An ExpTime Tableau Method for Dealing with Nominals and Quantified Number Restrictions in Deciding the Description Logic SHOQ

Linh Anh Nguyen[1,2] and Joanna Golińska-Pilarek[3]

[1] Institute of Informatics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
nguyen@mimuw.edu.pl
[2] Faculty of Information Technology, VNU University of Engineering and Technology
144 Xuan Thuy, Hanoi, Vietnam
[3] Institute of Philosophy, University of Warsaw
Krakowskie Przedmieście 3, 00-927 Warsaw, Poland
j.golinska@uw.edu.pl

Abstract. We present the first tableau method with an ExpTime (optimal) complexity for checking satisfiability of a knowledge base in the description logic $\mathcal{SHOQ}$, which extends $\mathcal{ALC}$ with transitive roles, hierarchies of roles, nominals and quantified number restrictions. The complexity is measured using binary representation for numbers. Our procedure is based on global caching and integer linear feasibility checking.

1 Introduction

Description logics (DLs) are formal languages suitable for representing terminological knowledge. They are of particular importance in providing a logical formalism for ontologies and the Semantic Web. Automated reasoning in DLs is useful, for example, in engineering and querying ontologies. One of basic reasoning problems in DLs is to check satisfiability of a knowledge base in a considered DL. Most of other reasoning problems in DLs are reducible to this one.

In this paper we study the problem of checking satisfiability of a knowledge base in the DL $\mathcal{SHOQ}$, which extends the basic DL $\mathcal{ALC}$ with transitive roles (S), hierarchies of roles (H), nominals (O) and quantified number restrictions (Q). It is known that this problem in $\mathcal{SHOQ}$ is ExpTime-complete [16] (even when numbers are coded in binary).

Nominals, interpreted as singleton sets, are a useful notion to express identity and uniqueness. However, when interacting with inverse roles (I) and quantified number restrictions in the DL $\mathcal{SHOIQ}$, they cause the complexity of the above mentioned problem to jump up to NExpTime-complete [15] (while that problem in any of the DLs $\mathcal{SHOQ}$, $\mathcal{SHIO}$, $\mathcal{SHIQ}$ is ExpTime-complete [16, 7, 15]).

In [8] Horrocks and Sattler gave a tableau algorithm for deciding the DL $\mathcal{SHOQ}$(D), which is the extension of $\mathcal{SHOQ}$ with concrete datatypes. Later, Pan and Horrocks [13] extended the method of [8] to give a tableau algorithm

for deciding the DL $\mathcal{SHOQ}(\mathrm{D}_n)$, which is the extension of $\mathcal{SHOQ}$ with n-ary datatype predicates. These algorithms use backtracking to deal with disjunction ($\sqcup$) and "or"-branching (e.g., the "choose"-rule) and use a straightforward way for dealing with with quantified number restrictions. They have a non-optimal complexity (NExpTime) when unary representation is used for numbers, and have a higher complexity (N2ExpTime) when binary representation is used. In [1] Faddoul and Haarslev gave an algebraic tableau reasoning algorithm for $\mathcal{SHOQ}$, which combines the tableau method with linear integer programming. The aim was to increase efficiency of handling quantified number restrictions. However, their algorithm still uses backtracking to deal with disjunction and "or"-branching and has a non-optimal complexity ("double exponential" [1]).

In this paper we present the first tableau method with an ExpTime (optimal) complexity for checking satisfiability of a knowledge base in the DL $\mathcal{SHOQ}$. The complexity is measured using binary representation for numbers. Our procedure is based on global caching and integer linear feasibility checking.

The idea of global caching comes from Pratt's work [14] on PDL. It was formally formulated for tableaux in some DLs in [3, 4] and has been applied to several modal and description logics (see [12] for references) to obtain tableau decision procedures with an optimal (ExpTime) complexity. A variant of global caching, called global state caching, was used to obtain cut-free optimal (ExpTime) tableau decision procedures for several modal logics with converse and DLs with inverse roles [5, 6, 9, 11].

Integer linear programming was exploited for tableaux in [2, 1] to increase efficiency of reasoning with quantified number restrictions. However, the first work that applied integer linear feasibility checking to tableaux was [10, 11]. In [10], Nguyen gave the first ExpTime (optimal) tableau decision procedure for checking satisfiability of a knowledge base in the DL $\mathcal{SHIQ}$, where the complexity is measured using binary representation for numbers. His procedure is based on global state caching and integer linear feasibility checking. In the current paper, we apply his method of integer linear feasibility checking to $\mathcal{SHOQ}$. It substantially differs from Farsiniamarj's method of exploiting integer programming for tableaux [2]. Our method of dealing with both nominals and quantified number restrictions is essentially different from the one by Faddoul and Haarslev [1].

Due to the lack of space, we restrict ourselves to introducing the problem of checking satisfiability of a knowledge base in the DL $\mathcal{SHOQ}$ and presenting some examples to illustrate our tableau method. For a full description of a tableau decision procedure with an ExpTime complexity we refer the reader to [12].

2 Notation and Semantics of $\mathcal{SHOQ}$

Our language uses a finite set $\mathbf{C}$ of *concept names*, a finite set $\mathbf{R}$ of *role names*, and a finite set $\mathbf{I}$ of *individual names*. A concept name stands for a unary predicate, a role name stands for a binary predicate, and an individual name stands for a constant. We use letters like A and B for concept names, r and s for role

names, and a and b for individual names. We also refer to A and B as *atomic concepts*, to r and s as *roles*, and to a and b as *individuals*.

An ($\mathcal{SHOQ}$) *RBox* $\mathcal{R}$ is a finite set of role axioms of the form $r \sqsubseteq s$ or $r \circ r \sqsubseteq r$. For example, $link \sqsubseteq path$ and $path \circ path \sqsubseteq path$ are such role axioms.

By $ext(\mathcal{R})$ we denote the least extension of $\mathcal{R}$ such that:

- $r \sqsubseteq r \in ext(\mathcal{R})$ for any role r
- if $r \sqsubseteq r' \in ext(\mathcal{R})$ and $r' \sqsubseteq r'' \in ext(\mathcal{R})$ then $r \sqsubseteq r'' \in ext(\mathcal{R})$.

