第 38 卷　第 5 期　电 子 科 技 大 学 学 报　Vol.38　No.5
2009年9月　Journal of University of Electronic Science and Technology of China　Sep. 2009

Quasi-Classical Description Logics and Paraconsistent Tableau Calculus for Reasoning with Acyclic TBox

WU Jin-zhao[1], HOU Hui[2], and LI Shao-rong[1]

(1. School of Optoelectronic Information, University of Electronic Science and Technology of China　Chengdu　610054

2. Chengdu Institute of Computer Applications, Chinese Academy of Sciences　Chengdu　610041)

Abstract　The forthcoming semantic Web evolving from the current World Wide Web is designed to define the semantics of information and services on the web, thereby endowing the web with intelligence to automatically reason about the web contents. Description logics (DLs) play a substantial role in the semantic Web, since they underlie the W3C-recommended Web ontology language (OWL), which is derived from ontology research in artificial intelligence (AI) in order to achieve the goal of the semantic Web. However, the knowledge and data in the Semantic Web are large-scale, dispersive, multi-authored, and therefore usually inconsistent. It is reasonable and imperative to develop practical reasoning techniques for inconsistent ontologies. This paper proposes a new type of paraconsistent description logics based on Hunter's quasi-classical logic (QCL), which are termed as quasi-classical description logics (QCDLs). QCDLs avoid logical explosion. A semantic tableau calculus is constructed in QCDLs for the reasoning on the knowledge bases with acyclic TBox. Furthermore, a sound, complete and decidable consequence relation based on the calculus is defined. These enable a complete framework for paraconsistent reasoning in the Semantic Web. A comparison with other key paraconsistent description logics is also given. It is shown that QCDLs possess more expressive semantics and stronger reasoning capability, and that its connectives behave classically at the object level.

Key words　acyclic TBox;　paraconsistent reasoning;　QCDLs;　semantic Web;　tableau calculus

非周期TBox框架下的推理——拟经典描述逻辑与超协调表演算

吴尽昭[1]，侯　晖[2]，李绍荣[1]

(1. 电子科技大学光电信息学院　成都　610054;　2. 中国科学院成都计算机应用研究所　成都　610041)

【摘要】源自当今互联网的语义网研究的目的是定义信息语义和网络服务，因此需要赋予网络智能以便能够自动对网络内容进行推理。各种描述逻辑(DLs)在语义网的研究中扮演着重要角色，构成了W3C推荐的网络本体语言(OWL)的基础，而OWL源于为达到语义网目标的人工智能(AI)本体论研究。语义网的知识和数据量巨大、分散、来源众多且因此通常不具有协调性。因此，必须开发针对非协调本体的实用的推理技术。该文基于Hunter的拟经典逻辑(QCL)，构造了新型超协调拟经典描述逻辑(QCDLs)，避免了逻辑爆炸问题，同时，针对基于非周期TBox的知识库推理问题，建立了QCDLs语义表演算，进而定义了一种可靠、完备且可判定的推理关系，从而构建了完整的语义网推理框架。与其他重要的超协调描述逻辑进行了比较，结果表明QCDLs具有更强的表达语义和推理能力，并且其相关行为在目标层次上表现出经典性。

关　键　词　非周期TBox;　超协调推理;　QCDLs;　语义网;　表演算

中图分类号　TN959.3　　**文献标识码**　A　　doi:10.3969/j.issn.1001-0548.2009.05.025

1　Introduction

The forthcoming semantic Web evolving from the current World Wide Web is designed to define the semantics of information and services on the web, thereby endowing the web with intelligence to automatically reason about the web contents[1]. Description Logics play a substantial role in the Semantic Web since they underlie the W3C-recommended Web ontology language, which is derived from ontology research in artificial intelligence in order to achieve the goal of the semantic Web[2-3].

Received date: 2009 – 08 – 08
收稿日期：2009 – 08 – 08
Foundation item: 863 Program(2007AA01Z143)
基金项目：国家863计划(2007AA01Z143)
Biography: WU Jin-zhao, born in 1965, male, professor, research interests include verification and evaluation of complex systems.
作者简介：吴尽昭(1965 –)，男，博士，教授，主要从事复杂系统验证与评估技术方面的研究.

However, a difficult dilemma is faced. It is rarely possible in practice to obtain consistent knowledge bases in a Semantic Web environment, since the knowledge and data are usually large-scale, dispersive and multi-authored. But classical logic is explosive when inconsistent knowledge is involved. So are description logics, because description logics are structured fragments of classical first-order logic[2]. Hence, it is necessary and significant to explore the ways of dealing with inconsistent knowledge in the Semantic Web.

Inconsistency handling is a well-known topic in AI. In tradition, there are two fundamentally different approaches to dealing with inconsistency, which are usually based on two different points of view on inconsistency. The first approach considers inconsistent data to be abnormal and incorrect knowledge, which should be eliminated or repaired in order to obtain a consistent knowledge base[4-5]. But this approach is usually unpractical and unfeasible in the Semantic Web environment. On the contrary, in the second approach, inconsistency is believed to be natural and reasonable, and therefore should be tolerated[6-12]. Because of the properties of knowledge in the Semantic Web mentioned above, it is usually difficult to determine which is true and which is false when inconsistency is presented. The second approach is therefore more suitable for the semantic Web applications. Many kinds of inconsistency-tolerated logics, usually called paraconsistent logics where ex contradictione ouodlibet (ECQ) is intrinsically avoided, have been proposed in AI. The two very useful and interesting types of such logics are Belnap's Four-Valued Logic[13] and Hunter's quasi-classical logic (QCL)[6,14-15]. Four-valued logic has been transferred to DLs to propose so-called four-valued description logics (FVDLs)[6,9,11], in which the additional truth values standing for undefined and overdefined are employed. However, FVDLs are too weak to infer enough conclusions which are inferable in the classical case. QCL grounds on different idea from Belnap's four-valued logic, where the order of applying compositional proof rules and decompositional proof rules is restricted. QCL is more expressive in semantics and stronger in reasoning capability than four-valued logic methodology. Moreover, its connectives behave in a "classical manner" at object level so that important proof rules such as modus tollens, modus ponens, and disjunctive syllogism hold again[14-17].

In this paper, we apply the methodology of QCL into DLs and propose a paraconsistent description logic, called quasi-classical description logics (QCDLs), in order that the useful and non-trivial conclusions might be inferred from inconsistent knowledge. QCDLs inherit merits from QCL and preserve more expressive and stronger semantics. Furthermore, a semantic tableau calculus for the reasoning problems on the knowledge bases with acyclic TBox is constructed. Based on the tableau calculus, a sound, complete and decidable consequence relation is defined. The description logic ALC is considered in chief because it is regarded as the most foundational one and covers the core of OWL-DL[3].

The rest of this paper is organized as follows: Section 2 provides a brief refresher on description logics, and analyzes the reason why ECQ happens in DLs. The formal semantics of QCDLs is defined in Section 3. In Section 4, the semantic tableau for QCDLs is presented and the properties of QCDLs are analyzed. In Section 5, we give a comparison with other key paraconsistent reasoning techniques. Section 6 concludes the paper.

2 Preliminaries

In this section, a brief review for the DL ALC is provided, and the reason of ECQ in DLs is analyzed.

2.1 Description Logic ALC

The reader is assumed to be familiar with description logics. Please refer to reference [2] if detailed background is needed.

Definition 1 The alphabet for ALC language is constructed as follows:

(1) A set of unary predicate symbols N_C denoting concept names;

(2) A set of binary predicate symbols N_R denoting role names;

(3) A set of constant symbols N_I denoting

individual names, in which there is assumedly at least one element;

(4) A set of logical symbols including connectives, quantifiers and auxiliary symbols.

ALC concepts are inductively defined as follows.

