Reasoning with Individuals for the Description Logic $\mathcal{SHIQ}$

Ian Horrocks[1], Ulrike Sattler[2], and Stephan Tobies[2]

[1] Department of Computer Science, University of Manchester, UK
horrocks@cs.man.ac.uk
[2] LuFg Theoretical Computer Science, RWTH Aachen, Germany
{sattler,tobies}@informatik.rwth-aachen.de

Abstract. While there has been a great deal of work on the development of reasoning algorithms for expressive description logics, in most cases only Tbox reasoning is considered. In this paper we present an algorithm for combined Tbox and Abox reasoning in the $\mathcal{SHIQ}$ description logic. This algorithm is of particular interest as it can be used to decide the problem of (database) conjunctive query containment w.r.t. a schema. Moreover, the realisation of an efficient implementation should be relatively straightforward as it can be based on an existing highly optimised implementation of the Tbox algorithm in the FaCT system.

1 Motivation

A description logic (DL) knowledge base (KB) is made up of two parts, a terminological part (the terminology or Tbox) and an assertional part (the Abox), each part consisting of a set of axioms. The Tbox asserts facts about *concepts* (sets of objects) and *roles* (binary relations), usually in the form of inclusion axioms, while the Abox asserts facts about *individuals* (single objects), usually in the form of instantiation axioms. For example, a Tbox might contain an axiom asserting that Man is subsumed by Animal, while an Abox might contain axioms asserting that both Aristotle and Plato are instances of the concept Man and that the pair $\langle \mathsf{Aristotle}, \mathsf{Plato} \rangle$ is an instance of the role Pupil-of.

For logics that include full negation, all common DL reasoning tasks are reducible to deciding KB consistency, i.e., determining if a given KB admits a non-empty interpretation [6]. There has been a great deal of work on the development of reasoning algorithms for expressive DLs [2; 12; 16; 11], but in most cases these consider only Tbox reasoning (i.e., the Abox is assumed to be empty). With expressive DLs, determining consistency of a Tbox can often be reduced to determining the satisfiability of a single concept [2; 23; 3], and—as most DLs enjoy the tree model property (i.e., if a concept has a model, then it has a tree model)—this problem can be decided using a tableau-based decision procedure.

The relative lack of interest in Abox reasoning can also be explained by the fact that many applications only require Tbox reasoning, e.g., ontological engineering [15; 20] and schema integration [10]. Of particular interest in this regard is the DL $\mathcal{SHIQ}$ [18], which is powerful enough to encode the logic $\mathcal{DLR}$ [10], and which can thus be used for reasoning about conceptual data models, e.g., Entity-Relationship (ER) schemas [9]. Moreover, if we think of the Tbox as a *schema* and the Abox as (possibly incomplete) *data*, then it seems reasonable to assume that realistic Tboxes will be of limited size, whereas realistic Aboxes could be of almost unlimited size. Given the high complexity of reasoning in most DLs [23; 7], this suggests that Abox reasoning could lead to severe tractability problems in realistic applications.[1]

However, $\mathcal{SHIQ}$ Abox reasoning is of particular interest as it allows $\mathcal{DLR}$ schema reasoning to be extended to reasoning about conjunctive query containment w.r.t. a schema [8]. This is achieved by using Abox individuals to represent variables and constants in the queries, and to enforce co-references [17]. In this context, the size of the Abox would be quite small (it is bounded by the number of variables occurring in the queries), and should not lead to severe tractability problems.

Moreover, an alternative view of the Abox is that it provides a restricted form of reasoning with *nominals*, i.e., allowing individual names to appear in concepts [22; 5; 1]. Unrestricted nominals are very powerful, allowing arbitrary co-references to be enforced and thus leading to the loss of the tree model property. This makes it much harder to prove decidability and to devise decision procedures (the decidability of $\mathcal{SHIQ}$ with unrestricted nominals is still an open problem). An Abox, on the other hand, can be modelled by a *forest*, a set of trees whose root nodes form an arbitrarily connected graph, where number of trees is limited by the number of individual names occurring in the Abox. Even the restricted form of co-referencing provided by an Abox is quite powerful, and can extend the range of applications for the DLs reasoning services.

In this paper we present a tableaux based algorithm for deciding the satisfiability of unrestricted $\mathcal{SHIQ}$ KBs (i.e., ones where the Abox may be non-empty) that extends the existing consistency algorithm for Tboxes [18] by making use of the forest model property. This should make the realisation of an efficient implementation relatively straightforward as it can be based on an existing highly optimised implementation of the Tbox algorithm (e.g., in the FaCT system [14]). A notable feature of the algorithm is that, instead of making a unique name assumption w.r.t. all individuals (an assumption commonly made in DLs [4]), increased flexibility is provided by allowing the Abox to contain axioms explicitly asserting inequalities between pairs of individual names (adding such an axiom for every pair of individual names is obviously equivalent to making a unique name assumption).