Let $r \sqsubseteq_{\mathcal{R}} s$ denote $r \sqsubseteq s \in ext(\mathcal{R})$, and $trans_{\mathcal{R}}(r)$ denote $(r \circ r \sqsubseteq r) \in ext(\mathcal{R})$. If $r \sqsubseteq_{\mathcal{R}} s$ then r is a *subrole* of s (w.r.t. $\mathcal{R}$). If $trans_{\mathcal{R}}(s)$ then s is a *transitive role* (w.r.t. $\mathcal{R}$). A role is *simple* (w.r.t. $\mathcal{R}$) if it is neither transitive nor has any transitive subrole (w.r.t. $\mathcal{R}$).

Concepts in $\mathcal{SHOQ}$ are formed using the following BNF grammar, where n is a nonnegative integer and s is a simple role:

$$C, D ::= \top \mid \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists r.C \mid \forall r.C \mid \{a\} \mid \geq n\, s.C \mid \leq n\, s.C$$

A concept stands for a set of individuals. The concept $\top$ stands for the set of all individuals (in the considered domain). The concept $\bot$ stands for the empty set. The constructors $\neg$, $\sqcap$ and $\sqcup$ stand for the set operators: complement, intersection and union. For the remaining forms, we give some examples: $\exists hasChild.Male$, $\forall hasChild.Female$, $\geq 2\, hasChild.Teacher$, $\leq 5\, hasChild.\top$.

We use letters like C and D to denote arbitrary concepts.

A *TBox* is a finite set of axioms of the form $C \sqsubseteq D$ or $C \doteq D$.

An *ABox* is a finite set of *assertions* of the form $a\!:\!C$, $r(a, b)$ or $a \neq b$.

An axiom $C \sqsubseteq D$ means C is a subconcept of D, while $C \doteq D$ means C and D are equivalent concepts. An assertion $a\!:\!C$ means a is an instance of concept C, and $a \neq b$ means a and b are distinct individuals.

A *knowledge base* in $\mathcal{SHOQ}$ is a tuple $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{R}$ is an RBox, $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox.

We say that a role s is *numeric* w.r.t. a knowledge base $KB = \langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ if:

- it is simple w.r.t. $\mathcal{R}$ and occurs in a concept $\geq n\, s.C$ or $\leq n\, s.C$ in KB, or
- $s \sqsubseteq_{\mathcal{R}} r$ and r is numeric w.r.t. KB.

We will simply call such an s a numeric role when KB is clear from the context.

A *formula* is defined to be either a concept or an ABox assertion. We use letters like φ, ψ and ξ to denote formulas. Let $\mathsf{null}\!:\!C$ stand for C. We use α to denote either an individual or null. Thus, $\alpha\!:\!C$ is a formula of the form $a\!:\!C$ or $\mathsf{null}\!:\!C$ (which means C).

An *interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ consists of a non-empty set $\Delta^{\mathcal{I}}$, called the *domain* of $\mathcal{I}$, and a function $\cdot^{\mathcal{I}}$, called the *interpretation function* of $\mathcal{I}$, that maps each concept name A to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each role name r to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, and each individual name a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The interpretation function is extended to complex concepts as follows, where $\sharp Z$

denotes the cardinality of a set Z:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad \bot^{\mathcal{I}} = \emptyset \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} - C^{\mathcal{I}}$$
$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}} \qquad \{a\}^{\mathcal{I}} = \{a^{\mathcal{I}}\}$$
$$(\exists r.C)^{\mathcal{I}} = \big\{x \in \Delta^{\mathcal{I}} \mid \exists y\big[\langle x, y\rangle \in r^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\big]\big\}$$
$$(\forall r.C)^{\mathcal{I}} = \big\{x \in \Delta^{\mathcal{I}} \mid \forall y\big[\langle x, y\rangle \in r^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}\big]\big\}$$
$$(\geq n\, s.C)^{\mathcal{I}} = \big\{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \mid \langle x, y\rangle \in s^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geq n\big\}$$
$$(\leq n\, s.C)^{\mathcal{I}} = \big\{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \mid \langle x, y\rangle \in s^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leq n\big\}.$$

For a set Γ of concepts, define $\Gamma^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid x \in C^{\mathcal{I}} \text{ for all } C \in \Gamma\}$.

The relational composition of binary relations R_1, R_2 is denoted by $\mathrm{R}_1 \circ \mathrm{R}_2$.

An interpretation $\mathcal{I}$ is a *model of an RBox* $\mathcal{R}$ if for every axiom $r \sqsubseteq s$ (resp. $r \circ r \sqsubseteq r$) of $\mathcal{R}$, we have that $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$ (resp. $r^{\mathcal{I}} \circ r^{\mathcal{I}} \subseteq r^{\mathcal{I}}$). Note that if $\mathcal{I}$ is a model of $\mathcal{R}$ then it is also a model of $ext(\mathcal{R})$.

An interpretation $\mathcal{I}$ is a *model of a TBox* $\mathcal{T}$ if for every axiom $C \sqsubseteq D$ (resp. $C \doteq D$) of $\mathcal{T}$, we have that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (resp. $C^{\mathcal{I}} = D^{\mathcal{I}}$).

An interpretation $\mathcal{I}$ is a *model of an ABox* $\mathcal{A}$ if for every assertion $a\!:\!C$ (resp. $r(a, b)$ or $a \neq b$) of $\mathcal{A}$, we have that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ (resp. $\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in r^{\mathcal{I}}$ or $a^{\mathcal{I}} \neq b^{\mathcal{I}}$).

An interpretation $\mathcal{I}$ is a *model of a knowledge base* $\langle \mathcal{R}, \mathcal{T}, \mathcal{A}\rangle$ if $\mathcal{I}$ is a model of $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{A}$. A knowledge base $\langle \mathcal{R}, \mathcal{T}, \mathcal{A}\rangle$ is *satisfiable* if it has a model.

