An ExpSpace Tableau-based Algorithm for $\mathcal{SHOIQ}$

Chan Le Duc[1], Myriam Lamolle[1], and Olivier Curé[2]

[1] LIASD Université Paris 8 - IUT de Montreuil, France
{chan.leduc, myriam.lamolle}@iut.univ-paris8.fr
[2] LIGM Université Paris-Est, France
ocure@univ-mlv.fr

Abstract. In this paper, we propose an ExpSpace tableau-based algorithm for $\mathcal{SHOIQ}$. The construction of this algorithm is founded on the standard tableau-based method for $\mathcal{SHOIQ}$ and the technique used for designing a NExpTime algorithm for the two-variable fragment of first-order logic with counting quantifiers $\mathcal{C}^2$.

1 Introduction

The ontology language OWL-DL [6] is widely used to formalize semantic resources on the Semantic Web. This language is mainly based on the description logic $\mathcal{SHOIQ}$ which is known to be decidable [9]. There have been several works on the consistency problem of a $\mathcal{SHOIQ}$ knowledge base. These works have not only shown decidability and complexity of the problem but also led to develop and implement efficient systems for reasoning on OWL-based ontologies. Tobies [9] has shown that the consistency problem of a $\mathcal{SHOIQ}$ knowledge base is NExpTime-complete. Horrocks *et al.* [2] have proposed a tableau-based algorithm that has been exploited to implement reasoners such as Pellet [8], which inherit from the success of early Description Logic reasoners such as FaCT [1].

It has been shown that when nominals are added to DLs the consistency problem is harder. In fact, the complexity jumps from ExpTime-complete for $\mathcal{SHIQ}$ to NExpTime-complete for $\mathcal{SHOIQ}$ [9]. Kazakov *et al.* [4] have indicated that when nominals are allowed in $\mathcal{SHIQ}$, the resolution-based approach yields a triple exponential decision procedure for the consistency problem. The authors have also identified that the interaction between nominals, inverse roles and number restrictions makes termination more difficult to be achieved, and thus, is responsible for this hardness.

Our approach is inspired from the technique that was employed by Pratt-Hartmann [7] to construct a NExpTime algorithm for the logic $\mathcal{C}^2$ that almost includes $\mathcal{SHOIQ}$. Unlike the existing tableau-based algorithms, this technique does not explicitly build a graph for representing a model but it builds a structure, called a *frame*, from *star-types* each of which represents a set of individuals. Pratt-Hartmann [7] shows that a model of a $\mathcal{C}^2$ knowledge base can be constructed from a frame tiled by *well selected* star-types.

The present paper is structured as follows. In the next section, we describe the logic $\mathcal{SHOIQ}$ and the consistency problem for a $\mathcal{SHOIQ}$ knowledge base. Section 3 describes a 2ExpSpace tableau-based algorithm for checking consistency of a $\mathcal{SHOIQ}$

knowledge base. An advantage of this algorithm is that a tree-like structure can be maintained to obtain termination. Section 4 transfers results from $\mathcal{C}^2$ [7] to $\mathcal{SHOIQ}$, and presents an EXPSPACE tableau-based algorithm for $\mathcal{SHOIQ}$. Finally, we discuss the results and future work. For the lack of place, we refer the reader to [5] for examples and full proofs.

2 The Description Logic $\mathcal{SHOIQ}$

In this section, we present the syntax and the semantics of $\mathcal{SHOIQ}$. We start by defining a role hierarchy and its semantics.

Definition 1 (role hierarchy). *Let $\mathbf{R}$ be a non-empty set of* role names *and $\mathbf{R}_+ \subseteq \mathbf{R}$ be a set of transitive role names. We use $\mathbf{R}_\mathsf{I} = \{P^- \mid P \in \mathbf{R}\}$ to denote a set of inverse roles. Each element of $\mathbf{R} \cup \mathbf{R}_\mathsf{I}$ is called a $\mathcal{SHOIQ}$-role. We define $R^\ominus := R^-$ if $R \in \mathbf{R}$, and $R^\ominus := R$ if $R \in \mathbf{R}_\mathsf{I}$. A* role hierarchy *$\mathcal{R}$ is a finite set of* role inclusion axioms *$R \sqsubseteq S$ where R and S are two $\mathcal{SHOIQ}$-roles. A relation $\underline{\circledast}$ is defined as the transitive-reflexive closure of $\sqsubseteq$ on $\mathcal{R} \cup \{R^\ominus \sqsubseteq S^\ominus \mid R \sqsubseteq S \in \mathcal{R}\}$. We define a function* $\mathsf{Trans}(R)$ *which returns* true *iff there is some $Q \in \mathbf{R}_+ \cup \{P^\ominus \mid P \in \mathbf{R}_+\}$ such that $Q \underline{\circledast} R$. A role R is called* simple *w.r.t. $\mathcal{R}$ if $\mathsf{Trans}(Q) = \mathsf{false}$. An* interpretation *$\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of a non-empty set $\Delta^\mathcal{I}$ (*domain*) and a function $\cdot^\mathcal{I}$ which maps each role name to a subset of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$ such that $R^{-\mathcal{I}} = \{\langle x, y\rangle \in \Delta^\mathcal{I} \times \Delta^\mathcal{I} \mid \langle y, x\rangle \in R^\mathcal{I}\}$ for all $R \in \mathbf{R}$, and $\langle x, z\rangle \in S^\mathcal{I}, \langle z, y\rangle \in S^\mathcal{I}$ implies $\langle x, y\rangle \in S^\mathcal{I}$ for each $S \in \mathbf{R}_+$. An interpretation $\mathcal{I}$ satisfies a role hierarchy $\mathcal{R}$ if $R^\mathcal{I} \subseteq S^\mathcal{I}$ for each $R \sqsubseteq S \in \mathcal{R}$. Such an interpretation is called a* model *of $\mathcal{R}$, denoted by $\mathcal{I} \models \mathcal{R}$.*

Definition 2 (terminology). *Let $\mathbf{C}$ be a non-empty set of* concept names *with a non-empty subset $\mathbf{C}_o \subseteq \mathbf{C}$ of nominals. The set of $\mathcal{SHOIQ}$-concepts is inductively defined as the smallest set containing all C in $\mathbf{C}$, $\top$, $C \sqcap D$, $C \sqcup D$, $\neg C$, $\exists R.C$, $\forall R.C$, $(\leq n\, S.C)$ and $(\geq n\, S.C)$ where n is a positive integer, C and D are $\mathcal{SHOIQ}$-concepts, R is an $\mathcal{SHOIQ}$-role and S is a simple role w.r.t. a role hierarchy. We denote $\bot$ for $\neg\top$. The interpretation function $\cdot^\mathcal{I}$ of an interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ maps each concept name to a subset of $\Delta^\mathcal{I}$ such that $\top^\mathcal{I} = \Delta^\mathcal{I}$, $(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$, $(C \sqcup D)^\mathcal{I} = C^\mathcal{I} \cup D^\mathcal{I}$, $(\neg C)^\mathcal{I} = \Delta^\mathcal{I} \backslash C^\mathcal{I}$, $\mathsf{card}\{o^\mathcal{I}\} = 1$ for all $o \in \mathbf{C}_o$, $(\exists R.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \exists y \in \Delta^\mathcal{I}, \langle x, y\rangle \in R^\mathcal{I} \wedge y \in C^\mathcal{I}\}$, $(\forall R.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \forall y \in \Delta^\mathcal{I}, \langle x, y\rangle \in R^\mathcal{I} \Rightarrow y \in C^\mathcal{I}\}$, $(\geq n\, S.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \mathsf{card}\{y \in C^\mathcal{I} \mid \langle x, y\rangle \in S^\mathcal{I}\} \geq n\}$, $(\leq n\, S.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \mathsf{card}\{y \in C^\mathcal{I} \mid \langle x, y\rangle \in S^\mathcal{I}\} \leq n\}$*
where $\mathsf{card}\{S\}$ *is denoted for the cardinality of a set S.*
$*$ $C \sqsubseteq D$ is called a general concept inclusion (GCI) where C, D are $\mathcal{SHOIQ}$-concepts (possibly complex), and a finite set of GCIs is called a terminology $\mathcal{T}$.
$*$ An interpretation $\mathcal{I}$ satisfies a GCI $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$ and $\mathcal{I}$ satisfies a terminology $\mathcal{T}$ if $\mathcal{I}$ satisfies each GCI in $\mathcal{T}$. Such an interpretation is called a model *of $\mathcal{T}$, denoted by $\mathcal{I} \models \mathcal{T}$.*