2 Preliminaries

In this section, we introduce the DL $\mathcal{SHIQ}$. This includes the definition of syntax, semantics, inference problems (concept subsumption and satisfiability, Abox consistency, and all of these problems with respect to terminologies[2]), and their relationships.

$\mathcal{SHIQ}$ is based on an extension of the well known DL $\mathcal{ALC}$ [24] to include transitively closed primitive roles [21]; we call this logic $\mathcal{S}$ due to its relationship with

[1] Although suitably optimised algorithms may make reasoning practicable for quite large Aboxes [13].

[2] We use *terminologies* instead of Tboxes to underline the fact that we allow for general concept inclusions axioms and do not disallow cycles.

the proposition (multi) modal logic $\mathbf{S4_{(m)}}$ [23].[3] This basic DL is then extended with inverse roles ($\mathcal{I}$), role hierarchies ($\mathcal{H}$), and qualifying number restrictions ($\mathcal{Q}$).

Definition 1. *Let* $\mathbf{C}$ *be a set of* concept names *and* $\mathbf{R}$ *a set of* role names *with a subset* $\mathbf{R}_+ \subseteq \mathbf{R}$ *of* transitive role names. *The set of* roles *is* $\mathbf{R} \cup \{R^- \mid R \in \mathbf{R}\}$. *To avoid considering roles such as* R^{--}*, we define a function* Inv *on roles such that* $\mathsf{Inv}(R) = R^-$ *if* R *is a role name, and* $\mathsf{Inv}(R) = S$ *if* $R = S^-$. *We also define a function* Trans *which returns* true *iff* R *is a transitive role. More precisely,* $\mathsf{Trans}(R) = \mathrm{true}$ *iff* $R \in \mathbf{R}_+$ *or* $\mathsf{Inv}(R) \in \mathbf{R}_+$.

A role inclusion axiom *is an expression of the form* $R \sqsubseteq S$*, where* R *and* S *are roles, each of which can be inverse. A* role hierarchy *is a set of role inclusion axioms. For a role hierarchy* $\mathcal{R}$*, we define the relation* $\underline{\circledast}$ *to be the transitive-reflexive closure of* $\sqsubseteq$ *over* $\mathcal{R} \cup \{\mathsf{Inv}(R) \sqsubseteq \mathsf{Inv}(S) \mid R \sqsubseteq S \in \mathcal{R}\}$. *A role* R *is called a* sub-role *(resp. super-role) of a role* S *if* $R \underline{\circledast} S$ *(resp.* $S \underline{\circledast} R$*). A role is* simple *if it is neither transitive nor has any transitive sub-roles.*

The set of $\mathcal{SHIQ}$-concepts *is the smallest set such that*

- *every concept name is a concept, and,*
- *if* C*,* D *are concepts,* R *is a role,* S *is a simple role, and* n *is a nonnegative integer, then* $C \sqcap D$*,* $C \sqcup D$*,* $\neg C$*,* $\forall R.C$*,* $\exists R.C$*,* $\geqslant nS.C$*, and* $\leqslant nS.C$ *are also concepts.*

A general concept inclusion axiom *(GCI) is an expression of the form* $C \sqsubseteq D$ *for two* $\mathcal{SHIQ}$*-concepts* C *and* D. *A* terminology *is a set of GCIs.*

Let $\mathbf{I} = \{a, b, c \ldots\}$ *be a set of* individual names. *An* assertion *is of the form* $a : C$*,* $(a, b) : R$*, or* $a \neq b$ *for* $a, b \in \mathbf{I}$*, a (possibly inverse) role* R*, and a* $\mathcal{SHIQ}$*-concept* C. *An* Abox *is a finite set of assertions.*

Next, we define semantics of $\mathcal{SHIQ}$ and the corresponding inference problems.

Definition 2. *An* interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *consists of a set* $\Delta^{\mathcal{I}}$*, called the* domain *of* $\mathcal{I}$*, and a* valuation $\cdot^{\mathcal{I}}$ *which maps every concept to a subset of* $\Delta^{\mathcal{I}}$ *and every role to a subset of* $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ *such that, for all concepts* C*,* D*, roles* R*,* S*, and non-negative integers* n*, the following equations are satisfied, where* $\sharp M$ *denotes the cardinality of a set* M *and* $(R^{\mathcal{I}})^+$ *the transitive closure of* $R^{\mathcal{I}}$*:*

