Computing and Informatics, Vol. 36, 2017, 1415–1446, doi: 10.4149/cai_2017_6_1415

ON THE SATISFIABILITY OF QUASI-CLASSICAL DESCRIPTION LOGICS

Xiaowang Zhang*, Zhiyong Feng

School of Computer Science and Technology
Tianjin University, Tianjin 300072, China
&
Tianjin Key Laboratory of Cognitive Computing and Application
Tianjin, China
e-mail: xiaowangzhang@tju.edu.cn

Wenrui Wu

School of Computer Science and Technology
Tianjin University, Tianjin 300072, China

Mokarrom Hossain, Wendy MacCaull

Department of Mathematics, Statistics and Computer Science
St. Francis Xavier University
PO Box 5000, Antigonish, NS, Canada

Abstract. Though quasi-classical description logic (QCDL) can tolerate the inconsistency of description logic in reasoning, a knowledge base in QCDL possibly has no model. In this paper, we investigate the satisfiability of QCDL, namely, QC-coherency and QC-consistency and develop a tableau calculus, as a formal proof, to determine whether a knowledge base in QCDL is QC-consistent. To do so, we repair the standard tableau for DL by introducing several new expansion rules and defining a new closeness condition. Finally, we prove that this calculus is sound and complete. Based on this calculus, we implement an OWL paraconsistent reasoner

* Corresponding author

called QC-OWL. Preliminary experiments show that QC-OWL is highly efficient in checking QC-consistency.

Keywords: Semantic web, description logics, quasi-classical logic, satisfiability, tableaux

1 INTRODUCTION

As a family of knowledge representation languages, description logics (DLs) can be used to represent the knowledge of an application domain in a structured and formally well-understood way [1]. DLs are the logical foundation of the Web Ontology Language (OWL) which represents concepts and properties in the Semantic Web [2], as an extension of the World Wide Web. Due to many reasons, such as modeling errors, migration from other formalisms, merging ontologies, and ontology evolution [3, 4], the Semantic Web is rarely perfect and the inconsistency arising from knowledge becomes unavoidable. However, description logic, as a fragment of first-order logic, could do nothing about this inconsistent knowledge. As a result, handling inconsistent knowledge in DLs has received extensive interests in the community in recent years [5, 3, 6, 7, 8, 4, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18].

There are several approaches to handling inconsistencies in DLs. As a functional type of those approaches, the inconsistency-tolerant (or paraconsistent) approach is not to simply avoid inconsistencies but to apply non-standard reasoning methods (e.g., non-standard inference or non-classical semantics) to obtain meaningful answers. Among those approaches, the approach based on four-valued semantics [19] is popular since four-valued semantics is more intuitive and concise than others [5, 3, 15]. However, the inference power of four-valued semantics is not strong. For instance, assume that *Wade* is a student or a staff member in a university and *Wade* is not a student. However, we do not conclude that *Wade* is a staff member in that university under four-valued semantics.

To strengthen the inference power of four-valued semantics for DLs, based on *quasi-classical propositional logic* proposed by [20], [21, 22] present a quasi-classical description logic (QCDL) where the quasi-classical semantics (QC semantics) of propositional logic is extended into description logic. In [22], a transformation-based algorithm is developed to reduce the classical entailment problem of QCDL into the entailment problem. In QCDL, a new negation called *quasi-classical negation* (QC negation) (e.g., $\overline{Bird}$, expressing the complement set of those which are known not to be in *Bird*) is presented to the complement of interpretations when the (classical) negation (e.g., $\neg Bird$, expressing the set of those which are known to be in $\neg Bird$) is weakened under the QC semantics. Though $\{Bird(a), \neg Bird(a)\}$ is satisfiable, $\{Bird(a), \overline{Bird}(a)\}$ is not satisfiable in QCDL.

For a logic, the satisfiability problem (SAT) is of central importance in various areas of computer science, including theoretical computer science, complexity theory, and artificial intelligence. It is interesting and important to develop an inherent proof (system) for a logic to determine whether a set of formulae in that logic is satisfiable. Indeed, this problem has been a recurring topic of proof theory. Specially, an inherent proof of a logic (that is, a proof directly works the language of this logic without translating it into other logics) is very important to this logic since the direct proof can embody the precise characteristics of this logic in reasoning. For instance, though the consistency problem of DL can be reduced to the consistency problem of propositional dynamic logic (PDL), developing some suitable tableau-based proofs for DLs has always been attracted in the recent years [1]. One may wonder what the inherent proof of QCDL is like although a transformation-based algorithm is presented to reduce the entailment problem of QCDL into the entailment problem of DL in [22]. The main goal of this paper is to answer this question. However, it is not trivial to develop a proof of QCDL for the satisfiability problem since the QC negation of conjunction and disjunction of concepts cannot be pushed inwards in a simple way (i.e., using DeMorgan law) which the negation obeys. Moreover, since most of standard tableau-based proofs of DLs work for a KB with the empty TBox, we must find a suitable transformation to reduce a KB into a KB without any TBox if we aim to build our proof on the standard tableau-based proof.

This paper systematically studies two satisfiability problems of QCDL, namely, *QC-coherency* and *QC-consistency*, as a complement of [22]. Firstly, we reduce the QC-coherency problem to the QC-consistency problem. Secondly, we prove that for any ABox $\mathcal{A}$, any TBox $\mathcal{T}$, and, an RBox $\mathcal{R}$, we can construct a new RBox $\mathcal{R}_U$ such that the problem determining whether $\mathcal{A}$ is QC-consistent w.r.t. $\mathcal{T}$ and $\mathcal{R}$ is equivalent to the problem determining whether $\mathcal{A}$ is QC-consistent w.r.t. $\mathcal{R}_U$. Finally, we develop a sound and complete tableau calculus for determining the QC-consistency of an ABox w.r.t. an RBox obtained from the standard tableau calculus of DL in the following way:

1. dividing the $\sqcup$-rule of the standard tableau calculus of DL into two new rules: R-rule for resolution and a new $\sqcup$-rule;
2. adding eight new expansion rules for pushing the QC negation inwards; and
3. redefining the closeness condition as $\{A, \overline{A}\}$ or $\{\neg A, \overline{\neg A}\}$ instead of the closeness condition as $\{A, \neg A\}$ of the standard tableau calculus.

Additionally, we show that the QC-entailment can be reduced to the QC-consistency and consequently our tableau calculus can be also taken as an alternative approach to the QC-entailment. In this paper, we select the DL $\mathcal{SHIQ}$ for the following reasons:

a) $\mathcal{SHIQ}$ is one of expressive DLs, which is powerful enough to encode the logic $\mathcal{DLR}$, and which can thus be used for reasoning on conceptual data models;
b) the standard tableau calculus presented [23] can be taken as a reference to construct our tableau calculus in an easy-to-understand way; and
c) the qualified number restrictions ($\mathcal{Q}$) is absent in our previous work [22].

The rest of this paper is organized as follows: Section 2 reviews the syntax and semantics of QC-$\mathcal{SHIQ}$. Section 3 discusses two satisfiability problems. Section 4 presents a tableau calculus for QC-$\mathcal{SHIQ}$ and Section 5 implements the paraconsistent OWL reasoner QC-OWL. Finally, Section 6 discusses related works and Section 7 summarizes this paper. All proofs are presented in the Appendix.

2 QUASI-CLASSICAL DESCRIPTION LOGICS

In this section, we introduce the syntax and semantics of an expressive *quasi-classical DL*: QC-$\mathcal{SHIQ}$. We omit the introduction of the syntax and semantics of $\mathcal{SHIQ}$. For more comprehensive background knowledge of DLs and QCDLs, we refer the reader to [1, 21, 22].

Let N_C, N_R, and N_I be countably infinite sets of concept names, role names, and individual names. Let $\mathbf{R}$ be a set of role names with a subset $\mathbf{R}_+ \subseteq \mathbf{R}$ of transitive role names. The set of roles is $\mathbf{R} \cup \{R^- \mid R \in \mathbf{R}\}$. The function $Inv(\cdot)$ is defined on the sets of roles as follows: $Inv(R) = R^-$ and $Inv(R^-) = R$, where R is a role name. The function $Trans(R) = true$ if and only if (iff) $R \in \mathbf{R}_+$ or $Inv(R) \in \mathbf{R}_+$.

For roles R_1 and R_2, a *role axiom* is a *role inclusion*, which is of the form $R_1 \sqsubseteq R_2$ for $R_1, R_2 \in \mathbf{R}$. An *RBox* or a *role hierarchy* $\mathcal{R}$ is a finite set of role axioms. Let $\sqsubseteq^{*}$ be the reflexive-transitive closure of $\sqsubseteq$ on $\mathcal{R} \cup \{Inv(R) \sqsubseteq Inv(S) \mid R \sqsubseteq S \in \mathcal{R}\}$ as follows: $\{(R_1, R_2) \mid R_1 \sqsubseteq R_2 \in \mathcal{R}$ or $Inv(R_1) \sqsubseteq Inv(R_2) \in \mathcal{R}\}$. A role R is called a *sub-role* (respectively, *super-role*) of a role S if $R \sqsubseteq^{*} S$ (respectively, $S \sqsubseteq^{*} R$). A role S is *simple* if it is neither transitive nor has any transitive sub-roles.

The set of complex concepts is the smallest set such that

- each concept name $A \in N_C$ is a concept;
- C, D are concepts, R is a role, S is a simple role, and n is a nonnegative integer, then $C \sqcap D$, $C \sqcup D$, $\neg C$, $\overline{C}$, $\forall R.C$, $\exists R.C$, $\leq n\, S.C$, and, $\geq n\, S.C$ are also concepts.

Note that the syntax of QC-$\mathcal{SHIQ}$ is slightly different from the syntax of classical $\mathcal{SHIQ}$ by introducing a new version of the negation called the *quasi-classical negation* (QC negation, $\overline{C}$) of a concept C when the negation is weakened to tolerate inconsistency. The QC negation is inspired from a so-called *total negation* which is not a syntax constructor but a semantic "operation" [15]. For instance, $\overline{\neg Bird}(tweety)$ means that *tweety* is known to not be a member of $\neg Bird$, while *tweety* is not necessarily known to be a member of *Bird* under non-classical semantics. Intuitively, the QC negation reverses both the information of being true and of being false.

In this paper, let A, B (or with A_i, B_i) be concept names, C, D (or with C_i, D_i) (general) concepts, R (or with R_i) a role name, S (or with S_i) a simple role, a, b (or with a_i, b_i) individual names, unless otherwise stated.

A *TBox* or a *terminology* $\mathcal{T}$ is a finite set of *general concept inclusion axioms* (GCIs) $C \sqsubseteq D$. It is the statement about how concepts are related to each other.

In an ABox, one describes a specific state of affairs of an application domain in terms of concepts and roles. An ABox $\mathcal{A}$ is a finite set of assertions of the forms $C(a)$ (*concept assertion*), $R(a,b)$ (role assertion), and $a \neq b$ (*inequality assertion*). In general, axioms are GCIs, role axioms, concept assertions, role assertions, and inequality assertions. A knowledge base (KB) $\mathcal{K}$ is a triple $(\mathcal{T}, \mathcal{R}, \mathcal{A})$.

Semantically, we propose to use two types of interpretations: *weak interpretations* and *strong interpretations*. The former is essentially a variant of four-valued interpretations [15]. Before introducing these two types of interpretations, we define a notion called *base interpretations* [22]. A *base interpretation* $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where the domain $\Delta^{\mathcal{I}}$ is a set of individuals and the assignment function $\cdot^{\mathcal{I}}$ assigns each concept name A to an ordered pair $\langle +A, -A\rangle$ where $\pm A \subseteq \Delta^{\mathcal{I}}$; each role R to $\langle +R, -R\rangle$ where $\pm R \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$; each inverse role R^- to an ordered pair $\langle +R^-, -R^-\rangle$ where $\pm R^- = \{(y,x) \mid (x,y) \in \pm R\}$. Note that $+X$ and $-X$ are not necessarily disjoint when $X \in \{A, R, R^-\}$. Intuitively, $+X$ is the set of elements known to be in the extension of X while $-X$ is the set of elements known to be not in the extent of X. Compared with classical interpretations, each base interpretation maps an object X to a pair of sets of elements but not to a set of elements.

A *weak interpretation* $\mathcal{I}$ is a base interpretation $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that the assignment function $\cdot^{\mathcal{I}}$ satisfies the conditions as follows:

1. $(C \sqcap D)^{\mathcal{I}} = \langle +C \cap +D, -C \cup -D\rangle$;
2. $(C \sqcup D)^{\mathcal{I}} = \langle +C \cup +D, -C \cap -D\rangle$;
3. $(\neg C)^{\mathcal{I}} = \langle -C, +C\rangle$;
4. $(\overline{C})^{\mathcal{I}} = \langle \Delta^{\mathcal{I}} \setminus +C, \Delta^{\mathcal{I}} \setminus -C\rangle$;
5. $(\exists R.C)^{\mathcal{I}} = \langle \{x \mid \exists y, (x,y) \in +R \text{ and } y \in +C\}, \{x \mid \forall y, (x,y) \in +R \text{ implies } y \in -C\}\rangle$;
6. $(\forall R.C)^{\mathcal{I}} = \langle \{x \mid \forall y, (x,y) \in +R \text{ implies } y \in +C\}, \{x \mid \exists y, (x,y) \in +R \text{ and } y \in -C\}\rangle$;
7. $(\geq n\, S.C)^{\mathcal{I}} = \langle \{x \mid \sharp(\{y.(x,y) \in +R\} \cap +C) \geq n\}, \{x \mid \sharp(\{y.(x,y) \in +R\} \cap (\Delta^{\mathcal{I}} \setminus -C)) < n\}\rangle$;
8. $(\leq n\, S.C)^{\mathcal{I}} = \langle \{x \mid \sharp(\{y.(x,y) \in +R\} \cap (\Delta^{\mathcal{I}} \setminus -C)) \leq n\}, \{x \mid \sharp(\{y.(x,y) \in +R\} \cap +C) > n\}\rangle$. We use $\overline{+C}$ to denote $\Delta^{\mathcal{I}} \setminus +C$ and we use $\overline{-C}$ to denote $\Delta^{\mathcal{I}} \setminus -C$, then, $(\overline{C})^{\mathcal{I}} = \langle \overline{+C}, \overline{-C}\rangle$, where $C^{\mathcal{I}} = \langle +C, -C\rangle$.