Definition 2 The smallest set $\mathcal{C}$ satisfying the following conditions constitutes ALC-concepts:

(1) The top concept $\top$, the bottom concept $\bot$ and every concept name $A \in N_{\mathrm{C}}$ are ALC-concepts;

(2) If C and D are ALC-concepts and $R \in N_{\mathrm{R}}$, then $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall R.C$ and $\exists R.C$ are ALC-concepts.

The formal semantics of $\mathcal{ALC}$ is defined in terms of an interpretation $\mathcal{I} = (\vartriangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$. The domain $\vartriangle^{\mathcal{I}}$ is a non-empty set of individuals and the interpretation function $\cdot^{\mathcal{I}}$ satisfies the definition in Table 1.

Table 1 Syntax and semantics of ALC

Name	Syntax	Semantics
Concept name	A	$A^I \subseteq \vartriangle^I$
Role name	R	$R^I \subseteq \vartriangle^I \times \vartriangle^I$
Individual name	a	$a^I \in \vartriangle^I$
Top concept	T	$\vartriangle^I$
Bottom concept	$\bot$	$\varnothing$
Conjunction	$C \sqcap D$	$C^I \cap D^I$
Disjunction	$C \sqcup D$	$C^I \cup D^I$
Negation	$\neg C$	$\vartriangle^I \backslash C^I$
Exists restriction	$\exists R.C$	$\{x \mid \exists y.(x,y) \in R^I$ and $y \in C^I\}$
Value restriction	$\forall R.C$	$\{x \mid \forall y.(x,y) \in R^I$ implies $y \in C^I\}$
Concept inclusion	$C \sqsubseteq D$	$C^I \subseteq D^I$
Concept assertion	$C(a)$	$a^I \in C^I$
Role assertion	$R(a,b)$	$(a^I, b^I) \in R^I$

An ALC knowledge base (or ontology) involves a set of terminological axioms, called TBox, and a set of assertional axioms, called ABox. The most general form of terminological axioms is general concept inclusion (GCI), which is of the form $C \sqsubseteq D$, where both C and D are concepts. It means that every individual of C is also the individual of D. An concept inclusion, which is of the form $A \sqsubseteq C$ where $A \in N_{\mathrm{C}}$ and C is any concept, is called *specialization.* A *definition* axiom of TBox, which is of the form $A \equiv C$ where $A \in N_{\mathrm{C}}$, can be viewed as an abbreviation of $A \sqsubseteq C$ and $C \sqsubseteq A$. Assertional axiom is of the form $C(a)$ or $R(a,b)$ where $C \in \mathcal{C}$, $R \in N_{\mathrm{R}}$ and $a, b \in N_{\mathrm{I}}$. An assertion $C(a)$ means that the individual a is an instance of concept C, and an assertion $R(a,b)$ means that there is a relationship R between the individuals a and b. The semantics of terminological and assertional axioms is also shown in Table 1. An interpretation I satisfies a knowledge base $\varSigma$, that is, I is a model of $\varSigma$, iff it satisfies each axiom in the TBox and the ABox of $\varSigma$.

In this paper, we focus on some special forms of TBox. A finite set of definitions $\mathcal{T}$ is called terminology if for every atomic concept $A \in N_{\mathrm{C}}$ there is at most one axiom in $\mathcal{T}$ whose left-hand side is A. A finite set of definitions and specializations T is called generalized terminology if for every atomic concept $A \in N_{\mathrm{C}}$ there is at most one axiom in $\mathcal{T}$ whose left-hand side is A. Let $A, B \in N_{\mathrm{C}}$ be concept names occurring in a terminology (or generalized terminology) $\mathcal{T}$. We say that A directly uses B in $\mathcal{T}$ if B appears on the right-hand side of the definition (or specialization) of A, and we call *uses* the transitive closure of the relation directly uses. Then $\mathcal{T}$ contains a cycle iff there exists an atomic concept $A \in N_{\mathrm{C}}$ in $\mathcal{T}$ that uses itself. Otherwise, $\mathcal{T}$ is called acyclic.

The purpose of a knowledge representation system is not only to represent and store existed knowledge but also to infer implicit knowledge. Various reasoning tasks are considered for DLs. Since the topic of our work is to discuss how to derive non-trivial inferences from inconsistent knowledge, it is sufficient to consider concept subsumption and instance checking:

(1) Concept Subsumption: a concept C is subsumed by another concept D w.r.t. a knowledge base $\varSigma$, written $\varSigma \vDash C \sqsubseteq D$, if $C^I \subseteq D^I$ holds for every model I of $\varSigma$.

(2) Instance Checking: an individual a is an instance of a concept C w.r.t. a knowledge base Σ, written $\varSigma \vDash C(a)$, if $a^I \in C^I$ holds for every model I of $\varSigma$. A pair of individuals (a,b) is an instance of a role R w.r.t. a knowledge base $\varSigma$, written

$\Sigma \models R(a,b)$, if $(a^I, b^I) \in R^I$ holds for every model I of Σ.

2.2 ECQ in Description Logics

Classical description logics preserve ECQ because description logics are structured fragments of classical first-order logic[2]. Any concept inclusion and any assertion are consequences of an inconsistent knowledge base in classical semantics and classical reasoning techniques of DLs. Let us consider an example.

Exmaple 1 Consider the following ALC-knowledge base $\Sigma = <\mathcal{T}, \mathcal{A}>$:

$$\mathcal{T} = \{\text{Driver} \sqsubseteq \text{Person} \sqcap \exists\text{PROFILE}$$
$$\text{DrivingLicense} \sqcap \exists\text{CONTEXT.Position}$$
$$\text{Traveler} \sqsubseteq (\text{Driver} \sqcup \text{Passenger}) \sqcap \exists\text{TO.Position}\}$$
$$\mathcal{A} = \{\text{Traveler(jack)}, \neg\text{Passenger(jack)},$$
$$\neg\exists\text{PROFILE.DrivingLicense(jack)}\}$$

This simplified knowledge base describes a possible situation of an urban traffic system. In this knowledge base, a driver is portryed as a person who has the driving license as his/her profile, the current position information as his/her dynamic context. A Traveler is designed as a driver or passenger who wants to go to some position. Jack was originally described as a traveler and not a passenger by engineers. However, an assertion describing that jack no longer has certified driving license is added into the knowledge base after some events have happened. This assertion may be added by importing knowledge from other sources such as police, or by another maintainer so that the knowledge base becomes inconsistent.

Users will find that what they obtain are totally meaningless answers when they raise a query to this knowledge base.

In model-theoretic view, a concept inclusion query or an assertion query is a semantic consequence of a knowledge base iff every model of the knowledge base is a model of the query. But it is obvious that no model can satisfy the inconsistent knowledge base Σ by the definition of classical semantics of ALC. In other words, there is no model of Σ that is not a model of the query. Vacuously, every model of Σ is a model of the query. So, any query is a semantic consequence of Σ.

In proof-theoretic view, it is a corresponding case in any complete reasoning approach. Consider semantic tableau calculus, the classical tableau for Σ and an arbitrary query is closed since the expanding process always encounter a classical clash, i.e., $\text{DrivingLicense}(i)$ and $\neg\text{DrivingLicense}(i)$ where i is an individual name generated in terms of the expension rule for $\exists$. It means the query is a logical consequence of Σ.

3 Quasi-Classical Description Logics

In this section, the semantics of the quasi-classical DL $\mathrm{QC\text{-}ALC}$ is defined. There are a number of challenges to transplant quasi-classical semantics from QCL into DLs, because of differences between them. Syntactically, for example, DLs have no function and explicit free variable which in fact implicitly exists in concept. Semantically, the descriptive semantics of DLs is dissimilar to the Herbrand interpretation of QCL.

We now consider the essential ideas behind QCDLs. Firstly, QCDLs separate the mutually exclusive relation between a formula and its negation, like some paraconsistent logics. This can achieve paraconsistency in the aspect of model theory. Secondly, applying of decompositional rules in proofs of QCDLs is forbidden after compositional rules have been applied. This can make ECQ be avoided in the aspect of proof theory. Thirdly, QCDLs preserve resolution to constitute the basis of useful paraconsistent reasoning. Hence, QCDLs possess more expressive and stronger semantics to capture resolution.

