A Technique for Handling the Right Hand Side of Complex RIAs

Milenko Mosurović[1] and Nenad Krdžavac[2]

[1] Faculty of Mathematics and Natural Science,
University of Montenegro, Podgorica,
ul. Džordža Vašingtona bb, 81000 Podgorica, Montenegro
[2] School of Electrical and Computer Engineering of Applied Studies in Belgrade,
University of Belgrade, Serbia,
ul. Vojvode Stepe 283, 11000 Belgrade, Serbia

Abstract. This paper examines a new technique based on tableau, that allows one to introduce composition of roles from the right hand side of complex role inclusion axioms (RIAs). Our motivation comes from modeling product models in manufacturing systems. The series of papers, so far, have studied the extension of tableau algorithm for Description Logics (DLs) to capture complex RIAs. However, such RIAs permit only the left hand side of the composition of roles. To illustrate the technique, we extend $\mathcal{RIQ}$ DL with one RIA of the form $R \dot{\sqsubseteq} Q \circ P$.

Keywords: Description Logic, Manufacturing system,Tableau.

1 Introduction

Description Logics [1] are a well-established branch of logics for knowledge representation and reasoning about it. Recent research in DLs has usually focused on the logics of the so-called $\mathcal{SH}$ family as basis for the standard Web Ontology Languages (OWL) [10]. In particular, the DL $\mathcal{SHIQ}$ [8] is closely related to OWL-Lite and extends the basic $\mathcal{ALC}$ [1](the minimal propositionally closed DL) with inverse roles and number restrictions, as well as with role inclusions and transitive roles. The DL known as $\mathcal{SHOIQ}$ [7], underlying OWL-DL, further extends $\mathcal{SHIQ}$ with nominals. Logics, $\mathcal{SHIQ}$ and $\mathcal{SHOIQ}$ were enhanced with regular role hierarchies in which the composition of a chain of roles may imply another role. These and other features were included in their extensions known as $\mathcal{SRIQ}$ [9] and $\mathcal{SROIQ}$ [5] respectively; the latter underlies the new OWL 2 [4] standard. For reasoning in them, the adaptations of the tableaux algorithms were proposed [9, 5]. In a pre-processing stage, the implications between roles, given by the role hierarchy, are captured in a set of non-deterministic finite state automata (NFA). The complexity of these logics is studied in [11]. Also, there exists another extensions of the logics with description graphs [14] and stratified ontologies [12]. Motivation for our research is based on modeling product models in manufacturing systems (see UML model on Figure 1) [3]. For example, when

an individual crankshaft in individual engine in an individual car, powers individual hubs in individual wheels in the same car, and not the hubs in the wheels in the other cars [13]. Such modeling example can be represented as RIAs with more then one role on the right hand side of role composition [13]. The aim of this paper is to show a technique that can allow the extension of $\mathcal{RIQ}$ DL [6] with a RIA of the form $R\dot{\sqsubseteq}Q \circ P$. The $\mathcal{RIQ}$ DL [6], is the fragment of $\mathcal{SRIQ}$ (without Abox, as well as, reflexive, symmetric, transitive, and irreflexive roles, disjoint roles, and the construct $\exists R.Self$) [9]. To avoid analysis of restrictions that roles must satisfy in new RIAs, we consider only one RIA of the form $R\dot{\sqsubseteq}Q \circ P$. Main idea is to define a new role (P^-, x) that remembers in which object is related to the role. We define new constructor which will deal with these roles.
The paper is organized as follows. Next section gives short overview of $\mathcal{RIQ}$ DL and its role hierarchy. Section (3) explains simple reduction problem and gives general idea. Section (4), outlines the extension of $\mathcal{RIQ}$ tableau, while section (5) gives formal proof of the correctness and termination of tableau algorithm. Finaly we give some remarks and explane future work.

2 Preliminaries

This section, in brief, outlines syntax and semantics of $\mathcal{RIQ}$ DL and regular role hierarchy. The alphabet of $\mathcal{RIQ}$ DL consists of set of concept names $\mathcal{N}_C$, set of role names $\mathcal{N}_R$ and finally, set of simple role names $\mathcal{N}_S \subset \mathcal{N}_R$. The set of roles is $\mathcal{N}_R \cup \{R^- | R \in \mathcal{N}_R\}$. According to [6], syntax and semantics of the $\mathcal{RIQ}$ DL concepts are given in definitions 1 and 2.

Definition 1. *Set of $\mathcal{RIQ}$ concepts is a smallest set such that*

- *every concept name and $\top$, $\bot$ are concepts, and,*
- *if C and D are concept and R is a role, S is simple role, n is non-negative integer, then $\neg C$, $C \sqcap D$, $C \sqcup D$, $\forall R.C$, $\exists R.C$, $(\leq nS.C)$, $(\geq nS.C)$ are concepts.* □

The semantics of the $\mathcal{RIQ}$ DL is defined by using interpretation. An interpretation is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set, called the domain of the interpretation. A valuation $\cdot^{\mathcal{I}}$ associates: every concept name C with a subset $C^I \subseteq \Delta^{\mathcal{I}}$; every role name R with a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ [6].

Definition 2. *An interpretation $\mathcal{I}$ extends to $\mathcal{RIQ}$ complex concepts and roles according to the following semantic rules:*

- *If R is a role name, then $(R^-)^{\mathcal{I}} = \{\langle x, y\rangle : \langle y, x\rangle \in R^{\mathcal{I}}\}$,*
- *If R_1, R_2,..., R_n are roles then $(R_1 R_2 \dots R_n)^{\mathcal{I}} = (R_1)^{\mathcal{I}} \circ (R_2)^{\mathcal{I}} \circ \dots \circ (R_n)^{\mathcal{I}}$, where sign $\circ$ is a composition of binary relations,*
- *If C and D are concepts, R is a role, S is a simple role and n is a non-negative integer, then* [3]

[3] $\sharp M$ denotes cardinality of set M.