Let $\mathcal{I}$ be a weak interpretation. A *weak satisfaction relation* $(\models_w)$ is determined by weak interpretation as follows: let $X^{\mathcal{I}} = \langle +X, -X\rangle$ where $X \in \{C, D, R, R_1, R_2\}$,

1. $\mathcal{I} \models_w R_1 \sqsubseteq R_2$ if $+R_1 \subseteq +R_2$;
2. $\mathcal{I} \models_w C \sqsubseteq D$ if $+C \subseteq +D$;
3. $\mathcal{I} \models_w C(a)$ if $a^{\mathcal{I}} \in +C$;
4. $\mathcal{I} \models_w R(a,b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in +R$; and
5. $\mathcal{I} \models_w a \neq b$ if $a^{\mathcal{I}} \neq b^{\mathcal{I}}$.

A *strong interpretation* $\mathcal{I}$ is a base interpretation $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that the assignment function $\cdot^{\mathcal{I}}$ satisfies the conditions in weak interpretations except that the conjunction and the disjunction of concepts are interpreted as follows: let $C^{\mathcal{I}} = \langle +C, -C \rangle$ and $D^{\mathcal{I}} = \langle +D, -D \rangle$, we define

$$(C \sqcap D)^{\mathcal{I}} = \left\langle +C \cap +D, (-C \cup -D) \cap \left(-C \cup \overline{+D}\right) \cap \left(\overline{+C} \cup -D\right) \right\rangle;$$

$$(C \sqcup D)^{\mathcal{I}} = \left\langle (+C \cup +D) \cap \left(\overline{-C} \cup +D\right) \cap \left(+C \cup \overline{-D}\right), -C \cap -D \right\rangle.$$

Compared with the weak interpretation, the strong interpretation of disjunction of concepts tightens the condition that an individual is known to belong to a concept. Similarly, we can define the strong satisfaction relation, denoted by $\models_s$, in terms of strong interpretations.

The definition of $\models_s$ is the same as the definition of $\models_w$ except for GCIs. Formally, let $\mathcal{I}$ be a strong interpretation and C, D be two concepts. We define $\mathcal{I} \models_s C \sqsubseteq D$ if $\overline{-C} \subseteq +D$ and $+C \subseteq +D$ and $-D \subseteq -C$ where $C^{\mathcal{I}} = \langle +C, -C \rangle$ and $D^{\mathcal{I}} = \langle +D, -D \rangle$.

Let $\mathcal{K}$ be a KB and ϕ be an axiom. We say $\mathcal{K}$ *quasi-classically entails* (QC entails) ϕ, denoted $\mathcal{K} \models_Q \phi$, if for each base interpretation $\mathcal{I}$, for each axiom φ in $\mathcal{K}$, $\mathcal{I} \models_s \varphi$ implies $\mathcal{I} \models_w \phi$. In this case, $\models_Q$ is called *QC-entailment*. The QC-entailment can bring more reasonable conclusions than other existing paraconsistent methods [22]. For instance, recalling the example in the Introduction, let $\mathcal{A} = \{\neg \mathit{Student}(\mathit{Wade}), \mathit{Student} \sqcup \mathit{Staff}(\mathit{Wade})\}$, we can conclude that $\mathcal{A} \models_Q \mathit{Staff}(\mathit{Wade})$. It is reasonable to expect that *Wade* is a staff member which can be inferred from two facts that *Wade* is a student or staff member in a university and we know that *Wade* is not a student. However, this conclusion cannot be inferred under four-valued DLs [15].

3 THE SATISFIABILITY PROBLEM OF QCDLS

In this section, we discuss two satisfiability problems in QCDLs, namely, *coherency* and *consistency*. To distinguish notions in QCDLs from similar notions in DLs, we will put the words "QC" in front of each notion.

To describe two such problems, we first define the notion of *QC-model* of a KB in QCDLs.

Let $\mathcal{K}$ be a KB and $\mathcal{I}$ be a base interpretation. $\mathcal{I}$ is a *QC-model* of $\mathcal{K}$ if for all axioms φ in $\mathcal{K}$, $\mathcal{I} \models_s \varphi$. We use $Mod^Q(\mathcal{K})$ to denote the collection of all models of $\mathcal{K}$.

Now, we can define two satisfiability problems as follows:

- Let $\mathcal{T}$ be a TBox and $\mathcal{R}$ be an RBox. A concept C is *QC-satisfiable* w.r.t. $\mathcal{T}$ and $\mathcal{R}$ if there exists some QC-model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{R}$ such that $+C \neq \emptyset$ where $C^{\mathcal{I}} = \langle +C, -C \rangle$; *QC-unsatisfiable* w.r.t. $\mathcal{T}$ and $\mathcal{R}$ otherwise. A KB is *QC-coherent* if there exists some QC-satisfiable concept names w.r.t. its TBox and its RBox and *QC-incoherent* otherwise.

- Let $\mathcal{K}$ be a KB. $\mathcal{K}$ is *QC-consistent* if there exists some QC-model $\mathcal{I}$ of $\mathcal{K}$, that is, $Mod^Q(\mathcal{K}) \neq \emptyset$; it is *QC-inconsistent* otherwise.

Additionally, we can define that an ABox is *QC-consistent* (or *QC-inconsistent*) w.r.t. a TBox and an RBox if the KB consisting of the three of them is QC-consistent (or QC-inconsistent).

For instance, $\mathcal{A}' = \{\neg Student(Wade), Student \sqcup Staff(Wade), \overline{\neg Staff}(Wade)\}$ is QC-consistent while $\mathcal{A}'' = \{\neg Student(Wade), Student \sqcup Staff(Wade), \overline{Staff}(Wade)\}$ is QC-inconsistent.

The QC-coherency problem can be reduced into the QC-consistency problem.

Proposition 3.1. Let $\mathcal{T}$ be a TBox and $\mathcal{R}$ an RBox. For any concept C, C is QC-unsatisfiable w.r.t. $\mathcal{T}$ and $\mathcal{R}$ iff the KB $(\mathcal{T}, \mathcal{R}, \{\overline{C}(\tau)\})$ is QC-consistent where τ is a fresh individual name.

The QC-entailment problem discussed in [22] determines whether a KB can quasi-classically entail an axiom. The following result shows that two basic QC-entailment problems can be reduced into the QC-inconsistency problem.

Proposition 3.2. Let $\mathcal{T}$ be a TBox, $\mathcal{R}$ an RBox, $\mathcal{A}$ an ABox, and, C, D concepts. The followings hold.

- $(\mathcal{T}, \mathcal{R}, \mathcal{A}) \models_Q C \sqsubseteq D$ iff $C \sqcap \overline{D}$ is QC-unsatisfiable w.r.t. $\mathcal{T}$ and $\mathcal{R}$.
- $(\mathcal{T}, \mathcal{R}, \mathcal{A}) \models_Q C(a)$ iff $(\mathcal{T}, \mathcal{R}, \mathcal{A} \cup \{\overline{C}(a)\})$ is QC-inconsistent.

For instance, consider the ABox $\mathcal{A} = \{\neg Student(Wade), Student \sqcup Staff(Wade)\}$. Let $Staff(Wade)$ be an axiom. We can conclude that $\mathcal{A} \models_Q Staff(Wade)$ since $(\mathcal{A}'' =)\ \mathcal{A} \cup \{\overline{Staff}(Wade)\}$ is QC-inconsistent. However, $\mathcal{A} \not\models_Q \neg Staff(Wade)$ since $(\mathcal{A}''' =)\ \mathcal{A} \cup \{\overline{\neg Staff}(Wade)\}$ is QC-consistent.

Two satisfiability problems of a KB can be reduced into two satisfiability problems of a KB with the empty TBox.

Let U be a transitive super-role of all roles occurring in $\mathcal{T}$ and their respective inverses called the *universal role* [23]. We define the base interpretation of U as follows: $U^{\mathcal{I}} = \langle \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}, \emptyset \rangle$ for any base interpretation $\mathcal{I}$.

Proposition 3.3. Let $\mathcal{T}$ be a TBox, $\mathcal{R}$ an RBox, and $\mathcal{A}$ an ABox. We define $C_{\mathcal{T}} := \bigsqcap_{C_i \sqsubseteq D_i \in \mathcal{T}} \neg C_i \sqcup D_i$. Let $\mathcal{R}_U := \mathcal{R} \cup \{R \sqsubseteq U, Inv(R) \sqsubseteq U \mid R$ occurs in $\mathcal{T}, C, D, \mathcal{A}$, or $\mathcal{R}\}$. Then the followings hold.

- C is QC-satisfiable w.r.t. $\mathcal{T}$ and $\mathcal{R}$ iff $C \sqcap C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}$ is QC-satisfiable w.r.t. $\mathcal{R}_U$.
- $(\mathcal{T}, \mathcal{R}, \mathcal{A}) \models_Q C \sqsubseteq D$ iff $C \sqcap \overline{D} \sqcap C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}$ is QC-unsatisfiable w.r.t. $\mathcal{R}_U$.
- $\mathcal{A}$ is QC consistent w.r.t. $\mathcal{R}$ and $\mathcal{T}$ iff $\mathcal{A} \cup \{C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}(a) \mid a$ occurs in $\mathcal{A}\}$ is QC-consistent w.r.t. $\mathcal{R}_U$.

In a short, the problem of determining the QC-coherency and the QC-consistency of a KB (even the QC-entailment between a KB and an axiom) can be equivalent to the problem of determining the QC-consistency of an ABox w.r.t. an RBox, which is in ExpTime-complete since this problem can be reduced into the problem of determining of the consistency of an ABox w.r.t. an RBox [22].

At the end of this section, we use an example in to illustrate that classical inconsistency can be tolerated in QCDL.

Example 3.1. Let $\mathcal{K}_t = (\mathcal{T}_t, \mathcal{A}_t)$ be a KB where the TBox $\mathcal{T}_t = \{\mathit{Fish} \sqsubseteq \exists \mathit{hasOrgan}.\mathit{Gill}, \exists\, \mathit{eat}.\mathit{Fish} \sqsubseteq \mathit{Piscivore}\}$ and the ABox $\mathcal{A}_t = \{\mathit{eat}(\mathit{ursidae}, \mathit{salmon}), \mathit{Fish}(\mathit{salmon}), \neg\mathit{Piscivore}(\mathit{ursidae})\}$. Here *Fish*, *Gill*, and *Piscivore* are concepts; *hasOrgan* and *eat* are roles; and *ursidae* and *salmon* are individuals. The KB $\mathcal{K}_t$ tells us: a fish has an organ named gill; a piscivore is an animal which eats primarily fish; a ursidae eats some salmon; and salmon is a fish.

We can infer $\mathit{Piscivore}(\mathit{ursidae})$ from $\mathit{Eat}(\mathit{ursidae}, \mathit{salmon})$ and $\mathit{Fish}(\mathit{salmon})$ while $\mathit{Piscivore}(\mathit{ursidae})$ is an axiom of $\mathcal{K}_t$. As a result, $\mathcal{K}_t$ is (classical) inconsistent.

Let $\Delta = \{\mathit{ursidae}, \mathit{salmon}, g, f, g_f, \ldots\}$ be a domain where $g, f, g_f, \ldots$ are fresh individuals and $\mathcal{I}_t$ be a base interpretation on Δ such that the followings hold.

1. $\mathit{Fish}^{\mathcal{I}_t} = \left\langle \left\{\mathit{salmon}^{\mathcal{I}_t}, f^{\mathcal{I}_t}\right\}, \Delta^{\mathcal{I}_t} \setminus \left\{f^{\mathcal{I}_t}\right\}\right\rangle$;
2. $\mathit{Gill}^{\mathcal{I}_t} = \left\langle \left\{g^{\mathcal{I}_t}, g_f^{\mathcal{I}_t}\right\}, \Delta^{\mathcal{I}_t} \setminus \left\{g^{\mathcal{I}_t}, g_f^{\mathcal{I}_t}\right\}\right\rangle$;
3. $\mathit{Piscivore}^{\mathcal{I}_t} = \left\langle \Delta^{\mathcal{I}_t}, \left\{\mathit{ursidae}^{\mathcal{I}_t}\right\}\right\rangle$;
4. $\mathit{hasOrgan}^{\mathcal{I}_t} = \left\langle \left\{(\mathit{salmon}^{\mathcal{I}_t}, g^{\mathcal{I}_t}), \left(f^{\mathcal{I}_t}, g_f^{\mathcal{I}_t}\right)\right\}, \emptyset\right\rangle$; and
5. $\mathit{eat}^{\mathcal{I}_t} = \left\langle \left\{\left(\mathit{ursidae}^{\mathcal{I}_t}, x\right) \mid x \in \Delta^{\mathcal{I}_t} \setminus \left\{f^{\mathcal{I}_t}\right\}\right\}, \emptyset\right\rangle$.

Then we can conclude:

1. $(\exists\, \mathit{eat}.\mathit{Fish})^{\mathcal{I}_t} = \left\langle \left\{\mathit{ursidae}^{\mathcal{I}_t}\right\}, \Delta^{\mathcal{I}_t} \setminus \left\{f^{\mathcal{I}_t}\right\}\right\rangle$ and
2. $(\exists\, \mathit{hasOrgan}.\mathit{Gill})^{\mathcal{I}_t} = \left\langle \left\{\mathit{salmon}^{\mathcal{I}_t}, f^{\mathcal{I}_t}\right\}, \emptyset\right\rangle$.

It is not difficult to show that $\mathcal{I}_t$ is a QC-model of $\mathcal{K}_t$. Therefore, $\mathcal{K}_t$ is QC-consistent.