The syntax of our quasi-classical description logic $\mathrm{QC\text{-}ALC}$ is the same as classical ALC. It is one of the merits of $\mathrm{QC\text{-}ALC}$ since users usually do not want any syntactical change in their knowledge bases, and paraconsistent reasoning can be executed in any classical ALC-knowledge base immediately. We first show some basic and necessary definitions.

Definition 3 Let $\mathcal{L}$ denote a set including concept inclusions, concept assertions and/or role assertions formed from $\mathcal{C}$, N_{R} and N_{I}.

Definition 4

(1) All concept names in N_C, the top concept $\top$ and the bottom concept $\bot$ are atomic concepts.

(2) If A is an atomic concept and R is a role name, then $A(a)$, $\neg A(a)$ and $R(a,b)$ are literals, where $a,b \in N_I$.

(3) A clause is a finite set of literals. A clause empty clause, denoted $\square$, if it has no literals.

(4) A clause set is a finite set of clauses.

(5) Let α be a clause $\{A_1, A_2, \cdots, A_n\}$, the focus of α by A_i, denoted $\otimes(\alpha, A_i)$, is defined as the clause obtained by removing A_i from the set $\{A_1, A_2, \cdots, A_n\}$. In the case of $n=1$, that is a clause just with one disjunct, we assume $\otimes(\{A_1\}, A_1) = \{A_1\}$. Intuitively, a clause is a disjunction of the literals in it and a clause set is a conjunction of the clauses in it.

Example 2 Let $\alpha = \{\text{Traveler}(\text{jack}), \neg\text{Passenger}(\text{jack}), \text{TO}(\text{jack}, \text{airport})\}$ be a clause, where Traveler(jack), $\neg$Passenger(jack) and TO(jack, airport) are literals. Then, $\otimes(\alpha, \text{Traveler}(\text{jack})) = \{\neg\text{Passenger}(\text{jack}), \text{TO}(\text{jack, airport})\}$. Sequentially, the semantics of QC-ALC can be defined based on a form of QC interpretation.

Definition 5 A QC interpretation $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty set $\triangle^{\mathcal{I}}$, called the domain of $\mathcal{I}$, and a function $\cdot^{\mathcal{I}}$ that maps every individual name to a element of $\triangle^{\mathcal{I}}$, every concept name (i.e. atomic concept) to a pair of subsets of $\triangle^{\mathcal{I}}$, and every role name to a pair of subsets of $\triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$, such that $\top^{\mathcal{I}} = <\triangle^{\mathcal{I}}, \varnothing>$ and $\bot^{\mathcal{I}} = <\varnothing, \triangle^{\mathcal{I}}>$.

Note that a QC interpretation is defined only for atomic concept names and role names but not for any compound concept. It is more similar to the interpretation in classical first-order logic but less to classical DLs in which compound concepts are also directly interpreted.

Example 3 Let $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a QC interpretation, where $\triangle^{\mathcal{I}} = \{jack, airport\}$, such that

$$\text{jack}^{\mathcal{I}} = \text{jack} \quad , \quad \text{airport}^{\mathcal{I}} = \text{airport}$$

$$\text{Traveler}^{\mathcal{I}} = <\{\text{jack}\}, \{\text{airport}\}>$$

$$\text{Passenger}^{\mathcal{I}} = <\varnothing, \{\text{jack}\}>$$

$$\text{Driver}^{\mathcal{I}} = <\{\text{jack}\}, \{\text{jack}\}>$$

$$\text{Position}^{\mathcal{I}} = <\{\text{airport}\}, \varnothing>$$

$$\text{TO}^{\mathcal{I}} = <\{(\text{jack}, \text{airport})\}, \varnothing>$$

Two functions proj^+ and proj^+ are defined as follows for simplicity of notation:

$$\text{proj}^+(<+A, -A>) = +A$$

and

$$\text{proj}^-(<+A, -A>) = -A$$

where both $+A$ and $-A$ are sets.

We say that $proj^+(A^{\mathcal{I}})$ (or $\text{proj}^+(R^{\mathcal{I}})$) is the *positive extension* of the concept name A (or role name R) in the QC interpretation $\mathcal{I}$, while $\text{proj}^-(A^{\mathcal{I}})$ (or $\text{proj}^-(R^{\mathcal{I}})$) is the negative extension. If $a \in \text{proj}^+(A^{\mathcal{I}})$, then we say that a is a *positive instance* of C in $\mathcal{I}$, while if $a \in \text{proj}^-(A^{\mathcal{I}})$, then we say that a is a negative instance of C in $\mathcal{I}$. Note that the negative extension for a role name is not necessary in this work because ALC does not contain negative role constructor. However, we still define it to retain consistency of notation with possible extensions to more expressive description logics.

With the QC interpretation, the notion of satisfiability for formulae in QC-ALC could be defined inductively. We first consider the following intuitive meaning for literals being satisfied or not in a QC interpretation $\mathcal{I}$:

$a \in \text{proj}^+(A^{\mathcal{I}})$ means $A(a)$ is “satisfiable” in $\mathcal{I}$;

$a \notin \text{proj}^+(A^{\mathcal{I}})$ means $A(a)$ is not “satisfiable” in $\mathcal{I}$;

$a \in \text{proj}^-(A^{\mathcal{I}})$ means $\neg A(a)$ is “satisfiable” in $\mathcal{I}$;

$a \notin \text{proj}^-(A^{\mathcal{I}})$ means $\neg A(a)$ is not “satisfiable” in $\mathcal{I}$.

It is easy to see that the mutually exclusive relation between a formula and its negation has been decoupled in the aspect of semantics since we allow simultaneous satisfaction for an assertion and its negation.

Following the above intuition, the decoupled satisfaction for literals is formally defined as follows. Then, the satisfactions for more complex formulae in $\mathcal{L}$ are defined based on the decoupled satisfaction.

Definition 6 For a QC interpretation $\mathcal{I}$, a satisfiability relation for literals, called Decoupled Satisfaction and denoted $\models_d$, is defined as follows, where $A(a)$ is an atomic concept assertion and

$R(a,b)$ is a role assertion.

$$\mathcal{I} \models_d A(a) \text{ iff } a^{\mathcal{I}} \in proj^{+}(A^{\mathcal{I}})$$

$$\mathcal{I} \models_d \neg A(a) \text{ iff } a^{\mathcal{I}} \in proj^{-}(A^{\mathcal{I}})$$

$$\mathcal{I} \models_d R(a,b) \text{ iff } (a^{\mathcal{I}},b^{\mathcal{I}}) \in proj^{+}(R^{\mathcal{I}})$$

Example 4 Consider again the QC interpretation $\mathcal{I}$ in Example 3. So

$$\mathcal{I} \models_d \text{Traveler(jack)},\ \mathcal{I}_d \not\models \neg\text{Traveler(jack)}$$

$$\mathcal{I}_d \not\models \text{Passenger(jack)},\ \mathcal{I} \models_d \neg\text{Passenger(jack)}$$

$$\mathcal{I} \models_d \text{Driver(jack)},\ \mathcal{I} \models_d \neg\text{Driver(jack)}$$

$$\mathcal{I} \models_d \text{TO(jack,airport)}$$

This definition of the decoupled satisfaction is the base case for the two further satisfaction relations, namely strong satisfaction and weak satisfaction, which allow us to define an entailment relation.

The main idea behind QCDLs is that proofs are separated two stages in which decompositional phase, including resolution, is followed by compositional phase. Hence, the semantics for these two stages is defined to capture this idea. Firstly, we show the inductive definition of strong satisfaction corresponding to the decompositional phase, in which the equivalences allow rewriting any formula in $\mathcal{L}$ into a set of assertions, and then into clause, which can be evaluated w.r.t. the QC interpretation.