$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}, \bot^{\mathcal{I}} = \emptyset, (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$,
$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}, (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$,
$(\exists R.C)^{\mathcal{I}} = \{x \ : \ \exists y. \langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$,
$(\forall R.C)^{\mathcal{I}} = \{x \ : \ \forall y. \langle x, y\rangle \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$,
$(\geq nS.C)^{\mathcal{I}} = \{x \ : \ \sharp\{y \ : \ \langle x, y\rangle \in S^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \geq n\}$,
$(\leq nS.C)^{\mathcal{I}} = \{x \ : \ \sharp\{y \ : \ \langle x, y\rangle \in S^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \leq n\}$.
Number restrictions $(\geq nS.C)$ *and* $(\leq nS.C)$*, are restricted to simple roles, in order to have* $\mathcal{RIQ}$ *decidability.* □

Strict partial order $\prec$ (irreflexive, transitive, and antisymmetric), on the set of roles, provides acyclicity [6]. Allowed RIAs in $\mathcal{RIQ}$ DL with respect to $\prec$, are expressions of the form $w \dot{\sqsubseteq} R$, where [6]:

1. R is a simple role name, $w = S$ and $S \prec R$ is a simple role,
2. $R \in \mathcal{N}_R \backslash \mathcal{N}_S$ is a role name and
   (a) $w = RR$, or
   (b) $w = R^-$, or
   (c) $w = S_1 \cdots S_n$ and $S_i \prec R$, for $1 \leq i \leq n$, or
   (d) $w = RS_1 \cdots S_n$ and $S_i \prec R$, for $1 \leq i \leq n$, or
   (e) $w = S_1 \cdots S_n R$ and $S_i \prec R$, for $1 \leq i \leq n$.

Note that the notion of simple role has the same meaning as defined in [5]. So, we use the simple role S carefully in allowed RIAs to avoid $R_1 \circ R_2 \sqsubseteq S$.

A $\mathcal{RIQ}$ RBox (role hierarchy) is a finite set $\mathcal{R}$ of RIAs. A role hierarchy $\mathcal{R}$ is regular if there exists strict partial order $\prec$ such that each RIA in $\mathcal{R}$ is regular [6]. An interpretation $\mathcal{I}$ satisfies a RIA $S_1 \cdots S_n \dot{\sqsubseteq} R$, if $S_1^{\mathcal{I}} \circ \cdots \circ S_n^{\mathcal{I}} \subseteq R^{\mathcal{I}}$. A $\mathcal{RIQ}$ concept C is satisfiable w.r.t. RBox $\mathcal{R}$ if there is an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$ and $\mathcal{I}$ satisfies all RIA in $\mathcal{R}$ [6, 11]. In this paper we extend regular $\mathcal{RIQ}$-RBox with one RIA of the form

$$w \dot{\sqsubseteq} Q \circ P \tag{1}$$

where $w = S_1 \circ S_2 \cdots S_n$, $S_i \prec Q$, $P \prec Q$ and there is no i, such that $P \prec S_i$. An interpretation $\mathcal{I}$ satisfies a RIA of the form $w \dot{\sqsubseteq} Q \circ P$, if $w^{\mathcal{I}} \subseteq Q^{\mathcal{I}} \circ P^{\mathcal{I}}$. In the rest of the paper we check satisfiability of concept C_0 w.r.t. defined RBox $\mathcal{R}$ and define $\mathcal{R}_{C_0} = \{R | R$ is role that occurs in C_0 or $\mathcal{R}\}$.

3 The simple reduction and general idea

Tableau algorithm in [6] tries to construct a tableau for $\mathcal{RIQ}$-concept C. In preprocessing step the role hierarchy is translated into NFA, that are used, both, in the definition of a tableau and in the tableau algorithm. Intuitively, an automaton is used to memorize path between an object x that has to satisfy a concept of the form $\forall R.C$ and other objects, and then to determine which of these objects must satisfy C [6]. Similar idea can be used in extension $\mathcal{RIQ}$ with a RIA of the form $w \dot{\sqsubseteq} Q \circ P$. If an object x should satisfies concept $\forall Q.C$ then we should define structure that will remember path $w \circ P^-$ from the object x to objects that must satisfy concept C. If we extend $\mathcal{RIQ}$ DL with Fun [11], then the next lemma holds:

Lemma 1. *Let C_0 be $\mathcal{RIQ}$ concepts and $\mathcal{R}$ regular RBox with a RIA of the form $w\dot{\sqsubseteq}QP$, where $Fun(P^-)$ holds. Let U be a new role name. We define*

$$C_1 := \forall U.(\forall w.(\exists P^-.\top)) \sqcap \forall w.(\exists P^-.\top),$$

and set

$$\mathcal{R}_1 := \mathcal{R}\backslash\{w\dot{\sqsubseteq}QP\} \cup \{UU\dot{\sqsubseteq}U, U^-\dot{\sqsubseteq}U\} \cup \{R\dot{\sqsubseteq}U | R \in \mathcal{R}_{C_0}\} \cup \{wP^-\dot{\sqsubseteq}Q\}.$$

Then, $\mathcal{RIQ}$ concept C_0 is satisfiable w.r.t. RBox $\mathcal{R}$ iff concept $C_0 \sqcap C_1$ is satisfiable w.r.t. Rbox $\mathcal{R}_1$.

Proof. The proof is based on transformation from one interpretation to another one. □

Without restriction $Fun(P^-)$, lemma (1) do not holds. It is illustrated in example (1).