Let us consider some reasoning tasks:

- $\mathcal{K}_t \models_Q \neg\mathit{Piscivore}(\mathit{ursidae})$ since the KB $\mathcal{K}_t^1 = \left(\mathcal{T}_t, \mathcal{A}_t \cup \left\{\overline{\neg\mathit{Piscivore}}(\mathit{ursidae})\right\}\right)$ is QC-inconsistent.
- $\mathcal{K}_t \models_Q \exists\,\mathrm{eat}.\mathit{Fish}(\mathit{ursidae})$ since the KB $\mathcal{K}_t^2 = \left(\mathcal{T}_t, \mathcal{A}_t \cup \left\{\overline{\exists\,\mathrm{eat}.\mathit{Fish}}(\mathit{ursidae})\right\}\right)$ is QC-inconsistent.
- $\mathcal{K}_t \models_Q \exists\, \mathit{eat}.\mathit{Fish} \sqsubseteq \mathit{Piscivore}$ since for some fresh individual τ, the KB $\mathcal{K}_t^3 = \left(\mathcal{T}_t, \mathcal{A}_t \cup \left\{\left(\exists\, \mathit{eat}.\mathit{Fish} \sqcap \overline{\mathit{Piscivore}}\right)(\tau)\right\}\right)$ is QC-inconsistent.
- $\mathcal{K}_t \not\models_Q \mathit{Fish}(\mathit{ursidae})$ since the KB $\mathcal{K}_t^4 = \left(\mathcal{T}_t, \mathcal{A}_t \cup \left\{\overline{\mathit{Fish}}(\mathit{ursidae})\right\}\right)$ is QC-consistent.

As Example 3.1 shows, in QCDL, all axioms in the KB can be inferred while other axioms are restricted in inferring new conclusions so that we can get more conclusions without the problem of explosive inference [22].

4 TABLEAU FOR THE QC-CONSISTENCY PROBLEM

In this section, we will develop a decidable, sound and complete tableau calculus for determining whether an ABox is QC-consistent w.r.t. an RBox.

The standard tableau calculus [23] generally consists of four modules as follows:

1. negation normal form (NNF, that is, the negation only occurs in the front of concept name) of concepts to be input;
2. blocking technique ensuring termination;
3. expansion rules as trigger mechanisms; and
4. closedness conditions as determining soundness.

To build a proof system for checking the QC-consistency based on tableau calculus, we need to do the followings:

1. select suitable NNF for concepts;
2. develop new expansion rules; and
3. define new closeness condition.

Firstly, we show that each QCDL concept is equivalent to its NNF. Let C, D be two concepts. We say C is *equivalent to* D, denoted by $C \equiv_s D$, if for any strong interpretation $\mathcal{I}$, $C^{\mathcal{I}} = D^{\mathcal{I}}$. That is, $+C = +D$ and $-C = -D$ where $C^{\mathcal{I}} = \langle +C, -C \rangle$ and $D^{\mathcal{I}} = \langle +D, -D \rangle$.

Proposition 4.1. The followings hold.

1. $\neg\overline{C} \equiv_s \overline{\neg C}$;
2. $\neg\neg C \equiv_s C$;
3. $\neg(C \sqcup D) \equiv_s \neg C \sqcap \neg D$;
4. $\neg(C \sqcap D) \equiv_s \neg C \sqcup \neg D$;
5. $\neg\exists R.C \equiv_s \forall R.\neg C$;
6. $\neg\forall R.C \equiv_s \exists R.\neg C$;
7. $\neg(\leq n\, S.C) \equiv_s \geq n{+}1\, S.C$; and
8. $\neg(\geq n{+}1\, S.C) \equiv_s \leq n\, S.C$.

It is not difficult to conclude that a concept can be translated into its NNF in a polynomial time [23]. However, it is difficult to push QC negation inward like negation since De Morgan's law is no longer true. Neither $\overline{C \sqcap D} \equiv_s \overline{C} \sqcup \overline{D}$ nor $\overline{C \sqcup D} \equiv_s \overline{C} \sqcap \overline{D}$ always holds. For instance, let $\Delta = \{a, b, c\}$ be a domain and $C^{\mathcal{I}} =$

$\langle\{a,b\},\{a\}\rangle$ and $D^{\mathcal{I}} = \langle\{b\},\{c\}\rangle$. Thus $\overline{C}^{\mathcal{I}} = \langle\{c\},\{b,c\}\rangle$ and $\overline{D}^{\mathcal{I}} = \langle\{a,c\},\{a,b\}\rangle$. $\overline{C \sqcap D}^{\mathcal{I}} = \langle\{c\},\{a,b\}\rangle$ while $(\overline{C} \sqcup \overline{D})^{\mathcal{I}} = \langle\{c\},\{b\}\rangle$. $\overline{C \sqcup D}^{\mathcal{I}} = \langle\{a,c\},\{a,b,c\}\rangle$ while $(\overline{C} \sqcap \overline{D})^{\mathcal{I}} = \langle\{c\},\{b\}\rangle$.

Fortunately, the QC negation can be pushed inward concepts except for conjunctions and disjunctions.

Proposition 4.2. The followings hold.

1. $\overline{\overline{C}} \equiv_s C$;
2. $\overline{\exists R.C} \equiv_s \forall R.\overline{C}$;
3. $\overline{\forall R.C} \equiv_s \exists R.\overline{C}$;
4. $\overline{\leq n\, S.C} \equiv_s \geq (n{+}1)\, S.\overline{\neg C}$; and
5. $\overline{(\geq n{+}1)\, S.C} \equiv_s \leq n\, S.\overline{\neg C}$.

Given a concept C, we denote the NNF of $\neg C$ by $\sim C$. Let $clos(C)$ denote the smallest set that contains C and it is closed under sub-concepts and $\sim$. Let $\mathcal{A}$ be an ABox and $\mathcal{R}$ an RBox. We denote $clos(\mathcal{A}) := \bigcup_{a:C\in\mathcal{A}} clos(C)$.

Note that the size of $clos(\mathcal{A})$ is polynomial in the size of $\mathcal{A}$. We denote by $\mathbf{R}_{\mathcal{A}}$ the set of roles occurring in $\mathcal{A}$ and $\mathcal{R}$ together with their inverse, and by $\mathbf{I}_{\mathcal{A}}$ the set of individuals occurring in $\mathcal{A}$.

Secondly, we define a new tableau called *quasi-classical tableau* (or QC-tableau) by modifying the standard tableau.

Definition 4.1. Let $\mathcal{R}$ be an RBox and $\mathcal{A}$ be an ABox whose concepts are all in NNF. A quadruple $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ is a *QC-tableau* for $\mathcal{A}$ w.r.t. $\mathcal{R}$ if the following conditions hold.

- $\mathbf{S}$: a non-empty set of individuals;
- $\mathcal{L} : \mathbf{S} \to 2^{clos(\mathcal{A})}$ maps each element in $\mathbf{S}$ to a set of concepts which is a subset of $clos(\mathcal{A})$;
- $\mathcal{E} : \mathbf{R}_{\mathcal{A}} \to 2^{\mathbf{S}\times\mathbf{S}}$ maps each role in $\mathbf{R}_{\mathcal{A}}$ to a set of pairs of elements in $\mathbf{S}$;
- $\mathcal{J} : \mathbf{I}_{\mathcal{A}} \to \mathbf{S}$ maps individuals occurring in $\mathcal{A}$ to elements in $\mathbf{S}$.

Let $S^{\mathbb{T}}(s, C) := \{t \in \mathbf{S} \mid \langle s,t\rangle \in \mathcal{E}(S) \text{ and } t \in C\}$.

Moreover, for all $s, t \in \mathbf{S}, C, C_1, C_2 \in clos(\mathcal{A})$, $R, S \in \mathbf{R}_{\mathcal{A}}$, $\mathbb{T}$ satisfies:

P1 if $C \in \mathcal{L}(s)$ then $\overline{C} \notin \mathcal{L}(s)$;

P2 if $C_1 \sqcap C_2 \in \mathcal{L}(s)$ then $\{C_1, C_2\} \subseteq \mathcal{L}(s)$;

P3 if $C_1 \sqcup C_2 \in \mathcal{L}(s)$ then if $\sim C_i \in \mathcal{L}(s)$ for some $(i \in \{1,2\})$, then $C_{3-i} \in \mathcal{L}(s)$; otherwise $C_1 \in \mathcal{L}(s)$ or $C_2 \in \mathcal{L}(s)$;

P4 if $\forall R.C \in \mathcal{L}(s)$ and $\langle s,t\rangle \in \mathcal{E}(R)$, then $C \in \mathcal{L}(t)$;

P5 if $\exists R.C \in \mathcal{L}(s)$ then there is some $t \in \mathbf{S}$ such that $\langle s,t\rangle \in \mathcal{E}(R)$ and $C \in \mathcal{L}(t)$;

P6 if $\forall R.C \in \mathcal{L}(s)$ and $\langle s,t\rangle \in \mathcal{E}(R)$ for some $R \underline{\circledast} S$ with $Trans(R)$, then $\forall R.C \in \mathcal{L}(t)$;

P7 $\langle s,t\rangle \in \mathcal{E}(R)$ iff $\langle t,s\rangle \in \mathcal{E}(Inv(R))$;

P8 if $\langle s,t\rangle \in \mathcal{E}(R)$ and $R \underline{\circledast} S$ then $\langle t,s\rangle \in \mathcal{E}(S)$;

P9 if $\leq n\, S.C \in \mathcal{L}(s)$ then $\sharp S^{\mathbb{T}}(s,C) \leq n$;

P10 if $\geq n\, S.C \in \mathcal{L}(s)$ then $\sharp S^{\mathbb{T}}(s,C) \geq n$;

P11 if $(\bowtie n\, S.C) \in \mathcal{L}(s)$ and $\langle s,t\rangle \in \mathcal{E}(S)$ then $C \in \mathcal{L}(s)$ or $\overline{C} \in \mathcal{L}(s)$;

P12 if $a : C \in \mathcal{A}$ then $C \in \mathcal{L}(\mathcal{J}(a))$;

P13 if $(a,b) : R \in \mathcal{A}$ then $\langle \mathcal{J}(a), \mathcal{J}(b)\rangle \in \mathcal{E}(R)$;

P14 if $a \neq b \in \mathcal{A}$, then $\mathcal{J}(a) \neq \mathcal{J}(b)$;

P15 if $\overline{\overline{C}} \in \mathcal{L}(s)$ then $C \in \mathcal{L}(s)$;

P16 if $\overline{C_1 \sqcap C_2} \in \mathcal{L}(s)$ then $\overline{C_1} \in \mathcal{L}(s)$ or $\overline{C_2} \in \mathcal{L}(s)$;

P17 if $\overline{C_1 \sqcup C_2} \in \mathcal{L}(s)$ then $\left\{\overline{C_1}, \overline{C_2}\right\} \subseteq \mathcal{L}(s)$, $\left\{\neg C_1, \overline{C_2}\right\} \subseteq \mathcal{L}(s)$, or, $\left\{\overline{C_1}, \neg C_2\right\} \subseteq \mathcal{L}(s)$

P18 if $\overline{\forall R.C} \in \mathcal{L}(s)$ then there is some $t \in \mathbf{S}$ such that $\langle s,t\rangle \in \mathcal{E}(R)$ and $\overline{C} \in \mathcal{L}(t)$;

P19 if $\overline{\exists R.C} \in \mathcal{L}(s)$ and $\langle s,t\rangle \in \mathcal{E}(R)$, then $\overline{C} \in \mathcal{L}(t)$;

P20 if $\overline{\leq n\, S.C} \in \mathcal{L}(s)$, then $\sharp S^{\mathbb{T}}\left(s, \overline{\neg C}\right) \geq n+1$;

P21 if $\overline{\geq n\, S.C} \in \mathcal{L}(s)$, then $\sharp S^{\mathbb{T}}\left(s, \overline{\neg C}\right) \leq n-1$;

P22 if $\overline{(\bowtie n\, S.C)} \in \mathcal{L}(s)$ then $\langle s,t\rangle \in \mathcal{E}(S)$ then $\overline{\neg C} \in \mathcal{L}(s)$ or $\neg C \in \mathcal{L}(s)$.

Here $\bowtie$ is a place-holder for both $\leq$ and $\geq$.

Intuitively, the mapping $\mathcal{J}$ is used to construct a finite model. **P1** ensures that both a concept and its QC negation do not occur in the same set together. It is not hard to show that **P2** and **P3** are introduced to characterize disjunction and conjunction concepts, **P4** and **P5** for exists restriction and value restriction concepts, **P7** for inverse roles, **P6** and **P8** for transitive roles, **P9–P11** for number restriction concepts, **P12** for concept assertions, **P13** for role assertions, **P14** for individual inequality assertions, and **P15–P22** for the QC negation of concepts, respectively. Compared with the standard tableau, we revise **P1**, **P3** and **P11** and add nine new expansion rules for QC negation of concepts **P15–P22**.

We can show that each QC-consistent ABox w.r.t. an RBox has at least one QC-tableau.

Theorem 4.1. Let $\mathcal{R}$ be an RBox and $\mathcal{A}$ an ABox. $\mathcal{A}$ is QC consistent w.r.t. $\mathcal{R}$ iff there is a QC-tableau for $\mathcal{A}$ w.r.t. $\mathcal{R}$.

Let $\mathcal{R}$ be an Rbox and $\mathcal{A}$ an ABox. Without loss of generality, we assume that all concepts occurring $\mathcal{A}$ and $\mathcal{R}$ are in QC-NNF. In the following, we develop a tableau calculus to decide whether the QC tableau of $\mathcal{A}$ w.r.t. $\mathcal{R}$ exists, called the *QC-tableau calculus*.

Following the standard tableau calculus, we first introduce a basic data structure so-called forest to support our QC-tableau calculus. A *completion forest* $\mathcal{F}$ for $\mathcal{A}$ w.r.t. $\mathcal{R}$ is a collection of trees whose distinguished root nodes are possibly connected by edges in an arbitrary way. $\mathcal{F}$ is constructed as follows:

1. each node x is labeled with a set $\mathcal{L}(x) \subseteq clos(\mathcal{A})$;
2. each edge $\langle x, y\rangle$ is labeled with a set $\mathcal{L}(\langle x, y\rangle) \in R$ of (possibly inverse) roles occurring in $\mathcal{A}$; and
3. an explicit inequality relation $\dot{\neq}$ on nodes is employed to capture the inequality between nodes while an explicit equality relation $\doteq$ is implicitly assumed to be symmetric.