An interpretation $\mathcal{I}$ *satisfies* a concept C (resp. a set X of concepts) if $C^{\mathcal{I}} \neq \emptyset$ (resp. $X^{\mathcal{I}} \neq \emptyset$). It *validates* C if $C^{\mathcal{I}} = \Delta^{\mathcal{I}}$. A set X of concepts is *satisfiable w.r.t. an RBox* $\mathcal{R}$ *and a TBox* $\mathcal{T}$ if there exists a model of $\mathcal{R}$ and $\mathcal{T}$ that satisfies X. For $X = \mathcal{A} \cup \mathcal{A}'$, where $\mathcal{A}$ is an ABox and $\mathcal{A}'$ is a set of assertions of the form $\neg r(a, b)$ or $a \doteq b$, we say that X is *satisfiable w.r.t. an RBox* $\mathcal{R}$ *and a TBox* $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\langle \mathcal{R}, \mathcal{T}, \mathcal{A}\rangle$ such that: $\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \notin r^{\mathcal{I}}$ for all $(\neg r(a, b)) \in \mathcal{A}'$, and $a^{\mathcal{I}} = b^{\mathcal{I}}$ for all $(a \doteq b) \in \mathcal{A}'$. In that case, we also say that $\mathcal{I}$ is a model of $\langle \mathcal{R}, \mathcal{T}, X\rangle$.

3 A Tableau Method for $\mathcal{SHOQ}$

We assume that concepts and ABox assertions are represented in negation normal form (NNF), where $\neg$ occurs only directly before atomic concepts. We use $\overline{C}$ to denote the NNF of $\neg C$, and for $\varphi = (a\!:\!C)$, we use $\overline{\varphi}$ to denote $a\!:\!\overline{C}$. For simplicity, we treat axioms of $\mathcal{T}$ as concepts representing global assumptions: an axiom $C \sqsubseteq D$ is treated as $\overline{C} \sqcup D$, while an axiom $C \doteq D$ is treated as $(\overline{C} \sqcup D) \sqcap (\overline{D} \sqcup C)$. That is, we assume that $\mathcal{T}$ consists of concepts in NNF. Thus, an interpretation $\mathcal{I}$ is a model of $\mathcal{T}$ iff $\mathcal{I}$ validates every concept $C \in \mathcal{T}$.

Let $\mathit{EdgeLabels} = \{\mathsf{testingCloseness}, \mathsf{checkingFeasibility}\} \times \mathcal{P}(\mathbf{R}) \times (\mathbf{I} \cup \{\mathsf{null}\})$. For $e \in \mathit{EdgeLabels}$, let $e = \langle \pi_t(e), \pi_r(e), \pi_i(e)\rangle$. Thus, $\pi_t(e)$ is the type of e, $\pi_r(e)$ is a set of roles, and $\pi_i(e)$ is either an individual or null.

We define a tableaux as a rooted graph. Such a graph is a tuple $G = \langle V, E, \nu\rangle$, where V is a set of nodes, $E \subseteq V \times V$ is a set of edges, $\nu \in V$ is the root, each

node $v \in V$ has a number of attributes, and each edge $\langle v, w\rangle$ may have a number of labels from *EdgeLabels*. Attributes of a tableau node v are:

- $\mathit{Type}(v) \in \{\mathsf{state}, \mathsf{non\text{-}state}\}$.
- $\mathit{SType}(v) \in \{\mathsf{complex}, \mathsf{simple}\}$ is called the subtype of v.
- $\mathit{Status}(v) \in \{\mathsf{unexpanded}, \mathsf{p\text{-}expanded}, \mathsf{f\text{-}expanded}, \mathsf{closed}, \mathsf{open}, \mathsf{blocked}\} \cup \{\mathsf{closed\text{-}wrt}(U) \mid U \subseteq V$ and $\mathit{Type}(u) = \mathsf{state} \wedge \mathit{SType}(u) = \mathsf{complex}$ for all $u \in U\}$, where p-expanded and f-expanded mean "partially expanded" and "fully expanded", respectively. $\mathit{Status}(v)$ may be p-expanded only when $\mathit{Type}(v) = \mathsf{state}$. If $\mathit{Status}(v) = \mathsf{closed\text{-}wrt}(U)$ then we say that the node v is closed w.r.t. the nodes from U.
- $\mathit{Label}(v)$ is a finite set of formulas, called the label of v.
- $\mathit{RFmls}(v)$ is a finite set of formulas, called the set of reduced formulas of v.
- $\mathit{IndRepl}(v) : \mathbf{I} \to \mathbf{I}$ is a partial mapping specifying replacements of individuals. It is available only when v is a complex node. If $\mathit{IndRepl}(v)(a) = b$ then, at the node v, we have $a \doteq b$ and b is the representative of its abstract class.
- $\mathit{ILConstraints}(v)$ is a set of integer linear constraints. It is available only when $\mathit{Type}(v) = \mathsf{state}$. The constraints use variables x_e indexed by labels e of edges outgoing from v such that $\pi_t(e) = \mathsf{checkingFeasibility}$. Such a variable specifies how many copies of the successor via e will be created for v.

If $\langle v, w\rangle \in E$ then we call v a *predecessor* of w, and w a *successor* of v. An edge outgoing from a node v has labels iff $\mathit{Type}(v) = \mathsf{state}$. When defined, the set of labels of an edge $\langle v, w\rangle$ is denoted by $\mathit{ELabels}(v, w)$. If $e \in \mathit{ELabels}(v, w)$ then $\pi_i(e) = \mathsf{null}$ iff $\mathit{SType}(v) = \mathsf{simple}$.

A node v is called a *state* if $\mathit{Type}(v) = \mathsf{state}$, and *non-state* otherwise. It is called a *complex node* if $\mathit{SType}(v) = \mathsf{complex}$, and a *simple node* otherwise. The label of a complex node consists of ABox assertions, while the label of a simple node consists of concepts. The root ν is a complex non-state.

A node may have status blocked only when it is a simple node with the label containing a nominal $\{a\}$. The status blocked can be updated only to closed or closed-wrt(...). We write closed-wrt(...) to mean closed-wrt(U) for some U.

The graph G consists of two layers: the layer of complex nodes and the layer of simple nodes. There are no edges from simple nodes to complex nodes. The edges from complex nodes to simple nodes are exactly the edges outgoing from complex states. That is: if $\langle v, w\rangle$ is an edge from a complex node v to a simple node w then $\mathit{Type}(v) = \mathsf{state}$; if $\mathit{Type}(v) = \mathsf{state}$ and $\langle v, w\rangle \in E$ then $\mathit{SType}(w) = \mathsf{simple}$. Each complex node of G is like an ABox (more formally: its label is an ABox), which can be treated as a graph whose vertices are named individuals. On the other hand, a simple node of G stands for an unnamed individual. If e is a label of an edge from a complex state v to a simple node w then the triple $\langle v, e, w\rangle$ can be treated as an edge from the named individual $\pi_i(e)$ (an inner node of the graph representing v) to the unnamed individual corresponding to w, and that edge is via the roles from $\pi_r(e)$.