Example 1. The UML[4] model of a car, shown on Figure (1a), describes Car with following parts: $Engine$, $Wheel$, $Crankshaft$ and Hub. Role name $powers$ is part-part relation [13, 2], but role names $engineInCar$, $wheelInCar$, $hubInWheel$ and $crankshaftInEngine$ are part-of relations [2]. The model corresponds to next RIA of the form [13]:

$$engineInCar \circ crankshaftInEngine \circ powers \dot{\sqsubseteq} wheelInCar \circ hubInWheel \quad (2)$$

Let $\mathcal{I}$ be an interpretation, shown on Figure (1b), of the RIA of the form (2). The interpretation $\mathcal{I}$ satisfies RIA of the form $w\dot{\sqsubseteq}wheelInCar \circ hubInWheel$, but it does not satisfy RIA of the form $w \circ hubInWheel^- \dot{\sqsubseteq} wheelInCar$, where $w = engineInCar \circ crankshaftInEngine \circ powers$. □

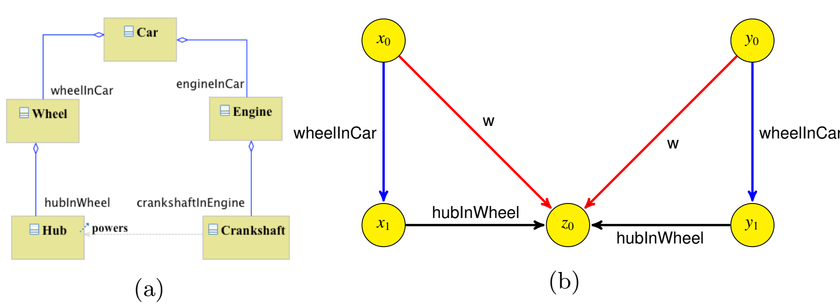

Fig. 1. (a) An UML product model (updated from [13]). (b) An interpretation $\mathcal{I}$ of RIA of the form (2).

[4] The Unified Modeling Language (http://www.uml.org/)

According to the Figure (1b), one can conclude that the restriction problems for reduction of RIAs is caused by the role $hubInWheel^{-}$. The role is not "unambiguously" determined. On the other side, by using the interpretation shown on the Figure (1b), it is obvious $\langle z_0, y_1\rangle \in (hubInWheel^{-})^{\mathcal{I}}$ corresponds to object y_0, while $\langle z_0, x_1\rangle \in (hubInWheel^{-})^{\mathcal{I}}$ corresponds to object x_0. In the other words, the condition of existence *unambiguously* role is connected to an object. To stressed which particular object corresponds to the role name, a new role (R, x) is defined. The role satisfies

$$(R, x)\dot{\sqsubseteq}R. \tag{3}$$

For example, in case of the interpretation, shown on Figure (1b), one can define new roles, as follows: $(hubInWheel^{-}, x_0)$, $(hubInWheel^{-}, y_0)$, which satisfy $\langle z_0, x_1\rangle \in (hubInWheel^{-}, x_0)^{\mathcal{I}}$, $\langle z_0, y_1\rangle \in (hubInWheel^{-}, y_0)^{\mathcal{I}}$, but $\langle z_0, x_1\rangle \notin (hubInWheel^{-}, y_0)^{\mathcal{I}}$. Now, one can define new tableau concept (constructor) denoted as $\underset{\forall}{\exists}$ $(hubInWheel^{-}, x).D$. This constructor is used in the label of nodes of the tableau (see definition 3). Intuitively, the constructor serves to write the set of sub-concepts of the concept C_0 which have to hold in some node, i.e. if $Z = \{D|\underset{\forall}{\exists}(hubInWheel^{-}, x_0).D \in \mathcal{L}(z_0)\} = \{D|\forall wheelInCar.D \in \mathcal{L}(x_0)\} \neq \emptyset$ then there exists x_1 such that $\langle z_0, x_1\rangle \in \mathcal{E}((hubInWheel^{-}, x_0))$ and $Z \subseteq \mathcal{L}(x_1)$.

4 The extension of $\mathcal{RIQ}$ tableau

This section examines how to extend tableau for the $\mathcal{RIQ}$ DL with the new constructor. We denote, as defined in [6], $\mathcal{B}_R$ as NFA that corresponds to role R. We use a special automaton for word w, denoted with $\mathcal{B}_w$. For $\mathcal{B}$ an NFA and q a state of $\mathcal{B}$, B^q denotes the NFA obtained from $\mathcal{B}$ by making q the (only) initial state of $\mathcal{B}$ [5]. The language recognized by NFA $\mathcal{B}$ is denoted by $\mathcal{L}(\mathcal{B})$. The $clos(C_0)$ is the smallest set of concepts in negation normal form (NNF) which contains C_0, that is closed under $\dot{\neg}$ and sub-concepts [6]. For a set S the $fclos(C_0, \mathcal{R})$ and $efc(C_0, \mathcal{R}, S)$ can be defined as follows:

$fclos(C_0, \mathcal{R}) = clos(C_0) \cup \{\forall \mathcal{B}_R^q.D|\forall R.D \in clos(C_0)$ and q is a state in $\mathcal{B}_R\}$,

$efc(C_0, \mathcal{R}, S) = fclos(C_0, \mathcal{R}) \cup \{\forall \mathcal{B}_w^q.\underset{\forall}{\exists}(P^{-}, s).D|s \in S,\ \forall Q.D \in clos(C_0)\} \cup \{\underset{\forall}{\exists}(P^{-}, s).D|s \in S,\ \forall Q.D \in clos(C_0)\}$.

Let's denote

$$PL(\mathcal{B}_w) = \left\{\langle w', q\rangle \,|q \text{ is a state in } \mathcal{B}_w, (\forall w'' \in \mathcal{L}(\mathcal{B}_w^q))\big(w'w'' \in \mathcal{L}(\mathcal{B}_w)\big)\right\}.$$

Definition 3. *T=(S, $\mathcal{L}$, $\mathcal{E}$) is tableau for concept C_0 w.r.t. $\mathcal{R}$ iff*[5]

- *S is non-empty set,*
- $\mathcal{L} : S \to 2^{efc(C_0, \mathcal{R}, S)}$,
- $\mathcal{E} : \mathcal{R}_{C_0} \cup \{(P^{-}, s)|s \in S\} \to 2^{S \times S}$
- $C_0 \in \mathcal{L}(s)$ *for some* $s \in S$

[5] w, Q and P refer to the RIA axiom of the form $w\dot{\sqsubseteq}Q \circ P$.