Definition 7 For a QC interpretation $\mathcal{I}$, a satisfiability relation, called strong satisfaction and denoted $\models_s$, is defined as follows, where $A_1, A_2, \cdots, A_n$ are literals in $\mathcal{L}$, $R \in N_{\mathrm{R}}$ is a role name and $a,b \in N_{\mathrm{I}}$ are individual names.

For a clause $\alpha = \{A_1, A_2, \cdots, A_n\}$, $I \models_s \alpha$ iff

(1) there is at least one $A_i \in \alpha$ such that $I \models_d A_i$, and

(2) for all $A_i \in \alpha$, $I \models_s \neg A_i$ implies $I \models_s \otimes(\alpha, A_i)$.

For $C, D \in C$, the definition is extended as follows.

$I \models_s C \sqsubseteq D$ iff for any individual $a \in N_{\mathrm{I}}$ $I \models_s \{\neg C \sqcup D(a)\}$.

$I \models_s C(a)$ iff $I \models_s \{C(a)\}$.

$I \models_s R(a,b)$ iff $I \models_s \{R(a,b)\}$.

Let β be a set of assertions. The definition is extended as follows.

if $C \sqcap D(a) \in \beta$, $I \models_s \beta$ iff $I \models_s (\beta - \{C \sqcap D(a)\}) \cup \{C(a)\}$ and $I \models_s (\beta - \{C \sqcap D(a)\}) \cup \{D(a)\}$.

if $C \sqcup D(a) \in \beta$, $I \models_s \beta$ iff $I \models_s (\beta - \{C \sqcup D(a)\}) \cup \{C(a), D(a)\}$.

if $\neg\neg C(a) \in \beta$, $I \models_s \beta$ iff $I \models_s (\beta - \{\neg\neg C(a)\}) \cup \{C(a)\}$.

if $\neg(C \sqcap D)(a) \in \beta$, $I \models_s \beta$ iff $I \models_s (\beta - \{\neg(C \sqcap D)(a)\}) \cup \{\neg C \sqcup \neg D(a)\}$.

if $\neg(C \sqcup D)(a) \in \beta$, $I \models_s \beta$ iff $I \models_s (\beta - \{\neg(C \sqcup D)(a)\}) \cup \{\neg C \sqcap \neg D(a)\}$.

if $\forall R.C(a) \in \beta$, $I \models_s \beta$ iff for all $b \in N_{\mathrm{I}}$, $I \models_s R(a,b)$ implies $I \models_s (\beta - \{\forall R.C(a)\}) \cup \{C(b)\}$.

if $\exists R.C(a) \in \beta$, $I \models_s \beta$ iff there is some $b \in N_{\mathrm{I}}$ such that $I \models_s (\beta - \{\exists R.C(a)\}) \cup \{R(a,b)\}$ and $I \models_s (\beta - \{\exists R.C(a)\}) \cup \{C(b)\}$.

if $\neg\forall R.C(a) \in \beta$, $I \models_s \beta$ iff $I \models_s (\beta - \{\neg\forall R.C(a)\}) \cup \{\exists R.\neg C(a)\}$.

if $\neg\exists R.C(a) \in \beta$, $I \models_s \beta$ iff $I \models_s (\beta - \{\neg\exists R.C(a)\}) \cup \{\forall R.\neg C(a)\}$.

Example 5 Consider again the QC interpretation $\mathcal{I}$ in *Example* 3. So

$\mathcal{I} \models_s \text{Traveler} \sqsubseteq \exists\text{TO.Position}$, $\mathcal{I} \not\models_s \text{Traveler} \sqsubseteq \text{Passenger}$,

$\mathcal{I} \models_s \text{Driver} \sqcup \neg\text{Passenger(jack)}$ and $\mathcal{I} \not\models_s \text{Driver} \sqcup \text{Passenger(jack)}$.

Since resolution rule is expected to be preserved in the case of the decoupling of the mutually exclusive relation between a formula and its negation, the items for disjunction in strong satisfaction are defined to capture a form of resolution in semantics.

Definition 8 Let $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base, where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is a ABox. A QC interpretation $\mathcal{I}$ is said to strongly satisfy Σ, written $\mathcal{I} \models_s \Sigma$, if $\mathcal{I}$ strongly satisfies every axiom of Σ. Such a QC interpretation $\mathcal{I}$ is said to be a strong model of Σ.

Example 6 Let $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a QC interpretation, where $\triangle^{\mathcal{I}} = \{jack, no123, college, airport\}$, such that $\text{jack}^{\mathcal{I}} = \text{jack}$, $\text{no123}^{\mathcal{I}} = \text{no123}$, $\text{colloge}^{\mathcal{I}} = \text{college}$, $\text{airport}^{\mathcal{I}} = \text{airport}$,

$\text{Driver}^{\mathcal{I}} = <\{\text{jack}\},\{\text{jack, no123, college, airport}\}>$,

$\text{Person}^{\mathcal{I}} = <\{\text{jack}\},\varnothing>$,

$\text{DriverLicense}^{\mathcal{I}} = <\{\text{no123}\},\{\text{no123}\}>$,
$\text{Position}^{\mathcal{I}} = <\{\text{college},\text{airport}\},\varnothing>$,
$\text{Traveler}^{\mathcal{I}} = <\{\text{jack}\},\{\text{jack},\text{no123},$
$\text{college},\text{airport}\}>$,
$\text{Passenger}^{\mathcal{I}} = <\{\text{jack}\},\{\text{jack}\}>$,
$\text{PROFILE}^{\mathcal{I}} = <\{(\text{jack},\text{no123})\},\varnothing>$,
$\text{CONTEXT}^{\mathcal{I}} = <\{(\text{jack},\text{college})\},\varnothing>$, and
$\text{TO}^{\mathcal{I}} = <\{(\text{jack},\text{airport})\},\varnothing>$.

So $\mathcal{I} \models_s \text{Traveler} \sqsubseteq \exists\text{TO.Position}$, $\mathcal{I} \models_s \text{Traveler} \sqsubseteq \text{Passenger}$,

$\mathcal{I} \models_s \text{Driver} \sqcup \neg\text{Passenger}(\text{jack})$, $\mathcal{I} \models_s \text{Driver} \sqcup \text{Passenger}(\text{jack})$, and

For the inconsistent knowledge base Σ in Example 1, $\mathcal{I} \models_s \Sigma$ holds, that is, $\mathcal{I}$ is a strong model of Σ.

The definition for weak satisfaction, corresponding to the compositional phase in proofs, is similar to strong satisfaction other than the definition for disjunction, which is no longer defined to capture resolution.

Definition 9 For a QC interpretation $\mathcal{I}$, a satisfiability relation, called weak satisfaction and denoted as $\models_w$, is defined as follows, where A is a literal in $\mathcal{L}$, $C,D \in \mathcal{C}$ are ALC-concepts, $R \in N_{\text{R}}$ is a role name and $a,b \in N_{\text{I}}$ are individual names.

$\mathcal{I} \models_w A$ iff $\mathcal{I} \models_d A$.

$\mathcal{I} \models_w C \sqsubseteq D$ iff for all $a \in U(L)$ $\mathcal{I} \models_w \neg C \sqcup D(a)$.

$\mathcal{I} \models_w C \sqcap D(a)$ iff $\mathcal{I} \models_w C(a)$ and $\mathcal{I} \models_w D(a)$.

$\mathcal{I} \models_w C \sqcup D(a)$ iff $\mathcal{I} \models_w C(a)$ or $\mathcal{I} \models_w D(a)$.

$\mathcal{I} \models_w \neg\neg C(a)$ iff $\mathcal{I} \models_w C(a)$.

$\mathcal{I} \models_w \neg(C \sqcap D)(a)$ iff $\mathcal{I} \models_w \neg C \sqcup \neg D(a)$.

$\mathcal{I} \models_w \neg(C \sqcup D)(a)$ iff $\mathcal{I} \models_w \neg C \sqcap \neg D(a)$.

$\mathcal{I} \models_w \forall R.C(a)$ iff for all $b \in N_{\text{I}}$, $\mathcal{I} \models_w R(a,b)$ implies $\mathcal{I} \models_w C(b)$.