$$
\begin{array}{rlr}
R^{\mathcal{I}} &= (R^{\mathcal{I}})^+ & \textit{for each role } R \in \mathbf{R}_+ \\
(R^-)^{\mathcal{I}} &= \{\langle x, y\rangle \mid \langle y, x\rangle \in R^{\mathcal{I}}\} & \textit{(inverse roles)} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} & \textit{(conjunction)} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} & \textit{(disjunction)} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} & \textit{(negation)} \\
(\exists R.C)^{\mathcal{I}} &= \{x \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \textit{ and } y \in C^{\mathcal{I}}\} & \textit{(exists restriction)} \\
(\forall R.C)^{\mathcal{I}} &= \{x \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \textit{ implies } y \in C^{\mathcal{I}}\} & \textit{(value restriction)} \\
(\geqslant nR.C)^{\mathcal{I}} &= \{x \mid \sharp\{y.\langle x, y\rangle \in R^{\mathcal{I}} \textit{ and } y \in C^{\mathcal{I}}\} \geqslant n\} & \textit{(}\geqslant\textit{-number restriction)} \\
(\leqslant nR.C)^{\mathcal{I}} &= \{x \mid \sharp\{y.\langle x, y\rangle \in R^{\mathcal{I}} \textit{ and } y \in C^{\mathcal{I}}\} \leqslant n\} & \textit{(}\leqslant\textit{-number restriction)}
\end{array}
$$

An interpretation $\mathcal{I}$ satisfies *a role hierarchy* $\mathcal{R}$ *iff* $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$ *for each* $R \sqsubseteq S$ *in* $\mathcal{R}$. *Such an interpretation is called a* model *of* $\mathcal{R}$ *(written* $\mathcal{I} \models \mathcal{R}$*).*

[3] The logic $\mathcal{S}$ has previously been called $\mathcal{ALC}_{R^+}$, but this becomes too cumbersome when adding letters to represent additional features.

An interpretation $\mathcal{I}$ satisfies *a terminology $\mathcal{T}$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for each GCI $C \sqsubseteq D$ in $\mathcal{T}$. Such an interpretation is called a* model *of $\mathcal{T}$ (written $\mathcal{I} \models \mathcal{T}$).*

A concept C is called satisfiable *with respect to a role hierarchy $\mathcal{R}$ and a terminology $\mathcal{T}$ iff there is a model $\mathcal{I}$ of $\mathcal{R}$ and $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$. A concept D* subsumes *a concept C w.r.t. $\mathcal{R}$ and $\mathcal{T}$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for each model $\mathcal{I}$ of $\mathcal{R}$ and $\mathcal{T}$. For an interpretation $\mathcal{I}$, an element $x \in \Delta^{\mathcal{I}}$ is called an* instance *of a concept C iff $x \in C^{\mathcal{I}}$.*

For Aboxes, an interpretation maps, additionally, each individual $a \in \mathbf{I}$ to some element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. An interpretation $\mathcal{I}$ satisfies an assertion

$$\begin{array}{rl} a\!:\!C & \textit{iff}\ \ a^{\mathcal{I}} \in C^{\mathcal{I}}, \\ (a,b)\!:\!R & \textit{iff}\ \ \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}},\ \textit{and} \\ a \dot{\neq} b & \textit{iff}\ \ a^{\mathcal{I}} \neq b^{\mathcal{I}} \end{array}$$

An Abox $\mathcal{A}$ is consistent *w.r.t. $\mathcal{R}$ and $\mathcal{T}$ iff there is a model $\mathcal{I}$ of $\mathcal{R}$ and $\mathcal{T}$ that satisfies each assertion in $\mathcal{A}$.*

For DLs that are closed under negation, subsumption and (un)satisfiability can be mutually reduced: $C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable, and C is unsatisfiable iff $C \sqsubseteq A \sqcap \neg A$ for some concept name A. Moreover, a concept C is satisfiable iff the Abox $\{a : C\}$ is consistent. It is straightforward to extend these reductions to role hierarchies, but terminologies deserve special care: In [2; 23; 3], the *internalisation* of GCIs is introduced, a technique that reduces reasoning w.r.t. a (possibly cyclic) terminology to reasoning w.r.t. the empty terminology. For $\mathcal{SHIQ}$, this reduction must be slightly modified. The following Lemma shows how general concept inclusion axioms can be *internalised* using a "universal" role U, that is, a transitive super-role of all roles occurring in $\mathcal{T}$ and their respective inverses.

Lemma 1. *Let C, D be concepts, $\mathcal{A}$ an Abox, $\mathcal{T}$ a terminology, and $\mathcal{R}$ a role hierarchy. We define*

$$C_{\mathcal{T}} := \bigsqcap_{C_i \sqsubseteq D_i \in \mathcal{T}} \neg C_i \sqcup D_i.$$

Let U be a transitive role that does not occur in $\mathcal{T}$, C, D, $\mathcal{A}$, or $\mathcal{R}$. We set

$$\mathcal{R}_U := \mathcal{R} \cup \{R \sqsubseteq U, \mathsf{Inv}(R) \sqsubseteq U \mid R \textit{ occurs in } \mathcal{T}, C, D, \mathcal{A}, \textit{ or } \mathcal{R}\}.$$