Next, $s,t \in S; C, C_1, C_2 \in fclos(C_0,R); R,S \in \mathcal{R}_{C_0}$ *and* $S^T(s,C)$ *[5] satisfies rules (P1a), (P1b), (P2), (P3), (P4a), (P4b), (P5), (P6), (P7), (P8), (P9), (P10), (P13) defined in [5], and satisfies new rules:*

- *(P6b) If* $\forall Q.C \in \mathcal{L}(s)$*, then* $\forall \mathcal{B}_w.\overset{\exists}{\forall}(P^-,s).C \in \mathcal{L}(s)$*.*
- *(P15a)* $\forall Q.\top \in \mathcal{L}(s)$ *for all* $s \in S$*.*
- *(P15b) If* $\overset{\exists}{\forall}(P^-,s).C_1 \in \mathcal{L}(v)$*, then there exists* t *with* $\langle v,t\rangle \in \mathcal{E}(P^-,s)$ *and* $C_1 \in \mathcal{L}(t)$*. Also, for all* $C_2 \in fclos(C_0)$*, if* $\overset{\exists}{\forall}(P^-,s).C_2$ $\in \mathcal{L}(v)$ *then* $C_2 \in \mathcal{L}(t)$*.*
- *(P15c) If* $\langle v,t\rangle \in \mathcal{E}(P^-,s)$*, then* $\langle v,t\rangle \in \mathcal{E}(P^-)$*.* □

Theorem 1. *Concept* C_0 *is satisfiable w.r.t.* $\mathcal{R}$ *iff there exists tableau for* C_0 *w.r.t.* $\mathcal{R}$*.*

Proof. For the if direction let $T = (S,\mathcal{L},\mathcal{E})$ be a tableau for C_0 w.r.t $\mathcal{R}$. We define interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, with: $\Delta^{\mathcal{I}} = S$, $C^{\mathcal{I}} = \{s | C \in \mathcal{L}(s)\}$, for concept names C in $clos(C_0)$, and for roles names $R \neq Q$ and Q, we set
$R^{\mathcal{I}} = \{\langle s_0, s_n\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} |$ there are $s_1,\ldots,s_{n-1}$ with $\langle s_i, s_{i+1}\rangle \in \mathcal{E}(S_{i+1})$ for $0 \le i \le n-1$ and $S_1 \cdots S_n \in \mathcal{L}(\mathcal{B}_R)\}$,
$Q^{\mathcal{I}} = \{\langle s_0, s_n\rangle \,|$ there are $s_1,\ldots,s_{n-1}$ with $\langle s_i,s_{i+1}\rangle \in \mathcal{E}(S_{i+1})$ for $0 \le i \le n-1$ and $S_1 \cdots S_n \in \mathcal{L}(\mathcal{B}_Q)\} \cup \{\langle x,y\rangle \,|(\exists z)(\langle x,z\rangle \in w^{\mathcal{I}}$ and $\langle z,y\rangle \in \mathcal{E}((P^-,x)))\}$.

Let's prove that $\mathcal{I}$ is model for $\mathcal{C}_0$ and $\mathcal{R}$.

<u>$\mathcal{I}$ is model for $\mathcal{R}$</u>. Let's consider RIA of the form $w \dot{\sqsubseteq} Q \circ P$. If $\langle x,y\rangle \in w^{\mathcal{I}}$. According to (P15a) and (P6b) then $\forall \mathcal{B}_w.\overset{\exists}{\forall}(P^-,x).\top \in \mathcal{L}(x)$ holds. According to (P4a), (P15b), (P15c) and definition of $Q^{\mathcal{I}}, P^{\mathcal{I}}$ we have $(\exists t)\,\langle y,t\rangle \in \mathcal{E}((P^-,x)), \langle y,t\rangle \in \mathcal{E}(P^-)$ and $\langle x,t\rangle \in Q^{\mathcal{I}}$, and finally implies $\langle x,y\rangle \in (Q \circ P)^{\mathcal{I}}$.
<u>$\mathcal{I}$ is model for C_0</u>. It is enough to prove that $C \in \mathcal{L}(s)$ implies $s \in C^{\mathcal{I}}$ for all $s \in S$ and $C \in clos(C_0)$. Let's consider $C \equiv \forall Q.D$. For other cases the proof is the same as proof in [6].

Let $\forall Q.D \in \mathcal{L}(s)$ and $(s,t) \in Q^{\mathcal{I}}$. If $\exists S_1 \cdots S_{n-1} \in \mathcal{L}(\mathcal{B}_Q)$, so $\langle s_i, s_{i+1}\rangle \in \mathcal{E}(S_{i+1}), i = 0,\ldots,n-1, s_0 = s, s_n = t$, then the proof is the same as proof in [6]. In case of $(\exists z)\,\langle s,z\rangle \in w^{\mathcal{I}}$ and $\langle z,t\rangle \in \mathcal{E}(P^-,s)$. Based on the definition of $\omega^{\mathcal{I}}$ and (P6b) we have $\overset{\exists}{\forall}(P^-,s).D \in \mathcal{L}(z)$. According to (P15b), we have $D \in \mathcal{L}(t)$. By induction $t \in D^{\mathcal{I}}$, so we have $s \in (\forall Q.D)^{\mathcal{I}}$.
For the converse, suppose that $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is model for C_0 w.r.t. $\mathcal{R}$. Let's define tableau $T = (S,\mathcal{L},\mathcal{E})$, as follows: $S = \Delta^{\mathcal{I}}$, $\mathcal{E}(R) = R^{\mathcal{I}}$, $\mathcal{E}((P^-,x)) = \{\langle y,z\rangle \in S^2 |\, \langle x,y\rangle \in w^{\mathcal{I}}, \langle x,z\rangle \in Q^{\mathcal{I}}, \langle z,y\rangle \in P^{\mathcal{I}}\}$, and