Definition 3 (knowledge base). *A pair $(\mathcal{T}, \mathcal{R})$ is called a $\mathcal{SHOIQ}$ knowledge base where $\mathcal{R}$ is a $\mathcal{SHOIQ}$ role hierarchy and $\mathcal{T}$ is a $\mathcal{SHOIQ}$ terminology. A knowledge* *base* $(\mathcal{T}, \mathcal{R})$ *is said to be consistent if there is a model* $\mathcal{I}$ *of both* $\mathcal{T}$ *and* $\mathcal{R}$*, i.e.,* $\mathcal{I} \models \mathcal{T}$ *and* $\mathcal{I} \models \mathcal{R}$*. A concept* C *is called* satisfiable *w.r.t.* $(\mathcal{T}, \mathcal{R})$ *iff there is some interpretation* $\mathcal{I}$ *such that* $\mathcal{I} \models \mathcal{R}$*,* $\mathcal{I} \models \mathcal{T}$ *and* $C^{\mathcal{I}} \neq \emptyset$*. Such an interpretation is called a* model *of* C *w.r.t.* $(\mathcal{T}, \mathcal{R})$*. A concept* D subsumes *a concept* C *w.r.t.* $(\mathcal{T}, \mathcal{R})$*, denoted by* $C \sqsubseteq D$*, if* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ *holds in each model* $\mathcal{I}$ *of* $(\mathcal{T}, \mathcal{R})$*.*

Thanks to the reductions between unsatisfiability, subsumption of concepts and knowledge base consistency, it suffices to study knowledge base consistency.

For the ease of construction, we assume all concepts to be in *negation normal form* (NNF), i.e., negation occurs only in front of concept names. Any $\mathcal{SHOIQ}$-concept can be transformed to an equivalent one in NNF by using DeMorgan's laws and some equivalences as presented in [3]. For a concept C, we denote the nnf of C by $\mathsf{nnf}(C)$ and the nnf of $\neg C$ by $\dot{\neg} C$. Let D be an $\mathcal{SHOIQ}$-concept in NNF. We define $\mathsf{cl}(D)$ to be the smallest set that contains all sub-concepts of D including D. For a knowledge base $(\mathcal{T}, \mathcal{R})$, we can define a set $\mathsf{cl}(\mathcal{T}, \mathcal{R})$. For the sake of brevity, we refer the reader to [2] for a more complete definition.

To prove soundness and completeness of our algorithms, we need a tableau structure that represents a model of a $\mathcal{SHOIQ}$ knowledge base. Regarding the definition of tableaux for $\mathcal{SHOIQ}$ presented in [2], we add a new property that imposes an exact number of S-neighbour individuals t of s if $(\leq nS.C) \in \mathcal{L}(s)$. This property makes explicit non-determinism implied from the semantics of $(\leq nS.C)$ and requires an extra expansion rule for the tableau-based algorithm.

3 A 2EXPSPACE Decision Procedure for $\mathcal{SHOIQ}$

In this section, we introduce a structure, called $\mathcal{SHOIQ}$-forest. We will show that such a forest is sufficient to represent a model of a $\mathcal{SHOIQ}$-knowledge base.

Definition 4 (tree). *Let* $(\mathcal{T}, \mathcal{R})$ *be a* $\mathcal{SHOIQ}$ *knowledge base. For each* $o \in \mathbf{C}_o$*, a* $\mathcal{SHOIQ}$*-tree for* $(\mathcal{T}, \mathcal{R})$*, denoted by* $\mathbf{T}_o = (V_o, E_o, \mathcal{L}_o, \widehat{x}_o, \neq_o)$*, is defined as follows:*

$*$ V_o *is a set of nodes containing a root node* $\widehat{x}_o \in V_o$*. Each node* $x \in V_o$ *is labelled with a function* $\mathcal{L}_o$ *such that* $\mathcal{L}_o(x) \subseteq \mathsf{cl}(\mathcal{T}, \mathcal{R})$ *and* $o \in \mathcal{L}_o(\widehat{x}_o)$*. A node* $x \in V_o$ *is called* nominal *if* $o' \in \mathcal{L}_o(x)$ *for some* $o' \in \mathbf{C}_o$*. In addition, the inequality relation* $\neq_o$ *is a symmetric binary relation over* V_o*.*

$*$ E_o *is a set of edges. Each edge* $\langle x, y\rangle \in E_o$ *is labelled with a function* $\mathcal{L}_o$ *such that* $\mathcal{L}_o(\langle x, y\rangle) \subseteq \mathbf{R}_{(\mathcal{T},\mathcal{R})}$*. If* $\langle x, y\rangle \in E_o$ *then* y *is called a successor of* x*, denoted by* $y \in \mathsf{succ}^1(x)$*, or* x *is called the predecessor of* y*, denoted by* $x = \mathsf{pred}^1(y)$*. In this case, we say that* x *is a neighbour of* y *or* y *is a neighbour of* x*. If* $z \in \mathsf{succ}^n(x)$ *(resp.* $z = \mathsf{pred}^n(x)$*) and* y *is a successor of* z *(resp.* y *is the predecessor of* z*) then* $y \in \mathsf{succ}^{(n+1)}(x)$ *(resp.* $y = \mathsf{pred}^{(n+1)}(x)$*) for all* $n \geq 0$ *where* $\mathsf{succ}^0(x) = \{x\}$ *and* $\mathsf{pred}^0(x) = x$*. A node* y *is called a* descendant *of* x *if* $y \in \mathsf{succ}^n(x)$ *for some* $n > 0$*. A node* y *is called an* ancestor *of* x *if* $y = \mathsf{pred}^n(x)$ *for some* $n > 0$*. To ensure that* $\mathbf{T}_o$ *is a tree, it is required that (i)* x *is a descendant of* $\widehat{x}_o$ *for all* $x \in V_o$ *with* $x \neq \widehat{x}_o$*, and (ii) each node* $x \in V_o$ *with* $x \neq \widehat{x}_o$ *has a unique predecessor. A node* y *is called an* R*-successor of* x*, denoted by* $y \in \mathsf{succ}^1_R(x)$ *(resp.* y *is called the* R*-predecessor* *of x, denoted by $y = \mathsf{pred}^1_R(x)$) if there is some role R' such that $R' \in \mathcal{L}_o(\langle x,y\rangle)$ (resp. $R' \in \mathcal{L}_o(\langle y,x\rangle)$) and $R' \underline{\circledast} R$. A node y is called a R-neighbour of x if y is either a R-successor or R-predecessor of x. If z is an R-successor of y (resp. z is the R-predecessor of y) and $y \in \mathsf{succ}^n_R(x)$ (resp. $y = \mathsf{pred}^n_R(x)$) then $z \in \mathsf{succ}^{(n+1)}_R(x)$ (resp. $z = \mathsf{pred}^{(n+1)}_R(x)$) for $n \geq 0$ with $\mathsf{succ}^0_R(x) = \{x\}$ and $x = \mathsf{pred}^0_R(x)$.*