- *C is satisfiable w.r.t. $\mathcal{T}$ and $\mathcal{R}$ iff $C \sqcap C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}$ is satisfiable w.r.t. $\mathcal{R}_U$.*
- *D subsumes C with respect to $\mathcal{T}$ and $\mathcal{R}$ iff $C \sqcap \neg D \sqcap C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}$ is unsatisfiable w.r.t. $\mathcal{R}_U$.*
- *$\mathcal{A}$ is consistent with respect to $\mathcal{R}$ and $\mathcal{T}$ iff $\mathcal{A} \cup \{a : C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}} \mid a \textit{ occurs in } \mathcal{A}\}$ is consistent w.r.t. $\mathcal{R}_U$.*

The proof of Lemma 1 is similar to the ones that can be found in [23; 2]. Most importantly, it must be shown that, (a) if a $\mathcal{SHIQ}$-concept C is satisfiable with respect to a terminology $\mathcal{T}$ and a role hierarchy $\mathcal{R}$, then $C, \mathcal{T}$ have a *connected* model, i. e., a model where any two elements are connect by a role path over those roles occuring in C and $\mathcal{T}$, and (b) if y is reachable from x via a role path (possibly involving inverse roles), then $\langle x, y \rangle \in U^{\mathcal{I}}$. These are easy consequences of the semantics and the definition of U.

Theorem 1. *Satisfiability and subsumption of $\mathcal{SHIQ}$-concepts w.r.t. terminologies and role hierarchies are polynomially reducible to (un)satisfiability of $\mathcal{SHIQ}$-concepts w.r.t. role hierarchies, and therefore to consistency of $\mathcal{SHIQ}$-Aboxes w.r.t. role hierarchies.*

Consistency of $\mathcal{SHIQ}$-Aboxes w.r.t. terminologies and role hierarchies is polynomially reducible to consistency of $\mathcal{SHIQ}$-Aboxes w.r.t. role hierarchies.

3 A $\mathcal{SHIQ}$-Abox Tableau Algorithm

With Theorem 1, all standard inference problems for $\mathcal{SHIQ}$-concepts and Aboxes can be reduced to Abox-consistency w.r.t. a role hierarchy. In the following, we present a tableau-based algorithm that decides consistency of $\mathcal{SHIQ}$-Aboxes w.r.t. role hierarchies, and therefore all other $\mathcal{SHIQ}$ inference problems presented.

The algorithm tries to construct, for a $\mathcal{SHIQ}$-Abox $\mathcal{A}$, a tableau for $\mathcal{A}$, that is, an abstraction of a model of $\mathcal{A}$. Given the notion of a tableau, it is then quite straightforward to prove that the algorithm is a decision procedure for Abox consistency.

3.1 A Tableau for Aboxes

In the following, if not stated otherwise, C, D denote $\mathcal{SHIQ}$-concepts, $\mathcal{R}$ a role hierarchy, $\mathcal{A}$ an Abox, $\mathbf{R}_\mathcal{A}$ the set of roles occurring in $\mathcal{A}$ and $\mathcal{R}$ together with their inverses, and $\mathbf{I}_\mathcal{A}$ is the set of individuals occurring in $\mathcal{A}$.

Without loss of generality, we assume all concepts C occurring in assertions $a:C \in \mathcal{A}$ to be in NNF, that is, negation occurs in front of concept names only. Any $\mathcal{SHIQ}$-concept can easily be transformed into an equivalent one in NNF by pushing negations inwards using a combination of DeMorgan's laws and the following equivalences:

$$\begin{array}{llll} \neg(\exists R.C) \equiv (\forall R.\neg C) & \neg(\forall R.C) & \equiv & (\exists R.\neg C) \\ \neg(\leqslant nR.C) \equiv \geqslant(n+1)R.C & \neg(\geqslant nR.C) & \equiv & \leqslant(n-1)R.C \quad \text{where} \\ & \leqslant(-1)R.C & := & A \sqcap \neg A \quad \text{for some } A \in \mathbf{C} \end{array}$$

For a concept C we will denote the NNF of $\neg C$ by $\sim C$. Next, for a concept C, $\mathsf{clos}(C)$ is the smallest set that contains C and is closed under sub-concepts and $\sim$. We use $\mathsf{clos}(\mathcal{A}) := \bigcup_{a:C\in\mathcal{A}} \mathsf{clos}(C)$ for the closure $\mathsf{clos}(C)$ of each concept C occurring in $\mathcal{A}$. It is not hard to show that the size of $\mathsf{clos}(\mathcal{A})$ is polynomial in the size of $\mathcal{A}$.