We will use also assertions of the form $a : (\preceq n\, s.C)$ and $a : (\succeq n\, s.C)$, where s is a numeric role. The difference between $a : (\preceq n\, s.C)$ and $a : (\leq n\, s.C)$ is that, for checking $a : (\preceq n\, s.C)$, we do not have to pay attention to assertions of the

form $s(a,b)$ or $r(a,b)$ with r being a subrole of s. The aim for $a:(\succeq n\, s.C)$ is similar. We use $a:(\preceq n\, s.C)$ and $a:(\succeq n\, s.C)$ only as syntactic representations of some expressions, and do not provide semantics for them. We define

$$\texttt{FullLabel}(v) = \mathit{Label}(v) \cup \mathit{RFmls}(v) - \{\text{formulas of the form } a\colon(\preceq n\, s.C) \text{ or } a\colon(\succeq n\, s.C)\}.$$

We apply global caching: if $v_1, v_2 \in V$, $\mathit{Label}(v_1) = \mathit{Label}(v_2)$ and ($\mathit{SType}(v_1) = \mathit{SType}(v_2) =$ simple or ($\mathit{SType}(v_1) = \mathit{SType}(v_2) =$ complex and $\mathit{Type}(v_1) = \mathit{Type}(v_2)$)) then $v_1 = v_2$. Due to global caching, an edge outgoing from a state may have a number of labels as the result of merging edges.

We say that a node v may affect the status of the root ν if there exists a path consisting of nodes $v_0 = \nu, v_1, \ldots, v_{n-1}, v_n = v$ such that, for every $0 \leq i < n$, $\mathit{Status}(v_i)$ differs from open and closed, and if it is closed-wrt(U) then U is disjoint from $\{v_0, \ldots, v_i\}$. In that case, if $u \in \{v_1, \ldots, v_n\}$ then we say that v may affect the status of the root ν via a path through u.

From now on, let $\langle \mathcal{R}, \mathcal{T}, \mathcal{A}\rangle$ be a knowledge base in NNF of the logic $\mathcal{SHOQ}$, with $\mathcal{A} \neq \emptyset$.[4] In this section we present a tableau calculus $C_{\mathcal{SHOQ}}$ for checking satisfiability of $\langle \mathcal{R}, \mathcal{T}, \mathcal{A}\rangle$. A $C_{\mathcal{SHOQ}}$-tableau for $\langle \mathcal{R}, \mathcal{T}, \mathcal{A}\rangle$ is a rooted graph $G = \langle V, E, \nu\rangle$ constructed as follows:

Initialization: $V := \{\nu\}$, $E := \emptyset$, $\mathit{Type}(\nu) :=$ non-state, $\mathit{SType}(\nu) :=$ complex, $\mathit{Status}(\nu) :=$ unexpanded, $\mathit{RFmls}(\nu) := \emptyset$, $\mathit{Label}(\nu) := \mathcal{A} \cup \{(a:C) \mid C \in \mathcal{T}$ and a is an individual occurring in $\mathcal{A}$ or $\mathcal{T}\}$, for each individual a occurring in $\mathit{Label}(\nu)$ set $\mathit{IndRepl}(\nu)(a) := a$.

Rules' Priorities and Expansion Strategies: The graph is then expanded by the following rules, which are specified in detail in [12]:

(UPS) rules for updating statuses of nodes,
(US) unary static expansion rules,
(DN) a rule for dealing with nominals,
(NUS) a non-unary static expansion rule,
(FS) the forming-state rule,
(TP) a transitional partial-expansion rule,
(TF) a transitional full-expansion rule.

Each of the rules is parametrized by a node v. We say that a rule is *applicable* to v if it can be applied to v to make changes to the graph. The rule (UPS) has a higher priority than (US), which has a higher priority than the remaining rules in the list. If neither (UPS) nor (US) is applicable to any node, then choose a node v with status unexpanded or p-expanded, choose the first rule applicable to v among the rules in the last five items of the above list, and apply it to v. Any strategy can be used for choosing v, but it is worth to choose v for expansion only when v may affect the status of the root ν of the graph. Note that the priorities of the rules are specified by the order in the above list, but the rules (UPS) and (US) are checked globally (technically, they are triggered immediately when possible), while the remaining rules are checked for a chosen node.

[4] If $\mathcal{A}$ is empty, we can add $a:\top$ to it, where a is a special individual.

Termination: The construction of the graph ends when the root ν receives status closed or open or when no more changes that may affect the status of ν can be made[5].

To check satisfiability of $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ one can construct a $C_{\mathcal{SHOQ}}$-tableau for it, then return "no" when the root of the tableau has status closed, or "yes" in the other case. It can be proved that (see [12]):

- A $C_{\mathcal{SHOQ}}$-tableau for $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ can be constructed in ExpTime.
- If $G = \langle V, E, \nu \rangle$ is an arbitrary $C_{\mathcal{SHOQ}}$-tableau for $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ then $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ is satisfiable iff $\mathit{Status}(\nu) \neq \mathsf{closed}$.

Remark 3.1. Our technique for dealing with quantified number restrictions is similar to Nguyen's technique used for $\mathcal{SHIQ}$ [10, 11]. There are some technical differences, which are caused by that we use global caching for $\mathcal{SHOQ}$, while Nguyen's work [10] uses global state caching for $\mathcal{SHIQ}$ (due to inverse roles) and inverse roles can interact with quantified number restrictions.

We briefly explain our technique for dealing with nominals. Suppose v is a simple node with $\mathit{Status}(v) \notin \{\mathsf{closed}, \mathsf{open}\}$ and $\{a\} \in \mathit{Label}(v)$, a complex state u is an ancestor of v, and v may affect the status of the root ν via a path through u. Let u_0 be a predecessor of u. The node u_0 has only u as a successor and it was expanded by the forming-state rule. There are three cases:

- If, for every $C \in \mathit{Label}(v)$, the formula obtained from $a{:}C$ by replacing every individual b by $\mathit{IndRepl}(u)(b)$ belongs to $\mathtt{FullLabel}(u)$, then v is "consistent" with u.
- If there exists $C \in \mathit{Label}(v)$ such that the formula obtained from $a : \overline{C}$ by replacing every individual b by $\mathit{IndRepl}(u)(b)$ belongs to $\mathtt{FullLabel}(u)$, then v is "inconsistent" with u. In this case, if $\mathit{Status}(v)$ is of the form $\mathsf{closed\text{-}wrt}(U)$ then we update it to $\mathsf{closed\text{-}wrt}(U \cup \{u\})$, else we update it to $\mathsf{closed\text{-}wrt}(\{u\})$.
- In the remaining case, the node u is "incomplete" w.r.t. v, which means that the expansion of u_0 was not appropriate. Thus, we delete the edge $\langle u_0, u \rangle$ and re-expand u_0 by an appropriate "or"-branching (see [12]).