$*$ *For a node x, a role S and $o \in \mathbf{C}_o$, we define the set $S^{\mathbf{T}_o}(x,C)$ of x's S-neighbours as follows: $S^{\mathbf{T}_o}(x,C) = \{y \in V_o \mid y \text{ is a } S\text{-neighbour of } x \text{ and } C \in \mathcal{L}_o(x)\}$.*

$*$ *A node x is called* iterated *by y w.r.t. a node x_o if x has no nominal ancestor except for $\widehat{x}_o$ and there are integers $n,m > 0$ and nodes x',y' such that : (i) $x_o = \mathsf{pred}^n(y)$, $y = \mathsf{pred}^m(x)$, (ii) $x' = \mathsf{pred}^1(x)$, $y' = \mathsf{pred}^1(y)$, (iii) $\mathcal{L}_o(x) = \mathcal{L}_o(y)$, $\mathcal{L}_o(x') = \mathcal{L}_o(y')$, (iv) $\mathcal{L}_o(\langle x',x\rangle) = \mathcal{L}_o(\langle y',y\rangle)$, and (v) if there are z,z' and $i > 0$ such that $z' = \mathsf{pred}^1(z)$, $\mathsf{pred}^i(z') = x_o$, $\mathcal{L}_o(z) = \mathcal{L}_o(y)$, $\mathcal{L}_o(z') = \mathcal{L}_o(y')$ and $\mathcal{L}_o(\langle z',z\rangle) = \mathcal{L}_o(\langle y',y\rangle)$ then $i \geq n$.*
A node x is called 1-iterated *by y if x is iterated by y w.r.t. $\widehat{x}_o$. A node x is called* blocked *by y, denoted by $y = \mathsf{b}(x)$, if x is iterated by y w.r.t. a 1-iterated node x_o.*

$*$ *In the following, we often use $\mathcal{L}(x)$, $\mathcal{L}(\langle x,y\rangle)$, $S^{\mathbf{T}}(x,C)$ and $\neq$ instead of $\mathcal{L}_o(x)$, $\mathcal{L}_o(\langle x,y\rangle)$, $S^{\mathbf{T}_o}(x,C)$ and $\neq_o$, respectively. This does not cause any confusion since $V_o \cap V_{o'} = \emptyset$ and $E_o \cap E_{o'} = \emptyset$ if $o \neq o'$. In addition, $x \neq_o y$ is never defined for $x \in V_o$ and $y \in V_{o'}$ with $o \neq o'$.*

We remark that the definition of 1-iterated nodes in Definition 4 for $\mathcal{SHOIQ}$-trees is very similar to the standard definition of blocked nodes for $\mathcal{SHIQ}$ completion trees (see [3]). Moreover, if we consider the sub-tree rooted at a 1-iterated node as a $\mathcal{SHIQ}$ completion tree then blocked nodes according to Definition 4 are also blocked nodes according to the standard definition for this $\mathcal{SHIQ}$ completion tree.

A $\mathcal{SHOIQ}$-tree consists of two layers : the first layer is formed of nodes from the root to 1-iterated nodes or nominal nodes, and the second layer consists of nodes from each 1-iterated node to blocked or nominal nodes. In addition, each node x in the layer 2 has a unique 1-iterated node, denoted $\widehat{\mathsf{b}}(x)$, such that $\widehat{\mathsf{b}}(x)$ is an ancestor of x.

Definition 5 (forest). *Let $(\mathcal{T},\mathcal{R})$ be a $\mathcal{SHOIQ}$ knowledge base. A $\mathcal{SHOIQ}$-forest for $(\mathcal{T},\mathcal{R})$ is a pair $\mathbf{G} = \langle \mathbf{T},\varphi\rangle$, where $\mathbf{T} = \{\mathbf{T}_o \mid o \in \mathbf{C}_o\}$ is a set of $\mathcal{SHOIQ}$-trees for $(\mathcal{T},\mathcal{R})$ with $\mathbf{T}_o = (V_o, E_o, \mathcal{L}_o, \widehat{x}_o, \neq_o)$, and φ is a* partitioning *function $\varphi : \mathcal{V} \to 2^{\mathcal{V}}$ with $\mathcal{V} = \bigcup_{o \in \mathbf{C}_o} V_o$. We denote $\mathcal{L}'(\langle x,y\rangle) = \mathcal{L}_o(\langle x,y\rangle)$ if $\langle x,y\rangle \in E_o$, and $\mathcal{L}'_o(\langle x,y\rangle) = \{S^{\ominus} \mid S \in \mathcal{L}_o(\langle y,x\rangle)\}$ if $\langle y,x\rangle \in E_o$ for some $o \in \mathbf{C}_o$. The partitioning function φ satisfies the following conditions:*

1. *For each $x \in \mathcal{V}$, $\varphi(x)$ is the* partition *of x with $x \in \varphi(x)$. There are $x_0, \cdots, x_n \in \mathcal{V}$ such that $\varphi(x_i) \cap \varphi(x_j) = \emptyset$ with $0 \leq i < j \leq n$ and $\bigcup_{0 \leq i \leq n} \varphi(x_i) = \mathcal{V}$;*
2. *For all $x,x' \in \mathcal{V}$, if $x' \in \varphi(x)$ then $\varphi(x) = \varphi(x')$ and $\mathcal{L}(x) = \mathcal{L}(x')$. We denote $\Lambda(\varphi(x)) = \mathcal{L}(x)$. In addition, an inequality relation over partitions can be described as follows : for $x,x' \in \mathcal{V}$ we define $\varphi(x) \neq \varphi(x')$ if there are two nodes $y \in \varphi(x)$ and $y' \in \varphi(x')$ such that $y \neq_o y'$ for some $o \in \mathbf{C}_o$;*
3. *For all $\varphi(x)$ and $\varphi(x')$, if there are two edges $\langle y,y'\rangle \in E_o$ and $\langle w,w'\rangle \in E_{o'}$ with $o,o' \in \mathbf{C}_o$ such that $y,w \in \varphi(x)$, $y',w' \in \varphi(x')$, $\mathcal{L}'(\langle y,y'\rangle) \neq \emptyset$, $\mathcal{L}'(\langle w,w'\rangle) \neq \emptyset$ then $\mathcal{L}'(\langle y,y'\rangle) = \mathcal{L}'(\langle w,w'\rangle)$.*

We define a function $\Lambda(\langle\cdot,\cdot\rangle)$ for labelling edges ended by two partitions as follows: $\Lambda(\langle\varphi(x),\varphi(x')\rangle)=\mathcal{L}'(\langle z,z'\rangle)$ where $z\in\varphi(x)$, $z'\in\varphi(x')$, $\mathcal{L}'(\langle z,z'\rangle)\neq\emptyset$, and $\{\langle z,z'\rangle,\langle z',z\rangle\}\cap E_{o'}\neq\emptyset$ for some $o'\in\mathbf{C}_o$. We say $\varphi(x')$ is a S-neighbour partition of $\varphi(x)$ if $S\in\Lambda(\langle\varphi(x),\varphi(x')\rangle)$.

