Tableau Systems for $\mathcal{SHIO}$ and $\mathcal{SHIQ}$

Jan Hladik*, Jörg Model
Chair for Automata Theory, TU Dresden
{hladik,model}@tcs.inf.tu-dresden.de

1 Introduction

Two prominent families of algorithms for the satisfiability test of DLs are *automata-based algorithms* (see e.g. [6]), which translate a concept C into an automaton A_C accepting all (abstractions of) models for C, and *tableau algorithms* (TAs) [2], which incrementally create a tree-shaped (pre-) model for C using a set of *completion rules*. In short, the advantages of automata algorithms are on the theoretical side, because in many cases the proofs are very elegant and provide tight complexity bounds (in particular for ExpTime-complete logics), whereas the advantages of tableau algorithms are on the practical side, since they are well suited for implementation and optimisation.

Thus, an approach combining both advantages is highly desirable. In [1], we introduced *tableau systems (TSs)*, a framework for tableau algorithms. From a TS for a DL $\mathcal{L}$, we can derive an automata algorithm deciding satisfiability of $\mathcal{L}$ inputs in exponential time, and a tableau algorithm for $\mathcal{L}$, including an appropriate blocking condition which ensures termination. As an application of this framework, we present in this paper tableau systems for two expressive DLs, the new DL $\mathcal{SHIO}$ and the well-known $\mathcal{SHIQ}$ [5]. Our main results are the following: firstly, we obtain that $\mathcal{SHIO}$ satisfiability is ExpTime-complete and can be decided by a tableau algorithm. Secondly, we will see that although these two logics share most of their constructs, they lead to quite different TSs, which demonstrates how the capabilities of our framework can be used to capture different language properties. Thirdly, the succinctness of the proofs demonstrates how our framework simplifies the design of TAs.

2 The Tableau Framework for ExpTime Logics

Although the term "DL tableau algorithm" is not formally defined, the following features can be considered as the common ground for existing algorithms: a TA operates on a *completion tree* which represents a model for the input (e.g. a concept term, possibly together with an RBox or TBox) under consideration. To generate this model, the TA starts with an initial tree, which is subsequently modified according to a set of *completion rules*, which may or may not be applicable to a certain node. These rules essentially describe subtrees of the completion tree before and after rule application, i.e. pre- and post-conditions. In many cases, they only operate on a node and its direct neighbours, but sometimes they also consider nodes which are arbitrarily

*The first author of this paper is supported by the DFG, Project No. GR 1324/3-4.

far apart (e.g. for nominals as in [4]) or global information which is relevant for all nodes (e.g. for concept or role inclusion axioms). In general, it is possible that several rules are applicable to a node at the same time, but the sequence of rule applications is *don't-care*-nondeterministic, i.e. every sequence will lead to the same result. In contrast, some rules, e.g. for disjunction, are *don't-know*-nondeterministic, i.e. it is possible that one disjunct may lead to a model, while another one may not.

An inconsistency in the generated completion tree is detected through *clash triggers*, subtrees containing an obvious contradiction, which means that a completion tree containing a clash trigger cannot be transformed into a model. Thus, a model is represented by a completion tree which is *saturated*[1], i.e. to which no rule is applicable, and *clash-free*, i.e. not containing a clash trigger.

In our framework, we formalise a completion tree as an *S-tree* $((V, E, n, \ell), \mu)$, where (V, E) is a bounded width tree, n and ℓ are node and edge labelling functions, and μ is a *global memory* which is used to store information relevant for all nodes. Rules are are represented by *S-Patterns*, which are essentially S-trees of bounded depth. For every S-pattern P, we describe the possible modifications by *all* rules as follows: P is mapped to a set of sets of patterns $\{S_1, S_2, \ldots, S_n\}$, where the choice of the set S_i represents the don't-care choice of the rule that is applied, and the choice of the pattern within the set S_i represents the don't-know choice of the alternative (a deterministic rule corresponds to a singleton set S_i). In particular, if P is saturated, then it is mapped to the empty set. As an example, consider a pattern P in which the node n is labelled with $\{C \sqcap D, E \sqcup F\}$. Here, one can define the rules as follows: $P \mapsto \{\{P_1\}, \{P_2, P_3\}\}$, and in P_1, C and D are added to the label of n, whereas E is added in P_2 and F in P_3.

S-patterns are also used to describe clash triggers: within our framework, a clash trigger is simply a pattern which contains a contradiction (in the context of the logic under consideration). Note that S-patterns also contain a μ component, thus rules and clash triggers can read the global memory, and rules can also modify it.

Since our intention was to define one tableau system for one logic rather than a particular one for every possible input, we have to include patterns for all possible node labels in these rules and clash triggers. For a particular input Γ, these sets are mapped to the appropriate subsets, which is necessary to ensure decidability in ExpTime.

Formally, a tableau system S is a tuple $(\mathsf{NLE}, \mathsf{GME}, \mathsf{EL}, \cdot^S, k, \mathcal{R}, \mathcal{C})$, where NLE is the set of all possible node label elements, GME is the set of global memory elements, EL is the set of edge labels, k is the maximum pattern depth, i.e. the number of edges on a longest path, and $\mathcal{R}$ and $\mathcal{C}$ are the sets of rules and clash triggers as defined above. The function $\cdot^S$ maps an input Γ to the tuple $\Gamma^S = (\mathsf{nle}, \mathsf{gme}, \mathsf{el}, \mathsf{ini})$, where nle, gme and el are finite subsets of the corresponding sets in S, and $\mathsf{ini} \subseteq \wp(\mathsf{nle}) \times \wp(\mathsf{gme})$ describes the possible initial states for Γ ($\wp$ denotes power set).

In order to obtain the results of our framework (ExpTime automata algorithm and a terminating TA) from a tableau system S, it has to satisfy three conditions: ExpTime-*admissibility, soundness and p-completeness*, which will be explained in the following paragraphs. Since the formal definitions require a rather complex notation, we will only explain the intuition behind these conditions here and refer the reader to [1] for the details of the definitions.