There are also treatments for dealing with assertions of the form $a{:}\{b\}$ and for updating statuses of nodes in the presence of $\mathsf{closed\text{-}wrt}(\ldots)$. □

4 Illustrative Examples

Example 4.1. Let us construct a $C_{\mathcal{SHOQ}}$-tableau for $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$, where

$$\mathcal{A} = \{a{:}A,\ a{:}\exists r.\exists r.(A \sqcup \{a\}),\ a{:}\geq 3\, r.\forall r.\neg A,\ a{:}\forall r.B,\ a{:}\leq 3\, r.B,\\ r(a,b),\ b{:}\forall r.\neg A,\ b{:}(\forall r.(\neg A \sqcap \neg\{a\}) \sqcup \neg B)\},$$

$\mathcal{R} = \emptyset$ and $\mathcal{T} = \emptyset$. An illustration is presented in Figure 1.

[5] That is, ignoring nodes that are unreachable from ν via a path without nodes with status closed or open, no more changes can be made to the graph.

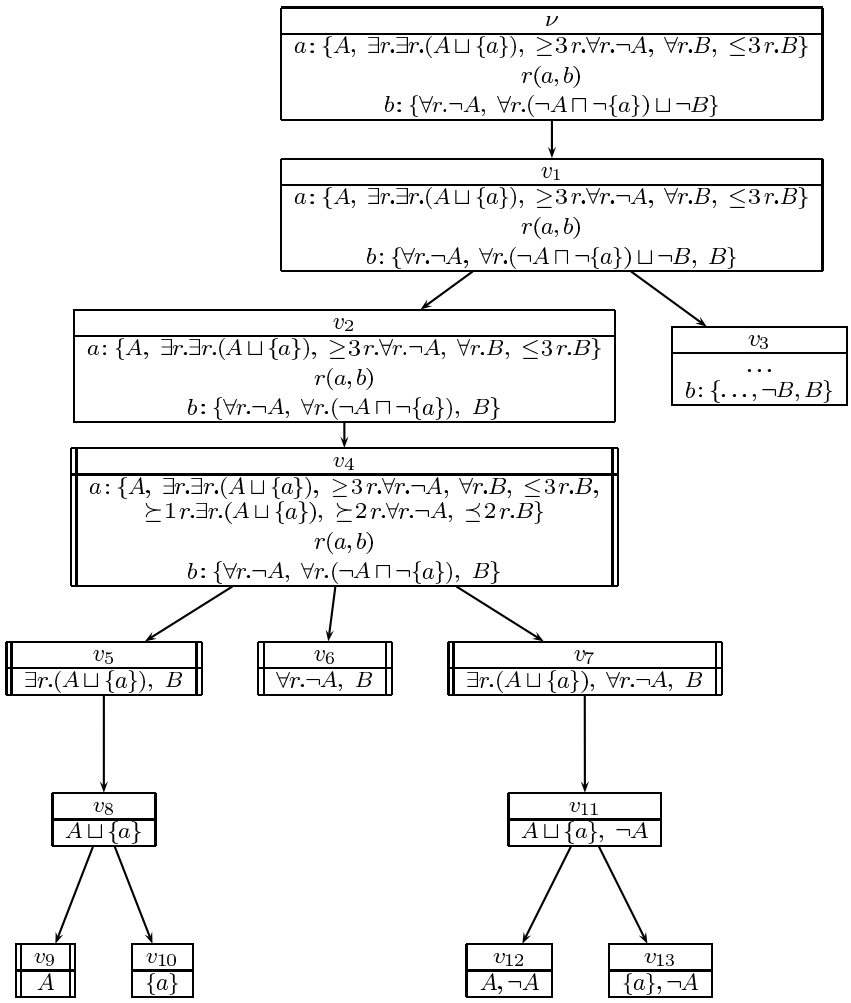

Fig. 1. An illustration of the tableau described in Example 4.1. The marked nodes v_4 – v_7 and v_9 are states. The nodes ν, v_1 – v_4 are complex nodes, the remaining are simple nodes. In each node, we display the formulas of its label.

At the beginning, the graph has only the root ν which is a complex non-state with $\mathit{Label}(\nu) = \mathcal{A}$. Since $\{a : \forall r.B, r(a,b)\} \subset \mathit{Label}(\nu)$, applying a unary static expansion rule to ν, we connect it to a new complex non-state v_1 with $\mathit{Label}(v_1) = \mathit{Label}(\nu) \cup \{b : B\}$.

Since $b : (\forall r.(\neg A \sqcap \neg\{a\}) \sqcup \neg B) \in \mathit{Label}(v_1)$, applying the non-unary static expansion rule to v_1, we connect it to new complex non-states v_2 and v_3 with

$$\mathit{Label}(v_2) = \mathit{Label}(v_1) - \{b : (\forall r.(\neg A \sqcap \neg\{a\}) \sqcup \neg B)\} \cup \{b : \forall r.(\neg A \sqcap \neg\{a\})\}$$
$$\mathit{Label}(v_3) = \mathit{Label}(v_1) - \{b : (\forall r.(\neg A \sqcap \neg\{a\}) \sqcup \neg B)\} \cup \{b : \neg B\}.$$

Since both $b : B$ and $b : \neg B$ belong to $Label(v_3)$, the node v_3 receives status closed. Applying the forming-state rule to v_2, we connect it to a new complex state v_4 with

$$Label(v_4) = Label(v_2) \cup \{a : \succeq 1\, r.\exists r.(A \sqcup \{a\}),\ a : \succeq 2\, r.\forall r.\neg A,\ a : \preceq 2\, r.B\}.$$

The assertion $a : \succeq 1\, r.\exists r.(A \sqcup \{a\}) \in Label(v_4)$ is due to $a : \exists r.\exists r.(A \sqcup \{a\}) \in Label(v_2)$ and the fact that the negation of $b : \exists r.(A \sqcup \{a\})$ in NNF belongs to $Label(v_2)$ (notice that $r(a, b) \in Label(v_2)$). The assertion $a : \succeq 2\, r.\forall r.\neg A \in Label(v_4)$ is due to $a : \geq 3\, r.\forall r.\neg A \in Label(v_2)$ and the fact that $\{r(a, b), b : \forall r.\neg A\} \subset Label(v_2)$. Similarly, the assertion $a : \preceq 2\, r.B \in Label(v_4)$ is due to $a : \leq 3\, r.B \in Label(v_2)$ and the fact $\{r(a, b), b : B\} \subset Label(v_2)$.