4. *For all $x,x'\in\mathcal{V}$, if $o\in\mathcal{L}(x)\cap\mathcal{L}(x')$ for some $o\in\mathbf{C}_o$ and $\varphi(x)\neq\varphi(x')$ does not hold then $\varphi(x)=\varphi(x')$;*
5. *If $(\leq nR.C)\in\Lambda(\varphi(x))$ for some $x\in\mathcal{V}$ and there exist $(n{+}1)$ nodes $x_0,\cdots,x_n\in\mathcal{V}$ such that (i) $\varphi(x_i)\cap\varphi(x_j)=\emptyset$ for all $0\leq i<j\leq n$, and (ii) $C\in\Lambda(\varphi(x_i))$, $R\in\Lambda(\langle\varphi(x),\varphi(x_i)\rangle)$ for all $i\in\{0,\cdots,n\}$, then $\varphi(x_l)\neq\varphi(x_m)$ for all $0\leq l<m\leq n$; and*
6. *If $(\geq nR.C)\in\Lambda(\varphi(x))$ for some $x\in\mathcal{V}$ then $\varphi(x)$ has n R-neighbour partitions $\varphi(x_1),\cdots,\varphi(x_n)$ such that $\varphi(x_i)\cap\varphi(x_j)=\emptyset$ and $C\in\Lambda(\varphi(x_i))$ for all $1\leq i<j\leq n$.*

$*$ ***Clashes:*** $\mathbf{T}$ *is said to contain a* clash *if one of the following conditions holds:*

1. *There is some node $x\in\mathcal{V}$ such that $\{A,\dot{\neg}A\}\subseteq\Lambda(\varphi(x))$ for some concept name $A\in\mathbf{C}$;*
2. *There are nodes $x,y\in\mathcal{V}$ such that $\varphi(x)\neq\varphi(y)$ and $o\in\Lambda(\varphi(x))\cap\Lambda(\varphi(y))$ for some $o\in\mathbf{C}_o$;*
3. *There is a node $x\in\mathcal{V}$ with $(\leq nR.C)\in\Lambda(\varphi(x))$ and there are $(n+1)$ nodes $x_0,\cdots,x_n\in\mathcal{V}$ such that $\varphi(x_i)\cap\varphi(x_j)=\emptyset$, $\varphi(x_i)\neq\varphi(x_j)$ with $0\leq i<j\leq n$, and $C\in\Lambda(\varphi(x_i))$, $R\in\Lambda(\langle\varphi(x),\varphi(x_i)\rangle)$ for $i\in\{0,\cdots,n\}$.*

We now describe the tableau-based algorithm whose goal is to construct from a knowledge base $(\mathcal{T},\mathcal{R})$ a $\mathcal{SHOIQ}$-forest $\mathbf{G}=\langle\mathbf{T},\varphi\rangle$. To do this, the algorithm applies expansion rules (as described in Fig. 1 and Fig. 2 in [5]), and terminates when none of the rules is applicable. The obtained $\mathbf{G}$ is called *complete*, and if $\mathbf{G}$ contains no clash then $\mathbf{G}$ is called *clash-free*. In this case, we also say $\mathbf{T}_o$ is complete and clash-free for all $\mathbf{T}_o\in\mathbf{T}$.

The expansion rules maintain the tree-like structure of $\mathcal{SHOIQ}$-forest and they are similar to those in [2] except that if a concept C is added to the label of a node x due to application of these rules then C is propagated to the label of each node $y\in\varphi(x)$. Moreover, all rules in Fig. 1 in [5] except for $\exists$- and $\geq$-rule update only the label of nodes or edges and do not change the partitioning function φ. In particular, when the $\leq$-rule is applied to a node x with two S-neighbours y,z of x, it must propagate the label of $\langle x,y\rangle$ to that of all $\langle x',z'\rangle$ (or $\langle z',x'\rangle$) where $x'\in\varphi(x)$ and $z'\in\varphi(z)$, and set the label of $\langle x,y\rangle$ to empty set. This may change φ only if $\varphi(y)$ is singleton. By a new rule, namely the $\bowtie$-rule, each node x containing a term $(\leq nS.C)$ has exactly m S-neighbours containing C with some $m\leq n$. As a result, this rule and $\geq$-rule ensure that if there are two nodes $y,y'\in\varphi(x)$ then y and y' have exactly m S-neighbours which contain C in their label. Finally, we can avoid infinite sequences of "merging-and-generating" without pruning nodes since all merges due to number restrictions or nominals are performed by updating the partitioning function. The following lemma establishes correctness and completeness of the algorithm.

Lemma 1. *Let $(\mathcal{T},\mathcal{R})$ be a $\mathcal{SHOIQ}$ knowledge base.*

1. *The tableau algorithm terminates and builds a $\mathcal{SHOIQ}$-forest whose the size is bounded by a doubly exponential function in the size of $(\mathcal{T},\mathcal{R})$.*
2. *If the tableau algorithm yields a clash-free and complete $\mathcal{SHOIQ}$-forest for $(\mathcal{T},\mathcal{R})$ then there is a tableau for $(\mathcal{T},\mathcal{R})$.*
3. *If there is a tableau for $(\mathcal{T},\mathcal{R})$ then the tableau algorithm yields a clash-free and complete $\mathcal{SHOIQ}$-forest for $(\mathcal{T},\mathcal{R})$.*

To prove soundness of the tableau algorithm, we can devise a model from a clash-free and complete $\mathcal{SHOIQ}$-forest by considering a partition as an individual and unravelling blocked nodes since we can show that each blocking node $\mathsf{b}(x)$ has no "core path" from $\mathsf{b}(x)$ to each nominal descendant y, i.e., there do not exist terms $(\leq m_iR_i.C_i) \in \mathsf{pred}^i(y)$, roles $R_i \in \mathcal{L}(\langle \mathsf{pred}^{i-1}(y), \mathsf{pred}^i(y)\rangle)$ and concepts $C_i \in \mathcal{L}(\mathsf{pred}^{i+1}(y))$ for $k < i \leq 0$, $\mathsf{b}(x) = \mathsf{pred}^k(y)$. The following theorem is a consequence of Lemma 1.

Theorem 1. *Let $(\mathcal{T},\mathcal{R})$ be a $\mathcal{SHOIQ}$ knowledge base. The tableau algorithm is a decision procedure for consistency of $(\mathcal{T},\mathcal{R})$ and it runs in* 2NEXPTIME *in the size of $(\mathcal{T},\mathcal{R})$.*

4 An EXPSPACE Tableau-based Algorithm for $\mathcal{SHOIQ}$

This section starts by translating some results presented in [7] for $\mathcal{C}^2$ into those for $\mathcal{SHOIQ}$.