[1] Usually, this property is called "completeness"; we use the word "saturated" to avoid confusion with the notion of completeness of the decision procedure.

ExpTime**-admissibility.** In order to be *admissible*, tableau systems have to satisfy four conditions:

1. Rules may never remove anything from a pattern, and they must add information (nodes, labels or global memory elements).
2. If $P_s = (V_s, E_s, n_s, \ell_s)$ is a saturated pattern and $P = (V, E, n, \ell)$ is a non-saturated sub-pattern of P_s (i.e. $V \subseteq V_s$ and, for all $v \in V$, $n(v) \subseteq n_s(v)$), then every applicable rule can be applied to P in such a way that the resulting pattern P' is a sub-pattern of P_s.
3. Rules may only add elements (to nodes, edges or the global memory) which appear in the subset nle, el or gme for the corresponding input Γ.
4. If P is a clash-trigger, then all super-patterns of P are also clash-triggers.

Conditions 1 and 2 are required to prove termination. However, not all existing TAs satisfy these conditions, e.g. sometimes there are rules which merge two nodes, e.g. the $\leqslant$-rule for $\mathcal{SHIQ}$ in [5], or create more nodes than necessary for a saturated pattern, e.g. the usual definition of the $\exists$-rule. However, for the logics we considered it was possible to reformulate these rules in an admissible way. For example, we can define the rules for $\exists$- and $\geqslant$-formulas non-deterministically (see below).

For ExpTime-*admissibility*, we require in addition to admissibility that the sets nle_S, el_S, gme_S and ini_S and the size of their elements are polynomial in the size of Γ and can be computed in time exponential in the size of Γ, and that it can be decided in exponential time whether a rule or a clash trigger is applicable to a pattern.

Soundness and p-completeness. For a tableau system S, it must be shown that if there exists a clash-free and saturated completion tree, then there exists a model (*soundness*), and conversely, that there exists a polynomial p such that for every satisfiable input Γ, there exists a clash-free and saturated completion-tree whose out-degree is bounded by $p(|\Gamma|)$ (p-*completeness*). Here, we distinguish between an S-tree *compatible with* Γ, a tree which is labelled in accordance with Γ^S, from an S-tree *for* Γ, a tree which can be constructed from an initial tree by rule application. If a saturated and complete S-tree compatible with Γ exists, then the existence of a saturated and complete S-tree for Γ follows from the framework (essentially from the admissibility condition). Thus in the proofs, we will only show the existence of an S-tree $((V, E, n, \ell), \mu)$ *compatible* with Γ, which is defined as follows:

- $\mu \subseteq \wp(\mathsf{gme}_S(\Gamma))$ and $n(x) \subseteq \wp(\mathsf{nle}_S(\Gamma))$ for each $x \in V$;
- $\ell(x, y) \in \mathsf{el}_S(\Gamma)$ for each $(x, y) \in E$;
- there exists $(\Lambda, \nu) \in \mathsf{ini}_S(\Gamma)$ such that $\nu \subseteq \mu$ and $\Lambda \subseteq n(v_0)$ for the root node v_0;
- the out-degree of T is bounded by $p(|\Gamma|)$ for a polynomial p.

3 The Description Logics $\mathcal{SHIO}$ and $\mathcal{SHIQ}^V$

Both $\mathcal{SHIO}$ and $\mathcal{SHIQ}$ are extensions of $\mathcal{SHI}$ (called $\mathcal{ALCHI}_{R^+}$ in [3]), which provides for transitive and inverse roles and role hierarchies. In addition to the $\mathcal{SHI}$ constructs, $\mathcal{SHIO}$ allows for *nominals*, i.e. concepts which have to be interpreted by singleton sets. This makes it possible to express that some concept can only have one instance (e.g. God), or to give names to individuals (e.g. Rome or John) and use these names in concept definitions (Roman or $\mathsf{Friend\_of\_John}$). The logic $\mathcal{SHIQ}$ allows for *qualifying number restrictions (QNR)* as described in [5]. For this paper, we introduce the syntactic variant $\mathcal{SHIQ}^V$, which does not include universal and

existential quantification and thus requires fewer rules. For both logics, the presence of transitive roles together with role hierarchies makes it possible to internalise general concept inclusion axioms [5], thus we do not include them in our syntax.

Definition 1 ($\mathcal{SHIO}$ and $\mathcal{SHIQ}^V$ syntax.) Let CON be a set of concept names, ROL be a set of role names, the set of nominal names $\mathsf{NOM} \subseteq \mathsf{CON}$, and the set of transitive role names $\mathsf{TRA} \subseteq \mathsf{ROL}$. If r is a role name, then both r and r^-, the inverse of r, are roles. To avoid multiple inverse operators as in r^{--}, we use the notation $\overline{r}$, with the meaning r^-, if r is a role name, and s, if r is an inverse role s^-. If r and s are roles, then $r \sqsubseteq s$ is a *role inclusion axiom*. An *RBox* is a finite set of role inclusion axioms. For an RBox B, we define the *role hierarchy* B^+ as $B^+ = B \cup \{\overline{r} \sqsubseteq \overline{s} \mid r \sqsubseteq s \in B\}$, and by $\sqsubseteq^*_B$ we denote the reflexive-transitive closure of $\sqsubseteq$ on B^+. A role r is called *simple* w.r.t. an RBox R if there exists no role $s \in \mathsf{TRA}$ with $s \sqsubseteq^*_B r$ or $\overline{s} \sqsubseteq^*_B r$.

The set of $\mathcal{SHIO}$ concepts is inductively defined as follows: every concept name is a concept; and if C and D are concepts and r is a role, then $\neg C$, $C \sqcap D$, $C \sqcup D$, $\forall r.C$, and $\exists r.C$ are also concepts.