As r is a numeric role, applying the transitional partial-expansion rule[6] to v_4, we just change the status of v_4 to p-expanded. After that, applying the transitional full-expansion rule to v_4, we connect it to new simple non-states v_5, v_6, v_7 using edges labeled by $e_{4,5}$, $e_{4,6}$, $e_{4,7}$, respectively, such that $e_{4,i} = \langle \mathsf{checkingFeasibility}, \{r\}, a\rangle$ for $5 \leq i \leq 7$, and $Label(v_5) = \{\exists r.(A \sqcup \{a\}),\ B\}$, $Label(v_6) = \{\forall r.\neg A,\ B\}$, $Label(v_7) = \{\exists r.(A \sqcup \{a\}),\ \forall r.\neg A,\ B\}$. The creation of v_5 is caused by $a : \succeq 1\, r.\exists r.(A \sqcup \{a\}) \in Label(v_4)$, while the creation of v_6 is caused by $a : \succeq 1\, r.\forall r.\neg A$. The node v_7 results from merging v_5 and v_6. Furthermore, $ILConstraints(v_4)$ consists of $x_{e_{4,i}} \geq 0$, for $5 \leq i \leq 7$, and

$$\begin{aligned} x_{e_{4,5}} + x_{e_{4,7}} &\geq 1 \\ x_{e_{4,6}} + x_{e_{4,7}} &\geq 2 \\ x_{e_{4,5}} + x_{e_{4,6}} + x_{e_{4,7}} &\leq 2. \end{aligned}$$

Applying the forming-state rule to v_5, the type of this node is changed from non-state to state. Next, applying the transitional partial-expansion rule to v_5, its status is changed to p-expanded. Then, applying the transitional full-expansion rule to v_5, we connect v_5 to a new simple non-state v_8 with $Label(v_8) = \{A \sqcup \{a\}\}$ using an edge labeled by $e_{5,8}$ and set $ILConstraints(v_5) = \{x_{e_{5,8}} \geq 0, x_{e_{5,8}} \geq 1\}$.

Applying the non-unary static expansion rule to v_8, we connect it to new simple non-states v_9 and v_{10} with $Label(v_9) = \{A\}$ and $Label(v_{10}) = \{\{a\}\}$. The status of v_9 is then changed to open, which causes the statuses of v_8 and v_5 to be updated to open. The node v_{10} is not expanded as it does not affect the status of the root node ν.

Applying the forming-state rule to v_6, the type of this node is changed from non-state to state. Next, applying the transitional partial-expansion rule and then the transitional full-expansion rule to v_6, its status is changed to f-expanded. The status of v_6 is then updated to open.

Applying the forming-state rule to v_7, the type of this node is changed from non-state to state. Next, applying the transitional partial-expansion rule to v_7, its status is changed to p-expanded. Then, applying the transitional full-expansion rule to v_7, we connect v_7 to a new simple non-state v_{11} with $Label(v_{11}) = \{A \sqcup$

[6] which is used for making transitions via non-numeric roles

$\{a\}$, $\neg A\}$ using an edge labeled by $e_{7,11}$ and set $\mathit{ILConstraints}(v_7) = \{x_{e_{7,11}} \geq 0$, $x_{e_{7,11}} \geq 1\}$.

Applying the non-unary static expansion rule to v_{11}, we connect it to new simple non-states v_{12} and v_{13} with $\mathit{Label}(v_{12}) = \{A, \neg A\}$ and $\mathit{Label}(v_{13}) = \{\{a\}, \neg A\}$. The status of v_{12} is then changed to closed. Since $a : A \in \mathit{Label}(v_4)$, the status of v_{13} is updated to closed-wrt($\{v_4\}$), which causes the status of v_{11} to be updated also to closed-wrt($\{v_4\}$). As the set $\mathit{ILConstraints}(v_7) \cup \{x_{e_{7,11}} = 0\}$ is infeasible, the status of v_7 is updated to closed-wrt($\{v_4\}$). Next, as the set $\mathit{ILConstraints}(v_4) \cup \{x_{e_{4,7}} = 0\}$ is infeasible, the status of v_4 is first updated to closed-wrt($\{v_4\}$) and then updated to closed. After that, the statuses of v_2, v_1, ν are sequentially updated to closed. Thus, we conclude that the knowledge base $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ is unsatisfiable. □

Example 4.2. Let us modify Example 4.1 by deleting the assertion $a : A$ from the ABox. That is, we are now constructing a $C_{\mathcal{SHOQ}}$-tableau for $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$, where

$$\mathcal{A} = \{a : \exists r.\exists r.(A \sqcup \{a\}),\ a : \geq 3\, r.\forall r.\neg A,\ a : \forall r.B,\ a : \leq 3\, r.B,$$
$$r(a, b),\ b : \forall r.\neg A,\ b : (\forall r.(\neg A \sqcap \neg\{a\}) \sqcup \neg B)\},$$

$\mathcal{R} = \emptyset$ and $\mathcal{T} = \emptyset$. The first stage of the construction is similar to the one of Example 4.1, up to the step of updating the status of v_{12} to closed. This stage is illustrated in Figure 2, which is similar to Figure 1 except that the labels of the nodes ν and v_1 – v_4 do not contain $a : A$. The continuation is described below and illustrated by Figure 3.