Definition 6 (star-type). *Let $(\mathcal{T},\mathcal{R})$ be a $\mathcal{SHOIQ}$ knowledge base. A* star-type *is a triplet $\sigma = \langle \lambda_\sigma, \bar{\nu}_\sigma, \bar{\mu}_\sigma\rangle$, where $\lambda_\sigma \in 2^{\mathsf{cl}(\mathcal{T},\mathcal{R})}$, $\bar{\mu}_\sigma = (\langle r_1, l_1\rangle, \cdots, \langle r_d, l_d\rangle)$ is a d-tuple over $2^{\mathbf{R}_{(\mathcal{T},\mathcal{R})}} \times 2^{\mathsf{cl}(\mathcal{T},\mathcal{R})}$, and $\bar{\nu}_\sigma = (\langle r', l'\rangle)$ with $\langle r', l'\rangle \in 2^{\mathbf{R}_{(\mathcal{T},\mathcal{R})}} \times 2^{\mathsf{cl}(\mathcal{T},\mathcal{R})}$. A pair $\langle r, l\rangle$ is a* ray *of σ if $\langle r, l\rangle$ is a component of $\bar{\mu}_\sigma$ or $\bar{\nu}_\sigma$. In particular, $\langle r, l\rangle$ is a* predecessor ray *if $(\langle r, l\rangle) = \bar{\nu}_\sigma$, and $\langle r, l\rangle$ is a* successor ray *if $\langle r, l\rangle$ is a component of $\bar{\mu}_\sigma$. We denote $\bar{\xi}_\sigma = (\langle r_1, l_1\rangle, \cdots, \langle r_d, l_d\rangle, \langle r_{d+1}, l_{d+1}\rangle)$ if $\bar{\nu}_\sigma = (\langle r', l'\rangle)$ where $r' = r_{d+1}$, $l' = l_{d+1}$, and $\bar{\xi}_\sigma = \bar{\mu}_\sigma$ if $\bar{\nu}_\sigma$ is empty.*

- *A ray $\langle r', l'\rangle$ of σ is* primary *w.r.t. a term $(\leq mR.C)$ if $(\leq mR.C) \in \lambda_\sigma$, $R \in r'$ and $C \in l'$. For a term $(\leq mR.C) \in \lambda_\sigma$, we denote $\mathcal{C}^\sigma_{\langle \leq mR.C\rangle}$ for the set of all rays $\langle r', l'\rangle$ of σ such that $R \in r'$, $C \in l'$.*
- *A star-type σ is* nominal *if $o \in \lambda_\sigma$ for some $o \in \mathbf{C}_o$.*
- *A star-type σ is* chromatic *if there is a term $(\geq nS.D) \in \lambda_\sigma$ and σ has n rays $\langle r'_1, l'_1\rangle, \cdots, \langle r'_n, l'_n\rangle$ such that $S \in l'_i$, $D \in l'_i$ for all $1 \leq i \leq n$, and $l'_i \neq l'_j$ for all $0 \leq i < j \leq n$ with $l'_0 = \lambda_\sigma$.*
- *A star-type σ is* homomorphic *(resp.* isomorphic*) to a star-type σ' if $\lambda_\sigma = \lambda_{\sigma'}$, and for each term $(\leq mR.C) \in \lambda_\sigma$, there is an injection (resp. a bijection) $\pi : \mathcal{C}^\sigma_{\langle \leq mR.C\rangle} \rightarrow \mathcal{C}^{\sigma'}_{\langle \leq mR.C\rangle}$ such that $\pi(\langle r, l\rangle) = \langle r', l'\rangle$ implies $r' = r$ and $l' = l$.*
- *Two star-types σ, σ' are* equivalent *if $\lambda_\sigma = \lambda_{\sigma'}$, and there is a bijection π between $\bar{\xi}_\sigma$ and $\bar{\xi}_{\sigma'}$ such that $\pi(\langle r, l\rangle) = \langle r', l'\rangle$ implies $r' = r$ and $l' = l$.*

We denote Σ for the set of all star-types for $(\mathcal{T},\mathcal{R})$. ◁

In the context of a $\mathcal{SHOIQ}$-forest, we can think of a star-type σ as the set of nodes x such that $\mathcal{L}(x) = \lambda_\sigma$, and each ray $\langle r_i, l_i \rangle$ of σ corresponds to a neighbour x_i of x such that $\mathcal{L}'(\langle x, x_i \rangle) = r_i$ and $\mathcal{L}(x_i) = l_i$. In this case, we say that x *satisfies* σ.

Remark 1. The notion of chromaticity introduced in Definition 6 implies an inequality relation $\neq$ over nodes. That stronger notion is needed to prevent "distinct" star-types from including nodes x, y which are neighbours or $x \neq y$. In order to make star-types chromatic, it is necessary to add to knowledge bases some new concepts and axioms as follows. Let $(\mathcal{T}, \mathcal{R})$ be a $\mathcal{SHOIQ}$ knowledge base. For each term $(\geq nS.D) \in \mathsf{cl}(\mathcal{T}, \mathcal{R})$, we add to $\mathsf{cl}(\mathcal{T}, \mathcal{R})$ n new concept names $C^0_{(\geq nS.D)}, \cdots, C^n_{(\geq nS.D)}$, and to $\mathcal{T}$ the following axioms: $C^i_{(\geq nS.D)} \sqcap C^j_{(\geq nS.D)} \sqsubseteq \perp$ for all $0 \leq i < j \leq n$. It is straightforward to prove that the terminology $(\mathcal{T}', \mathcal{R})$ is consistent iff $(\mathcal{T}, \mathcal{R})$ is consistent where $\mathcal{T}'$ is obtained from $\mathcal{T}$ by adding these new axioms. Thanks to these new concepts and axioms, the following definition points out how to build chromatic star-types.

Definition 7 (valid star-type). *Let $(\mathcal{T}, \mathcal{R})$ be a $\mathcal{SHOIQ}$ knowledge base. Let σ be a star-type for $(\mathcal{T}, \mathcal{R})$ where $\sigma = \langle \lambda_\sigma, \bar{\nu}, \bar{\mu} \rangle$ with $\bar{\mu} = (\langle r_1, l_1 \rangle, \cdots, \langle r_d, l_d \rangle)$ and $\lambda_\sigma = l_0$, $\bar{\nu} = \{\langle r_{d+1}, l_{d+1} \rangle\}$. σ is* valid *with an inequality relation $\neq$ over $\mathcal{C}^\sigma$ if the following conditions are satisfied:*