The set of $\mathcal{SHIQ}^V$ concepts is inductively defined as for $\mathcal{SHIO}$, with the restriction that the set NOM is empty and the quantifiers $\exists$ and $\forall$ are not allowed. In addition, $\mathcal{SHIQ}^V$ provides the following constructors: if C is a concept, m is a non-negative integer and r is a *simple* role, then $(\leqslant m\ r\ C)$ and $(\geqslant m\ r\ C)$ are $\mathcal{SHIQ}^V$ roles. If r is an arbitrary role, then $(\leqslant 0\ r\ C)$ and $(\geqslant 1\ r\ C)$ are also $\mathcal{SHIQ}^V$ concepts.

The concepts $\forall r.C$ and $\exists r.C$ can be expressed in $\mathcal{SHIQ}^V$ by $(\leqslant 0\ r\ \neg C)$ and $(\geqslant 1\ r\ C)$, respectively. Thus, our syntax allows for non-simple roles in QNRs, if they are equivalent to an $\forall$ or $\exists$ formula, but not in the general case.

Definition 2 ($\mathcal{SHIO}$ and $\mathcal{SHIQ}^V$ semantics.) An *interpretation* of a concept C w.r.t. an RBox B is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set of individuals and $\cdot^\mathcal{I}$ maps every concept name C to a set $C^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and every role name r to a set $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. For all $O \in \mathsf{NOM}$, it holds that $\#O^\mathcal{I} = 1$, where $\#S$ denotes the cardinality of a set S. For all $t \in \mathsf{TRA}$, it holds that $t^\mathcal{I} = (t^\mathcal{I})^+$, where $\cdot^+$ denotes the transitive closure of a relation t. Complex roles and concepts are interpreted as follows:

- $(r^-)^\mathcal{I} = \{(x,y) \mid (y,x) \in r^\mathcal{I}\}$,
- $(\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I}$, $\quad (C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$, $\quad (C \sqcup D)^\mathcal{I} = C^\mathcal{I} \cup D^\mathcal{I}$,
- $(\exists r.C)^\mathcal{I} = \{d \in \Delta^\mathcal{I} \mid$ there is an $e \in \Delta^\mathcal{I}$ with $(d,e) \in r^\mathcal{I}$ and $e \in C^\mathcal{I}\}$,
- $(\forall r.C)^\mathcal{I} = \{d \in \Delta^\mathcal{I} \mid$ for all $e \in \Delta^\mathcal{I}$, if $(d,e) \in r^\mathcal{I}$, then $e \in C^\mathcal{I}\}$,
- $(\leqslant m\ r\ C)^\mathcal{I} = \big\{x \mid \sharp\{(x,y) \in r^\mathcal{I}$ and $y \in C^\mathcal{I}\} \leq m\big\}$,
- $(\geqslant m\ r\ C)^\mathcal{I} = \big\{x \mid \sharp\{(x,y) \in r^\mathcal{I}$ and $y \in C^\mathcal{I}\} \geq m\big\}$.

An interpretation $\mathcal{I}$ is a *model* for an RBox B if, for all $r \sqsubseteq s \in B$, it holds that $r^\mathcal{I} \subseteq s^\mathcal{I}$. A *model* for C w.r.t. B is a model for B where $C^\mathcal{I}$ is a nonempty set. If such a model exists, we say that C is *satisfiable* w.r.t. B.

4 The Tableau Systems $S_\mathcal{O}$ and $S_\mathcal{Q}$

Before defining the tableau systems, we fix some notation. Firstly, in an S-pattern $P = ((V, E, n, \ell), \mu)$ with $\{m, n\} \subseteq V$, we call m an *r-neighbour* of n if $\ell(n,m) = r$ or

$\ell(m,n) = \overline{r}$. Secondly, to capture roles which are implicitly declared to be transitive (e.g. $\overline{r}$ if $r \in \mathsf{TRA}$), we use, for an RBox B, the predicate Trans_B, and define that for a role r, $\mathsf{Trans}_B(r)$ is true iff there exists a role s such that $s \in \mathsf{TRA}, s' \sqsubseteq^*_B r$ and $r \sqsubseteq^*_B s''$ for some $s', s'' \in \{s, s^-\}$.

For the sake of simplicity, we only deal with concepts in *negation normal form (NNF)*, i.e. where negation appears only directly before concept names. Every concept can be transformed into an equivalent one in NNF in linear time using the duality of $\wedge$ to $\vee$, $\exists$ to $\forall$, and $\geqslant$ to $\leqslant$. By $\sim C$ we denote the NNF of $\neg C$. The *closure* $\mathsf{clos}(C,B)$ of a concept term C and an RBox B is defined as follows: $C \in \mathsf{clos}(C,B)$; if $\neg D \in \mathsf{clos}(C,B)$, then $D \in \mathsf{clos}(C,B)$; if $D \sqcap E$ or $D \sqcup E \in \mathsf{clos}(C,B)$, then $\{D,E\} \subseteq \mathsf{clos}(C,B)$; if $\exists r.D \in \mathsf{clos}(C,B)$, then $D \in \mathsf{clos}(C,B)$; and if $\forall r.D \in \mathsf{clos}(C,B)$ and the role s appears in C or B, then $\{D, \forall s.D, \forall \overline{s}.D\} \subseteq \mathsf{clos}(C,B)$.[2] For QNR, we need the following addition: if $(\leqslant m\ r\ D)$ or $(\geqslant m\ r\ D) \in \mathsf{clos}(C,B)$, then $\{D, \sim D\} \subseteq \mathsf{clos}(C,B)$, and if $(\leqslant 0\ r\ D) \in \mathsf{clos}(C,B)$ and the role s appears in C or B, then $\{(\leqslant 0\ s\ D), (\leqslant 0\ \overline{s}\ D)\} \subseteq \mathsf{clos}(C,B)$.