Definition 3. *$T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{I})$ is a* tableau *for $\mathcal{A}$ w.r.t. $\mathcal{R}$ iff*

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

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

(P1) if $C \in \mathcal{L}(s)$, then $\neg C \notin \mathcal{L}(s)$,
(P2) if $C_1 \sqcap C_2 \in \mathcal{L}(s)$, then $C_1 \in \mathcal{L}(s)$ and $C_2 \in \mathcal{L}(s)$,

(P3) *if* $C_1 \sqcup C_2 \in \mathcal{L}(s)$*, then* $C_1 \in \mathcal{L}(s)$ *or* $C_2 \in \mathcal{L}(s)$*,*
(P4) *if* $\forall S.C \in \mathcal{L}(s)$ *and* $\langle s,t\rangle \in \mathcal{E}(S)$*, then* $C \in \mathcal{L}(t)$*,*
(P5) *if* $\exists S.C \in \mathcal{L}(s)$*, then there is some* $t \in \mathbf{S}$ *such that* $\langle s,t\rangle \in \mathcal{E}(S)$ *and* $C \in \mathcal{L}(t)$*,*
(P6) *if* $\forall S.C \in \mathcal{L}(s)$ *and* $\langle s,t\rangle \in \mathcal{E}(R)$ *for some* $R \mathrel{\underline{\circledast}} S$ *with* $\mathsf{Trans}(R)$*, then* $\forall R.C \in \mathcal{L}(t)$*,*
(P7) $\langle x,y\rangle \in \mathcal{E}(R)$ *iff* $\langle y,x\rangle \in \mathcal{E}(\mathsf{Inv}(R))$*,*
(P8) *if* $\langle s,t\rangle \in \mathcal{E}(R)$ *and* $R \mathrel{\underline{\circledast}} S$*, then* $\langle s,t\rangle \in \mathcal{E}(S)$*,*
(P9) *if* ${\leqslant}nS.C \in \mathcal{L}(s)$*, then* $\sharp S^T(s,C) \leqslant n$*,*
(P10) *if* ${\geqslant}nS.C \in \mathcal{L}(s)$*, then* $\sharp S^T(s,C) \geqslant n$*,*
(P11) *if* $(\bowtie\ n\ S\ C) \in \mathcal{L}(s)$ *and* $\langle s,t\rangle \in \mathcal{E}(S)$ *then* $C \in \mathcal{L}(t)$ *or* ${\sim}C \in \mathcal{L}(t)$*,*
(P12) *if* $a{:}C \in \mathcal{A}$*, then* $C \in \mathcal{L}(\mathcal{J}(a))$*,*
(P13) *if* $(a,b){:}R \in \mathcal{A}$*, then* $\langle \mathcal{J}(a), \mathcal{J}(b)\rangle \in \mathcal{E}(R)$*,*
(P14) *if* $a \neq b \in \mathcal{A}$*, then* $\mathcal{J}(a) \neq \mathcal{J}(b)$*,*

where $\bowtie$ *is a place-holder for both* $\leqslant$ *and* $\geqslant$*, and* $S^T(s,C) := \{t \in \mathbf{S} \mid \langle s,t\rangle \in \mathcal{E}(S)$ *and* $C \in \mathcal{L}(t)\}$*.*

Lemma 2. *A* $\mathcal{SHIQ}$*-Abox* $\mathcal{A}$ *is consistent w.r.t.* $\mathcal{R}$ *iff there exists a tableau for* $\mathcal{A}$ *w.r.t.* $\mathcal{R}$*.*

Proof: For the *if* direction, if $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ is a tableau for $\mathcal{A}$ w.r.t. $\mathcal{R}$, a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{A}$ and $\mathcal{R}$ can be defined as follows:

$$\begin{array}{rl}
 & \Delta^{\mathcal{I}} := \mathbf{S} \\
\text{for concept names A in } \mathsf{clos}(\mathcal{A}): & A^{\mathcal{I}} := \{s \mid A \in \mathcal{L}(s)\} \\
\text{for individual names } a \in \mathbf{I}: & a^{\mathcal{I}} := \mathcal{J}(a) \\
\text{for role names } R \in \mathcal{R}: & R^{\mathcal{I}} := \begin{cases} \mathcal{E}(R)^+ & \text{if } \mathsf{Trans}(R) \\ \mathcal{E}(R) \cup \bigcup\limits_{P \mathrel{\underline{\circledast}} R, P \neq R} P^{\mathcal{I}} & \text{otherwise} \end{cases}
\end{array}$$

where $\mathcal{E}(R)^+$ denotes the transitive closure of $\mathcal{E}(R)$. The interpretation of non-transitive roles is recursive in order to correctly interpret those non-transitive roles that have a transitive sub-role. From the definition of $R^{\mathcal{I}}$ and (P8), it follows that, if $\langle s,t\rangle \in S^{\mathcal{I}}$, then either $\langle s,t\rangle \in \mathcal{E}(S)$ or there exists a path $\langle s,s_1\rangle, \langle s_1,s_2\rangle, \ldots, \langle s_n,t\rangle \in \mathcal{E}(R)$ for some R with $\mathsf{Trans}(R)$ and $R \mathrel{\underline{\circledast}} S$.