1. *If $C \sqsubseteq D \in \mathcal{T}$ then $\mathsf{nnf}(\neg C \sqcup D) \in l_i$ for all $0 \leq i \leq d+1$;*
2. *$\{A, \neg A\} \not\subseteq l_i$ for every concept name A with $0 \leq i \leq d+1$;*
3. *If $C_1 \sqcap C_2 \in l_i$ then $\{C_1, C_2\} \subseteq l_i$ for all $0 \leq i \leq d+1$;*
4. *If $C_1 \sqcup C_2 \in l_i$ then $\{C_1, C_2\} \cap l_i \neq \emptyset$ for all $0 \leq i \leq d+1$;*
5. *If $\exists R.C \in \lambda_\sigma$ then there is some $1 \leq i \leq d+1$ such that $C \in l_i$ and $R \in r_i$;*
6. *If $(\leq nS.C) \in \lambda_\sigma$ and there is some $1 \leq i \leq d+1$ such that $S \in r_i$ then $C \in l_i$ or $\dot{\neg} C \in l_i$;*
7. *If $(\leq nS.C) \in \lambda_\sigma$ and there is some $1 \leq i \leq d+1$ such that $C \in l_i$ and $S \in r_i$ then there is some $1 \leq m \leq n$ such that $\{(\leq mS.C), (\geq mS.C)\} \subseteq \lambda$;*
8. *For each $1 \leq i \leq d+1$, if $R \in r_i$ and $R \underline{\circledast} S$ then $S \in r_i$;*
9. *If $\forall R.C \in \lambda_\sigma$ and $R \in r_i$ for some $1 \leq i \leq d+1$ then $C \in l_i$;*
10. *If $\forall R.D \in \lambda_\sigma$, $S \underline{\circledast} R$, $\mathsf{Trans}(S)$ and $R \in r_i$ for some $1 \leq i \leq d+1$ then $\forall S.D \in l_i$;*
11. *If $(\geq nS.C) \in \lambda_\sigma$ then $C^0_{(\geq nS.C)} \in \lambda_\sigma$ and there are $1 \leq i_1 < \cdots < i_n \leq d+1$ such that $\{C, C^j_{(\geq nS.C)}\} \subseteq l_{i_j}$, $S \in r_{i_j}$ for all $1 \leq j \leq n$.*
12. *If $(\leq nS.C) \in \lambda_\sigma$ and there are no $1 \leq i_1 < \cdots < i_{n+1} \leq d+1$ such that $C \in l_{i_j}$ and $S \in r_{i_j}$ for all $1 \leq j \leq n+1$.* $\lhd$

Notice that a valid star-type according to Definition 7 is chromatic. If we think of a star-type σ as a node x satisfying σ in a $\mathcal{SHOIQ}$-forest then σ is valid if no expansion rule is applicable to x. Moreover, due to the conditions 7, 11 and 12 in Definition 7, if there is a term $(\leq nS.D) \in \lambda_\sigma$ for a valid star-type σ then σ has exactly n primary rays $\langle r_i, l_i \rangle, \cdots, \langle r_n, l_n \rangle$ w.r.t. $(\leq nS.D)$.

Definition 8 (frame). *Let $(\mathcal{T}, \mathcal{R})$ be a $\mathcal{SHOIQ}$ knowledge base. A* frame *for $(\mathcal{T}, \mathcal{R})$ is a tuple $\mathcal{F} = \langle (\mathcal{N}_0, \cdots, \mathcal{N}_H), \delta, \Phi, \widehat{\delta} \rangle$, where $H \in \mathbb{N}$ is the* dimension *of $\mathcal{F}$; $\mathcal{N}_i \subseteq \Sigma$ for* *all* $0 \leq i \leq H$, *and all star-types in* $\mathcal{N}_0$ *are nominal. We denote* $\mathfrak{N} = \bigcup_{i \in \{1, \cdots, H\}} \mathcal{N}_i$*;* δ *is a function* $\delta : \mathfrak{N} \rightarrow \mathbb{N}$*;* Φ *is a function* $\Phi : \bar{\mathfrak{N}} \rightarrow \Sigma$ *where* $\bar{\mathfrak{N}}$ *is denoted for the set of all star-types* $\sigma \in \mathfrak{N}$ *such that one of the three condition holds : (i)* σ *is nominal; (ii)* σ *has a ray* $\langle r, l \rangle$ *such that there is a term* $(\leq mR.C)$ *with* $(\leq mR.C) \in \lambda_\sigma$*,* $C \in l$ *and* $R \in r$*; (iii)*σ *has a ray* $\langle r, l \rangle$ *such that there is a term* $(\leq mR.C)$ *with* $(\leq mR.C) \in l$*,* $C \in \lambda_\sigma$ *and* $R \in r$*;* $\widehat{\delta}$ *is a function* $\widehat{\delta} : \Phi(\bar{\mathfrak{N}}) \rightarrow \mathbb{N}$ *which is defined as follows:*

> *For two star-types* $\sigma, \sigma' \in \bar{\mathfrak{N}}$*,* $\Phi(\sigma) = \Phi(\sigma')$ *iff either* σ *is isomorphic to* σ'*, or there is a star-type* $\omega \in \mathfrak{N} \setminus \mathcal{N}_H$ *such that* σ *and* σ' *are homomorphic to* ω*.*

Additionally, a star-type $\sigma \in \mathcal{N}_k$ *(*$0 < k < H$*) is* linkable *with a star-type* $\sigma' \in \mathcal{N}_{k-1}$ *by a ray* $\langle r, l \rangle$ *of* σ *if* σ' *has a ray* $\langle r', l' \rangle$ *such that* $\lambda_{\sigma'} = l$*,* $r' = r^-$ *and* $l' = \lambda_\sigma$ *where* $r^- = \{R^\ominus \mid R \in r\}$*.*

Remark 2. For $\sigma, \sigma' \notin \mathcal{N}_H$ and σ, σ' are valid, if σ is homomorphic to σ' then σ is isomorphic to σ'. In fact, if there is a term $(\leq mR.C) \in \lambda_\sigma$ then both σ, σ' have exactly m primary rays w.r.t. $(\leq mR.C)$. If there is a homomorphism between the two sets of primary rays w.r.t. $(\leq mR.C)$ then it is an isomorphism as well.

The frame structure, as introduced in Definition 8, allows us to tile a forest structure by star-types. Such a structure is crucial to obtain termination when designing a tableau-based algorithm. An important difference between a frame and a $\mathcal{SHOIQ}$-forest is that a frame does not represent nodes corresponding to individuals but stores the number of individuals satisfying a star-type. The function $\delta(\sigma)$ is used for this purpose. In the context of a $\mathcal{SHOIQ}$-forest, we can think of $\Phi(\sigma)$ as a star-type which is satisfied by nodes forming a set of partitions. In fact, the function Φ maps star-types forming a $\mathcal{SHOIQ}$-forest into another set of star-types that regroups non-neighbour partitions.

Notice that the function Φ introduced in Definition 8 does not transfer the relation of linkability from $\bar{\mathfrak{N}}$ to $\Phi(\bar{\mathfrak{N}})$, and that chromaticity of star-types prevents chromatically linkable star-types from collapsing into a unique star-type by the function Φ. The function $\widehat{\delta}(\Phi(\sigma))$ counts the number of partitions that are mapped to a star-type by Φ. Such a function can be defined if all star-types "covering" a partition must be mapped to a unique star-type by Φ. This is a consequence of the Φ's definition.