$\mathcal{L}(s) = \{C \in clos(C_0) | s \in C^{\mathcal{I}}\} \cup \{\forall \mathcal{B}_R.C | \forall R.C \in clos(C_0)$ and $s \in (\forall R.C)^{\mathcal{I}}\} \cup \{\forall \mathcal{B}^q_R.C \in fclos(C_0,\mathcal{R})|$ for all $S_1 \cdots S_n \in \mathcal{L}(\mathcal{B}^q_R), s \in (\forall S_1.$ $\forall S_2 \cdots \forall S_n.C)^{\mathcal{I}}$, and if $\varepsilon \in \mathcal{L}(\mathcal{B}^q_R)$ then $s \in C^{\mathcal{I}}\} \cup \{\forall Q.\top\} \cup \{\forall \mathcal{B}_w.\overset{\exists}{\forall}(P^-$ $,s).C | s \in (\forall Q.C)^{\mathcal{I}}\} \cup \{\forall \mathcal{B}^q_w.\overset{\exists}{\forall}(P^-,t).C | (\exists w')\,\langle w',q\rangle \in PL(\mathcal{B}_w)$ and $\langle t,s\rangle \in (w')^{\mathcal{I}}$ and $t \in (\forall Q.C)^{\mathcal{I}}\} \cup \{\overset{\exists}{\forall}(P^-,t).C |\, \langle t,s\rangle \in w^{\mathcal{I}}$ and $t \in (\forall Q.C)^{\mathcal{I}}\}$.

Let's prove that T is tableau for C_0 w.r.t. $\mathcal{R}$. We consider only new rules (see definition 3). From definition $\mathcal{E}(P^-,x)$ and $\mathcal{E}(P)$ we prove (P15c). From the definition of $\mathcal{L}(s)$ we have that (P15a) and (P6b) holds. Let's prove rule (P15b). Suppose that $\overset{\exists}{\forall}(P^-,s).C_1 \in \mathcal{L}(v)$. From the definition of $\mathcal{L}(v)$ follows

$\langle s,v\rangle \in w^{\mathcal{I}}$ and $s \in (\forall Q.C)^{\mathcal{I}}$. Because of $\mathcal{I} \models w\dot{\sqsubseteq}Q \circ P$ we have that there exists z such that $\langle s,z\rangle \in Q^{\mathcal{I}}$ and $\langle z,v\rangle \in P^{\mathcal{I}}$ i.e. $\langle v,z\rangle \in \mathcal{E}((P^{-},s))$. On the other hand, from $s \in (\forall Q.C)^{\mathcal{I}}$ and $\langle s,z\rangle \in Q^{\mathcal{I}}$ follows $z \in C_1^{\mathcal{I}}$, so $C_1 \in \mathcal{L}(z)$. If $\frac{\exists}{\forall}(P^{-},s).C_2 \in \mathcal{L}(v)$ then $s \in (\forall Q.C_2)^{\mathcal{I}}$, so $z \in C_2^{\mathcal{I}}$, i.e. $C_2 \in \mathcal{L}(z)$. □

5 Tableau algorithm

The tableau algorithm generates completion tree.

Definition 4. *(Completion tree) Completion tree for C_0 w.r.t $\mathcal{R}$ is labelled tree $G = (V, E, \mathcal{L}, \dot{\neq})$ where each node $x \in V$ is labelled with a set $\mathcal{L}(x) \subseteq efc(C_0, \mathcal{R}, V) \cup \{\leq mR.C | \leq nR.C \in clos(C_0), m \leq n\}$. Each edge $\langle x,y\rangle \in E$ is labelled with a set $\mathcal{L}\langle x,y\rangle \subseteq \mathcal{R}_{C_0} \cup \{(P^{-},s)|s \in V\}$. Additionally, we care of inequalities between nodes in V, of the tree G, with a symmetric binary relation $\dot{\neq}$.*
If $\langle x,y\rangle \in E$, then y is called successor of the x, but x is called predecessor of y. Ancestor is the transitive closure of predecessor, and descendant is the transitive closure of successor. A node y is called an R-successor of a node x if, for some R' with $R' \sqsubseteq_* R$, $R' \in \mathcal{L}(\langle x,y\rangle)$. A node y is called a neighbour (R-neighbour) of a node x if y is a successor (R-successor) of x or if x is a successor ($Inv(R)$-successor) of y. For $S \in \mathcal{R}_{C_0}$, $x \in V$, $C \in clos(C_0)$ we define set $S^G(x,C) = \{y | y \text{ is } S-neighbour \text{ of } x \text{ and } C \in \mathcal{L}(y)\}$ □

Definition 5. *A tree G is said to contain a clash if there is a node x such that:*

- *$\perp \in \mathcal{L}(x)$, or*
- *for a concept name A, $\{A, \neg A\} \subseteq \mathcal{L}(x)$, or*
- *there exists a concept $(\leq nS.C) \in \mathcal{L}(x)$ and $\{y_0, \ldots, y_n\} \in S^G(x,C)$ with $y_i \dot{\neq} y_j$ for all $0 \leq i < j \leq n$.* □

In order to provide termination of the algorithm, in [6] blocking techniques are used, and fact that the set of nodes' labels is finite. In our tableau definition (3), if S is infinite set then $efc(C_0, \mathcal{R}, S)$ is also infinite. So, number of different $\mathcal{L}(s)$ is infinite. Also, sets $\mathcal{L}(s)$ can be infinite. To ensure that sets $\mathcal{L}(s)$ are finite, we define additional restriction on the set of RIA of the form $w\dot{\sqsubseteq}Q \circ P$. Let's suppose that language $\mathcal{L}(\mathcal{B}_w)$ is finite.

If $\forall \mathcal{B}_w^q.\frac{\exists}{\forall}(P^{-},s).C \in \mathcal{L}(t)$ then there exists $(w',q) \in PL(\mathcal{B}_w)$ and $(s,t) \in \mathcal{E}(w')$. If $n = \sharp fclos(C_0, \mathcal{R})$, $l = max\{len(w')|(\exists q)(w',q) \in PL(\mathcal{B}_w)\}$ and number of $successors$ is less than m (different than $P - neighbours$), then [6]: $\sharp\mathcal{L}(t) \leq n \cdot m^l \cdot \sharp PL(\mathcal{B}_w)$. To illustrate the technique in an understandable way, we consider only special case, when $\mathcal{L}(\mathcal{B}_w) = \{R\}$.

