1 \text{memb}^-.(\leq 1 \text{memb}.T))\}; \\ \text{tuple}(K) &= \{[r_1, \dots, r_k] \equiv t(\exists r_1.T \cap (\leq 1 r_1.T)) \cap \dots \cap t(\exists r_k.T \cap (\leq 1 r_k.T))\}; \\ \text{key-tuple}(K) &= \{ \langle C | r_1, \dots, r_k \rangle \subseteq [r_1, \dots, r_k] \cap t(C) \}\end{aligned}$$

When we use the transformation rules above, we can transport the CATSbqr complex constructor to CQI. Then we can use the tableau algorithm of CQI to reason description logic CATSbqr.

The important characteristic of the description logic is the reasoning service which can be decision, it can reason implicit knowledge from known knowledge. The semantics of knowledge base is equivalence with the axiom set of first-order logic. The DLs provides different kinds of reasoning services, the first is to discuss the concept of reasoning, then the TBOX and ABOX reasoning. Finally, these inference can boil down to a major problem of reasoning, that is the ABOX consistency checking.

The basic reasoning task of TBOX is to verify the inclusion of the concept, to construct the hierarchical structure of terminology by including the relationship between the concepts, which can make the reasoning more easily. The reason method of description logic CATSbqr is two steps, firstly, transport the CATSbqr to CQI by transformation rules, then use the tableau decision algorithms of CQI to detect the ABOX consistency.

The tableau decision Algorithms of CATSbqr

The tableau algorithm is proposed by Schmidt-Schaub and Smolka order to satisfy the concept of ALC. Since then, this method has been widely used in many of the description logics from the extended ALC to obtain a reliable and complete algorithm.

The tableau algorithm[5] is also called table deduction algorithm, widely used in the field of automatic reasoning, its algorithm idea is to construct a complete tree, the tree node represents the knowledge base concept set, tree branches represents the relationship between concepts.

The program flow of tableau algorithms is below:

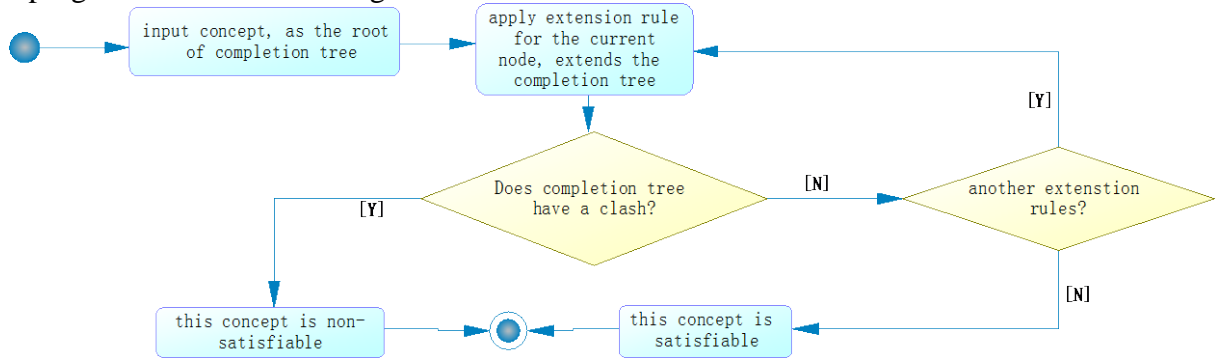


Fig. 2 The tableau algorithm program flow diagram

Assume the algorithm starts with a completion tree consisting of a single node x with $L(x) = \{D\}$. It applies the expansion rules, stopping when a clash occurs. The extension rules of CQI is :