Since $\mathit{Label}(v_{13}) = \{\{a\}, \neg A\}$, applying the rule for dealing with nominals to v_{13}, we delete the edge $\langle v_2, v_4 \rangle$ (from E) and re-expand v_2 by connecting it to new complex non-states v_{14} and v_{15} with $\mathit{Label}(v_{14}) = \mathit{Label}(v_2) \cup \{a : \neg A\}$ and $\mathit{Label}(v_{15}) = \mathit{Label}(v_2) \cup \{a : A\}$ as shown in Figure 3. The status of v_{13} is updated to blocked. The node v_4 is not deleted, but we do not display it in Figure 3.

Applying the forming-state rule to v_{14} we connect it to a new complex state v_{16}. The label of v_{16} is computed using $\mathit{Label}(v_{14})$ in a similar way as in Example 4.1 when computing $\mathit{Label}(v_4)$.

Applying the transitional partial-expansion rule to v_{16} we change its status to p-expanded. After that, applying the transitional full-expansion rule to v_{16} we connect it to the existing nodes v_5, v_6, v_7 by using edges labeled by $e_{16,5}$, $e_{16,6}$, $e_{16,7}$, respectively, which are the same tuple $\langle \mathsf{checkingFeasibility}, \{r\}, a \rangle$. The set $\mathit{ILConstraints}(v_{16})$ consists of $x_{e_{16,i}} \geq 0$, for $5 \leq i \leq 7$, and

$$x_{e_{16,5}} + x_{e_{16,7}} \geq 1$$
$$x_{e_{16,6}} + x_{e_{16,7}} \geq 2$$
$$x_{e_{16,5}} + x_{e_{16,6}} + x_{e_{16,7}} \leq 2.$$

Applying the forming-state rule to v_{15} we connect it to a new complex state v_{17}. The label of v_{17} is computed using $\mathit{Label}(v_{15})$ in a similar way as in Example 4.1 when computing $\mathit{Label}(v_4)$.

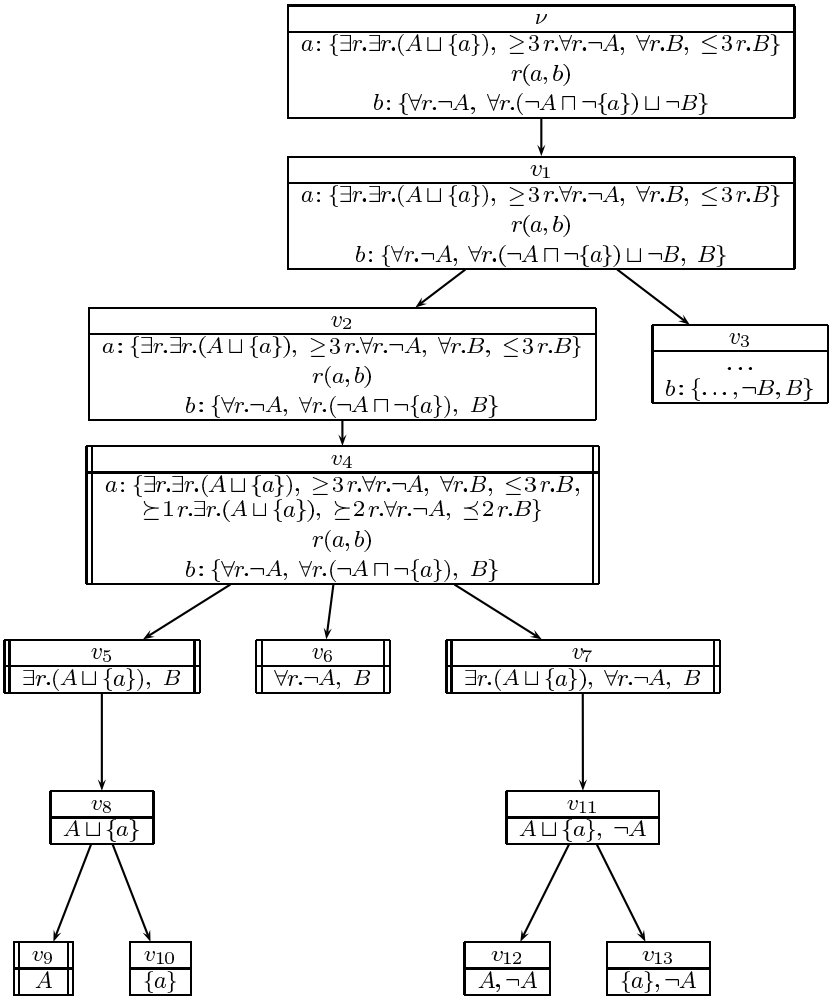

Fig. 2. An illustration for Example 4.2 – Part I.

The expansion of v_{17} is similar to the expansion of v_{16}. The set $\mathit{ILConstraints}(v_{17})$ is like $\mathit{ILConstraints}(v_{16})$, with the subscripts 16 replaced by 17. Analogously to updating the statuses of the nodes v_{13}, v_{11}, v_7 in Example 4.1 to $\mathsf{closed\text{-}wrt}(\{v_4\})$, the statuses of v_{13}, v_{11}, v_7 are updated to $\mathsf{closed\text{-}wrt}(\{v_{17}\})$. Next, as $\mathit{ILConstraints}(v_{17}) \cup \{x_{e_{17,7}} = 0\}$ is infeasible, the status of v_{17} is first updated to $\mathsf{closed\text{-}wrt}(\{v_{17}\})$ and then updated to closed. After that, the status of v_{15} is also updated to closed. As no more changes that may affect the status of ν can be made and $\mathit{Status}(\nu) \neq \mathsf{closed}$, we conclude that the knowledge base $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ is satisfiable. □