Definition 6. *Let $G = (V, E, \mathcal{L}, \dot{\neq})$ be completion tree and $f : V \to V$ is a function.*

1. *We say that $\mathcal{L}(x)$ $f-match$ with $\mathcal{L}(y)$, denoted as $\mathcal{L}(x) \sim^f \mathcal{L}(y)$, if*

[6] Because of $\mathcal{L}(\mathcal{B}_w)$ is finite, then l, $\sharp PL(\mathcal{B}_w)$ are also finite.

- $f(x) = y$,
- $\mathcal{L}(x) \cap fclos(C_0, \mathcal{R}) = \mathcal{L}(y) \cap fclos(C_0, \mathcal{R})$,
- $R \in \mathcal{L}(\langle z, x\rangle) \Leftrightarrow R \in \mathcal{L}(\langle f(z), y\rangle)$,
- $\overset{\exists}{\forall}(P^-, z).C \in \mathcal{L}(x) \Leftrightarrow \overset{\exists}{\forall}(P^-, f(z)).C \in \mathcal{L}(y)$.

2. *We say that* $\mathcal{L}(\langle x, y\rangle)$ $f - match$ *with* $\mathcal{L}(\langle u, v\rangle)$, *denoted with* $\mathcal{L}(\langle x, y\rangle) \sim^f \mathcal{L}(\langle u, v\rangle)$, *if*
   - $\mathcal{L}(\langle x, y\rangle) \cap \mathcal{R}_{C_0} = \mathcal{L}(\langle u, v\rangle) \cap \mathcal{R}_{C_0}$,
   - $(\forall s \in V)((P^-, s) \in \mathcal{L}(\langle x, y\rangle) \Leftrightarrow (P^-, f(s)) \in \mathcal{L}(\langle u, v\rangle))$. □

Definition 7. *(Blocking) A node x is label blocked if there is a function* $f : V \to V$ *and there are predecessors* x', y, y' *of the node x, such that*

- $x' \neq y$,
- *x is successor of* x' *and y is successor of* y',
- $\mathcal{L}(x) \sim^f \mathcal{L}(y)$, $\mathcal{L}(x') \sim^f \mathcal{L}(y')$,
- $\mathcal{L}(\langle x, x'\rangle) \sim^f \mathcal{L}(\langle y, y'\rangle)$.

In this case we say that y blocks x. □

A node is blocked if it is label blocked or its predecessor is blocked. If the predecessor of a node x is blocked, then we say that x is indirectly blocked [5]. There is an algorithm that checks whether a node y blocks node x. It is enough to consider nodes x, y and their predecessors x' and y' and (finite number of) R-neighbours of these four nodes. For the nodes, function f can be nondeterministically defined and check the rules in the definition (7). It is also possible to check the rules algorithmically, because the rules use only finite sets.

The non-deterministic tableau algorithm can be described as follows:

- Input: Concept C_0 and RBox $\mathcal{R}$,
- Output: "Yes" if concept C_0 is satisfiable w.r.t. RBox $\mathcal{R}$, otherwise "No"
- Method:
  1. step: Construct tree $G = (V, E, \mathcal{L}, \dot{\neq})$, where $V = \{x_0\}$, $E = \emptyset$, $\mathcal{L}(x_0) = \{C_0\}$. Go to step 2.
  2. step: Apply an expansion rule (see table 1) to the tree G, while it is possible. Otherwise, go to step 3.
  3. step: If the tree does not contain *clash* return "Yes", otherwise return "No".

Theorem 2. *1. Tableau algorithm terminates when started with* C_0 *and* $\mathcal{R}$,
2. Tableau algorithm returns answer "Yes" iff there exists tableau of the concept C_0 *w.r.t* $\mathcal{R}$.

Proof. (a) $\exists$-rule and $\geq$-rule generate finite number of successors of node x. So, the set $\mathcal{L}(x)$ is finite and the number of (P^-, y)-successors of node x is finite. There is limited number the possible labels of pairs $(x', x) \in E$ that will lead the blocking of tree nodes. It means, the tree generated by the algorithm is finite. According to [6], the rule which generates node y and remove rule $\leq$, will not be applied, again. This means that the algorithm can applied only finite number of expansion rules.