A node y is called an *R-successor* of x if for some R' with $R' \underline{\sqsubseteq}^{*} R$, either y is a successor of x or $R' \in \mathcal{L}(\langle x, y\rangle)$. In this sense, x is called an *$Inv(R)$-predecessor* of y. *R-neighbors* and *R-ancestors* are defined in the usual way. For a role R, a concept C and a node x in $\mathcal{F}$, we let $S^{\mathcal{F}}(x, C) := \{y \mid y$ is S-successor of x and $C \in \mathcal{L}(y)\}$. Note that a so-called *blocking* technique is developed to ensure termination and correctness. Because of the presence of transitive roles and inverse roles, we need to employ the technique of *dynamic blocking* presented by [23].

Next, we briefly recall the blocking technique. A node is *blocked* iff it is not a root node and it is either directly or indirectly blocked. A node x is *directly blocked* iff none of its ancestors is blocked, and it has ancestors x', y and y' such that

1. y is not a root node;
2. x is a successor of x' and y is a successor of y';
3. $\mathcal{L}(x) = \mathcal{L}(y)$ and $\mathcal{L}(x') = \mathcal{L}(y')$; and
4. $\mathcal{L}(\langle x', x\rangle) = \mathcal{L}(\langle y', y\rangle)$.

A node y is *indirectly blocked* iff one of its ancestors is blocked, or it is a successor of a node x and $\mathcal{L}(\langle x, y\rangle) = \emptyset$. Now, we refine the conditions of clash which is a key notion of tableau calculus. A node $\mathcal{L}(x)$ contains a *clash* if:

1. for some concept name $A \in N_C$, $\{A, \overline{A}\} \subseteq \mathcal{L}(x)$ or $\{\neg A, \overline{\neg A}\} \subseteq \mathcal{L}(x)$;
2. for some role S, $\leq n\, S.C \in \mathcal{L}(x)$ and x has $n{+}1$ S-neighbours $y_0, \ldots, y_n$ with $C \in \mathcal{L}(y_i)$ such that $y_i \dot{\neq} y_j)$ for all $0 \leq i < j \leq n$.

Note that the first condition of clash $\{A, \neg A\} \subseteq \mathcal{L}(x)$ for the standard tableau calculus is replaced with $\{A, \overline{A}\} \subseteq \mathcal{L}(x)$ or $\{\neg A, \overline{\neg A}\} \subseteq \mathcal{L}(x)$.

A completion forest is *clash-free* if none of its nodes contains a clash, and it is complete if no *QC expansion rules* from Table 2 (see below) can be applied to it.

The QC-tableau calculus initializes a completion forest $\mathcal{F}_{\mathcal{A}}$ consisting only of root nodes. More precisely, $\mathcal{F}_{\mathcal{A}}$ contains a root node x_0^i for each individual $a^i \in \mathbf{I}_{\mathcal{A}}$, and an edge $\langle x_0^i, x_0^j\rangle$ if $\mathcal{A}$ contains an assertion $(a_i, a_j) : R$ for some R. The labels of these nodes and edges and the relations $\dot{\neq}$ and $\doteq$ are initialized as follows:

1. $\mathcal{L}(x_i^0) := \{C \mid a_i : C \in \mathcal{A}\}$;
2. $\mathcal{L}(\langle x_0^i, x_0^j \rangle) := \{R \mid (a_i, a_j) : R \in \mathbf{R}_{\mathcal{A}}\}$; and
3. $x_0^i \neq x_0^j \Leftrightarrow a_i \neq a_j \in \mathcal{A}$.

Here the $\doteq$-relation is initialized to be the empty set. $\mathcal{F}_{\mathcal{A}}$ is then expanded by repeatedly applying the QC expansion rules.

Comparing with that of the standard tableau calculus, there are three changes:

- The $\sqcup$-rule in the standard tableau calculus is replaced with two new rules, namely, the $\sqcup$-rule and the R-rule, which are applied to treat two cases with disjunction concepts shown in Table 2. When a node in a branch contains a disjunction concept $C_1 \sqcup C_2$ without any negation of C_1 and C_2, two new branches which contain C_1 and C_2, respectively, are added by using the $\sqcup$-rule. On the other hand, when a node in a branch contains a disjunction concept $C_1 \sqcup C_2$ and some negation of sub-concept C_1 or C_2, the sub-concept whose negation does not occur in the node will be added in the node by using the R-rule. For example, if $\{C_1 \sqcup C_2, \neg C_2\} \subseteq \mathcal{L}(x)$, then $\{C_1 \sqcup C_2, \neg C_2, C_1\} \subseteq \mathcal{L}(x)$ by using the R-rule. In this sense, the R-rule could capture resolution of disjunction concepts.
- The *choose*-rule is revised by replacing $\{C, \sim C\}$ by $\{C, \overline{C}\}$ shown in Table 2. Though $\{A, \neg A\}$ is a clash in the standard tableau calculus, it is no longer a clash in the QC-tableau calculus where $\{A, \overline{A}\}$ or $\{A, \overline{\neg A}\}$ is a clash. Moreover, we add the *c-choose*-rule to analogously capture $\overline{\bowtie n\, S.C}$.
- Seven new rules (*double*, $\overline{\sqcap}$-rule, $\overline{\sqcup}$-rule, $\overline{\exists}$-rule, $\overline{\forall}$-rule, $\overline{\leq}$-rule, and $\overline{\geq}$-rule) are introduced to push the QC negation inwards.

Additionally, though other expansion rules are not changed formally, they can handle concepts with the QC negation technically.

Now, we are ready to present the QC-tableau calculus as follows: it starts with the completion forest $\mathcal{F}_{\mathcal{A}}$ and it then exhaustively applies the expansion rules till it terminates in the situation that no rule can be applied or a clash occurs, it answers "there is a QC-tableau of $\mathcal{A}$ w.r.t. $\mathcal{R}$" iff the expansion rules can be applied in such a way that they yield a complete and clash-free completion forest; and it answers "there is not any QC-tableau of $\mathcal{A}$ w.r.t. $\mathcal{R}$" otherwise.

The QC-tableau calculus always terminates.

Theorem 4.2. Let $\mathcal{R}$ be an RBox and $\mathcal{A}$ an ABox. The QC-tableau calculus terminates when started for $\mathcal{A}$ and $\mathcal{R}$.

The following theorem states that the QC-tableau calculus is sound and complete.

Theorem 4.3. Let $\mathcal{R}$ be an RBox and $\mathcal{A}$ an ABox. $\mathcal{A}$ has a QC-tableau w.r.t. $\mathcal{R}$ iff the QC-tableau calculus applied to $\mathcal{A}$ w.r.t. $\mathcal{R}$ yields a complete and clash-free completion forest.

$\sqcap$-rule	If	1. $C_1 \sqcap C_2 \in \mathcal{L}(x)$, x is not blocked, and 2. $\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$.
	Then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{C_1, C_2\}$.
$\exists$-rule	If	1. $\exists R.C \in \mathcal{L}(x)$, x is not blocked, and 2. x has no R-neighbor y with $C \in \mathcal{L}(y)$.
	Then	create a new node y with $\mathcal{L}(\langle x, y\rangle) := \{R\}$ and $\mathcal{L}(y) := \{C\}$.
$\forall$-rule	If	1. $\forall R.C \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. there is some R-neighbor y of x with $C \notin \mathcal{L}(y)$
	Then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{C\}$
$\forall_+$-rule	If	1. $\forall R.C \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. there is some R with $Trans(R)$ and $R \underline{\boxast} S$, and 3. there is an R-neighbor y of x with $\forall R.C \notin \mathcal{L}(y)$.
	Then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{\forall R.C\}$.
$\geq$-rule	If	1. $\geq n\, S.C \in \mathcal{L}(x)$, x is not blocked, and; 2. there is no n S-neighbors $y_1, \ldots, y_n$ such that $C \in \mathcal{L}(y_i)$ and $y_i \neq y_j$ for $1 \leq i < j \leq n$.
	Then	create n new nodes $y_1, \ldots, y_n$ with $\mathcal{L}(\langle x, y_i\rangle) := \{S\}$ $\mathcal{L}(y_i) := \{C\}$, and $y_i \neq y_j$ for $1 \leq i < j \leq n$.
$\leq$-rule	If	(1) $\leq n\, S.C \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. $\sharp S^{\mathcal{F}}(x, C) > n$, there are S-neighbors y, z of x with not $y \neq z$, y is neither a root node nor an ancestor of z and $C \in \mathcal{L}(y) \cap \mathcal{L}(z)$.
	Then	1. $\mathcal{L}(z) := \mathcal{L}(z) \cup \mathcal{L}(y)$; and 2. if z is an ancestor of x then $\mathcal{L}(\langle z, x\rangle) := \mathcal{L}(\langle z, x\rangle) \cup Inv(\mathcal{L}(\langle x, y\rangle))$ else $\mathcal{L}(\langle x, z\rangle) := \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}(\langle x, y\rangle)$; 3. $\mathcal{L}(\langle x, y\rangle) := \emptyset$; 4. set $u \neq z$ for all u with $u \neq y$.
$\leq_r$-rule	If	1. $\leq n\, S.C \in \mathcal{L}(x)$, and 2. $\sharp S^{\mathcal{F}}(x, C) > n$; 3. there are S-neighbors y, z of x which are both root nodes, $C \in \mathcal{L}(y) \cap \mathcal{L}(z)$ with not $y \neq z$.
	Then	1. $\mathcal{L}(z) := \mathcal{L}(z) \cup \mathcal{L}(y)$; and 2. for all edge $\langle y, w\rangle$: (i) if the edge $\langle z, w\rangle$ does not exist, create it with $\mathcal{L}(\langle z, w\rangle) := \emptyset$; (ii) $\mathcal{L}(\langle z, w\rangle) := \mathcal{L}(\langle z, w\rangle) \cup \mathcal{L}(\langle y, w\rangle)$; 3. for all edge $\langle w, y\rangle$: (i) if the edge $\langle w, z\rangle$ does not exist, create it with $\mathcal{L}(\langle w, z\rangle) := \emptyset$; (ii) $\mathcal{L}(\langle w, z\rangle) := \mathcal{L}(\langle w, z\rangle) \cup \mathcal{L}(\langle w, y\rangle)$; 4. set $\mathcal{L}(y) := \emptyset$ and remove all edges to/front y; 5. set $u \neq z$ for all u with $u \neq y$; 6. set $y \doteq z$.

Table 1. QC Expansion Rules

choose	If	1. $\bowtie n\, S.C \in \mathcal{L}(x)$, x is not indirectly blocked, 2. there is some S-neighbor y of x with $\{C, \approx C\} \cap \mathcal{L}(y) = \emptyset$.
	Then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{E\}$ for some $E \in \{C, \overline{C}\}$.
R-rule	If	1. $C_1 \sqcup C_2 \in \mathcal{L}(x)$, x is not blocked, and 2. $\sim C_i \in \mathcal{L}(x)$ for some $i \in \{1,2\}$).
	Then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{C_{3-i}\}$.
$\sqcup$-rule	If	1. $C_1 \sqcup C_2 \in \mathcal{L}(x)$, x is not blocked, and 2. $\{C_1, C_2, \sim C_1, \sim C_2\} \cap \mathcal{L}(x) = \emptyset$.
	Then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{E\}$ for some $E \in \{C_1, C_2\}$.
double	If	1. $\overline{\overline{C}} \in \mathcal{L}(x)$, x is not blocked, and
	Then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{C\}$.
$\overline{\sqcap}$-rule	If	1. $\overline{C_1 \sqcap C_2} \in \mathcal{L}(x)$, x is not blocked, and 2. $\{\overline{C_1}, \overline{C_2}\} \not\subseteq \mathcal{L}(x) = \emptyset$.
	Then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{\overline{E}\}$ for some $E \in \{C_1, C_2\}$.
$\overline{\sqcup}$-rule	If	1. $\overline{C_1 \sqcup C_2} \in \mathcal{L}(x)$, x is not blocked, and
	Then	$\mathcal{L}(x) := \mathcal{L}(x) \cup W$ for some $W \in \{\{\overline{C_1}, \overline{C_2}\}, \{\sim C_1, \overline{C_2}\}, \{\overline{C_1}, \sim C_2\}\}$.
$\overline{\exists}$-rule	If	1. $\overline{\exists R.C} \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. there is some R-neighbor y of x with $\overline{C} \notin \mathcal{L}(y)$.
	Then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{\overline{C}\}$.
$\overline{\forall}$-rule	If	1. $\overline{\forall R.C} \in \mathcal{L}(x)$, x is not blocked, and 2. x has no R-neighbor y with $\overline{C} \in \mathcal{L}(y)$.
	Then	create a new node y with $\mathcal{L}(\langle x, y\rangle) := \{R\}$ and $\mathcal{L}(y) := \{\overline{C}\}$.
$\overline{\geq}$-rule	If	1. $\overline{\geq n\, S.C} \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. $\sharp S^{\mathcal{F}}\left(x, \overline{\sim C}\right) > n$, there are S-neighbors y, z of x with not $y \dot{\neq} z$, y is neither a root node nor an ancestor of z and $\overline{\sim C} \in \mathcal{L}(y) \cap \mathcal{L}(z)$.
	Then	1. $\mathcal{L}(z) := \mathcal{L}(z) \cup \mathcal{L}(y)$; and 2. if z is an ancestor of x then $\mathcal{L}(\langle z, x\rangle) := \mathcal{L}(\langle z, x\rangle) \cup Inv(\mathcal{L}(\langle x, y\rangle))$ else $\mathcal{L}(\langle x, z\rangle) := \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}(\langle x, y\rangle)$; 3. $\mathcal{L}(\langle x, y\rangle) := \emptyset$; 4. set $u \dot{\neq} z$ for all u with $u \dot{\neq} y$.
$\overline{\leq}$-rule	If	1. $\overline{\leq n\, S.C} \in \mathcal{L}(x)$, x is not blocked, and; 2. there is no n S-neighbors $y_1, \ldots, y_n$ such that $\overline{\sim C} \in \mathcal{L}(y_i)$ and $y_i \dot{\neq} y_j$ for $1 \leq i < j \leq n$.
	Then	create n new nodes $y_1, \ldots, y_n$ with $\mathcal{L}(\langle x, y_i\rangle) := \{S\}$ $\mathcal{L}(y_i) := \{\overline{\sim C}\}$, and $y_i \dot{\neq} y_j$ for $1 \leq i < j \leq n$.
c-choose	If	1. $\bowtie \overline{n\, S.C} \in \mathcal{L}(x)$, x is not indirectly blocked, 2. there is some S-neighbor y of x with $\{\overline{\sim C}, \sim C\} \cap \mathcal{L}(y) = \emptyset$.
	Then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{E\}$ for some $E \in \{\overline{\sim C}, \sim C\}$.