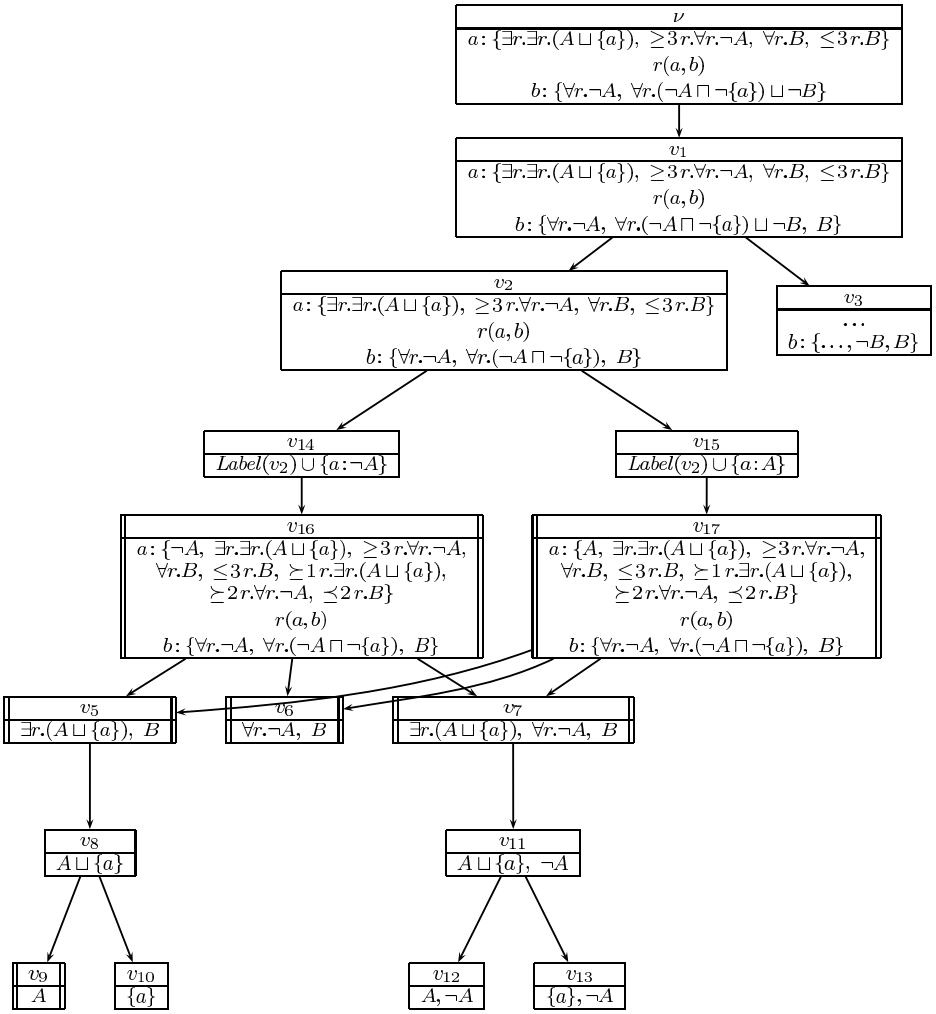

Fig. 3. An illustration for Example 4.2 – Part II.

5 Conclusions

We have presented the first tableau method with an ExpTime (optimal) complexity for checking satisfiability of a knowledge base in the DL $\mathcal{SHOQ}$. The complexity is measured using binary representation for numbers. Our detailed tableau decision procedure for $\mathcal{SHOQ}$ is given in [12].

This work differs from Nguyen's work [10] on $\mathcal{SHIQ}$ in that nominals are allowed instead of inverse roles. Without inverse roles, global caching is used instead of global state caching to allow more cache hits. To deal with nominals, we use additional statuses closed-wrt(...) for nodes of the graph to be constructed.

Acknowledgments. This work was supported by Polish National Science Centre (NCN) under Grants No. 2011/01/B/ST6/02759 (for the first author) and 2011/02/A/HS1/00395 (for the second author).

References

1. J. Faddoul and V. Haarslev. Algebraic tableau reasoning for the description logic SHOQ. *J. Applied Logic*, 8(4):334–355, 2010.
2. N. Farsiniamarj. Combining integer programming and tableau-based reasoning: a hybrid calculus for the description logic SHQ. Master's thesis, Concordia University, 2008.
3. R. Goré and L.A. Nguyen. ExpTime tableaux with global caching for description logics with transitive roles, inverse roles and role hierarchies. In *Proceedings of TABLEAUX 2007*, volume 4548 of *LNAI*, pages 133–148. Springer, 2007.
4. R. Goré and L.A. Nguyen. Exptime tableaux for ALC using sound global caching. *J. Autom. Reasoning*, 50(4):355–381, 2013.
5. R. Goré and F. Widmann. Sound global state caching for $\mathcal{ALC}$ with inverse roles. In M. Giese and A. Waaler, editors, *Proceedings of TABLEAUX 2009*, volume 5607 of *LNCS*, pages 205–219. Springer, 2009.
6. R. Goré and F. Widmann. Optimal and cut-free tableaux for propositional dynamic logic with converse. In J. Giesl and R. Hähnle, editors, *Proceedings of IJCAR 2010*, volume 6173 of *LNCS*, pages 225–239. Springer, 2010.
7. J. Hladik and J. Model. Tableau systems for SHIO and SHIQ. In *Proceedings of DL'2004*, volume 104 of *CEUR Workshop Proceedings*, pages 168–177, 2004.
8. I. Horrocks and U. Sattler. Ontology reasoning in the SHOQ(D) description logic. In *Proceedings of IJCAI'2001*, pages 199–204. Morgan Kaufmann, 2001.
9. L.A. Nguyen. A cut-free ExpTime tableau decision procedure for the description logic SHI. In *Proceedings of ICCCI'2011 (1)*, volume 6922 of *LNCS*, pages 572–581. Springer, 2011 (see also the long version `http://arxiv.org/abs/1106.2305v1`).
10. L.A. Nguyen. ExpTime tableaux for the description logic SHIQ based on global state caching and integer linear feasibility checking. arXiv:1205.5838, 2012.
11. L.A. Nguyen. A tableau method with optimal complexity for deciding the description logic SHIQ. In *Proceedings of ICCSAMA'2013*, volume 479 of *Studies in Computational Intelligence*, pages 331–342. Springer, 2013.
12. L.A. Nguyen and J. Golińska-Pilarek. A long version of the current paper. `http://www.mimuw.edu.pl/~nguyen/shoq-long.pdf`, 2013.
13. J.Z. Pan and I. Horrocks. Reasoning in the SHOQ(D_n) description logic. In *Proc. of DL'2002*, volume 53 of *CEUR Workshop Proceedings*, pages 53–62, 2002.
14. V.R. Pratt. A near-optimal method for reasoning about action. *J. Comp. Syst. Sci.*, 20(2):231–254, 1980.
15. S. Tobies. *Complexity results and practical algorithms for logics in knowledge representation*. PhD thesis, RWTH-Aachen, 2001.
16. `http://owl.cs.manchester.ac.uk/navigator/`.