Definition 9 (valid frame). *Let* $(\mathcal{T}, \mathcal{R})$ *be a* $\mathcal{SHOIQ}$ *knowledge base. A frame* $\mathcal{F} = \langle (\mathcal{N}_0, \cdots, \mathcal{N}_H), \delta, \Phi, \widehat{\delta} \rangle$ *is* valid *if the following conditions are satisfied:*

1. *For each* $\sigma \in \mathfrak{N}$*, if* $\delta(\sigma) \geq 1$ *then* σ *is valid;*
2. *For each* $\sigma \in \bar{\mathfrak{N}}$*, if* $\delta(\sigma) \geq 1$ *then* $\widehat{\delta}(\Phi(\sigma)) \geq 1$*;*
3. *For each* $o \in \mathbf{C}_o$ *there is a unique* $\sigma_o \in \mathcal{N}_0$ *such that* $o \in \lambda_{\sigma_o}$ *and* $\delta(\sigma_o) = 1$*;*
4. *For each* $\sigma \in \mathfrak{N}$*, if* $o \in \lambda_\sigma$ *with some* $o \in \mathbf{C}_o$ *then* $\Phi(\sigma) = \Phi(\sigma_o)$ *and* $\widehat{\delta}(\Phi(\sigma_o)) = 1$ *with* $\sigma_o \in \mathcal{N}_0$ *such that* $o \in \lambda_{\sigma_o}$ *and* $\delta(\sigma_o) = 1$*;*
5. *For each* $0 \leq k < H$ *and* $\langle \lambda, r, \lambda' \rangle \in 2^{\mathsf{cl}(\mathcal{T}, \mathcal{R})} \times 2^{\mathbf{R}_{(\mathcal{T}, \mathcal{R})}} \times 2^{\mathsf{cl}(\mathcal{T}, \mathcal{R})}$ *with* $r^- = \{R^\ominus \mid R \in r\}$*,*

$$\sum_{\sigma \in \mathcal{N}_k} \delta(\sigma) |\bar{\mu}_\sigma|_{\langle \lambda, r, \lambda' \rangle} = \sum_{\sigma' \in \mathcal{N}_{k+1}} \delta(\sigma') |\bar{\nu}_{\sigma'}|_{\langle \lambda', r^-, \lambda \rangle}$$

where $|\bar{\nu}_\sigma|_{\langle\lambda,r,\lambda'\rangle}$ and $|\bar{\mu}_\sigma|_{\langle\lambda,r,\lambda'\rangle}$ are denoted for the number of components $\langle r',l'\rangle$ of respective $\bar{\nu}_\sigma$ and $\bar{\mu}_\sigma$ such that $\lambda_\sigma = \lambda$, $r' = r$ and $l' = \lambda'$;

6. *For each $\langle\lambda, r, \lambda'\rangle \in 2^{\mathsf{cl}(\mathcal{T},\mathcal{R})} \times 2^{\mathbf{R}_{(\mathcal{T},\mathcal{R})}} \times 2^{\mathsf{cl}(\mathcal{T},\mathcal{R})}$ with $r^- = \{R^\ominus \mid R \in r\}$,*

$$\sum_{\sigma\in\bar{\mathfrak{N}}} \widehat{\delta}(\Phi(\sigma))\big|\bar{\xi}_{\Phi(\sigma)}\big|_{\langle\lambda,r,\lambda'\rangle} = \sum_{\sigma'\in\bar{\mathfrak{N}}} \widehat{\delta}(\Phi(\sigma'))\big|\bar{\xi}_{\Phi(\sigma')}\big|_{\langle\lambda',r^-,\lambda\rangle}$$

where $\big|\bar{\xi}_{\Phi(\sigma)}\big|_{\langle\lambda,r,\lambda'\rangle} = \big|\bar{\nu}_{\Phi(\sigma)}\big|_{\langle\lambda,r,\lambda'\rangle} + \big|\bar{\mu}_{\Phi(\sigma)}\big|_{\langle\lambda,r,\lambda'\rangle}$ $\lhd$

Remark 3. It is not required that star-types $\Phi(\sigma)$ are valid. We will use function Φ to trim rays $\langle r,l\rangle$ of star-types such that (i) $\langle r,l\rangle$ or $\langle r^-,l\rangle$ is not a primary ray of every star-type. The images of star-types σ by Φ, i.e. trimmed star-types $\Phi(\sigma)$, are employed to represent partitions obtained from merge processes. As described in Section 3, in order to govern partitions, it suffices to deal with nodes x containing a term $(\leq mR.C)$ and the R-neighbours of x containing C. For this reason, the function Φ maps only star-types in $\bar{\mathfrak{N}}$ by trimming non-primary rays.

The notion of validity for a frame is crucial to establish a connection with the tableau-based algorithm presented in Section 3, i.e., how to build a $\mathcal{SHOIQ}$-forest from a valid frame, and inversely. The condition 1 in Definition 9 requires that every star-type satisfied by at least one node must be valid. The condition 2 implies that each valid star-type including a primary ray will be mapped by Φ. The condition 3 ensures that each nominal is counted exactly once while the condition 4 imposes that all nominal star-types containing some $o \in \mathbf{C}_o$ are mapped into a unique star-type by Φ. In the context of a $\mathcal{SHOIQ}$-forest, these conditions imply that for each nominal $o \in \mathbf{C}_o$ there is exactly one tree whose root contains o and there is exactly one partition containing o. The condition 5 allows for linking star-types at level k with star-types at level $k-1$ and $k+1$. It ensures that each node x satisfying (or counted for) a star-type σ at level k is linked by its rays to neighbours satisfying star-types at level $k-1$ and $k+1$. The number of these neighbours corresponds exactly to the number of σ's rays. Finally, the condition 6 deals with partitions. In the context of a $\mathcal{SHOIQ}$-forest where $\Phi(\sigma)$ represents the image of a set of partitions, the condition 6 points out how star-types $\Phi(\sigma)$ would be interconnected.

Lemma 2. *Let $(\mathcal{T},\mathcal{R})$ be a $\mathcal{SHOIQ}$ knowledge base.*

1. *If the tableau algorithm can build a clash-free and complete $\mathcal{SHOIQ}$-forest for $(\mathcal{T},\mathcal{R})$ then there is a valid frame for $(\mathcal{T},\mathcal{R})$.*
2. *If there is a valid frame $\mathcal{F} = \langle(\mathcal{N}_0,\cdots,\mathcal{N}_H),\delta,\Phi,\widehat{\delta}\rangle$ for $(\mathcal{T},\mathcal{R})$ then the tableau algorithm can build a clash-free and complete $\mathcal{SHOIQ}$-forest for $(\mathcal{T},\mathcal{R})$.*

Lemma 2 points out the equivalence between a clash-free and complete $\mathcal{SHOIQ}$-forest and a valid frame for $(\mathcal{T},\mathcal{R})$. The following lemma affirms that there is an exponential structure, a valid frame, which can represent a $\mathcal{SHOIQ}$-forest whose size may be doubly exponential.

Lemma 3. *Let $(\mathcal{T},\mathcal{R})$ be a $\mathcal{SHOIQ}$ knowledge base. The size of a valid frame $\mathcal{F} = \langle(\mathcal{N}_0,\cdots,\mathcal{N}_H),\delta,\Phi,\widehat{\delta}\rangle$ is bounded by an exponential function in the size of $(\mathcal{T},\mathcal{R})$.*

We can sketch a proof of the lemma here. We have $H \leq K$ where $K = 2^{2m+k} \times 2$ with $m = \mathsf{card}\{\mathsf{cl}(\mathcal{T}, \mathcal{R})\}$ and $k = \mathsf{card}\{\mathbf{R}_{(\mathcal{T},\mathcal{R})}\}$. Moreover, each star-type has at most M distinct rays where $M = \sum m_i + E$, m_i occurs in a number restriction term $(\geq m_i R.C)$ appearing in $\mathcal{T}$, and E is the number of distinct terms $\exists R.C$ appearing in $\mathcal{T}$. If we denote Σ for the set of all star-types then $\mathsf{card}\{\Sigma\} \leq ((\mathsf{card}\{\mathsf{cl}(\mathcal{T}, \mathcal{R})\})^2 \times \mathsf{card}\{\mathbf{R}_{(\mathcal{T},\mathcal{R})}\})^M$. Since $\delta(\sigma)$ is bounded by $M\delta(\sigma')$ where σ', σ are respective star-types at level $k-1$ and k, it holds that $\delta(\sigma) \leq M^{2^{2m+k} \times 2}$. If $\delta(\sigma)$ is represented as a binary number then it takes an exponential number of bits.