Due to (P8) and by definition of $\mathcal{I}$, we have that $\mathcal{I}$ is a model of $\mathcal{R}$.

To prove that $\mathcal{I}$ is a model of $\mathcal{A}$, we show that $C \in \mathcal{L}(s)$ implies $s \in C^{\mathcal{I}}$ for any $s \in \mathbf{S}$. Together with (P12), (P13), and the interpretation of individuals and roles, this implies that $\mathcal{I}$ satisfies each assertion in $\mathcal{A}$. This proof can be given by induction on the length $\|C\|$ of a concept C in NNF, where we count neither negation nor integers in number restrictions. The only interesting case is $C = \forall S.E$: let $t \in \mathbf{S}$ with $\langle s,t\rangle \in S^{\mathcal{I}}$. There are two possibilities:

– $\langle s,t\rangle \in \mathcal{E}(S)$. Then (P4) implies $E \in \mathcal{L}(t)$.
– $\langle s,t\rangle \notin \mathcal{E}(S)$. Then there exists a path $\langle s,s_1\rangle, \langle s_1,s_2\rangle, \ldots, \langle s_n,t\rangle \in \mathcal{E}(R)$ for some R with $\mathsf{Trans}(R)$ and $R \mathrel{\underline{\circledast}} S$. Then (P6) implies $\forall R.E \in \mathcal{L}(s_i)$ for all $1 \leq i \leq n$, and (P4) implies $E \in \mathcal{L}(t)$.

In both cases, $t \in E^{\mathcal{I}}$ by induction and hence $s \in C^{\mathcal{I}}$.

For the converse, for $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ a model of $\mathcal{A}$ w.r.t. $\mathcal{R}$, we define a tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ for $\mathcal{A}$ and $\mathcal{R}$ as follows:

$$\mathbf{S} := \Delta^{\mathcal{I}}, \quad \mathcal{E}(R) := R^{\mathcal{I}}, \quad \mathcal{L}(s) := \{C \in \mathsf{clos}(\mathcal{A}) \mid s \in C^{\mathcal{I}}\}, \quad \text{and} \quad \mathcal{J}(a) = a^{\mathcal{I}}.$$

It is easy to demonstrate that T is a tableau for D. □

3.2 The Tableau Algorithm

In this section, we present a completion algorithm that tries to construct, for an input Abox $\mathcal{A}$ and a role hierarchy $\mathcal{R}$, a tableau for $\mathcal{A}$ w.r.t. $\mathcal{R}$. We prove that this algorithm constructs a tableau for $\mathcal{A}$ and $\mathcal{R}$ iff there exists a tableau for $\mathcal{A}$ and $\mathcal{R}$, and thus decides consistency of $\mathcal{SHIQ}$ Aboxes w.r.t. role hierarchies.

Since Aboxes might involve several individuals with arbitrary role relationships between them, the completion algorithm works on a *forest* rather than on a *tree*, which is the basic data structure for those completion algorithms deciding satisfiability of a concept. Such a forest is a collection of trees whose root nodes correspond to the individuals present in the input Abox. In the presence of transitive roles, *blocking* is employed to ensure termination of the algorithm. In the additional presence of inverse roles, blocking is *dynamic*, i.e., blocked nodes (and their sub-branches) can be un-blocked and blocked again later. In the additional presence of number restrictions, *pairs* of nodes are blocked rather than single nodes.

Definition 4. *A* completion forest *$\mathcal{F}$ for a $\mathcal{SHIQ}$ Abox $\mathcal{A}$ is a collection of trees whose distinguished root nodes are possibly connected by edges in an arbitrary way. Moreover, each node x is labelled with a set $\mathcal{L}(x) \subseteq \mathsf{clos}(\mathcal{A})$ and each edge $\langle x, y\rangle$ is labelled with a set $\mathcal{L}(\langle x, y\rangle) \subseteq \mathcal{R}_{\mathcal{A}}$ of (possibly inverse) roles occurring in $\mathcal{A}$. Finally, completion forests come with an explicit inequality relation $\dot{\neq}$ on nodes and an explicit equality relation $\doteq$ which are implicitly assumed to be symmetric.*

*If nodes x and y are connected by an edge $\langle x, y\rangle$ with $R \in \mathcal{L}(\langle x, y\rangle)$ and $R \,\underline{\circledast}\, S$, then y is called an S-*successor *of x and x is called an $\mathsf{Inv}(S)$-*predecessor *of y. If y is an S-successor or an $\mathsf{Inv}(S)$-predecessor of x, then y is called an S-neighbour of x. A node y is a successor (resp. predecessor or neighbour) of y if it is an S-successor (resp. S-predecessor or S-neighbour) of y for some role S. Finally,* ancestor *is the transitive closure of* predecessor.