$\mathcal{I} \models_w \exists R.C(a)$ iff there is some $b \in N_{\text{I}}$ such that $\mathcal{I} \models_w R(a,b)$ and $\mathcal{I} \models_w C(b)$.

$\mathcal{I} \models_w \neg\forall R.C(a)$ iff $\mathcal{I} \models_w \exists R.\neg C(a)$.

$\mathcal{I} \models_w \neg\exists R.C(a)$ iff $\mathcal{I} \models_w \forall R.\neg C(a)$.

Example 7 Consider the QC interpretation $\mathcal{I}$ in Example 3 again. So

$\mathcal{I} \models_w \text{Traveler} \sqsubseteq \exists\text{TO.Position}$, $\mathcal{I}_w \not\models \text{Traveler} \sqsubseteq \text{Passenger}$,

$\mathcal{I} \models_w \text{Driver} \sqcup \neg\text{Passenger}(\text{jack})$, $\mathcal{I} \models_w \text{Driver} \sqcup \text{Passenger}(\text{jack})$.

Example 8 Consider the QC interpretation $\mathcal{I}$ in Example 6 again. So

$\mathcal{I} \models_w \text{Traveler} \sqsubseteq \exists\text{TO.Position}$, $\mathcal{I} \models_w \text{Traveler} \sqsubseteq \text{Passenger}$,

$\mathcal{I} \models_w \text{Driver} \sqcup \neg\text{Passenger}(\text{jack})$, $\mathcal{I} \models_w \text{Driver} \sqcup \text{Passenger}(\text{jack})$

Sequentially, based on the two satisfaction defined above, QCDLs entailment is established, which is of the same form as classical DLs entailment, except that strong satisfaction is used for the assumptions and weak satisfaction is used for the conclusion.

Definition 10 For a knowledge base Σ, let $\models_Q$ denote an entailment relation, called the QC Entailment Relation, which is defined as follows, where $C,D \in C$ are concept names, $R \in N_{\text{R}}$ and $a,b \in N_{\text{I}}$ are individual names.

$\Sigma \models_Q C \sqsubseteq D$ iff for any QC interpretation $\mathcal{I}$, $\mathcal{I} \models_s \Sigma$ implies $\mathcal{I} \models_w C \sqsubseteq D$.

$\Sigma \models_Q C(a)$ iff for any QC interpretation $\mathcal{I}$, $\mathcal{I} \models_s \Sigma$ implies $\mathcal{I} \models_w C(a)$.

$\Sigma \models_Q R(a,b)$ iff for any QC interpretation $\mathcal{I}$, $\mathcal{I} \models_s \Sigma$ implies $\mathcal{I} \models_w R(a,b)$.

Example 9 Consider the inconsistent knowledge base Σ in Example 1 again. The followings can be verified.

$\Sigma \models_Q \text{Traveler} \sqsubseteq \exists\text{TO.Position}$, $\Sigma_Q \not\models \text{Traveler} \sqsubseteq \text{Passenger}$,

$\Sigma \models_Q \exists\text{TO.Position}(\text{jack})$, and $\Sigma_Q \not\models \neg\text{Person}(\text{jack})$.

This example shows that $\models_Q$ is non-trivializable in the sense that it is not the case that every formula in $\mathcal{L}$ is entailed by Σ even if Σ is classically inconsistent.

4 A Tableau Calculus for QCDLs

The tableau-based reasoning technique is most widely used to solve the reasoning problems in DLs. It was first introduced to the area of DLs in reference [19], and developed and extended by many other work such as reference [19]. In this section, the tableau approach for classical DLs is adapted to provide a paraconsistent automated proof procedure – the QC semantic tableau, which can be used to handle reasoning problems on the knowledge base with acyclic TBox.

In classical semantics, the reasoning problems on the knowledge base with acyclic TBox can be reduced to problems on the knowledge base with the empty TBox by means of normalization and expanding concepts[2]. However, the normalization is no longer valid in QC semantics. So, it is needed in QC semantic tableau to develop particular expansion rules for acyclic inclusion axioms.

The following definitions are needed for the adaptation.

Definition 11 The set of signed formulae of $\mathcal{L}$ is defined as $\mathcal{L}^* = \mathcal{L} \cup \{\varphi^* \mid \varphi \in \mathcal{L}\}$.

We regard the formulae in $\mathcal{L}^*$ without the symbol * as satisfiable and the formulae in $\mathcal{L}^*$ with the symbol * as unsatisfiable.

For the sake of convince, it can be assumed without loss of generality that all of the concepts occurring in the given knowledge base and query are transformed in negation normal form (NNF), i.e., that negation is applied only to concept names. An arbitrary ALC concept can be transformed to an equivalent NNF by pushing negations inwards using a series of de Morgan's laws, the duality between quantifiers and double negation elimination, which are guaranteed by both the strong and weak satisfaction. For a concept or an assertion C, we will use $\dot{\neg}C$ to denote the NNF of $\neg C$. For example, the NNF of the concept $\neg(\forall R.A \sqcup \exists P.B)$, where A and B are concept names, is $(\exists R.\neg A) \sqcap (\forall P.\neg B)$.

4.1 Constraint System

Before describing the semantic tableau more formally, it is needed to introduce the notion of *constraint system*, which was originally proposed in reference [19]. In order to provide an appropriate data structure representing constraints such as "$C(a)$ is strongly satisfiable" and "$C(b)$ is weakly unsatisfiable", the traditional constraint system is adapted, and signed ABox assertions is used to represent the constraints in this paper. In order to gain the ability of processing acyclic TBox, inclusion axioms are included in our constraint system in which an inclusion axiom is viewed as a cluster of assertion constraints, and is finally transformed to these constraints by applying expansion rules.

Definition 12 A constraint system S is defined as a finite nonempty set of sets of constraints which are signed assertions and signed GCIs, i.e., $S \neq \varnothing$ and $S \subseteq \wp(\mathcal{L}^*)$, where $\wp(\mathcal{L}^*)$ is the power set of $\mathcal{L}^*$. Initially, a given $\mathcal{ALC}$-knowledge base $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ is translated into a constraint system $S_\Sigma = \{\{\alpha\} \mid \alpha \in \mathcal{T} \ or \ \alpha \in \mathcal{A}\}$.

Example 10 The initial constraint system translated from the knowledge base Σ in Example 1 is as follows:

$$S_\Sigma = \{\{\text{Driver} \sqsubseteq \text{Person} \sqcap \exists\text{PROFILE.DrivingLicense} \sqcap \exists\text{CONTEXT.Position}\}, \{\text{Traveler} \sqsubseteq (\text{Driver} \sqcup \text{Passenger}) \sqcap \exists\text{TO.Position}\}, \{\text{Traveler}(jack)\}, \{\neg\text{Passenger}(jack)\}, \{\forall\text{PROFILE.}\neg\text{DrivingLicense}(jack)\}\}.$$

The satisfiability relation for constraint system is defined as follows.

Definition 13 The strong satisfaction and weak satisfaction relations are further extended for constraint system as follows, where $\varphi \in L$ and S is a constraint system.

$\mathcal{I} \models_s \{\varphi^*\}$ iff $\mathcal{I} \models_s \varphi^*$ iff $\mathcal{I} \not\models_s \varphi$

$\mathcal{I} \models_w \{\varphi^*\}$ iff $\mathcal{I} \models_w \varphi^*$ iff $\mathcal{I} \not\models_w \varphi$

$\mathcal{I} \models_s S$ iff for every $\beta \in S$, $\mathcal{I} \models_s \beta$

$\mathcal{I} \models_w S$ iff for every $\beta \in S$, $\mathcal{I} \models_w \beta$

We say S is satisfiable iff there is a QC interpretation $\mathcal{I}$ such that $\mathcal{I} \models_s S \cap \wp(\mathcal{L})$ and $\mathcal{I} \models_w S - S \cap \wp(\mathcal{L})$.