Based on Lemma 3 and 2, we can present straightforwardly an optimal worst-case algorithm for checking the consistency of a $\mathcal{SHOIQ}$ knowledge base. However, such an algorithm cannot be used in practice since the non-determinism is not sufficiently constrained to obtain termination in feasible time. In the sequel, based on the results obtained so far, we try to design an algorithm which has more goal-directed behaviour.

Blocking Condition for a Frame Let $\mathcal{F} = \langle(\mathcal{N}_0, \cdots, \mathcal{N}_H), \delta, \Phi, \widehat{\delta}\rangle$ be a frame. A star-type $\sigma_k \in \mathcal{N}_k$ with $0 < k \leq H$ is *blocked* if there are $\sigma_i \in \mathcal{N}_i$ with $0 \leq i \leq k$ such that σ_i is linkable with σ_{i-1} for all $i \in \{1, \cdots, k\}$, then there are $0 < k_1 < k_2 < k_3 < k_4 \leq k$ such that:

1. $\lambda_{\sigma_{k_1}} = \lambda_{\sigma_{k_2}}$, $\bar{\nu}_{\sigma_{k_1}} = \bar{\nu}_{\sigma_{k_2}}$, and there is no $0 < j < k_2$ such that $j \neq k_1$, $\lambda_{\sigma_j} = \lambda_{\sigma_{k_2}}$ and $\bar{\nu}_{\sigma_j} = \bar{\nu}_{\sigma_{k_2}}$;
2. $\lambda_{\sigma_{k_3}} = \lambda_{\sigma_{k_4}}$, $\bar{\nu}_{\sigma_{k_3}} = \bar{\nu}_{\sigma_{k_4}}$, and there is no $k_2 < j < k_4$ such that $j \neq k_3$, $\lambda_{\sigma_j} = \lambda_{\sigma_{k_4}}$ and $\bar{\nu}_{\sigma_j} = \bar{\nu}_{\sigma_{k_4}}$.

Notice that this blocking condition is looser than the blocking condition introduced in Definition 5 for a $\mathcal{SHOIQ}$-forest. Since we cannot determine the path from root to a node satisfying a star-type over a frame, it is not possible to check blocking condition in the same way as for a $\mathcal{SHOIQ}$-forest. The blocking condition for a frame, as described above, implies that a node satisfying a blocked star-type must have an ancestor which is blocked according to the blocking condition for a $\mathcal{SHOIQ}$-forest.

We are now ready to propose an EXPSPACE tableau-based algorithm for $\mathcal{SHOIQ}$. It starts by generating nominal star-types at level 0. The goal of the algorithm is to replace progressively non-valid star-types with those which are "nearer" the validity. When a non-valid star-type σ at a level h is replaced with a "better" one σ' at the same level, the algorithm adds $\delta(\sigma)$ to $\delta(\sigma')$ and sets $\delta(\sigma) = 0$. This may lead to update $\delta(\omega)$ where ω is linkable with σ. In addition, the algorithm has to maintain two functions $\Phi(\sigma)$ and $\widehat{\delta}(\Phi(\sigma))$ such that the conditions 5 and 6 in Definition 9 always hold after each update of star-types.

An important difference between the tableau algorithm presented in Section 3 and the tableau algorithm for constructing a valid frame is that the latter adds star-types to a frame and updates functions $\delta(\sigma)$ and $\widehat{\delta}(\Phi(\sigma))$ instead of adding nodes for representing individuals. Again, we refer the readers to [5] where the rules for building a valid frame, called *frame rules*, can be found.

Soundness of the tableau-based algorithm for building a frame can be established thanks to Lemma 2. Since each frame rule has its counterpart in the expansion rules, completeness of the algorithm can be shown by using the same arguments as those employed to prove Lemma 1. From these results and Lemma 3, we obtain the following main result of the section:

Theorem 2. *Let $(\mathcal{T}, \mathcal{R})$ be a $\mathcal{SHOIQ}$ knowledge base. The tableau algorithm for constructing a frame is a decision procedure for consistency of $(\mathcal{T}, \mathcal{R})$ and it runs in* EXPSPACE *in the size of $(\mathcal{T}, \mathcal{R})$.*

5 Conclusion and Discussion

We have presented in this paper a practical EXPSPACE decision procedure for the logic $\mathcal{SHOIQ}$. The construction of this algorithm is founded on the well-known results for $\mathcal{SHOIQ}$ and $\mathcal{C}^2$. First, we have based our approach on a technique that constructs tree-like structures for representing a model. This allows us to reuse the standard blocking technique over these tree-like structures to obtain termination. Second, we have transferred to $\mathcal{SHOIQ}$ the method used for constructing a NEXPTIME algorithm for $\mathcal{C}^2$. This enables us to represent a doubly exponential $\mathcal{SHOIQ}$-forest by an exponential structure.

The tableau algorithms proposed in the present paper have introduced several new non-deterministic rules, e.g., $\bowtie$- or $\leq_o$-rules. In particular, the most non-deterministic behaviour of the tableau algorithm for building a valid frame is to update the function δ for star-types ω when applying frame rules to a star-type σ which is linkable with ω. Such an update may lead to choose a subset from an exponential set of linkable star-types. An open issue consists in investigating whether the complexity resulting from this behaviour is comparable to that caused by the $\leq$- and $\leq_o$-rules.

Acknowledgements. Thanks to the reviewers for their helpful comments.

References

1. Horrocks, I.: The FaCT system. In: de Swart, H. (ed.) Proc. of the 2nd Int. Conf. on Analytic Tableaux and Related Methods (TABLEAUX'98). Lecture Notes in Artificial Intelligence, vol. 1397, pp. 307–312. Springer (1998), download/1998/Horr98b.pdf
2. Horrocks, I., Sattler, U.: A tableau decision procedure for $\mathcal{SHOIQ}$. Journal Of Automated Reasoning 39(3), 249–276 (2007)
3. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for expressive description logics. In: Proceedings of the International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR 1999). Springer (1999)
4. Kazakov, Y., Motik, B.: A resolution-based decision procedure for $\mathcal{SHOIQ}$. Journal of Automated Reasoning 40(2–3), 89–116 (2008)
5. Le Duc, C., Lamolle, M., Curé, O.: An EXPSPACE tableau-based algorithm for $\mathcal{SHOIQ}$. In: Technical Report. http://www.iut.univ-paris8.fr/∼leduc/papers/RR2012a.pdf (2012)
6. Patel-Schneider, P., Hayes, P., Horrocks, I.: OWL web ontology language semantics and abstract syntax. In: W3C Recommendation (2004)
7. Pratt-Hartmann, I.: Complexity of the two-variable fragment with counting quantifiers. Journal of Logic, Language and Information 14(3), 369–395 (2005)
8. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: a pratical OWL-DL reasoner. Journal of Web Semantics 5(2), 51–53 (2007)
9. Tobies, S.: The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. Journal of Artificial Intelligence Research 12, 199–217 (2000)