We can now define a TS for $\mathcal{SHIO}$, $S_\mathcal{O} = (\mathsf{NLE}_\mathcal{O}, \mathsf{GME}_\mathcal{O}, \mathsf{EL}_\mathcal{O}, 1, \cdot^{S_\mathcal{O}}, \mathcal{R}_\mathcal{O}, \mathcal{C}_\mathcal{O})$. Here, we use the global memory for three purposes: for transitive roles, role inclusion axioms, and for information about concepts appearing in a node label together with a nominal.

- $\mathsf{NLE}_\mathcal{O}$ is the set of all $\mathcal{SHIO}$ concepts,
- $\mathsf{GME}_\mathcal{O} = \{(O,C) \mid O \in \mathsf{NOM}$ and $C \in \mathsf{NLE}_\mathcal{O}\} \cup$ $\{\mathsf{Trans}(r) \mid r$ is a role$\} \cup \{r \sqsubseteq^* s \mid r$ and s are roles$\}$,
- $\mathsf{EL}_\mathcal{O}$ is the set of all $\mathcal{SHIO}$ roles, and
- for an input $\Gamma = (C,B)$, where C is a concept and B is an RBox, the function $\cdot^{S_\mathcal{O}}$ maps Γ to a tuple $\Gamma^{S_\mathcal{O}} = (\mathsf{nle}_\Gamma, \mathsf{gme}_\Gamma, \mathsf{el}_\Gamma, \mathsf{ini}_\Gamma)$ with
  - $\mathsf{nle}_\Gamma = \mathsf{clos}(C,B)$,
  - $\mathsf{el}_\Gamma = \{r \mid r$ or $\overline{r}$ appears in C or $B\}$,
  - $\mathsf{gme}_\Gamma = \{(O,D) \mid O \in \mathsf{NOM} \cap \mathsf{clos}(C,B)$ and $D \in \mathsf{clos}(C,B)\} \cup$ $\{\mathsf{Trans}(r) \mid r \in \mathsf{el}_\Gamma\} \cup \{r \sqsubseteq^* s \mid \{r,s\} \subseteq \mathsf{el}_\Gamma\}$, and
  - $\mathsf{ini}_\Gamma = \{(\{C\}, \{\mathsf{Trans}(r) \mid \mathsf{Trans}_B(r)$ holds$\} \cup \{r \sqsubseteq^* s \mid r \sqsubseteq^*_B s$ holds$\})\}$.

The set of rules $\mathcal{R}_\mathcal{O}$ is defined in Figure 1. For each pattern $P = (t, \mu)$, where $t = (V, E, n, \ell)$ has v_0 as root and depth at most 1, $\mathcal{R}(P)$ contains the described sets. Most of these rules correspond directly to the "standard" rules known from DL tableaus, with the exception of R$\exists$, which in our framework is non-deterministic. The reason for this is that with a deterministic rule which simply adds a new son node, this TS would violate condition 2 of admissibility (see Section 2). In an implementation, a deterministic rule would be preferable due to efficiency considerations, since the creation of duplicate nodes does not compromise completeness of the decision procedure. Finally, the set $\mathcal{C}_\mathcal{O}$ of clash patterns contains all patterns $((V,E,n,\ell),\mu)$ of depth 0 with node v_0 such that $\{D, \neg D\} \subseteq n(v_0)$ for some concept $D \in \mathsf{clos}(C,R)$. This completes $S_\mathcal{O}$, and we can obtain our first result:

Lemma 3 The TS $S_\mathcal{O}$ is admissible, sound and q-complete for $\mathcal{SHIO}$ satisfiability, where $q = (x \mapsto x^2)$.

Proof. Since admissibility of the tableau systems is easy to see, we prove only soundness and p-completeness.

[2] The slightly unusual definition for the $\forall$ quantifier is motivated by the $\forall_+$-rule (see below), which in turn is necessary to capture transitive sub-roles of non-transitive roles.

R⊓ If $C \sqcap D \in n(v)$ for a node $v \in V$ and $\{C, D\} \not\subseteq n(v)$, then $\mathcal{R}(P)$ contains $\{((V, E, n', \ell), \mu)\}$, where $n'(x) = n(x)$ for all $x \neq v$ and $n'(v) = n(v) \cup \{C, D\}$.

R⊔ If $C \sqcup D \in n(v)$ and $\{C, D\} \cap n(v) = \emptyset$, then $\mathcal{R}(P)$ contains $\{((V, E, n', \ell), \mu), ((V, E, n'', \ell), \mu)\}$, where $n'(x) = n''(x) = n(x)$ for all $x \neq v$, $n'(v) = n(v) \cup \{C\}$ and $n''(v) = n(v) \cup \{D\}$.

R∃ If $\exists r.C \in n(v_0)$, $v_1, \ldots, v_m$ are all the sons of v_0 with $\ell(v_0, v_i) = r$, and $C \notin n(v_i)$ for all $i, 1 \leq i \leq m$, then $\mathcal{R}(P)$ contains the set $\{P_0, P_1, \ldots, P_m\}$ with

- $P_0 = ((V_0, E_0, n_0, \ell_0), \mu)$, where $v' \notin V$, $V_0 = V \cup \{v'\}, E_0 = E \cup \{(v_0, v')\}, n_0 = n \cup \{v' \mapsto \{C\}\}, \ell_0 = \ell \cup \{(v_0, v') \mapsto r\}$.
- for all $i, 1 \leq i \leq m, P_i = ((V, E, n_i, \ell), \mu)$, where $n_i(x) = n(x)$ for all $x \neq v_i$ and $n_i(v_i) = n(v_i) \cup \{C\}$.

R∀ If $\forall r.C \in n(v)$ for some $v \in V$, v' is an s-neighbour of v with $C \notin n(v')$ and $s \sqsubseteq^* r \in \mu$, then $\mathcal{R}(P)$ contains $\{((V, E, n', \ell), \mu)\}$ with $n'(x) = n(x)$ for $x \neq v'$ and $n'(v') = n(v') \cup \{C\}$.