$\sqcap$-rule:	If then	$C_1 \sqcap C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and $\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$, $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C_1, C_2\}$
$\sqcup$-rule:	If then	$C_1 \sqcup C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and $\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$, $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{E\}$, for some $E \in \{C_1, C_2\}$
$\exists$-rule:	If then	$\exists S.C \in \mathcal{L}(x)$, x is not blocked, and x has no S-negihbour y where $C \in \mathcal{L}(y)$ create new node y where $\mathcal{L}(\langle x, y\rangle) := \{S\}$ and $\mathcal{L}(y) := \{C\}$
$\forall_1$-rule:	If then	$\forall S.C \in \mathcal{L}(x)$, x is not indirectly blocked, and $\forall \mathcal{B}_S.C \notin \mathcal{L}(x)$ $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{\forall \mathcal{B}_S.C\}$
$\forall_2$-rule:	If then	$\forall \mathcal{B}^p.C \in \mathcal{L}(x)$, x is not indirectly blocked, $p \xrightarrow{S} q \in \mathcal{B}^p$ and there is an S-neighbour y of x with $\forall \mathcal{B}^q.C \notin \mathcal{L}(y)$ $\mathcal{L}(y) \rightarrow \mathcal{L}(y) \cup \{\forall \mathcal{B}^q.C\}$
$\forall_3$-rule:	If then	$\forall \mathcal{B}.C \in \mathcal{L}(x)$, x is not indirectly blocked, $\varepsilon \in \mathcal{L}(\mathcal{B})$ and $C \notin \mathcal{L}(x)$ $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C\}$
$\forall_4$-rule:	If then	$\forall Q.C \in \mathcal{L}(x)$, x is not indirectly blocked, and $\forall \mathcal{B}_w.\frac{\exists}{\forall}(P^-, x).C \notin \mathcal{L}(x)$ $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{\forall \mathcal{B}_w.\frac{\exists}{\forall}(P^-, x).C\}$
$\forall_5$-rule:	If then	$\forall Q.\top \notin \mathcal{L}(x)$, x is not indirectly blocked $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{\forall Q.\top\}$
$(\frac{\exists}{\forall})_1$-rule:	If then	$\frac{\exists}{\forall}(P^-, x).C \in \mathcal{L}(y)$, y is not blocked and there is no z with $(P^-, x) \in \mathcal{L}(\langle y, z\rangle)$ create new node z with $(P^-, x) \in \mathcal{L}(\langle y, z\rangle)$, $P^- \in \mathcal{L}(\langle y, z\rangle)$
$(\frac{\exists}{\forall})_2$-rule:	If then	$\frac{\exists}{\forall}(P^-, x).C \in \mathcal{L}(y)$, y is not blocked, there is z with $(P^-, x) \in \mathcal{L}(\langle y, z\rangle)$ and $C \notin \mathcal{L}(z)$ $\mathcal{L}(z) \rightarrow \mathcal{L}(z) \cup \{C\}$
choose-rule:	If then	$(\leq nS.C) \in \mathcal{L}(x)$, x is not indirectly blocked, and there is an S-neighbour y of x $\{C, \dot{\neg} C\} \cap \mathcal{L}(y) = \emptyset$ $\mathcal{L}(y) \rightarrow \mathcal{L}(y) \cup \{E\}$, for some $E \in \{C, \dot{\neg} C\}$
$\geq$-rule:	If then	(1) $(\geq nS.C) \in \mathcal{L}(x)$, x is not blocked, and (2) there are not n S-neighbours $y_1, \ldots, y_n$ of x with $C \in \mathcal{L}(y_i)$ and $y_i \dot{\neq} y_j$ for $1 \leq i < j \leq n$, create n new nodes $y_1, \ldots y_n$ with $\mathcal{L}(\langle x, y_i\rangle) = \{S\}$, $\mathcal{L}(y_i) = \{C\}$ and $y_i \dot{\neq} y_j$ for $1 \leq i < j \leq n$
$\leq$-rule:	If then	(1) $(\leq nS.C) \in \mathcal{L}(x)$, x is not indirectly blocked (2) $\sharp S^{\mathbf{G}}(x, C) > n$ and there are $y, z \in S^{\mathbf{G}}(x, C)$ with not $y \dot{\neq} z$ and y is not root node nor an ancestor of z (1) $\mathcal{L}(z) \rightarrow \mathcal{L}(z) \cup \mathcal{L}(y)$ (2) if z is an ancestor of x, then $\mathcal{L}(\langle z, x\rangle) \rightarrow \mathcal{L}(\langle z, x\rangle) \cup Inv(\mathcal{L}(\langle x, y\rangle))$ else $\mathcal{L}(\langle x, z\rangle) \rightarrow \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}(\langle x, y\rangle)$ (3) set $u \dot{\neq} z$, for all u with $u \dot{\neq} y$ (4) remove y and sub-tree below y from G

Table 1. Expansion rules for a tableau algorithm (updated from [5])

(b) For the if direction, suppose that the algorithm returns "Yes". It means that the algorithm generated tree $G = (V, E, \mathcal{L}, \dot{\neq})$ without clash and there are no expansion rules (see table 1) that can be applied.

A path [5, 8] is a sequence of pairs of nodes of G of the form

$$p = \langle (x_0, x_0'), \ldots, (x_n, x_n') \rangle \,. \tag{4}$$

For such a path, we define $Tail(p) = x_n$ and $Tail'(p) = x_n'$. We denote the path

$$\big\langle (x_0, x_0'), (x_1, x_1'), \ldots, (x_n, x_n'), (x_{n+1}, x_{n+1}') \big\rangle \,. \tag{5}$$

with $\big\langle p, (x_{n+1}, x_{n+1}') \big\rangle$. If node x is label blocked then corresponding function f is denoted with f_x. The node x is blocked with node $f_x(x)$. We use function: $G(x, z) = z$, if x is not blocked, or $G(x, z) = f_x(z)$, if x is blocked.

The set $Paths(G)$ is defined inductively, as follows:

- If $x_0 \in V$ is the root of tree then $\langle x_0, x_0 \rangle \in Paths(G)$,
- If $p \in Paths(G)$ and $z \in V$and z is not indirectly blocked, such that $\langle Tail(p), z \rangle \in E$, then $\langle p, (G(z, z), z) \rangle \in Paths(G)$.

Let's define structure $\mathcal{T} = \{\mathcal{S}, \mathcal{L}', \mathcal{E}\}$ as follows:

$\mathcal{S} = Paths(G)$,

$\mathcal{E}(S) = \{\langle p, q \rangle \in Paths(G) \times Paths(G) | q = \langle p, (G(z, z), z) \rangle$ and $S \in \mathcal{L}(\langle Tail(p), z \rangle)$ or $p = \langle q, (G(z, z), z) \rangle$ and $Inv(S) \in \mathcal{L}(\langle Tail(q), z \rangle)\}$, for $S \in \mathcal{R}_{C_0}$,

$\mathcal{E}(P^-, r) = \{\langle p, q \rangle \in \mathcal{E}(P^-) | \langle r, p \rangle \in \mathcal{E}(R)$ and $(P^-, G(Tail'(p), Tail'(r)))$ $\in \mathcal{L}(G(Tail'(p), Tail'(p)), Tail'(q))\}$,

$\mathcal{L}'(p) = \mathcal{L}(Tail(p)) \cap fclos(C_0, \mathcal{R}) \cup \{\forall R. \overset{\exists}{\forall}(P^-, p).C | \forall Q.C \in \mathcal{L}(Tail(p))\} \cup \{\overset{\exists}{\forall}(P^-, r).C | \langle r, p \rangle \in \mathcal{E}(R)$ and $\overset{\exists}{\forall}(P^-, Tail'(r)).C \in \mathcal{L}(Tail'(p))\}$.