Table 2 The extension rules of CQI

Rules	Extension
\cap -rule	If 1. $C_1 \cap C_2 \in L(x)$ and 2. $\{C_1, C_2\} \not\subseteq L(x)$ then $L(x) \rightarrow L(x) \cup \{C_1, C_2\}$
\cup -rule	If 1. $C_1 \cup C_2 \in L(x)$ and 2. $\{C_1, C_2\} \cap L(x) = \emptyset$ then $L(x) \rightarrow L(x) \cup \{C\}$ for some $C \in \{C_1, C_2\}$
Choose-rule	If 1. $(\bowtie n R C) \in L(x)$ and 2. there is an R-predecessor y of x with $\{C, \sim C\} \cap L(x) = \emptyset$ Then $L(y) \rightarrow L(y) \cup \{E\}$ for some $E \in \{C, \sim C\}$ and delete all descendants of y
\geq -rule	If 1. $(\geq n R C) \in L(x)$, x is not blocked and no non-generating rule is applicable to x or any of its ancestors, and 2. $\#R^T(x, C) < n$ then create a new node y with $L(\langle x, y \rangle) = R$ and $L(y) = B(y) = \{C, E_1, \dots, E_n\}$ where $\{D_1, \dots, D_n\} = \{D \mid (\bowtie n R D) \in L(x)\}$ and $E_i \in \{D_i, \sim D_i\}$

Using RacPro on CATSbqr

The RacPro[3] was a reasoner on ontologies and description logics. It was developed based on tableau algorithm, it supported description logic $ALCNH_{R+}(D)^-$. The RacPro was the first system which efficiently supported concrete domains for TBOX and ABOX reasoning. RacPro used the client / server model, supported TCP/IP protocol, you could put the RacPro as a server, and it gives the api for file, socket and the HTTP, UI interface can be used RacPorter, which can input description logic and ontology by an interactive way. It uses nRQL language to query subsumption, satisfiability and consistency between concepts, and the racpro server return a query answer to racerporter to show whether the concepts is subsumption, satisfiability and consistency.

If we want to input a knowledge base on RacPro, we should use the language of the RacPro, the correspondence between RacPro language and the description logic constructor is as follows:

Table 3 The correspondence between RacPro and DL constructor

DL constructor	RacPro language	DL constructor	RacPro language
C	Atomic-concept (C)	$\geq nr.C$ $\leq nr.C$	(at-least n R) (at-most n R)
R	Atomic-concept (C)	$c1 \sqsubseteq c2$	(implies c1 c2)
C(a)	(instance a C)	$c1 \equiv c2$	(equivalent c1 c2)
$\forall r.C$	(ALL R C)	$R1 \equiv R2$	(equivalent R1 R2)
$\exists r.C$	(some R C)		

To use RacPro to reason CATSbqr, first use the transformation rules on CATSbqr, then use RacPro language to input CQI knowledge base, input nRQL to query subsumption, satisfiability and consistency, etc, if RacPro return T, the concept is to be satisfied, if return F, the concept is not satisfied.

Summary

The work described in this paper provides a reason method for description logic CATSbqr, first we use description logic CATSbqr to describe UML composition association model, then we use the transformation rule to transport the CATSbqr constructors to CQI, finally, we use the tableau algorithm of CQI by RacPro to reason CATSbqr. This formal method can reduce the software designer to check the semantic correctness of the model, so it improve the efficiency and reliability of the design.

References

- [1] Calvanese D, Giacomo GD, Nardi D, Lenzerini M. reasoning expressive description logics[M]. In: Robinson A, Voronkov A, eds. Handbook of Automated Reasoning. Milan: Elsevier Science Publishers, (2001). 1581–1634.
- [2] Calvanese D, Lenzerini M, Nardi D. Unifying class-based representation formalisms[J]. arXiv preprint arXiv: (2011).1105.5452.
- [3] Haarslev V, Hidde K, Ller R, et al. The racpro knowledge representation and reasoning system[J]. Semantic Web, (2012), 3(3):267-277. Calvanese D
- [4] Calvanese D, Ortiz M, Simkus M. Evolving Graph Databases under Description Logic Constraints[C]//Description Logics. (2013): 120-131
- [5] Lenzerini M. TBox and ABox Reasoning in Expressive Description Logics[J]. In Proc. of KR-96, (1996):316--327.
- [6] De Giacomo G, Lenzerini M. What's in an aggregate: Foundations for description logics with tuples and sets[J]. International Joint Conference on Artificial Intelligence. Lawrence Erlbaum Associates Ltd, (1995), 14: 801-807.