R∀$_+$ If $\forall r.C \in n(v)$, $\{\mathsf{Trans}(s), s \sqsubseteq^* r, q \sqsubseteq^* s\} \subseteq \mu$ and v' is an q-neighbour of v with $\forall s.C \notin n(v')$, then $\mathcal{R}(P)$ contains $\{((V, E, n', \ell), \mu)\}$ with $n'(x) = n(x)$ for $x \neq v'$ and $n'(v') = n(v') \cup \{\forall s.C\}$.

R↑ If $\{O, C\} \subseteq n(v)$ for some $O \in \mathsf{NOM}$ and $(O, C) \notin \mu$, then $\mathcal{R}(P)$ contains $\{((V, E, n, \ell), \mu')\}$, where $\mu' = \mu \cup \{(O, C)\}$.

R↓ If $O \in n(v)$ for an $O \in \mathsf{NOM}, (O, C) \in \mu$ and $C \notin n(v)$, then $\mathcal{R}(P)$ contains $\{((V, E, n', \ell), \mu)\}$, where $n'(x) = n(x)$ for $x \neq v$ and $n'(v) = n(v) \cup \{C\}$.

Figure 1: Tableau rules for $\mathcal{SHIO}$.

Soundness. From a saturated and clash-free S-tree (t, μ) with $t = (V, E, n, \ell)$, we generate a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ as follows: $\Delta^{\mathcal{I}} = \{d_O \mid O \in \mathsf{NOM} \cap \mathsf{clos}(C, B)\} \cup \{d_v \mid v \in V \text{ and } n(v) \cap \mathsf{NOM} = \emptyset\}$, i.e. we have one individual for each nominal name and one individual for every tree node that is not labelled with a nominal. A concept name C is interpreted as follows: for every $O \in \mathsf{NOM} \cap \mathsf{clos}(C, B)$, $d_O \in C^{\mathcal{I}}$ iff there is a node $v \in V$ with $\{O, C\} \subseteq n(v)$. Since R↑ and R↓ are not applicable, all nodes whose labels contain the same nominal symbol have exactly the same label, and thus $\cdot^{\mathcal{I}}$ is well-defined. For all other individuals, $d_v \in C^{\mathcal{I}}$ iff $C \in n(v)$. For a role r, $r^{\mathcal{I}}$ is the smallest set satisfying the following conditions: if $\ell(v, w) = r$ or $\ell(w, v) = \overline{r}$, then $(d_v, d_w) \in r^{\mathcal{I}}$; if $s \sqsubseteq^* r \in \mu$, then $s^{\mathcal{I}} \subseteq r^{\mathcal{I}}$; if $\mathsf{Trans}(r) \in \mu$, then $r^{\mathcal{I}}$ is closed under transitivity.

We will now show by induction that complex concepts are interpreted correctly. By definition, all individuals belong to the interpretation of the concept names in their labels, and the interpretation of a nominal contains exactly one element. From our construction, it follows directly that the role hierarchy is respected and transitive roles are interpreted correctly. For a conjunct $C \sqcap D$ (disjunct $C \sqcup D$) in a node label $n(v)$, since R⊓ (R⊔) is not applicable, it follows that C and D (C or D) are contained in $n(v)$, and by induction, d_v is contained in $C^{\mathcal{I}} \cap D^{\mathcal{I}}$ ($C^{\mathcal{I}} \cup D^{\mathcal{I}}$).

If $\exists r.C \in n(v)$, we assume w.l.o.g. that r is a role name (if it is an inverse role, the argument is analogous). Since R∃ is not applicable, there exists an r-son w of v with $C \in n(w)$. By construction, $(d_v, d_w) \in r^{\mathcal{I}}$ and $d_w \in C^{\mathcal{I}}$. If $\forall r.C \in n(v)$, we again assume that r is a role name. There are two possible reasons why (d_v, d_w) can be contained in $r^{\mathcal{I}}$: firstly, if w is an s-neighbour of v for some s with $s \sqsubseteq^* r \in \mu$. In

this case, it follows that $C \in n(w)$ because R$\forall$ is not applicable, and thus $d_w \in C^{\mathcal{I}}$. Secondly, if there exist roles $s, s_1, \ldots, s_k$ such that $\{\mathsf{Trans}(s), s \sqsubseteq^* r, s_i \sqsubseteq^* s\} \subseteq \mu$ for all $i \in \{1, \ldots, k\}$ and there is an s_i-chain from v to w, i.e. a sequence of nodes $v_1, v_2, \ldots, v_n$ such that, for all edges $e \in \{(v, v_1), (v_1, v_2), \ldots, (v_n, w)\}$, it holds that $e \in E$ and $\ell(e) = s_i$ for some i. In this case, since R$\forall_+$ is not applicable, all nodes $v_1, \ldots, v_k$ are labelled with $\forall s.C$ and, since R$\forall$ is not applicable to v_k, $n(w)$ contains C. By induction, it follows that $d_v \in (\forall r.C)^{\mathcal{I}}$.

Completeness. We have to show that if there exists a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ for an input $\Gamma = (C, B)$, then there also exists a clash-free and saturated S-tree (t, μ) with $t = (V, E, n, \ell)$ for Γ, whose width is at most quadratic in $|\Gamma|$. We will create (t, μ) by unravelling $\mathcal{I}$: firstly, we add the appropriate transitivity axioms ($\mathsf{Trans}(r)$ if $\mathsf{Trans}_B(r)$ holds) and role inclusion axioms ($r \sqsubseteq^* s$ if $r \sqsubseteq^*_B s$ holds) to μ. The tree t is inductively defined as follows: since $\mathcal{I} \models \Gamma$, there is an individual d_0 in $\Delta^{\mathcal{I}}$ which satisfies C. We start with $V = \{v_0\}$ and define $n(v_0)$ as the set of all concepts in $\mathsf{clos}(C, B)$ which d_0 satisfies. We define a function $\pi : V \to \Delta^{\mathcal{I}}$ and set $\pi(v_0) = d_0$.