Table 2. QC Expansion Rules (cont'd)

As a result, the QC-tableau calculus can determine whether an ABox is QC-consistent w.r.t. an RBox in the following way: Given an RBox $\mathcal{R}$ and an ABox $\mathcal{A}$, $\mathcal{A}$ is QC-consistent w.r.t. $\mathcal{R}$ if the QC-tableau calculus applied to $\mathcal{A}$ w.r.t. $\mathcal{R}$ yields a complete and clash-free completion forest; and $\mathcal{A}$ is QC-inconsistent w.r.t. $\mathcal{R}$ otherwise.

At the end of this section, we use an example to illustrate how the QC-tableau calculus works in checking if a KB is QC-consistent.

Example 4.1. Let us consider an ABox $\mathcal{A}_t^*$ by revising slightly the KB $\mathcal{K}_t^1$ in Example 3.1 as follows: $\mathcal{A}_t^* = \{(\neg \mathit{Fish} \sqcup \exists \mathit{hasOrgan.Gill})(\mathit{salmon}), (\exists\, \mathit{eat.Fish} \sqcup \underline{\mathit{Piscivore}})(\mathit{ursidae}), \mathit{eat}(\mathit{ursidae}, \mathit{salmon}), \mathit{Fish}(\mathit{salmon}), \neg \mathit{Piscivore}(\mathit{ursidae}), \overline{\neg \mathit{Piscivore}}(\mathit{ursidae})\}$.

Now, we apply the QC-tableau calculus to check whether $\mathcal{A}_t^*$ is QC-consistent using the following procedures:

- By initializing the ABox $\mathcal{A}_t^*$, we have
  - $\mathcal{L}(\mathit{salmon}) = \{(\neg \mathit{Fish} \sqcup \exists \mathit{hasOrgan.Gill}), \mathit{Fish}\}$;
  - $\mathcal{L}(\mathit{ursidae}) = \left\{(\exists\, \mathit{eat.Fish} \sqcup \mathit{Piscivore}), \neg \mathit{Piscivore}, \overline{\neg \mathit{Piscivore}}\right\}$;
  - $\mathcal{L}(\langle \mathit{ursidae}, \mathit{salmon}\rangle) = \{\mathit{eat}\}$.
- By applying the R-rule in Table 2, we have
  - $\mathcal{L}(\mathit{salmon}) = \{(\exists \mathit{hasOrgan.Gill}), \mathit{Fish}\}$;
  - $\mathcal{L}(\mathit{ursidae}) = \left\{(\exists\, \mathit{eat.Fish}, \neg \mathit{Piscivore}, \overline{\neg \mathit{Piscivore}}\right\}$;
  - $\mathcal{L}(\langle \mathit{ursidae}, \mathit{salmon}\rangle) = \{\mathit{eat}\}$.
- By applying the $\exists$-rule in Table 2, we have
  - $\mathcal{L}(\mathit{salmon}) = \{(\exists \mathit{hasOrgan.Gill}), \mathit{Fish}\}$;
  - $\mathcal{L}(g) = \{\mathit{Gill}\}$;
  - $\mathcal{L}(\langle \mathit{salmon}, g\rangle) = \{\mathit{hasOrgan}\}$
  - $\mathcal{L}(\mathit{ursidae}) = \left\{(\exists\, \mathit{eat.Fish}, \neg \mathit{Piscivore}, \overline{\neg \mathit{Piscivore}}\right\}$;
  - $\mathcal{L}(\langle \mathit{ursidae}, \mathit{salmon}\rangle) = \{\mathit{eat}\}$.
- The QC-tableau calculu terminates, we obtain the completion forest $\mathcal{F}_{\mathcal{A}}$.
- $\mathcal{F}_{\mathcal{A}}$ is closed since it contains a clash $\left\{\neg \mathit{Piscivore}, \overline{\neg \mathit{Piscivore}}\right\}$.
- We can conclude that $\mathcal{A}_t^*$ is QC-inconsistent by Theorem 4.3. Indeed, we also conclude that $\mathcal{A}_t^*$ is QC-inconsistent from $\mathcal{K}_t^*$ is QC-inconsistent by Proposition 3.3.

5 EXPERIMENTS

In this section, we implement the QC-tableau calculus in the OWL inconsistency-tolerant reasoner and analyze the test results.

Implementation. Based on the QC-tableau algorithm described in the previous section, we have developed an inconsistency-tolerant reasoner named *QC-OWL*. QC-OWL can perform reasoning over both consistent and inconsistent ontologies with acceptable to very good performance for $\mathcal{SHIQ}$. It is designed and developed by following the Strategy Pattern, a behavioral design pattern, and it is based on the core framework of Pellet [26].

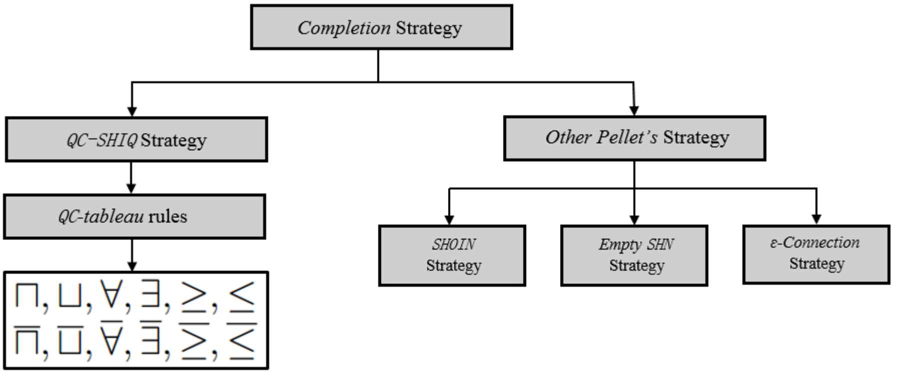


Figure 1. Different completion strategies in QC-OWL

Pellet, a widely used complete OWL-DL reasoner, is a tableau-based reasoner. It has a core reasoning engine which is suitable for extension. The core functionality of any tableau-based reasoner is checking the consistency of a KB. In order to check the consistency, the tableaux reasoner searches for a model through a process of completion. The completion algorithm in the Pellet reasoning engine is built on an extensible architecture where different completion strategies can be plugged in. Since we have a new tableau algorithm for the QC-consistency problem, we have implemented a new completion strategy called the *QC-$\mathcal{SHIQ}$ Strategy* and incorporated this strategy into Pellet's generic completion strategy. All the QC-tableau expansion rules have been implemented inside the QC-SHIQ Strategy. Figure 1 depicts how the QC-SHIQ Strategy has been incorporated with Pellet. QC-OWL is implemented in Java 7 with the aforementioned design and is built on Pellet 2.3.1. It supports OWL-APIs (versions 3 to 3.4.3) [25], Java APIs for creating, manipulating and serializing OWL Ontologies. In the current version of QC-OWL, only one functionality has been provided, checking the QC-consistency of an ontology.

Evaluation. In this section, we evaluate the performance of QC-OWL for the task of consistency checking and compare it with that of PROSE [22, 27]. PROSE is a transformation based prototype system which is designed using the Decorator Pattern, a structural design pattern, in order to perform paraconsistent reasoning on inconsistent KBs. In PROSE, the input KB $\mathcal{K}$, and the input query φ,

KB Name	DL Expressivity	Concept Count	Axiom Count
heart.owl	SHI	75	448
tambis-patched.owl	SHIN	395	1 090
bad-food.owl	ALCO (D)	18	52
amino-acid.owl	ALCF (D)	46	563
buggyPolicy.owl	ALCHO	15	41
uma-025-arctan.owl	ALCRIF (D)	366	14 816
uma-random-0.01-arctan.owl	ALCRF (D)	366	14 816
uma-random-0.03-arctan.owl	ALCRIF (D)	366	26 421
uma-random-0.01-arctan-inc.owl	ALCRF (D)	366	14 829
uma-random-0.03-arctan-inc.owl	ALCRIF (D)	366	26 421
uma-random-0.04-arctan-inc.owl	ALCRF (D)	366	32 231
uma-random-0.07-arctan-inc.owl	ALCRIF (D)	366	49 592

Table 3. Characteristics of the Benchmark KBs

KB Name	Con	PROSE	PROSE (NoOpi)	QC-OWL
amino-acid.owl	Y	615	70 333	33
uma-random-0.01-arctan.owl	Y	733	904	36
uma-random-0.03-arctan.owl	Y	1 248	111 776	39
heart.owl	Y	time-out	time-out	27
tambis-patched.owl	Y	time-out	time-out	39
uma-025-arctan.owl	Y	time-out	time-out	30

Table 4. QC-consistency test results for consistent ontologies

are transformed into a new KB $\mathcal{S}(\mathcal{K})$, and a new query $\mathcal{W}(\varphi)$, respectively, then a classical reasoner, e.g. Pellet, is used to do reasoning. Since a classical reasoner is used in PROSE, all optimization techniques provided in that reasoner also work in PROSE. In this experiment, the inner classical reasoner of PROSE was Pellet. Pellet implements most of the state-of-the-art optimization techniques, e.g., Optimized Blocking, Dependency-directed Backjumping etc. On the other hand, these optimization techniques have not been investigated yet for QCDL. Therefore, most state-of-the-art optimization techniques are absent in QC-OWL. So, in order to make a proper comparison, we have also observed PROSE's performance by turning off those optimizations; here we denote this by *PROSE(NoOpi)*. Since the core system of a tableau-based reasoner is checking the consistency of a KB, we have observed the time of QC-consistency checking only in this experiment. We have conducted our experiment for both consistent and inconsistent ontologies (see Table 3). Three of these have been collected from *TONES Ontology Repository* [28] while the others were from various sources. Some of the inconsistent ontologies have been generated by taking existing ontologies and inserting inconsistencies into them. Table 4 and Table 5 show the preliminary results for consistent and inconsistent ontologies, respectively. In Tables 4 and 5, the column *Con* is for the consistency and *NoOpi* stands for no optimization in Pellet. Displayed time was measured in *milliseconds* and *time-out* indicates that the program was manually terminated after 30 minutes.

KB Name	Con	PROSE	PROSE(NoOpi)	QC-OWL
bad-food.owl	N	31	127	34
buggyPolicy.owl	N	16	78	22
uma-random-0.01-arctan-inc.owl	N	3 328	59 510	14 761
uma-random-0.03-arctan-inc.owl	N	1 404	152 080	10 342
uma-random-0.04-arctan-inc.owl	N	4 836	88 322	20 044
uma-random-0.07-arctan-inc.owl	N	time-out	time-out	185 803

Table 5. QC-consistency test results for inconsistent ontologies

5.1 Results Analysis

As it is observed from Table 4, QC-OWL significantly outperforms PROSE for consistent ontologies even both without and with the state-of-the-art optimizations in Pellet. The reason is that when a KB is transformed into a new KB in PROSE, lots of negations and disjunctions incur in the KB which complicates the overall reasoning process. Moreover, each inclusion is transformed into three inclusions at every step which increases the size of KB. It is also interesting to note that the performance of QC-OWL is relatively consistent for different ontologies contrary to the performance of PROSE(NoOpi). For example, in case of the *heart* ontology, PROSE cannot give any answer within 30 minutes whereas QC-OWL takes only 27 milliseconds to answer. Figure 2 displays a bar graph that illustrates the time required to check QC-consistency for consistent ontologies in QC-OWL, PROSE(NoOpi) and PROSE, with a logarithmic scale on the vertical axis.

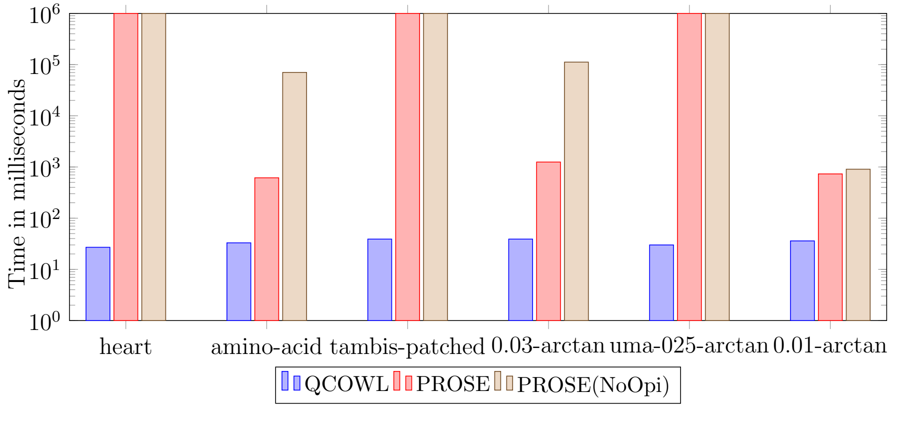


Figure 2. QC-OWL, PROSE, PROSE(NoOpi) testing for results for consistent ontologies

In Table 5, the results show that QC-OWL is more effcient than PROSE(NoOpi) for inconsistent ontologies. As an example, for the *bad-food* ontology PROSE (NoOpi) takes 127 milliseconds whereas QC-OWL takes only 34 milliseconds. One might have noticed that QC-OWL is not as effcient as PROSE. This is expected because PROSE inherits most of the state-of-the-art optimization techniques from Pellet whereas QC-OWL has not implemented any optimization yet. In QC-OWL, reasoning w.r.t. any general TBox has been reduced to reasoning w.r.t. an empty TBox with the internalization technique [1] that introduces many disjunctions, hence the reasoning process is somewhat slower. However, if we implement these Pellet optimizations in QC-OWL, then we expect it to outperform PROSE analogous with the behavior for consistent ontologies. Implementing state-of-the-art optimization techniques in QC-OWL is an interesting topic for the future work. It is motivating to note in Figure 3 that for the *0.07-arctan-inc* ontology, both PROSE and PROSE(NoOpi) get time out whereas QC-OWL gets results approximately in three minutes.