Let's prove that $\mathcal{T}$ is tableau for C_0 w.r.t R. We consider only (P15b) property, and avoid already defined properties in [6]. New properties (P6b), (P15a), (P15c) imply from $\forall_4$, $\forall_5$ and $(\overset{\exists}{\forall})_1$.

Suppose $\overset{\exists}{\forall}(P^-, r).C \in \mathcal{L}'(p)$ then $\langle r, p \rangle \in \mathcal{E}(R)$ and $\overset{\exists}{\forall}(P^-, Tail'(r)).C \in \mathcal{L}(Tail'(p))$. Because of $\langle r, p \rangle \in \mathcal{E}(R)$, four cases are possible:

1. $p = \langle r, (G(z, z), z) \rangle$ and $G(z, z) = z$
2. $p = \langle r, (G(z, z), z) \rangle$ and $G(z, z) \neq z$
3. $r = \langle p, (G(z, z), z) \rangle$ and $G(z, z) = z$
4. $r = \langle p, (G(z, z), z) \rangle$ and $G(z, z) \neq z$

The subcases above are analyzing on the similar way and we consider the most complex of them i.e. case (2). The $Tail'(r)$ is not blocked, so $Tail'(r) = Tail(r)$, while z is blocked by $G(z, z)$. From $\langle r, p \rangle \in \mathcal{E}(R)$ blocking definition we have $R \in \mathcal{L}(\langle Tail(r), z \rangle)$ and $R \in \mathcal{L}(G(z, Tail(r)), G(z, z))$, while, from $\overset{\exists}{\forall}(P^-, Tail'(r)).C \in \mathcal{L}(z)$ we have $\overset{\exists}{\forall}(P^-, G(z,\ Tail(r)).C \in \mathcal{L}(G(z, z))$. According to the rule $(\overset{\exists}{\forall})_1$, we have that there exists node y such that $P^-, (P^-, G(z, Tail(r)) \in \mathcal{L}(\langle G(z, z), y \rangle)$. Let $q = \langle p, (G(y, y), y) \rangle$ then $\langle p, q \rangle \in \mathcal{E}(P^-)$ and $(P^-, G(Tail'(p), Tail'(r))) \in \mathcal{L}(G(Tail'(p),\ Tail'(p)), Tail'(q))$, so $\langle p, q \rangle \in \mathcal{E}(P^-, r)$. Having regard to the rule $(\overset{\exists}{\forall})_2$ we conclude that property (P15b) holds.

For the only-if direction, the proof is the same as proof in [6] (i.e., we take a tableau and use it to steer the application of the non-deterministic rules). □

6 Conclusions and future works

This paper shortly examines how to handle complex RIAs with more than one role from the right hand side of the composition of roles in $\mathcal{RIQ}$ DL. Although the proof was conducted for RIA of the form $R\dot{\sqsubseteq}Q{\circ}P$, we can apply the technique to RIA of the form $S_1S_2\cdots S_n\dot{\sqsubseteq}R_1R_2\cdots R_m$, with restriction that corresponding languages are finite. Our future work will be focused on the problem which conditions should satisfy role if we have more than one RIAs, to be mention technique could be applied. Also, we will do research on RIA of the form $w\dot{\sqsubseteq}QP$ when the language $\mathcal{L}(\mathcal{B}_w)$ is infinite.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook - Theory, Implementation and Application.* Cambridge University Press, second edition, 2007.
2. C. Bock. UML 2 Composition Model. *Journal of Object Technology*, 3(10):47-73, 2004.
3. C. Bock, X. F. Zha, H-W. Suh, J-H. Lee, Ontological Product Modeling for Collaborative Design, U.S. National Institute of Standards and Technology, Technical Report NISTIR 7643, 2009.
4. B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. P. Schneider, and U. Sattler. OWL 2: The next step for OWL. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web.* 6(4): 309-322, 2008.
5. I. Horrocks, O. Kutz, and U. Sattler. The Even More Irresistible $\mathcal{SROIQ}$. *In Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006).* pages 57-67, 2006.
6. I. Horrocks, U. Sattler. Decidability of $\mathcal{SHIQ}$ with Complex Role Inclusion Axioms. *Artificial Intelligence*, 160(1-2):79-104, 2004.
7. I. Horrocks, U. Sattler. A Tableaux Decision Procedure for $\mathcal{SHOIQ}$. *Journal of Automated Reasoning.* Springer Verlag, 39(3): 245-429, 2007.
8. I. Horrocks, and U. Sattler. Optimised Reasoning for SHIQ. *In Proceedings of the 15th European Conf. on Artificial Intelligence (ECAI 2002)*, pages 277-281, 2002.
9. I. Horrocks, O. Kutz, and U. Sattler. The irresistible $\mathcal{SRIQ}$. *In Proceedings of the OWLED*05 Workshop on OWL: Experiences and Directions*, 2005.
10. I. Horrocks, , P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The Making of a Web Ontology Language. *Journal of Web Semantics.* 1(1):7-26, 2003.
11. Y. Kazakov. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are Harder than $\mathcal{SHOIQ}$. *In Proceedings of Description Logics Workshop (DL 2008), CEUR-Workshop.* Vol. 353, 2008.
12. Y. Kazakov. An Extension of Complex Role Inclusion Axioms in the Description Logic $\mathcal{SROIQ}$. *In Proceedings of the 5th International Joint Conference on Automated Reasoning (IJCAR 2010).* pages 472-487, Edinburgh, UK, 2010.
13. N. Krdžavac, C. Bock. Reasoning in Manufacturing Part-Part Examples with OWL 2. U.S. National Institute of Standards and Technology, Technical Report NISTIR 7535, 2008.
14. B. Motik, R. Shearer, and I. Horrocks. Hypertableau Reasoning for Description Logics. *Journal of Articial Intelligence Research.* 36: 165-228, 2009.