For every individual $a \in N_I$ occurring in a constraint system S, we call a explicit if there is a constraint on a in S that is involved in a single constraint set containing only one constraint, and we call it implicit if there is no such constraint. For example, in a constraint system $S = \{\{A(a), R(a,b)\}, \{B(a)\}\}$, a is explicit and b is implicit.

4.2 Expansion Rules

After obtaining a constraint system from a knowledge base, the QC semantic tableau then applies it to the so-called expansion rules, which transform the sets of constraints into clauses, then syntactically decompose the clauses into literals and generate one or more expanded constraint systems with new constraints in every step. There are two types of expansion rules in the definition of the QC semantic tableau. The first type is given in Definition 14 and called S-rules, which is to process the constraints without the symbol * that are believed to be satisfiable. The second type is given in Definition 15 and called U-rules, which is to process the constraints with the symbol * that are believed to be unsatisfiable.

Definition 14 The S-rules for QC semantic tableau are defined as follows.

$\sqcap$-rule $S \to S \cup \{(\beta - \{C \sqcap D(a)\}) \cup \{C(a)\}, (\beta - \{C \sqcap D(a)\}) \cup \{D(a)\}\}$

if (1) $C \sqcap D(a) \in \beta$ where $\beta \in S$, and

(2) $\{(\beta - \{C \sqcap D(a)\}) \cup \{C(a)\}, (\beta - \{C \sqcap D(a)\}) \cup \{D(a)\}\} \not\subseteq S$.

$\sqcup_1$-rule $S \to S \cup \{(\beta - \{C \sqcup D(a)\}) \cup \{C(a), D(a)\}\}$

if (1) $C \sqcup D(a) \in \beta$ where $\beta \in S$, and

(2) $(\beta - \{C \sqcup D(a)\}) \cup \{C(a), D(a)\} \notin S$.

$\sqcup_2$-rule $S \to S \cup \{\otimes(\alpha, A_i)\}$

if (1) $\alpha = \{A_1, A_2, \cdots, A_n\} \in S$ where α is a clause, and

(2) There is a $A_i \in \alpha$ such that $\{\dot{\neg} A_i\} \in S$ and $\otimes(\alpha, A_i) \notin S$.

$\sqcup_3$-rule $S \to S \cup \{\{A_1\}, S \cup \{A_2\}\}, \cdots, S \cup \{\{A_n\}\}$

if (1) $\alpha = \{A_1, A_2, \cdots, A_n\} \in S$ where α is a clause, and

(2) $\{\{A_i\} \mid A_i \in \alpha\} \cap S = \varnothing$.

$\sqsubseteq$-rule $S \to S \cup \{\{\neg A \sqcup C(a)\}\}$

if 1. $\{A \sqsubseteq C\} \in S$, and

(2) There is an explicit individual a such that $\{\neg A \sqcup C(a)\} \notin S$.

$\forall$-rule $S \to S \cup \{(\beta - \{\forall R.C(a)\}) \cup \{C(b)\}\}$

if (1) $\forall R.C(a) \in \beta$ where $\beta \in S$, and

(2) There is an individual b such that $\{R(a,b)\} \in S$, and $(\beta - \{\forall R.C(a)\}) \cup \{C(b)\} \notin S$.

$\exists$-rule $S \to S \cup \{(\beta - \{\exists R.C(a)\}) \cup \{R(a,b)\}, (\beta - \{\exists R.C(a)\}) \cup \{C(b)\}\}$

if (1) $\exists R.C(a) \in \beta$ where $\beta \in S$,

(2) b is a new individual name, and

(3) There is no individual i such that $(\beta - \{\exists R.C(a)\}) \cup \{R(a,i)\} \in S$ and $(\beta - \{\exists R.C(a)\}) \cup \{C(i)\} \in S$.

Definition 15 The U-rules for QC semantic tableau are defined as follows.

$\sqcap^*$-rule $S \to S \cup \{\{C(a)^*\}\}, S \cup \{\{D(a)^*\}\}$

if (1) $\{C \sqcap D(a)^*\} \in S$, and

(2) $\{\{C(a)^*\}, \{D(a)^*\}\} \cap S = \varnothing$.

$\sqcup^*$-rule $S \to S \cup \{\{C(a)^*\}, \{D(a)^*\}\}$

if (1) $\{C \sqcup D(a)^*\} \in S$, and

(2) $\{\{C(a)^*\}, \{D(a)^*\}\} S$.

$\sqsubseteq^*$-rule $S \to S \cup \{\{\neg A \sqcup C(b)^*\}\}$

if (1) $\{(A \sqsubseteq C)^*\} \in S$,

(2) b is a new individual name, and

(3) there is no individual i such that $\{\neg A \sqcup C(i)^*\} \in S$.

$\forall^*$-rule $S \to S \cup \{\{R(a,b)\}, \{C(b)^*\}\}$

if (1) $\{\forall R.C(a)^*\} \in S$,

(2) b is a new individual name, and

(3) There is no individual i such that $\{\{R(a,i)\}, \{C(i)^*\}\} \subseteq S$.

$\exists^*$-rule $S \to S \cup \{\{C(b)^*\}\}$

if (1) $\{\exists R.C(a)^*\} \in S$, and

(2) There is an individual b such that $\{R(a,b)\} \in S$ and $\{C(b)^*\} \notin S$.

We refer to $\sqcup_3$ -rule and $\sqcap^*$ -rule as nondeterministic rules, since they can be applied in different ways to the same constraint system (intuitively, they correspond to branching rules of tableau). All the other rules are called deterministic rules. Among them, we refer to $\sqcap$-rule, $\sqcup_1$ -rule, $\sqsubseteq$ -rule, $\sqcup^*$ -rule and $\sqsubseteq^*$ -rule as *rewriting* rules, $\sqcup_2$ -rule as resolution rule, and $\forall$ -rule, $\exists$ -rule, $\forall^*$ -rule and $\exists^*$ -rule as quantification rules. In addition, we refer to $\exists$-rule, $\sqsubseteq^*$-rule and $\forall^*$-rule as generating rules since they introduce new individuals in the constraint system. All the other rules are called non-generating rules.

4.3 The QC Semantic Tableau and QC Consequence Relation

Definition 16 A QC semantic tableau for a knowledge base Σ and a query $\varphi \in \mathcal{L}$ is a tree such that

(1) The root of the tree is labeled by $S_\Sigma \cup \{\{\phi^*\}\}$, and

(2) All child nodes are a constraint system

obtained by applying expansion rules on their parents. This definition is similar to the one for the classical semantic tableau in refrences [2, 20]. The major differences include: (1) The root of the classical tableau is labeled by $S_{\Sigma} \cup \{\neg\phi\}$ if the query $\phi = C(a)$, or is labeled by $S_{\Sigma} \cup \{C \sqcap \neg D(b)\}$, where b is a new individual, if the query $\phi = C \sqsubseteq D$; (2) The constraint system of the classical tableau is a set of non-signed constraints, while the constraint system of the QC tableau is a set of constraint sets. The reason causing the differences is that the link between a formula and its complement has been decoupled.

Definition 17 A constraint system is complete iff no expansion rule is applicable to it. A complete system derived from a constraint system S is also called a completion of S. A constraint system S contains a clash iff $\{\{\bot(a)\},\{\neg\top(a)\},\{\top(a)^*\},\{\neg\bot(a)^*\}\} \cap S \neq \varnothing$ or $\{\{A\},\{A^*\}\} \subseteq S$, where A is a literal.

Definition 18 A QC tableau is closed iff all its branches are closed. A branch in such a tableau is closed iff its leaf node contains a clash. A tableau is open iff at least one of its branches is open. A branch is open iff its leaf node is complete and does not contain clash.

Now, the QC consequence relation can be defined using the QC semantic tableau.

Definition 19 For a knowledge base with acyclic TBox Σ, let $\vdash_Q$ denote a consequence relation, called the QC Consequence Relation, which is defined as follows, where $C \in \mathcal{C}$ are concept names, $R \in N_R$ is a role name and $a, b \in N_I$ are individual names.