Then we iterate, for every node v, the following procedure: for every existential formula $\exists r.D \in n(v)$ we choose a witness individual $d \in \Delta^{\mathcal{I}}$ with $d \in D^{\mathcal{I}}$ and $(\pi(v), d) \in r^{\mathcal{I}}$ (such a witness exists by definition of $n(v)$). We create a new node w with $\pi(w) = d$, $(v, w) \in E$ and $\ell(v, w) = r$. Again, we label w with the appropriate concepts in $\mathsf{clos}(C, B)$ and then continue the iteration. For every nominal concept O, we add to μ the pair (O, D) for every concept $D \in \mathsf{clos}(C)$ which the unique element d_O of $O^{\mathcal{I}}$ satisfies.

It is easy to see that (t, μ) is compatible with Γ and clash-free. We will now show that it is also saturated: from the definition of clos, it follows that R$\sqcap$ and R$\sqcup$ are not applicable. If a node label $n(v)$ contains a concept $\exists r.D$, then by construction of t, there is an r-successor of v which is labelled with D. Likewise, if $\forall r.D \in n(v)$, all r-neighbours of v are labelled with D. If $\forall r.D \in n(v)$, μ contains $s \sqsubseteq^* r$, $q \sqsubseteq^* s$ and $\mathsf{Trans}(s)$, and there is an q-neighbour w of v, then, since $\mathcal{I}$ is a model, $(\pi(v), \pi(w)) \in s^{\mathcal{I}}$ and, since $s^{\mathcal{I}}$ is transitive, for every node u with $(\pi(w), \pi(u)) \in s^{\mathcal{I}}$, it also holds that $(\pi(v), \pi(u)) \in s^{\mathcal{I}}$, and therefore $\pi(u) \in D^{\mathcal{I}}$. Thus, $\pi(w) \models \forall r.D$ and, since $s \sqsubseteq^*_B r$, $v(w)$ contains $\forall s.D$, which means that R$\forall_+$ is not applicable. Finally, since every node n with $\pi(n) = d_O$ for a nominal O is labelled with exactly those concepts for which μ contains (O, C), R$\uparrow$ and R$\downarrow$ are not applicable.

The width of the S-tree is quadratic in the length of Γ, because we create for every node at most one successor for every existential formula in $\mathsf{clos}(\Gamma)$ and the number of such formulas is bounded by the product of the number of roles appearing in C or B and the number of existential subformulas of C. ∎

From this, we can derive that $\mathcal{SHIO}$ satisfiability is decidable through a tableau algorithm, and we know that for the blocking condition, equality blocking suffices, i.e. we do not need pair-wise blocking as e.g. for $\mathcal{SHIQ}$ [5], since we use only patterns of depth at most 1. We can also derive a complexity result:

Theorem 4 Satisfiability for $\mathcal{SHIO}$ concepts w.r.t. RBoxes is ExpTime-complete.

Proof. It is easy to see that S is ExpTime-admissible: e.g. the size of $\mathsf{nle}_{\mathcal{S}}(\Gamma)$ and $\mathsf{gme}_{\mathcal{S}}(\Gamma)$ is quadratic in the size of the input. Soundness and completeness have been shown above. ExpTime-hardness follows from the fact that $\mathcal{SHIO}$ is an extension of $\mathcal{ALC}$ with TBoxes, for which satisfiability is known to be ExpTime-hard [7]. ∎

R⊓/R⊔ See $\mathcal{R}_\mathcal{O}$.

RC If $(\gtrless\ m\ r\ C) \in n(v)$ (where $\gtrless$ is a placeholder for $\geqslant$ or $\leqslant$) for some m and a node $v \in V$ and $\{C, \sim C\} \cap n(w) = \emptyset$ for an r-neighbour w of v, then $\mathcal{R}$ contains the set $\{((V, E, n', \ell), \mu),\ ((V, E, n'', \ell), \mu)\}$ with $n'(x) = n''(x) = n(x)$ for all $x \in V \setminus \{w\}$ and $n'(w) = n(w) \cup \{C\}$ and $n''(w) = n(w) \cup \{\sim C\}$.

R∀$_+$ If $(\leqslant 0\ r\ C) \in n(v)$ for a node $v \in V$ and there is a role s with $\{\mathsf{Trans}(s),\ q \sqsubseteq^* s,\ s \sqsubseteq^* r\} \subseteq \mu$ and an q-neighbour w of v with $(\leqslant 0\ s\ C) \notin n(w)$, then $\mathcal{R}$ contains $\{((V, E, n', \ell), \mu)\}$ with $n'(x) = n(x)$ for $x \neq w$ and $n'(w) = n(w) \cup \{(\leqslant 0\ s\ C)\}$.

R$\geqslant$ If P is a pattern of depth 2 and $(\geqslant m\ r\ C) \in n(w)$ for one successor w of v_0 and there are less than m s-neighbours of w with $s \sqsubseteq^* r \in \mu$ and $C \in n(u_i)$, then $\mathcal{R}$ contains the set $\{P_0, P_1 \dots P_n\}$, where $u_1 \dots u_n$ are the s-neighbours of w with $s \sqsubseteq^* r \in \mu$ and $C \notin n(u_i)$ and

- $P_0 = ((V_0, E_0, n_0, \ell_0), \mu)$ with $u_0 \notin V$,$V_0 = V \cup \{u_0\}$, $E_0 = E \cup \{(w, u_0)\}$, $n_0(x) = n(x)$ for all $x \in V$ and $n_0(u_0) = \{C\}$ and $\ell_0 = \ell \cup \{(w, u_0) \mapsto s\}$.
- For $1 \leq i \leq n$, $P_i = ((V, E, n_i, \ell), \mu)$ with $n_i(x) = n(x)$ for all $x \in V \setminus \{u_i\}$ and $n_i(u_i) = n(u_i) \cup \{C\}$.