For a role S, a concept C and a node x in $\mathcal{F}$ we define $S^{\mathcal{F}}(x, C)$ by

$$S^{\mathcal{F}}(x, C) := \{y \mid y \text{ is } S\text{-neighbour of } x \text{ and } C \in \mathcal{L}(y)\}.$$

A node is blocked *iff it is not a root node and it is either directly or indirectly blocked. A node x is* directly blocked *iff none of its ancestors are blocked, and it has ancestors x', y and y' such that*

1. *y is not a root node* and
2. *x is a successor of x' and y is a successor of y'* and

3. $\mathcal{L}(x) = \mathcal{L}(y)$ *and* $\mathcal{L}(x') = \mathcal{L}(y')$ and
4. $\mathcal{L}(\langle x', x\rangle) = \mathcal{L}(\langle y', y\rangle)$.

In this case we will say that y blocks x.

A node y *is* indirectly blocked *iff one of its ancestors is blocked, or it is a successor of a node* x *and* $\mathcal{L}(\langle x, y\rangle) = \emptyset$*; the latter condition avoids wasted expansions after an application of the* $\leqslant$*-rule.*

Given a $\mathcal{SHIQ}$*-Abox* $\mathcal{A}$ *and a role hierarchy* $\mathcal{R}$*, the algorithm initialises a completion forest* $\mathcal{F}_\mathcal{A}$ *consisting only of root nodes. More precisely,* $\mathcal{F}_\mathcal{A}$ *contains a root node* x_0^i *for each individual* $a_i \in \mathbf{I}_\mathcal{A}$ *occurring in* $\mathcal{A}$*, and an edge* $\langle x_0^i, x_0^j\rangle$ *if* $\mathcal{A}$ *contains an assertion* $(a_i, a_j){:}R$ *for some* R*. The labels of these nodes and edges and the relations* $\not\doteq$ *and* $\doteq$ *are initialised as follows:*

$$\begin{aligned} \mathcal{L}(x_0^i) &:= \{C \mid a_i{:}C \in \mathcal{A}\}, \\ \mathcal{L}(\langle x_0^i, x_0^j\rangle) &:= \{R \mid (a_i, a_j) : R \in \mathcal{A}\}, \\ x_0^i \not\doteq x_0^j &\text{ iff } a_i \not\doteq a_j \in \mathcal{A}, \text{ and} \end{aligned}$$

the $\doteq$*-relation is initialised to be empty.* $\mathcal{F}_\mathcal{A}$ *is then expanded by repeatedly applying the rules from Figure 1.*

For a node x*,* $\mathcal{L}(x)$ *is said to contain a* clash *if, for some concept name* $A \in \mathbf{C}$*,* $\{A, \neg A\} \subseteq \mathcal{L}(x)$*, or if there is some concept* $\leqslant nS.C \in \mathcal{L}(x)$ *and* x *has* $n+1$ S*-neighbours* $y_0, \ldots, y_n$ *with* $C \in \mathcal{L}(y_i)$ *and* $y_i \not\doteq y_j$ *for all* $0 \leq i < j \leq n$*. A completion forest is* clash-free *if none of its nodes contains a clash, and it is* complete *if no rule from Figure 1 can be applied to it.*

For a $\mathcal{SHIQ}$*-Abox* $\mathcal{A}$*, the algorithm starts with the completion forest* $\mathcal{F}_\mathcal{A}$*. It applies the expansion rules in Figure 1, stopping when a clash occurs, and answers "*$\mathcal{A}$ *is consistent w.r.t.* $\mathcal{R}$*" iff the completion rules can be applied in such a way that they yield a complete and clash-free completion forest, and "*$\mathcal{A}$ *and is inconsistent w.r.t.* $\mathcal{R}$*" otherwise.*

Since both the $\leqslant$-rule and the $\leqslant_r$-rule are rather complicated, they deserve some more explanation. Both rules deal with the situation where a concept $\leqslant nR.C \in \mathcal{L}(x)$ requires the identification of two R-neighbours y, z of x that contain C in their labels. Of course, y and z may only be identified if $y \not\doteq z$ is not asserted. If these conditions are met, then one of the two rules can be applied. The $\leqslant$-rule deals with the case where at least one of the nodes to be identified, namely y, is not a root node, and this can lead to one of two possible situations, both shown in Figure 2. The upper situation occurs when both y and z are successors of x. In this case, we add the label of y to that of z, and the label of the edge $\langle x, y\rangle$ to the label of the edge $\langle x, z\rangle$. Finally, z inherits all inequalities from y, and $\mathcal{L}(\langle x, y\rangle)$ is set to $\emptyset$, thus blocking y and all its successors.

The second situation occurs when both y and z are neighbours of x, but z is the predecessor of x. Again, $\mathcal{L}(y)$ is added to $\mathcal{L}(z)$, but in this case the inverse of $\mathcal{L}(\langle x, y\rangle)$ is added to $\mathcal{L}(\langle z, x\rangle)$, because the edge $\langle x, y\rangle$ was pointing away from x while $\langle z, x\rangle$ points towards it. Again, z inherits the inequalities from y and $\mathcal{L}(\langle x, y\rangle)$ is set to $\emptyset$.