$\Sigma \vdash_Q A \sqsubseteq C$ iff the QC semantic tableau for Σ and $A \sqsubseteq C$ is closed;

$\Sigma \vdash_Q C(a)$ iff the QC semantic tableau for Σ and $C(a)$ is closed;

$\Sigma \vdash_Q R(a,b)$ iff the QC semantic tableau for Σ and $R(a,b)$ is closed.

The reasoning tasks formalized in the end of Subsection 2.1 can be finished using the QC entailment, by checking whether the corresponding QC consequence relations hold.

Example 11 Consider the inconsistent knowledge base Σ in Example 1 and the query $\neg\text{Person}(\text{jack})$ in Example 9. The root of the semantic tableau for Σ and the query is:

$$\begin{aligned} S_0 = S_{\Sigma} \cup \{\{\neg\text{Person}(\text{jack})^*\}\} = \\ &\{\{\text{Driver} \sqsubseteq \text{Person} \sqcap \\ &\exists\text{PROFILE.DrivingLicense} \sqcap \\ &\exists\text{CONTEXT.Position}\}, \\ &\{\text{Traveler} \sqsubseteq (\text{Driver} \sqcup \\ &\text{Passenger}) \sqcap \exists\text{TO.Position}\}, \\ &\{\text{Traveler}(\text{jack})\}, \\ &\{\neg\text{Passenger}(\text{jack})\}, \\ &\{\forall\text{PROFILE.}\neg\text{DrivingLicense} \\ &(\text{jack})\},\{\neg\text{Person}(\text{jack})^*\}\} \end{aligned}$$

A sequence of applications of the expansion rules to S_0 is in Appendix A.

It can be verified that S_{63} is a complete clash-free constraint system. Hence, the semantic tableau for Σ and the query $\neg\text{Person}(\text{jack})$ is open, that is, $\Sigma \nvDash_Q \neg\text{Person}(\text{jack})$. Corresponding with $\vDash_Q$ in Example 9, this example shows that $\vdash_Q$ is non-trivializable in the proof-theoretic view.

4.4 Properties of Quasi-Classical Description Logics

Quasi-classical description logic QC- ALC possesses some elegant properties, such as that its QC consequence relation is paraconsistent, sound, complete and decidable. Some classically logical properties are preserved in QCDLs, while others no langer holds. Now we discuss those properties.

Soundness, completeness and decidability are three of the most important properties of any formal logical system. The following theorem shows the soundness and completeness of the QC consequence relation.

Theorem 1 Let Σ be a knowledge base with acyclic TBox and $\varphi \in \mathcal{L}$ a query. $\Sigma \vdash_Q \varphi$ iff $\Sigma \vDash_Q \varphi$.

Theorem 2 Let Σ be a finite knowledge base with acyclic TBox and $\varphi \in \mathcal{L}$. $\Sigma \vdash_Q \varphi$ can be determined in a finite number of steps. In other words, the QC consequence relation is decidable.

Proposition 1 Quasi-classical description logics are paraconsistent.

QCDLs are paraconsistent in the sense that they does not allow trivial inferences. That is, it is not the case that any conclusion in the language is entailed by a given classical inconsistent knowledge base. It is shown by Examples 9 and 11. QCDLs provide a paraconsistent logical fundament for the OWL, so that knowledge reasoning in the Semantic Web no longer needs the usually unachivable requirement for consistency.

Proposition 2 The QC semantic tableau collapses to a classical semantic tableau if the following rule are added to the expansion rules,

$\neg^*$-rule $S \to S \cup \{\{\dot{\neg}\phi\}\}$

if 1. $\{\phi^*\} \in S$ where ϕ is an assertion, and

2. $\{\dot{\neg}\phi\} \notin S$.

and we can use the classical definition for clash (i.e. a constraint system contains both $\{A\}$ and $\{\neg A\}$ for a literal A).

This proposition shows that the QC semantic tableau can be viewed as a special form of the classical tableau for DLs. If the given knowledge base is consistent, then all expansion rules in the QC semantic tableau and $\neg^*$-rule can be applied in a classical way to finish classical reasoning tasks.

Proposition 3 Reflexivity, monotonicity, and consistency preservation hold in QCDLs. However, transitivity fails in any tautology cannot be inferred from the empty knowledge base.

5 Related Work

In this section, we provide a detailed comparison with related work on paraconsistent reasoning in description logics.

5.1 Comparing with Four-Valued Description Logics

Four-Valued Description Logics stems from Belnap's four-valued logic and have been studied in many literature[9,11]. The similarities and differences between QCDLs and FVDLs resemble those between QCL and Belnap's four-valued logic respectively.

We first consider their similarities. Of course, they are both paraconsistent. In model-theoretical view, it is an idea shared by several paraconsistent logics to decouple the link between a formula and its negation at the level of model to obtain a paraconsistent semantics. So are QCDLs and FVDLs. For the four-valued semantics of ALC4 defined in reference [9], we show its correspondence with our quasi-classical semantics through the following definition.

Definition 20 Let $\mathcal{I}$ be a quasi-classical model and $\mathcal{I}_4$ a four-valued model. $\mathcal{I}_4$ is a corresponding model for $\mathcal{I}$ iff $a^{I_4} \in proj^+(A^{I_4})$ if $+A(a) \in \mathcal{I}$, $a^{I_4} \in proj^-(A^{I_4})$ if $-A(a) \in \mathcal{I}$ and $(a^{I_4}, b^{I_4}) \in proj^+(R^{I_4})$ if $+R(a,b) \in \mathcal{I}$.

For the general concept inclusion axioms in TBox, ALC4 allows three kinds of inclusions.

Now, let us consider the differences. In model-theoretic view, QCDLs have a stronger semantics because the quasi-classical semantics define more restriction, in addition to decoupling the link between a formula and its complement, for strong satisfaction of disjunction. Those constraints ensure that if the negation of a disjunct holds in a strong model, then the resolvent should also hold in the model. Therefore, the number of strong models of QCDLs is not greater than that of FVDLs, and the number of weak models of QCDLs is actually equal to that of FVDLs because the additional restriction for semantics of disjunction is not defined in weak satisfaction. Even those numbers may be infinite. If all inclusion axioms in a knowledge base are considered material inclusion axioms, the following proposition holds.

Proposition 4 Let Σ be an ALC-knowledge base. For all quasi-classical model $\mathcal{I}$, if $\mathcal{I} \models_s \Sigma$ holds then there is a corresponding four-valued model $\mathcal{I}_4$ for $\mathcal{I}$ such that $\mathcal{I}_4 \models_4 \Sigma$ holds. Moreover, For all quasi-classical model $\mathcal{I}$, $\mathcal{I} \models_w \Sigma$ holds if there is a corresponding four-valued model $\mathcal{I}_4$ for $\mathcal{I}$ such that $\mathcal{I}_4 \models_4 \Sigma$ holds.

Since QCDLs have reduced strong models, a knowledge base often has more non-tautological inferences in QC semantics than those in four-valued semantics, and it is never less. Consider an ABox $\mathcal{A} = \{A \sqcup B(a), \neg A(a)\}$ for an example, $B(a)$ is an inference of $\mathcal{A}$ in QC semantics, but not in four-valued semantics. Furthermore, the inferences of a knowledge base in QC semantics almost coincidering with in classical DLs when the knowledge base is classically consistent. In fact, only tautologies and formulae containing tautologies cannot be inferred.

However, tautologies contain no information whatsoever since a knowledge base is often used as a domain knowledge model.

In proof-theoretic view, it reflects FVDLs' weaker semantics that they cannot support disjunctive syllogism, modus ponens and modus tollens, all of which are supported by QCDLs.

5.2 On Quasi-Classical Semantics of Description Logics

When the primary version of this work was being peer-reviewed, we noticed that another paper on quasi-classical semantics of description logics was accepted by another conference reference [12]. After publishing of that work, we carefully read it and found that there may be some serious glitches in it. We will discuss them in this subsection.