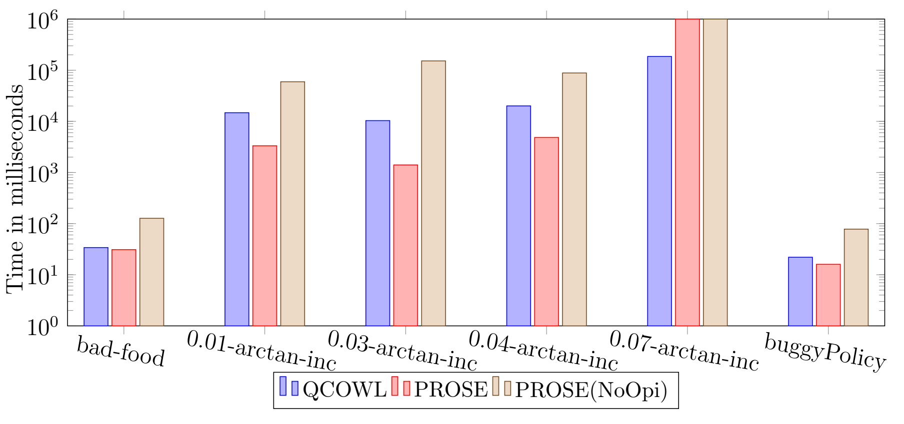


Figure 3. QC-OWL, PROSE, PROSE(NoOpi) testing for results for inconsistent ontologies

5.2 Summary

While the classical reasoner fails to perform reasoning over inconsistent ontologies, both PROSE and QC-OWL can give some meaningful results under non-classical semantics. Though both of them are intended for paraconsistent reasoning, QC-OWL significantly differs from PROSE. PROSE uses a classical reasoner to perform reasoning on the transformed ontology while QC-OWL directly applies tableau expansion rules on the loaded ontology. Moreover, QC-OWL has couple of key advantages over PROSE:

1. PROSE is a transformation based prototype system whereas QC-OWL is based on QC-tableau that can handle inconsistency directly; transformation adds some overhead.
2. Not counting the transformation time, PROSE is still slower than QC-OWL.

This is due to two reasons:

1. The transformation introduces many negations and disjunctions which complicates the reasoning process;
2. During transformation, each inclusion is replaced by three inclusions which increases the size of the ontology.

It was noted in [26] that a classical reasoner takes more time to work on a transformed ontology.

Our initial motivation was to develop a truly paraconsistent reasoner that can handle inconsistency directly with acceptable speed and reasonable inference power. The preliminary results suggest that QC-OWL could be a new benchmark in paraconsistent reasoning. If most of the state-of-the-art optimization techniques are implemented in QC-OWL, then it could perform as optimally as a classical reasoner; we leave this for future work.

6 RELATED WORKS

In this section, we compare this work with the existing works about QCDL [21, 22].

[21] introduces primarily quasi-classical semantics for $\mathcal{ALC}$, a simple member of description logics, and presents some properties of it and the relationship with the method by using Belnap's four-valued logic. However, as investigated in [22], there exist three insufficiencies as follows:

- The approach presented in [21] is not enough to characterize all features of DLs. As is well known, DL is used in artificial intelligence for formal reasoning on the concepts of an application domain [1]. However, [21] did not really introduce a new logic with the QC negation, but rather viewed negation as a transformation on formulas (axioms). This makes it impossible to directly represent the "opposite" concept of a given concept, because the negation of a concept $\neg C$ is not taken as the "opposite" concept but rather as a concept unrelated to C. In the current QCDL, we can directly introduce the QC negation of a concept as the "opposite" concept. Moreover, this approach cannot capture the natural relationship between "$\sqcup$" and "$\sqsubseteq$".
- The basic approach of [21] cannot be generalized to more expressive description logics such as $\mathcal{SHIQ}$ which can be obtained by generalizing [22]. One of important reasons is the complement of axioms $\approx \phi$ presented in [21] cannot capture expressive DL axioms. For instance, $\approx ((\geq\, n\, S.C)(a))$ cannot

be represented by both $(\leq (n\text{-}1)\, S)(a)$ and $\approx (C(b))$. Instead, the QC negation of concept $\overline{C}$ introduced in [22] can capture expressive DL axioms, e.g., $(\overline{\geq n\, S.C})(a) \equiv (\leq (n\text{-}1)\, S.\overline{C})(a)$.

- The complement of inclusions $\approx (C \sqsubseteq D)$ can be no longer translated into a corresponding DL concept inclusions. Because of this, it is impossible to transform this logic into classical DL. On the contrary, our proposed QCDL can be exactly transformed into DL [22].
- In [22], a transformation-based approach is proposed to reduce the entailment problem in QCDL into the entailment problem in DL. Comparing with the approach by applying some transformation in [22], this paper proposed an inherent proof for the QC-consistency without translating it into other logics. Moreover, the technique in this paper is totally different from the technique adopted in [22] in principle. In some sense, this paper can be taken as a complement of [22] in the insight of QCDL. Additionally, we found that the transformation process is not always efficient since each inclusion will be transformed into three inclusions at every step. Our experimental results indicate that our QC-OWL based on QC-tableau calculus is indeed more efficient than PROSE based on QC-transformation even in handling consistent ontologies.

Recently, [24] presents a quasi-classical semantics for DL where each quasi-classical model is a subset of Herbrand base, which is obtained by grounding all concepts and roles in a Herbrand Universe (a set of constants). In this sense, we think that the semantics could be taken as some kind of restricted version of the semantics as discussed in [21, 22].

7 CONCLUSIONS

This paper presented a theoretical research of the satisfiability problem of QCDL and developed a tableau calculus called QC-tableau for satisfiability problems. Based on this tableau calculus, we implemented an OWL paraconsistent reasoner named QC-OWL. We show that QC-OWL has higher efficiency than PROSE in checking QC-consistency. In the future work, we will attempt to implement most optimization techniques used for classical tableau reasoning to make QC-OWL more efficient and employ a high performance computing enviroment [29]. As another direction, we will provide more QC expansion rules so more expressive OWL ontologies can be handled with QC-OWL.

Acknowledgments

We would like to thank to anonymous referees for their critical comments which helped us to improve the paper. Xiaowang Zhang would like to thank Zuoquan Lin, Yue Ma, Guilin Qi, Kewen Wang, and Guohui Xiao for their helpful and constructive discussions and also thanks Ulrike Sattler for a helpful e-mail communication. This work is partly funded by the program of the National Key Research and Development Program of China (2016YFB1000603), the National Natural Science Foundation of China (61502336), and the Key Technology Research and Development Program of Tianjin (16YFZCGX00210). Wendy MacCaull is funded by the Natural Sciences and Engineering Research Council of Canada. Mokarrom Hossain is partly funded by the St. Francis Xavier University Graduate Research Assistantship.

A APPENDIX: PROOFS

Proof of Proposition 3.1. Let Δ be a domain. By the definition, C is QC-unsatisfiable w.r.t. $\mathcal{T}$ and $\mathcal{R}$ iff for any QC-model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{R}$ such that $+C = \emptyset$ where $C^{\mathcal{I}} = \langle +C, -C\rangle$ iff for any QC-model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{R}$ such that $+\overline{C} = \overline{+C} = \Delta^{\mathcal{I}}$ where $\overline{C}^{\mathcal{I}} = \langle +\overline{C}, -\overline{C}\rangle$ iff $\mathcal{I}$ for any QC-model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{R}$, $\mathcal{I}$ is a QC-model of $\overline{C}(\tau)$ where τ is a fresh individual name, that is, the KB $(\mathcal{T}, \mathcal{R}, \{\overline{C}(\tau)\})$ is QC-consistent where τ is a fresh individual name.

Proof of Proposition 3.2.

- $(\mathcal{T}, \mathcal{R}, \mathcal{A}) \models_Q C \sqsubseteq D$ iff, by definition, for any QC-model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ (that is, for any QC-model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{R}$), $\mathcal{I} \models_w C \sqsubseteq D$ iff, by the property stated in [22] that for each base interpretation $\mathcal{I}$ on Δ, $\mathcal{I} \models_w C \sqsubseteq D$ iff $\mathcal{I} \models_w \overline{C} \sqcup D(a)$ for any $a \in \Delta$, we can conclude that for any QC-model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{R}$, $\mathcal{I} \models_w \overline{C} \sqcup D(a)$ for any $a \in \Delta$ iff for any QC-model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{R}$, $\overline{+C} \cup +D = \Delta^{\mathcal{I}}$ where $C^{\mathcal{I}} = \langle +C, -C\rangle$ and $D^{\mathcal{I}} = \langle +D, -D\rangle$ iff for any QC-model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{R}$, $+C \cap +\overline{D} = \emptyset$ since $+\overline{D} = \overline{+D}$ where $C^{\mathcal{I}} = \langle +C, -C\rangle$ and $D^{\mathcal{I}} = \langle +D, -D\rangle$ iff $C \sqcap \overline{D}$ is QC-unsatisfiable w.r.t. $\mathcal{T}$ and $\mathcal{R}$.
- $(\mathcal{T}, \mathcal{R}, \mathcal{A}) \models_Q C(a)$ iff, by definition, for any QC-model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{R}, \mathcal{A})$, $\mathcal{I} \models_w C(a)$ iff, by definition, for any QC-model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{R}, \mathcal{A})$, $a^{\mathcal{I}} \in +C$ where $C^{\mathcal{I}} = \langle +C, -C\rangle$ iff for any QC-model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{R}, \mathcal{A})$, $a^{\mathcal{I}} \notin +\overline{C}$ since $+\overline{C} = \overline{+C}$ where $C^{\mathcal{I}} = \langle +C, -C\rangle$ iff $(\mathcal{T}, \mathcal{R}, \mathcal{A} \cup \{\overline{C}(a)\})$ is QC-inconsistent.

Proof of Proposition 3.3. To prove Proposition 3.3, we need two lemmas.

Lemma A.1. [22] Let Δ be a domain. For each base interpretation $\mathcal{I}$ on Δ, $\mathcal{I} \models_s C \sqsubseteq D$ iff $\mathcal{I} \models_s \neg C \sqcup D(a)$ for any $a \in \Delta$.

Lemma A.2. Let T be a TBox and R an RBox. We can conclude that the KB $(\mathcal{T}, \mathcal{R}, \emptyset)$ is QC-consistent iff $\{C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}(\tau)\}$ is QC-consistent w.r.t. $\mathcal{R}_U$ where τ is a fresh individual.

Proof. Let Δ be a domain with containing τ. It is not hard to prove the only-if direction. Now, we prove the if direction. That is, if $\{C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}(\tau)\}$ is QC-consistent w.r.t. $\mathcal{R}_U$ where τ is a fresh individual then $(\mathcal{T}, \mathcal{R}, \emptyset)$ is QC-consistent. Assume that $\mathcal{I}$ be a QC-model of $\mathcal{R}_U$ and $\mathcal{I} \models_s C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}(\tau)$ where τ is a fresh individual and the domain of $\mathcal{I}$ only contains those instances *connected from/to* τ (*Because the inverse of role exists, we must consider whose instances connect to* τ.), that is, there exists a path from/to τ to connect those instances [23]. Since $\mathcal{R} \subseteq \mathcal{R}_U$, $\mathcal{I}$ is a QC-model of $\mathcal{R}$. Now, we prove that $\mathcal{I}$ is a QC-model of $\mathcal{T}$. We have $\mathcal{I} \models_s C_{\mathcal{T}}(\tau)$. Because $\mathcal{I} \models_s R \sqsubseteq U$ for any $R \in \mathcal{R}_U$, for any $\tau' \in \Delta$, $\mathcal{I} \models_s R(\tau, \tau')$ implies $\mathcal{I} \models_s U(\tau, \tau')$. Because $\mathcal{I} \models_s \forall U.C_{\mathcal{T}}(\tau)$, $\mathcal{I} \models_s \forall U.C_{\mathcal{T}}(\tau')$. For each instance b occurring in the domain of $\mathcal{I}$, $\mathcal{I} \models_s C_{\mathcal{T}}(b)$ since b is connected from/to τ. By Lemma A.1, $\mathcal{I}$ is a QC model of $\mathcal{T}$. $\square$

Now, we are ready to prove Proposition 3.3.

1. This directly follows from the definition of strong interpretation of conjunction and Lemma A.2.
2. This directly follows from the first item of Proposition 3.2 and Lemma A.2.
3. This is analogous to the proof of Lemma A.2 where we select a QC model whose domain are those instances connected from/to some individual occurring in $\mathcal{A}$.

Proof of Proposition 4.1. This directly follows from the definition of strong interpretation since $\neg$ only exchanges two parts of strong interpretations of all concepts.

Proof of Proposition 4.2. We only prove that $\overline{\leq n\, S.C} \equiv_s \geq (n{+}1)\, S.\overline{\neg C}$ and $\overline{(\geq n{+}1)\, S.C} \equiv_s \leq n\, S.\overline{\neg C}$ since others directly follow the definition of strong interpretations.

Let $\mathcal{I}$ be a strong interpretation. Assume that $C^{\mathcal{I}} = \langle +C, -C \rangle$ and $R^{\mathcal{I}} = \langle +R, -R \rangle$. Let $\overline{\neg C}^{\mathcal{I}} = \left\langle +\overline{\neg C}, -\overline{\neg C} \right\rangle$. Thus $+\overline{\neg C} = \overline{-C}$ and $-\overline{\neg C} = \overline{+C}$. $\left(\overline{\geq n\, S.C}\right)^{\mathcal{I}} = \langle \{x \mid \sharp(\{y.(x,y) \in +R\} \cap +C) \leq n{-}1\}, \{x \mid \sharp\left(\{y.(x,y) \in +R\} \cap \overline{-C}\right) \geq n\} \rangle = \langle \{x \mid \sharp\left(\{y.(x,y) \in +R\} \cap \overline{-\overline{\neg C}}\right) \leq n{-}1\}, \{x \mid \sharp\left(\{y.(x,y) \in +R\} \cap +\overline{\neg C}\right) \geq n\} \rangle = \left(\leq (n-1)\, S.\overline{\neg C}\right)^{\mathcal{I}}$ and $\left(\overline{\leq n\, S.C}\right)^{\mathcal{I}} = \langle \{x \mid \sharp\left(\{y.(x,y) \in +R\} \cap \overline{-C}\right) \leq n\}, \{x \mid \sharp(\{y.(x,y) \in +R\} \cap +C) > n\} \rangle = \langle \{x \mid \sharp\left(\{y.(x,y) \in +R\} \cap +\overline{\neg C}\right) \geq n+1\}, \{x \mid \sharp\left(\{y.(x,y) \in +R\} \cap \overline{-\overline{\neg C}}\right) \geq n\} \rangle = \left(\geq (n+1)\, S.\overline{\neg C}\right)^{\mathcal{I}}$.