R$\geqslant_{\mathsf{ROOT}}$ If $\{(\geqslant m\ r\ C), \mathsf{ROOT}\} \in n(v_0)$ of the root node v_0 and there are less than m s-successors of v_0 with $s \sqsubseteq^* r \in \mu$ and $C \in n(u_i)$, then $\mathcal{R}$ contains the set $\{P_0, P_1 \dots P_n\}$, where $u_1 \dots u_n$ are the s-successors of v_0 with $s \sqsubseteq^* r \in \mu$ and $C \notin n(u_i)$ and $P_0, \dots, P_n$ are defined as for R$\geqslant$.

Figure 2: Tableau Rules for $\mathcal{SHIQ}^V$

For $\mathcal{SHIQ}^V$, we need a TS with quite different properties: to handle QNR in the presence of inverse roles correctly, we need patterns of size 2. However, this makes a special treatment for the root node necessary, since it does not have a predecessor. Thus, we need an additional concept name ROOT and a special $\geqslant$-rule for the root node. Moreover, in contrast to the algorithm in [5], we do not have a $\leqslant$-rule, but a nondeterministic $\geqslant$-rule, which recycles neighbour nodes if necessary.

The TS $S_\mathcal{Q} = (\mathsf{NLE}_\mathcal{Q}, \mathsf{GME}_\mathcal{Q}, \mathsf{EL}_\mathcal{Q}, 2, \cdot^{S_\mathcal{Q}}, \mathcal{R}_\mathcal{Q}, \mathcal{C}_\mathcal{Q})$ is defined as $S_\mathcal{O}$, with the exception that $\mathsf{NLE}_\mathcal{Q}$ contains the additional element ROOT, $\mathsf{GME}_\mathcal{Q}$ and gme_Γ do not contain any "nominal elements" (O, C) and $\mathsf{ini}_\Gamma = \{(\{C, \mathsf{ROOT}\}, \{\mathsf{Trans}(r) \mid \mathsf{Trans}_B(r) \text{ holds}\} \cup \{r \sqsubseteq^* s \mid r \sqsubseteq^*_B s \text{ holds}\})\}$. The rules $\mathcal{R}_\mathcal{Q}$ are given in Figure 2. Note that no rule modifies μ and that R$\geqslant$ applies only to patterns whose depth is exactly 2, whereas the other rules apply to patterns of depth at most 2. Here, we obtain an explanation why double-blocking [5] is needed for $\mathcal{SHIQ}$: the maximum required pattern depth is 2.

We do not need an extra rule for concepts of the form $(\leqslant 0\ s\ D)$ to propagate $\sim D$ to all the appropriate neighbours (analogous to R∀), since this is performed by the rule RC. We also do not have a $\leqslant$-rule, but only a corresponding clash trigger: the set $\mathcal{C}_\mathcal{Q}$ contains all patterns in $\mathcal{C}_\mathcal{O}$ and additionally all patterns of depth at most 2 such that $(\leqslant m\ s\ C) \in n(v)$ with $v \in V$ and there are at least $m+1$ r-neighbours of v with $r \sqsubseteq^* s \in \mu$. For the proof of p-completeness, we require that the numbers in number restrictions are coded unary, since otherwise the width of a model can be exponential in the size of the input. We can then obtain alternative proofs for the known results of $\mathcal{SHIQ}$ decidability and complexity:

Lemma 5 If unary coding is used in number restrictions, the TS $S_{\mathcal{Q}}$ is ExpTime-admissible, sound and q-complete for $\mathcal{SHIQ}^V$ satisfiability, where $q = (x \mapsto x^2)$.

Proof. The soundness proof is easier than for $S_{\mathcal{O}}$, since a completion tree corresponds directly to a model, and we do not have to "merge" nodes labelled with nominals.

Soundness. From a saturated and clash-free S-tree (t, μ) with $t = (V, E, n, \ell)$, we generate a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ as follows: $\Delta^{\mathcal{I}} = \{d_v \mid v \in V\}$. For any concept name D and any role r we have the same interpretation as in the proof of soundness for $\mathcal{SHIO}$. Thus, for $\sqcap$ and $\sqcup$ concepts, the proof is analogous to the one for Lemma 3.

We will now show that the QNR $(\leqslant m\ r\ D)$ and $(\geqslant m\ r\ D)$ are also interpreted correctly. Let $(\geqslant m\ r\ C)$ be a concept in $n(v)$ of a node v. Then, since the S-tree is saturated, there exist at least m r-neighbours $u_1 \dots u_m$ of v with $C \in n(u_i)$ and hence $d_{u_i} \in C^{\mathcal{I}}$. From our construction, it follows that $(d_v, d_{u_i}) \in r^{\mathcal{I}}$ and thus $\sharp\{d \mid (d_v, d) \in r^{\mathcal{I}} \text{ and } d \in C^{\mathcal{I}}\} \geq m$. If $(\leqslant m\ r\ C) \in n(v)$ for a node v and a simple role r, then, since the S-tree is clash-free, there exist at most m r-neighbours $u_1 \dots u_m$ of v with $C \in n(u_i)$ and hence $d_{u_i} \in C^{\mathcal{I}}$. All other r-neighbours w contain the concept $\sim D$ because the rule RC is not applicable. Since r is simple, our construction of $r^{\mathcal{I}}$ does not introduce any further $r^{\mathcal{I}}$-neighbours. Thus, it follows that $\sharp\{d \mid (d_v, d) \in r^{\mathcal{I}} \text{ and } d \in C^{\mathcal{I}}\} \leq m$.