First, the quasi-classical semantics of ALC defined in reference [12] leads to a contradiction. In both QCL and our work on QCDLs, the resolution rule applied only to clauses in the definition of strong satisfaction[7,14-16]. This restriction is necessary. However, the strong interpretation in reference [12] weakens this restriction to arbitrary concept. We demonstrate on the following example that the strong interpretation defined in reference [12] leads to a contradiction.

Example 12 Let C, D and E be ALC concepts, and $I=(\{a\},\cdot^I)$ a strong interpretation defined in reference [12] such that $C^I=<\varnothing,\varnothing>$, $D^I=<\{a\},\{a\}>$ and $E^I=<\{a\},\varnothing>$. By the definition in reference [20], $(C\sqcup(D\sqcap E))^I=<\{a\},\varnothing>$ but $((C\sqcup D)\sqcap(C\sqcup E))^I=<\varnothing,\varnothing>$. It contradicts distributivity of disjunction.

Second, corresponding with the problem in semantics, the tableau algorithm defined in reference [12] also does not restrict the applying of the resolution rule to clauses. This leads a similar contradiction illustrated above.

In addition, even if the above glithes are repaired, the method proposed in reference [12] only can deal with the problem of instance checking. The paraconsistent reasoning with inconsistent knowledge base contained concept inclusions are not discussed in reference [12].

6 Conclusion and Future Work

In this paper, we have proposed the quasi-classical description logics which can achieve paraconsistent reasoning in DLs. The semantics and the sound, complete and decidable QC semantic tableau for QCDLs have been elaborately introduced. This work provides a novel approach for paraconsistent reasoning with ontologies represented in OWL in the Semantic Web. We have provided a comparison between QCDLs and other key paraconsistent DLs to explain that QCDLs are more appropriate than other paraconsistent DLs for applications in the Semantic Web.

Of course, it remains much work to be done. In order to make our approach more practicable, it is obvious to optimize the tableau calculus. Secondly, we have to refine the decidability theorem to provide a precise complexity analysis. Moreover, we intend to extend the QC semantics and QC semantic tableau to more expressive DLs to cover more parts of OWL-DL. In addition, we are developing a paraconsistent OWL reasoner based on classical OWL reasoner Pellete using our approach.

References

[1] BERNERS-LEE T, HENDLER J, LASSILA O. The semantic Web[J]. Scientific American, 2001, 284(5): 34-43.

[2] BAADER F, CALVANESE D, MCGUINNESS D L, et al. The description logic handbook: Theory, implementation, and applications[M]. Cambridge: Cambridge University Press, 2003.

[3] W3C. OWL Web Ontology Language Overview. W3C Recommendation (10 February 2004). Available at http://www.w3.org/TR/owl-features/.

[4] HAASE P, HARMELEN F van, HUANG Zhi-sheng, et al. e. A framework for handling inconsistency in changing ontologies[C]//International Semantic Web Conference. [S.l.]: Springer, 2005, 3729: 353-367..

[5] SCHLOBACH S, CORNET R. Non-standard reasoning services for the debugging of description logic terminologies[C]//IJCAI. [S.l.]: [s.n.], 2003: 355-362.

[6] PATEL-SCHNEIDER P F. A four-valued semantics for terminological logics[J]. Artificial Intel-Ligence, 1989, 38:319-351.

[7] HOU Hui, WU Jin-zhao. Quasi-classical description logics and paraconsistent reasoning in the semantic Web[C]//In 2nd International Workshop on New Forms of Reasoning for the Semantic Web (NEFORS 2008) at ASWC 2008. [S.l.]: [s.n.], 2008.

[8] HUANG Zhi-sheng, HARMELEN F van, TEIJE A ten . Reasoning with inconsistent ontologies[C]//In Leslie Pack Kaelbling and Alessandro Saffiotti. [S.l.]: IJCAI, Professional Book Center, 2005: 454- 459.

[9] MA Yue, HITZLER P, LIN Zuo-quan. Algorithms for paraconsistent reasoning with OWL[J]. LNCS, 2007, 4519: 399-413.

[10] MA Yue, LIN Zuo-quan, LIN Zhan-gang. Inferring with inconsistent OWL DL ontology: a multi-valued logic approach[C]//In Current Trends in Database Technology - EDBT 2006 Work- shops. [S.l.]: [s.n.], 2006.

[11] STRACCIA U, MARIA V S. A sequent calculus for reasoning in four-valued description logics[C]//In Proc. of the Int. Conf. on Analytic Tableaux and Related Methods (TABLEAUX-97), in Lecture Notes in Artificial Intelligence. [S.l.]: Springer-Verlag, 1997, (1227): 343-357.

[12] ZHANG Xiao-wang, LIN Zuo-quan. Paraconsistent reasoning with quasi-classical semantic in ALC[J]. LNCS, 2008, 5341: 222-229.

[13] BELNAP N D. A useful four-valued logic[M]. [S.l.]: Modern Uses of Multiple-Valued Logic, 1977.

[14] BESNARD P, HUNTER A. Quasi-classical logic: non-trivializable classical reasoning from inconsistent information[C]// In Proceedings of the European Conference on Symbolic and Quantitative Approaches To Reasoning and Uncertainty: Lecture Notes In Computer Science. London: Springer-Verlag, 1995, 946: 44-51.

[15] HUNTER A. Reasoning with contradictory information using quasi-classical logic[J]. Journal of Logic and Computation, 2000, 10(5): 677-703.

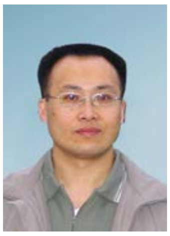
[16] HUNTER A . A semantic tableau version of first-order ouasi-classical logic[C]//In Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches To Reasoning with Uncertainty: Lecture Notes In Computer Science, London: Springer-Verlag, 2001, 2143: 544-555.

[17] HUNTER A. Measuring inconsistency in knowledge via quasi-classical models[C]//The Proceedings of the 18th American National Conference on Artificial Intelligence (AAAI’2002). [S.l.]: MIT Press, 2002: 68-73.

[18] SCHMIDT-SCHAU M, SMOLKA G. Attributive concept descriptions with complements[J]. Artificial Intelligence, 1991, 48(1): 1-26.

[19] BUCHHEIT M, DONNI F M, SCHAERF A. Decidable reasoning in terminological knowledge representation systems[J]. J of Artificial Intelligence Research, 1993, 1: 109-138

编 辑 漆 蓉



吴尽昭，电子科技大学光电信息学院教授、博士生导师。中国科学院“百人计划(国外引入杰出人才)”入选者(2000年)，国家人事部等七部委的首批“百千万人才工程国家级人选”(2004年)，四川省学术与技术带头人(2005年)。

1991.09～1994.07月，中科院系统科学研究所，理学博士；1994.08～1996.07月，北京大学数学学院信息科学系，博士后；1997.1～1998.1月，美国Texas A&M大学电子工程系，研究助理；1998.1～1999.12月，德国Max-Planck计算机科学研究所，研究员；2000.1～2005.12月，德国Mannheim大学计算机科学系，客座教授。

长期从事计算与推理基础理论研究及形式化验证等方面的研究。作为课题负责人，近五年来来，先后承担了国家自然科学基金、“863”项目、“973”重大基础研究项目、德国DFG项目及省部级科研项目十余项，研发了集成电路系统验证软件原型系统；在《Acta Informatica》、《Information Processing Letters》、《Discrete Math.》、《ISSAC》、《J.Automated Reasoning》、《中国科学》、《ISMVL》、《J.Computrt Sci. & Technol.》、《SEFM》、《ICECCS》和《ICEFEM》等国内外重要刊物和国际学术会议论文集上发表论文102余篇，其中SCI、EI收录6余篇；出版学术专著2部，获得软件著作权2项。研究方向为复杂电子系统、集成电路验证技术和专用集成系统设计及其应用。