In the following proofs, we apply similar proof techniques developed in [23].

Proof of Theorem 4.1. (*The if direction*) If $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{I})$ is a QC-tableau for $\mathcal{A}$ w.r.t. $\mathcal{R}$, then we can define a QC-model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{A}$ and $\mathcal{R}$ as follows:

1. $\Delta^{\mathcal{I}} := \mathbf{S}$;
2. $A^{\mathcal{I}} := \langle \{s \mid A \in \mathcal{L}(s)\}, \{s \mid \neg A \in \mathcal{L}(s)\} \rangle$ for concept name $A \in clos(\mathcal{A})$;
3. $a^{\mathcal{I}} := \mathcal{J}(a)$ for any individual $a \in N_I$;
4. $R^{\mathcal{I}} := \begin{cases} \langle \mathcal{E}(R)^+, (\mathbf{S} \times \mathbf{S}) \backslash \mathcal{E}(R)^+ \rangle, & \text{if } Trans(R), \\ \langle \mathcal{E}(R) \cup \bigcup_{P \,\underline{\circledast}\, R, P \neq R} +P, (\mathbf{S} \times \mathbf{S}) \backslash (\mathcal{E}(R) \cup \bigcup_{P \,\underline{\circledast}\, R, P \neq R} +P) \rangle, & \text{otherwise.} \end{cases}$

Here let $\mathcal{E}(R)^+$ denote the transitive closure of $\mathcal{E}(R)$ and $P^{\mathcal{I}} = \langle +P, -P \rangle$. By the definition of $R^{\mathcal{I}}$ and **P8**, we can conclude that if $\langle s, t \rangle \in S^{\mathcal{I}}$ then either $\langle s, t \rangle \in \mathcal{E}(S)$ or there exists some path $\langle s, s_1 \rangle, \langle s_1, s_2 \rangle, \ldots, \langle s_n, t \rangle \in \mathcal{E}(S)$ for some R with Trans(R) and $R \underline{\boxast} S$. By **P8**, it follows that $\mathcal{I}$ is a QC-model of $\mathcal{R}$.

To prove that $\mathcal{I}$ is also a QC-model of $\mathcal{A}$, we need to show that $C \in \mathcal{L}(s)$ implies $s \in +C$ where $C^{\mathcal{I}} = \langle +C, = C \rangle$ for any $s \in \mathbf{S}$. By **P12** and **P13**, and the interpretation of individuals and roles, we can prove that $\mathcal{I}$ satisfies each assertion in $\mathcal{A}$ by induction on the length $\|C\|$ of a concept C in NNF, where we count neither negation nor integers in number restrictions.

An important case is $C = \forall S.E$; let $t \in \mathbf{S}$ with $\langle s, t \rangle \in S^{\mathcal{I}}$. There are two possibilities:

1. $\langle s, t \rangle \in \mathcal{E}(S)$. Then **P4** implies $E \in \mathcal{L}(t)$.
2. $\langle s, t \rangle \notin \mathcal{E}(S)$.

Then there exists a path $\langle s, s_1 \rangle, \langle s_1, s_2 \rangle, \ldots, \langle s_n, t \rangle \in \mathcal{E}(S)$ for some R with Trans(R) and $R \underline{\boxast} S$. Then **P6** implies $\forall R.E \in \mathcal{L}(s_i)$ for all $1 \leq i \leq n$, and **P4** implies $E \in \mathcal{L}(t)$. In both cases, $t \in E^{\mathcal{I}}$ by induction and hence $s \in C^{\mathcal{I}}$.

(*The only-if direction*) Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a QC-model of $\mathcal{A}$, we define a QC-tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ for $\mathcal{A}$ and $\mathcal{R}$ as follows: $\mathbf{S} := \Delta^{\mathcal{I}}, \mathcal{E}(R) := +R, \mathcal{L}(s) := \{C \in \mathit{clos}(\mathcal{A}) \mid s \in +C\}$, and $\mathcal{J}(a) := a^{\mathcal{I}}$. It concludes that $\mathbb{T}$ is a QC-tableau for $\mathcal{A}$ and $\mathcal{R}$.

Proof of Theorem 4.2. Assume that $|\mathit{clos}(\mathcal{A})| = m$, $|\mathbf{R}_{\mathcal{A}}| = n$, $\max\{\geq nS.C \in \mathit{clos}(\mathcal{A})\} = n_{max}$. The QC-tableau calculus satisfies the following three properties:

- No node is removed in the calculus. Moreover, the only rules that remove elements from the labels of edges or nodes are the $\leq$-rule and $\leq_r$-rule, which sets them to $\emptyset$. If the $\leq_r$-rule sets an edge label to $\emptyset$, the node below the edge is blocked forever. If the $\leq_r$-rule sets a root x label to $\emptyset$, then after this, $x's$ label is never changed again. Because no root node is generated, this removal might only happen a finite number of times, and the new edges generated by the $\leq_r$-rule guarantee that the resulting structure is still a completion forest.
- Because there are at most 2^{mn} different possible labelings for a pair of nodes and an edge, the pair-wise blocking condition implies the existence of two nodes on P such that one directly blocks the other if P is a path of length at least 2^{2mn}. Paths are of length at most 2^{2mn} since a path on which nodes are blocked cannot become longer.
- Only four rules: the $\exists$-rule, the $\geq$-rule, the $\overline{\forall}$-rule, and, the $\overline{\leq}$-rule generate new nodes, and each generation is triggered by a concept of the form $\exists R.C$, $\geq n\, S.C$, $\overline{\forall R.C}$ (by the $\overline{\forall}$-rule, it turns to $\exists R.\overline{C}$), or $\overline{\leq n\, S.C}$ (by the $\overline{\forall}$-rule, it turns to $\geq n\, S.\overline{\neg C}$) in $\mathit{clos}(\mathcal{A})$. Each of those concepts trigger the generation of at most n_{max} successors y_i: x will have some S-neighbor z with $\mathcal{L}(z) \subseteq \mathcal{L}(y)$ if the $\leq$-rule or the $\leq_r$-rule subsequently causes $\mathcal{L}(\langle x, y_i \rangle)$ to be set to $\emptyset$. By the

definition of clashes, the rule application leading to the generation of y_i will not be repeated. The out-degree of the forest is bounded by mnn_{max} since $clos(\mathcal{A})$ contains a total of at most m $\exists R.C$.

Based on three properties above, we conclude that the QC-tableau calculus terminates.

Proof of Theorem 4.3. (*The if direction*) Let $\mathcal{F}$ be a complete and clash-free completion forest. The definition of a QC-tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ from $\mathcal{F}$ works as follows: A path is a sequence of pairs of nodes of $\mathcal{F}$ of the form $p = \left[\frac{x_0}{x'_0}, \ldots, \frac{x_n}{x'_n}\right]$. We also define $Tail(p) := x_n$ and $Tail'(p) := x'_n$. With $\left[p \mid \frac{x_{n+1}}{x'_{n+1}}\right]$, we denote the path $\left[\frac{x_0}{x'_0}, \ldots, \frac{x_n}{x'_n}, \frac{x_{n+1}}{x'_{n+1}}\right]$. We define a set $Paths(\mathcal{F})$ in an inductive way:

1. For each root node x^i_0 of $\mathcal{F}$, $\left[\frac{x^i_0}{x^i_0}\right] \in Paths(\mathcal{F})$, and
2. for each path $p \in Paths(\mathcal{F})$ and each node z in $\mathcal{F}$:
   (a) if z is a successor of $Tail(p)$ and z is neither blocked nor a root node, then $\left[p \mid \frac{z}{z}\right] \in Paths(\mathcal{F})$, or
   (b) if, for some node y in $\mathcal{F}$, y is a successor of $Tail(p)$ and z blocks y, then $\left[p \mid \frac{z}{y}\right] \in Paths(\mathcal{F})$.

Because root nodes neither are blocked nor are they blocking other nodes, they only occur in the first place of a path. Moreover, if $p \in Paths(\mathcal{F})$, then $Tail(p)$ is not blocked, $Tail(p) = Tail'(p)$ iff $Tail'(p)$ is not blocked, and $\mathcal{L}(Tail(p)) = \mathcal{L}(Tail(p))$.

Now, we define a QC-tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ as follows:

1. $\mathbf{S} = Paths(\mathcal{F})$;
2. $\mathcal{L}(p) = \mathcal{L}(Tail(p))$;
3. $\mathcal{E}(R) = \{\langle p, \left[p|\frac{x}{x'}\right]\rangle \in \mathbf{S} \times \mathbf{S} \mid x'$ is an R-successor of $Tail(p)\} \cup \{\langle \left[q|\frac{x}{x'}\right], q_i\rangle \in \mathbf{S} \times \mathbf{S} \mid x'$ is an R-predecessor of $Tail(q)\} \cup \{\langle \left[\frac{x}{x}\right], \left[\frac{y}{y}\right]\rangle \in \mathbf{S} \times \mathbf{S} \mid x, y$ are root nodes, and y is an R-neighbor of $x\}$; and
4. $\mathcal{J}(a_i) = \begin{cases} \left[\frac{x^i_0}{x^i_0}\right], & \text{if } x^i_0 \text{ is a root node in } \mathcal{F} \text{ with } \mathcal{L}(x^i_0) \neq 0; \\ \left[\frac{x^j_0}{x^j_0}\right], & \text{if } \mathcal{L}(x^i_0) = \emptyset, x^j_0 \text{ a root node in } \mathcal{F} \text{ with } \mathcal{L}(x^j_0) \neq \emptyset \text{ and } x^i_0 \doteq x^j_0. \end{cases}$

If $\mathcal{L}(x) = \emptyset$ then x is a root node and that there is another root node y with $\mathcal{L}(y) \neq \emptyset$ and $x = y$. Now we prove that $\mathbb{T}$ is a QC-tableau for $\mathcal{A}$ and $\mathcal{R}$.

- $\mathbb{T}$ satisfies **P1** because $\mathcal{F}$ is clash-free. **P2** and **P3** are satisfied by $\mathbb{T}$ because $\mathcal{F}$ is complete. **P4** follows the definition of R-neighbors and R-successor.

- For **P4**, let $p, q \in \mathbf{S}$ with $\forall R.C \in \mathcal{L}(p)$, $\langle p, q\rangle \in \mathcal{E}(R)$. If $q = \left[p \mid \frac{x}{x'}\right]$, then x' is an R-successor of $\mathit{Tail}(p)$ and, due to completeness of $\mathcal{F}$, $C \in \mathcal{L}(x') = \mathcal{L}(x) = \mathcal{L}(q)$. If $p = \left[q|\frac{x}{x'}\right]$, then x' is an R-predecessor of $\mathit{Tail}(q)$ and, due to completeness of F, $C \in \mathcal{L}(\mathit{Tail}(q)) = \mathcal{L}(q)$. If $p = \left[\frac{x}{x}\right]$ and $q = \left[\frac{y}{y}\right]$ for two root nodes x and x', then y is an R-neighbor of x, and completeness of $\mathcal{F}$ yields $C \in \mathcal{L}(y) = \mathcal{L}(q)$. **P6**, **P11**, **P19**, and **P22** hold for similar reasons. For **P5**, let $\exists R.C \in \mathcal{L}(p)$ and $\mathit{Tail}(p) = x$. Since x is not blocked and $\mathcal{F}$ is complete, x has some R-neighbor y with $C \in \mathcal{L}(y)$. If y is a successor of x, then y can either be a root node or not $\emptyset$.

  1. If y is not a root node: if y is not blocked, then $q := \left[p|\frac{y}{y}\right] \in \mathbf{S}$; if y is blocked by some node z, then $q := \left[p|\frac{z}{y}\right] \in \mathbf{S}$.
  2. If y is a root node: since y is a successor of x, x is also a root node. This implies $p = \left[\frac{x}{x}\right]$ and $q = \left[\frac{y}{y}\right] \in \mathbf{S}$.

  So $\langle p, q\rangle \in \mathcal{E}(R)$ and $C \in \mathcal{L}(q)$. **P18** holds for similar reasons. **P7** holds because of the symmetric definition of the mapping $\mathcal{E}$. **P8** follows from the transitivity of $\mathcal{E}$.

- Suppose, for the sake of contradiction, that **P9** is not satisfied. Hence there is some $p \in \mathbf{S}$ with $(\leq nS.C) \in \mathcal{L}(p)$ and $\sharp(S^T(p, C)) > n$. We will show that this implies $\sharp S^{\mathcal{F}}(\mathit{Tail}(p), C) > n$, contradicting either clash-freeness or completeness of $\mathcal{F}$. Let $x := \mathit{Tail}(p)$ and $P := S^T(p, C)$. We distinguish two cases:

  1. P contains only paths of the form $\left[p \mid \frac{y}{y'}\right]$ and $\left[\frac{x_0^i}{x_0^i}\right]$. Then $\sharp(P) > n$ is impossible since the function Tail' is injective on P: if we assume that there are two distinct paths $q_1, q_2 \in P$ and $\mathit{Tail}'(q_1) = \mathit{Tail}'(q_2) = y'$, then this implies that each q_i is of the form $q_i = \left[p \mid \frac{y_i}{y'}\right]$ or $q_i = \left[\frac{y'}{y'}\right]$. From $q_1 \neq q_2$, we have that $q_i = \left[p \mid \frac{y_i}{y'}\right]$ holds for some $i \in \{1, 2\}$. Since root nodes occur only in the beginning of paths and $q_1 \neq q_2$, we have $q_1 = [p \mid (y_1, y')]$ and $q_2 = [p \mid (y_2, y')]$. If y' is not blocked, then $y_1 = y' = y_2$, contradicting $q_1 \neq q_2$. If y' is blocked in $\mathcal{F}$, then both y_1 and y_2 block y', which implies $y_1 = y_2$, again a contradiction. Hence Tail' is injective on P and thus $\sharp P = \sharp(\mathit{Tail}'(P))$. Moreover, for each $y' \in \mathit{Tail}'(P)$, y' is an R-successor of x and $C \in \mathcal{L}(y')$. This implies $\sharp(S^{\mathcal{F}}(x, C)) > n$.
  2. P contains a path q where $p = \left[q \mid \frac{x}{x'}\right]$. Obviously, P may only contain one such path. As in the previous case, Tail' is an injective function on the set $P' := P \backslash \{q\}$, each $y' \in \mathit{Tail}'(P')$ is an S-successor of x, and $C \in \mathcal{L}(y')$ for each $y' \in \mathit{Tail}'(P')$.