For a concept $(\leqslant 0\ r\ C) \in n(v)$ and a non-simple role r, the proof is similar to the one for $\forall$-concepts in $\mathcal{SHIO}$: for roles $s, s_1, \dots, s_n$ with $s \in \mathsf{TRA}$ and $s_i \sqsubseteq^* s \sqsubseteq^* r$, it follows from saturatedness of the S-tree that $(\leqslant 0\ r\ C) \in n(w)$ for every node w that is reachable from v via an s_i-chain, and thus all r-neighbours of w are labelled with $\sim C$ since the tree is clash-free and the rule RC is not applicable.

Completeness. We have to show that if there exists a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ for an input $\Gamma = (C, B)$, then there also exists a clash-free and saturated S-tree (t, μ) with $t = (V, E, n, \ell)$ for Γ. We will create (t, μ) by unravelling $\mathcal{I}$: the global memory μ is created as in the case of $\mathcal{SHIO}$ by adding all appropriate transitivity and role inclusion axioms. The tree is inductively defined as follows: we start with an individual $d_0 \in C^{\mathcal{I}}$, create a node v_0, and a function π with $\pi(v_0) = d_0$. Moreover, we define, for this and all further nodes v, $n(v)$ as the set of all concepts $D \in \mathsf{clos}(C, B)$ which the individual $\pi(v)$ satisfies. For the root node, we add the marker concept ROOT to $n(v_0)$.

New nodes are added to the tree if there is a node v and a formula $(\geqslant m\ r\ D) \in n(v)$: if there exist only k r-neighbours of v and $k < m$, we choose appropriate individuals $d_{k+1} \dots d_m$, i.e. individuals which are not yet in the range of π, with $d_i \in D^{\mathcal{I}}$ and $(\pi(v), d_i) \in r^{\mathcal{I}}$. For these nodes, we create r-neighbours $u_1 \dots u_m$ of v and set $\pi(u_i) = d_i$. Since $\mathcal{I}$ is a model, it always is possible to find appropriate individuals. From this construction, it follows that R$\geqslant$ and R$\geqslant_{\mathsf{ROOT}}$ are not applicable. The rule RC is not applicable by definition, since every node satisfies either C or $\sim C$ for any concept C, and for R$\sqcap$, R$\sqcup$ and R$\forall_+$, saturatedness follows analogous to the proof for $S_{\mathcal{O}}$. Note that the out-degree of the S-tree is bounded by the number of concepts of the form $(\geqslant m\ r\ D) \in \mathsf{clos}(C, B)$ and the highest number m occurring in such a concept.

The resulting S-tree can neither contain a clash trigger with $\{C, \sim C\} \subseteq n(v)$ for a node v and a concept C, since $\mathcal{I}$ is a model, nor a clash trigger with a number restriction of the form $D = (\leqslant m\ r\ C) \in n(v)$ of a node $v \in V$ with $\pi(v) = d$, because $d \in D^{\mathcal{I}}$ and we create at most one r-neighbour of v for every $r^{\mathcal{I}}$-neighbour of d.

It is easy to see that (t, μ) is compatible with Γ and the width is at most quadratic in length of Γ since we create only m successors for a formula $(\geqslant\ m\ r\ C) \in \mathsf{clos}(C, B)$, the number m is coded unary and the number of such formulas is quadratic in the length of Γ. ∎

Theorem 6 Satisfiability for $\mathcal{SHIQ}^V$ concepts w.r.t. RBoxes is ExpTime-complete.

Proof. The tableau system $S_{\mathcal{Q}}$ is ExpTime-admissible: the size of $\mathsf{nle}_S(\Gamma)$ and of $\mathsf{gme}_S(\Gamma)$ is quadratic in the size of the input. Soundness and p-completeness have been shown above. ExpTime-hardness follows as for $S_{\mathcal{O}}$. ∎

5 Conclusion and Outlook

We have described the main features of the tableau framework for ExpTime logics and defined tableau systems for the new DL $\mathcal{SHIO}$ and the known DL $\mathcal{SHIQ}$. It turned out that these two logics make use of different features of the tableau framework: to capture nominals, we need the global memory, whereas large patterns are needed to handle QNR properly. From the tableau systems, we can derive automata algorithms deciding satisfiability of $\mathcal{SHIO}/\mathcal{SHIQ}$ concepts w.r.t. RBoxes in ExpTime and terminating tableau algorithms, which are well suited for implementation and include appropriate blocking conditions. We believe that the simplicity of the proofs justifies the additional overhead resulting from the formalisation of the algorithm within the tableau framework.

We aim at extending our framework to cover NExpTime logics, e.g. $\mathcal{ALCQIO}$ [8]. One way of achieving this could consist in replacing looping automata (for which the emptiness problem is in P) with an automata model with an NP-complete emptiness problem, e.g. Rabin automata. However, we do not yet see a way of capturing the $\mathcal{ALCQIO}$-specific problems with a Rabin acceptance condition.

References

[1] F. Baader, J. Hladik, C. Lutz, and F. Wolter. From tableaux to automata for description logics. *Fundamenta Informaticae*, 57:1–33, 2003.

[2] F. Baader and U. Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69, 2001.

[3] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. Technical Report 98-05, LuFg Theoretical Computer Science, RWTH Aachen, 1998.

[4] I. Horrocks and U. Sattler. Ontology reasoning in the $\mathcal{SHOQ}(D)$ description logic. In *Proc. of IJCAI-01*, pages 199–204. Morgan Kaufmann, 2001.

[5] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proc. of LPAR'99*, number 1705 in LNAI, pages 161–180. Springer-Verlag, 1999.

[6] U. Sattler and M. Y. Vardi. The hybrid μ-calculus. In *IJCAR-01*, volume 2083 of *LNAI*, pages 76–91. Springer-Verlag, 2001.

[7] K. Schild. Terminological cycles and the propositional μ-calculus. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proc. of KR-94*, pages 509–520, Bonn, 1994. Morgan Kaufmann.

[8] S. Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. of Artificial Intelligence Research*, 12:199–217, May 2000.