  Let $z := \mathit{Tail}(q)$. We distinguish two cases:

1. $x = x'$. Hence x is not blocked, and thus x is an S-predecessor of z. Since $\mathit{Tail}'(P')$ contains only successors of x we have that $z \notin \mathit{Tail}'(P')$ and, by construction, z is an R-neighbor of x with $C \in \mathcal{L}(z)$.
2. $x \neq x'$. This implies that x' is blocked by x and that x' is an R-predecessor of z. Due to the definition of pairwise-blocking this implies that x is an S-successor of some node u with $\mathcal{L}(u) = \mathcal{L}(z)$.

Again, $u \notin \mathit{Tail}'(P')$ and, by construction, u is an S-neighbor of x and $C \in \mathcal{L}(u)$. **P21** holds for similar reasons.

- For **P10**, let $(\geq nS.C) \in \mathcal{L}(p)$. Hence there are n S-neighbors $y_1, \ldots, y_n$ of $x = \mathit{Tail}(p)$ in $\mathcal{F}$ with $C \in \mathcal{L}(y_i)$. For each y_i there are two possibilities:

  1. y_i is an S-successor of x and y_i is not blocked in $\mathcal{F}$. Then $q_i := [p \mid \frac{y_i}{y_i}]$ or y_i is a root node and $q_i := [\frac{y_i}{y_i}]$ is in $\mathbf{S}$; and
  2. y_i is an S-successor of x and y_i is blocked in $\mathcal{F}$ by some node z. Then $q_i = [p \mid \frac{z}{y_i}]$ is in $\mathbf{S}$.

  Since the same z may block several of the y_js, it is indeed necessary to include y_i explicitly into the path to make them distinct.

  Hence for each y_i there is a different path q_i in $\mathbf{S}$ with $S \in \mathcal{L}(\langle p, q_i \rangle)$ and $C \in \mathcal{L}(q_i)$, and thus $\sharp(S^T(p, C)) > n$. **P20** holds for similar reasons.

- **P12** is due to the fact that, when the completion calculus is started for an ABox $\mathcal{A}$, the initial completion forest $\mathcal{F}_{\mathcal{A}}$ contains, for each individual name a_i occurring in $\mathcal{A}$, a root node x_0^i with $\mathcal{L}(x_0^i) = \{C \in \mathit{clos}(\mathcal{A}) \mid a_i : C \in \mathcal{A}\}$. The calculus never blocks root individuals, and, for each root node x_0^i whose label and edges are removed by the $\leq_r$-rule, there is another root node x_0^j with $x_0^i = x_0^j$ and $\{C \in \mathit{clos}(\mathcal{A}) \mid a_i : C \in \mathcal{A}\} \subseteq \mathcal{L}(x_0^j)$. Together with the definition of $\mathcal{I}$, this yields **P12**. **P13** is satisfied for similar reasons. **P14** is satisfied because the $\leq_r$-rule does not identify two root nodes x_0^i, y_0^i when $x_0^i \neq y_0^i$ holds. Finally, **P15**, **P16**, and **17** directly follow from the definition of strong interpretations and the expansion rules: *double*, $\overline{\sqcap}$-rule, and $\overline{\sqcup}$-rule in Table 2, respectively.

(*The only-if direction*) We use $\mathbb{T}$ to trigger the application of the expansion rules such that they yield a completion forest $\mathcal{F}$ that is both complete and clash-free. For this purpose, a function π is used to map the nodes of $\mathcal{F}$ to elements of $\mathbf{S}$. The mapping π is defined as follows:

1. For individuals a_i in $\mathcal{A}$, we define $\pi(x_0^i) := \mathcal{J}(a_i)$.
2. If $\pi(x) = s$ is already defined, and a successor y of x was generated for $\exists R.C \in \mathcal{L}(x)$, then $\pi(y) = t$ for some $t \in \mathbf{S}$ with $C \in \mathcal{L}(t)$ and $\langle s, t \rangle \in \mathcal{E}(R)$.
3. If $\pi(x) = s$ is already defined, and successors y_i of x were generated for $\geq n\, R \in \mathcal{L}(x)$, then $\pi(y_i) = t_i$ for n distinct $t_i \in \mathbf{S}$ with $\langle s, t_i \rangle \in \mathcal{E}(R)$.

Clearly, the mapping for the initial completion forest for $\mathcal{A}$ satisfies the following condition: $\mathcal{L}(x) \in \mathcal{L}(\pi(x))$ if y is an S-neighbor of x then $\langle \pi(x), \pi(y) \rangle \in \mathcal{E}(R)$ and $x \neq y$ implies $\pi(x) \neq \pi(y)$.

It can be shown that the following claim holds:

Claim: Let $\mathcal{F}$ be generated by the completion calculus for $\mathcal{A}$ and let π satisfy (1). If an expansion rule is applicable to $\mathcal{F}$, then this rule can be applied such that it yields a completion forest $\mathcal{F}'$ and a (possibly extended) π that satisfies the condition above.

As a consequence of this claim, (**P1**), and (**P9**), if $\mathcal{A}$ has a QC-tableau, then the expansion rules can be applied to $\mathcal{A}$ such that they yield a complete and clash-free completion forest.

REFERENCES

[1] Baader, F.—Calvanese, D.—McGuinness, D. L.—Nardi, D.—Patel-Schneider, P. F. (Eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2003.

[2] Berners-Lee, T.—Hendler, J.—Lassila, O.: The Semantic Web. Scientific American, 2001, doi: 10.1038/scientificamerican0501-34.

[3] Straccia, U.: A Sequent Calculus for Reasoning in Four-Valued Description Logics. Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 1997). Lecture Notes in Computer Science, Vol. 1227, 1997, pp. 343–357, doi: 10.1007/BFb0027425.

[4] Flouris, G.—Huang, Z.—Pan, J. Z.—Plexousakis, D.—Wache, H.: Inconsistencies, Negations and Changes in Ontologies. Proceeding of the 21st National Conference on Artificial Intelligence, AAAI, 2006.

[5] Patel-Schneider, P. F.: A Four-Valued Semantics for Terminological Logics. Artificial Intelligence, Vol. 38, 1989, No. 3, pp. 319–351, doi: 10.1016/0004-3702(89)90036-2.

[6] Schlobach, S.—Cornet, R.: Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. Proceeding of the 18th International Joint Conference on Artificial Intelligence (IJCAI '03), AAAI, 2003, pp. 355–360.

[7] Parsia, B.—Sirin, E.—Kalyanpur, A.: Debugging OWL Ontologies. Proceeding of the 14th International Conference on World Wide Web (WWW '05), 2005, pp. 633–640, doi: 10.1145/1060745.1060837.

[8] Huang, Z.—van Harmelen, F.—ten Teije, A.: Reasoning with Inconsistent Ontologies. Proceeding of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05), 2005, pp. 454–459.

[9] Kalyanpur, A.—Parsia, B.—Horridge, M.—Sirin, E.: Finding All Justifications of OWL DL Entailments. The Semantic Web (ISWC 2007 + ASWC 2007). Lecture Notes in Computer Science, Vol. 4825, 2007, pp. 267–280, doi: 10.1007/978-3-540-76298-0_20.

[10] Qi, G.—Du, J.: Model-Based Revision Operators for Terminologies in Description Logics. Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI '09), 2009, pp. 891–897.

[11] Nguyen, L. A.—Szałas, A.: Three-Valued Paraconsistent Reasoning for Semantic Web Agents. Agent and Multi-Agent Systems: Technologies and Applications (KES AMSTA 2010). Lecture Notes in Computer Science, Vol. 6070, 2010, pp. 152–162, doi: 10.1007/978-3-642-13480-7_17.

[12] Lembo, D.—Lenzerini, M.—Rosati, R.—Ruzzi, M.—Savo, D. F.: Query Rewriting for Inconsistent DL-Lite Ontologies. Web Reasoning and Rule Systems (RR 2011). Lecture Notes in Computer Science, Vol. 6902, 2011, pp. 155–169, doi: 10.1007/978-3-642-23580-1_12.

[13] Borgwardt, S.—Peñaloza, R.: Description Logics over Lattices with Multi-Valued Ontologies. Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI '11), 2011, pp. 768–773.

[14] Kamide, N.: Embedding-Based Approaches to Paraconsistent and Temporal Description Logics. Journal of Logic and Computation, Vol. 22, 2012, No. 5, pp. 1097–1124, doi: 10.1093/logcom/exr016.

[15] Maier, F.—Ma, Y.—Hitzler, P.: Paraconsistent OWL and Related Logics. Semantic Web, Vol. 4, 2013, No. 4, pp. 395–427.

[16] Huang, S.—Li, Q.—Hitzler, P.: Reasoning with Inconsistencies in Hybrid MKNF Knowledge Bases. Logic Journal of the IGPL, Vol. 21, 2013, No. 2, pp. 263–290.

[17] Zhang, X.—Lin, Z.: An Approach to Generating Arguments over DL-Lite Ontologies. Computing and Informatics, Vol. 32, 2013, No. 5, pp. 924–948.

[18] Cocos, C.—Imam, F.—MacCaull, W.: Ontology Merging and Reasoning Using Paraconsistent Logics. International Journal of Knowledge-Based Organizations (IJKBO), Vol. 2, 2012, No. 4, pp. 35–53.

[19] Belnap, N. D.: A Useful Four-Valued Logic. In: Dunn, J. M., Epstein, G. (Eds.): Modern Uses of Multiple-Valued Logics. Springer, Dordrecht, 1977, pp. 7–73, doi: 10.1007/978-94-010-1161-7_2.

[20] Hunter, A.: Reasoning with Contradictory Information Using Quasi-Classical Logic. Journal of Logic and Computation, Vol. 10, 2000, No. 5, pp. 677–703, doi: 10.1093/logcom/10.5.677.

[21] Zhang, X.—Lin, Z.: Quasi-Classical Description Logic. Multiple-Valued Logic and Soft Computing, Vol. 18, 2012, No. 3-4, pp. 291–327.

[22] Zhang, X.—Xiao, G.—Lin, Z.—Van den Bussche, J.: Inconsistency-Tolerant Reasoning with OWL DL. International Journal of Approximate Reasoning, Vol. 55, 2014, No. 2, pp. 557–584.

[23] Horrocks, I.—Sattler, U.—Tobies, S.: Reasoning with Individuals for the Description Logic SHIQ. Automated Deduction – CADE-17. Lecture Notes in Computer Science, Vol. 1831, 2000, pp. 482–496, doi: 10.1007/10721959_39.

[24] Hou, H.—Wu, J.: Quasi-Classical Semantics and Tableau Calculus of Description Logics for Paraconsistent Reasoning in the Semantic Web. Proceeding of the Inter-

national Conference on Computational Science and Engineering (CSE '09), Canada, 2009, pp. 703–708, doi: 10.1109/CSE.2009.311.

[25] Horridge, M.—Bechhofer, S.: The OWL API: A Java API for OWL Ontologies. Semantic Web, Vol. 2, 2011, No. 1, pp. 11–21.

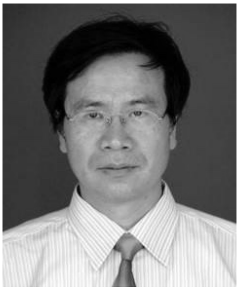
[26] Sirin, E.—Parsia, B.—Cuenca Grau, B.—Kalyanpur, A.—Katz, Y.: Pellet: A Practical OWL-DL Reasoner. Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 5, 2007, No. 2, pp. 51–53.

[27] PROSE: A Paraconsistent OWL-DL Reasoner. `http://prose-web.appspot.com/`, 2014.

[28] TONES: Ontology Repository. University of Manchester, `http://owl.cs.manchester.ac.uk/repository/`, 2008.

[29] Hossain, M.: Parallelization of Inconsistent-Tolerant DL-Reasoning. M.Sc. Thesis, Saint Francis Xavier University, Canada, 2016.

Xiaowang Zhang is Associate Professor at School of Computer Science and Technology in Tianjin University. He received his Ph.D. degree from Peking University in 2011. His research interests include artificial intelligence, databases and knowledge graph, etc.



Zhiyong Feng is Professor at School of Computer Software in Tianjin University. He received his Ph.D. degree from Tianjin University in 1996. His main research interests include knowledge engineering, service technology and security software engineering.

Wenrui Wu received his M.Sc. degree from the School of Computer Science and Technology in Tianjin University in 2017. His main research interests include parallel processing and RDF data management. He is currently working as an engineer in the Standard Chartered Bank.

Mokarrom Hossain received his B.Sc. degree in computer science and engineering from Shahjalal University of Science and Technology, Sylhet, Bangladesh, in 2012 and the M.Sc. degree in computer science from St. Francis Xavier University, Antigonish, NS, Canada, in 2016. His research interests include description logic, parallel reasoning, inconsistency-tolerant reasoning, and high-performance computing. He is currently working as a Software Engineer in the R & D team of a reputed company.

Wendy MacCaull is Professor and Chair of the Mathematics, Statistics and Computer Science Department of St. Francis Xavier University, Antigonish, Canada. She received her M.Sc. (1980) and Ph.D. (1984) in pure mathematics from McGill University, and joined the faculty at StFX in 1984. Her research interests include nonclassical logics, including paraconsistent and substructural logics, automated theorem proving, model checking, model driven software engineering and OWL ontologies.