The $\leqslant_r$ rule handles the identification of two root nodes. An example of the whole procedure is given in the lower part of Figure 2. In this case, special care has to be taken to preserve the relations introduced into the completion forest due to role assertions in

$\sqcap$-rule:	if 1. $C_1 \sqcap C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. $\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$ then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{C_1, C_2\}$
$\sqcup$-rule:	if 1. $C_1 \sqcup C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. $\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$ then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{E\}$ for some $E \in \{C_1, C_2\}$
$\exists$-rule:	if 1. $\exists S.C \in \mathcal{L}(x)$, x is not blocked, and 2. x has no S-neighbour y with $C \in \mathcal{L}(y)$ then create a new node y with $\mathcal{L}(\langle x, y\rangle) := \{S\}$ and $\mathcal{L}(y) := \{C\}$
$\forall$-rule:	if 1. $\forall S.C \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. there is an S-neighbour y of x with $C \notin \mathcal{L}(y)$ then $\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{C\}$
$\forall_+$-rule:	if 1. $\forall S.C \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. there is some R with $\mathsf{Trans}(R)$ and $R \,\underline{\circledast}\, S$, 3. there is an R-neighbour y of x with $\forall R.C \notin \mathcal{L}(y)$ then $\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{\forall R.C\}$
choose-rule:	if 1. $(\bowtie\ n\ S\ C) \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. there is an S-neighbour y of x with $\{C, {\sim}C\} \cap \mathcal{L}(y) = \emptyset$ then $\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{E\}$ for some $E \in \{C, {\sim}C\}$
$\geqslant$-rule:	if 1. $\geqslant nS.C \in \mathcal{L}(x)$, x is not blocked, and 2. there are no n S-neighbours $y_1, \ldots, y_n$ such that $C \in \mathcal{L}(y_i)$ and $y_i \neq y_j$ for $1 \leq i < j \leq n$ then create n new nodes $y_1, \ldots, y_n$ with $\mathcal{L}(\langle x, y_i\rangle) = \{S\}$, $\mathcal{L}(y_i) = \{C\}$, and $y_i \neq y_j$ for $1 \leq i < j \leq n$.
$\leqslant$-rule:	if 1. $\leqslant nS.C \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. $\sharp S^{\mathcal{F}}(x, C) > n$, there are S-neighbours y, z of x with not $y \neq z$, y is neither a root node nor an ancestor of z, and $C \in \mathcal{L}(y) \cap \mathcal{L}(z)$, then 1. $\mathcal{L}(z) \longrightarrow \mathcal{L}(z) \cup \mathcal{L}(y)$ and 2. if z is an ancestor of x then $\mathcal{L}(\langle z, x\rangle) \longrightarrow \mathcal{L}(\langle z, x\rangle) \cup \mathsf{Inv}(\mathcal{L}(\langle x, y\rangle))$ else $\mathcal{L}(\langle x, z\rangle) \longrightarrow \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}(\langle x, y\rangle)$ 3. $\mathcal{L}(\langle x, y\rangle) \longrightarrow \emptyset$ 4. Set $u \neq z$ for all u with $u \neq y$
$\leqslant_r$-rule:	if 1. $\leqslant nS.C \in \mathcal{L}(x)$, and 2. $\sharp S^{\mathcal{F}}(x, C) > n$ and there are two S-neighbours y, z of x which are both root nodes, $C \in \mathcal{L}(y) \cap \mathcal{L}(z)$, and not $y \neq z$ then 1. $\mathcal{L}(z) \longrightarrow \mathcal{L}(z) \cup \mathcal{L}(y)$ and 2. For all edges $\langle y, w\rangle$: i. if the edge $\langle z, w\rangle$ does not exist, create it with $\mathcal{L}(\langle z, w\rangle) := \emptyset$ ii. $\mathcal{L}(\langle z, w\rangle) \longrightarrow \mathcal{L}(\langle z, w\rangle) \cup \mathcal{L}(\langle y, w\rangle)$ 3. For all edges $\langle w, y\rangle$: i. if the edge $\langle w, z\rangle$ does not exist, create it with $\mathcal{L}(\langle w, z\rangle) := \emptyset$ ii. $\mathcal{L}(\langle w, z\rangle) \longrightarrow \mathcal{L}(\langle w, z\rangle) \cup \mathcal{L}(\langle w, y\rangle)$ 4. Set $\mathcal{L}(y) := \emptyset$ and remove all edges to/from y. 5. Set $u \neq z$ for all u with $u \neq y$. 6. Set $y \doteq z$.

Fig. 1. The Expansion Rules for $\mathcal{SHIQ}$-Aboxes.

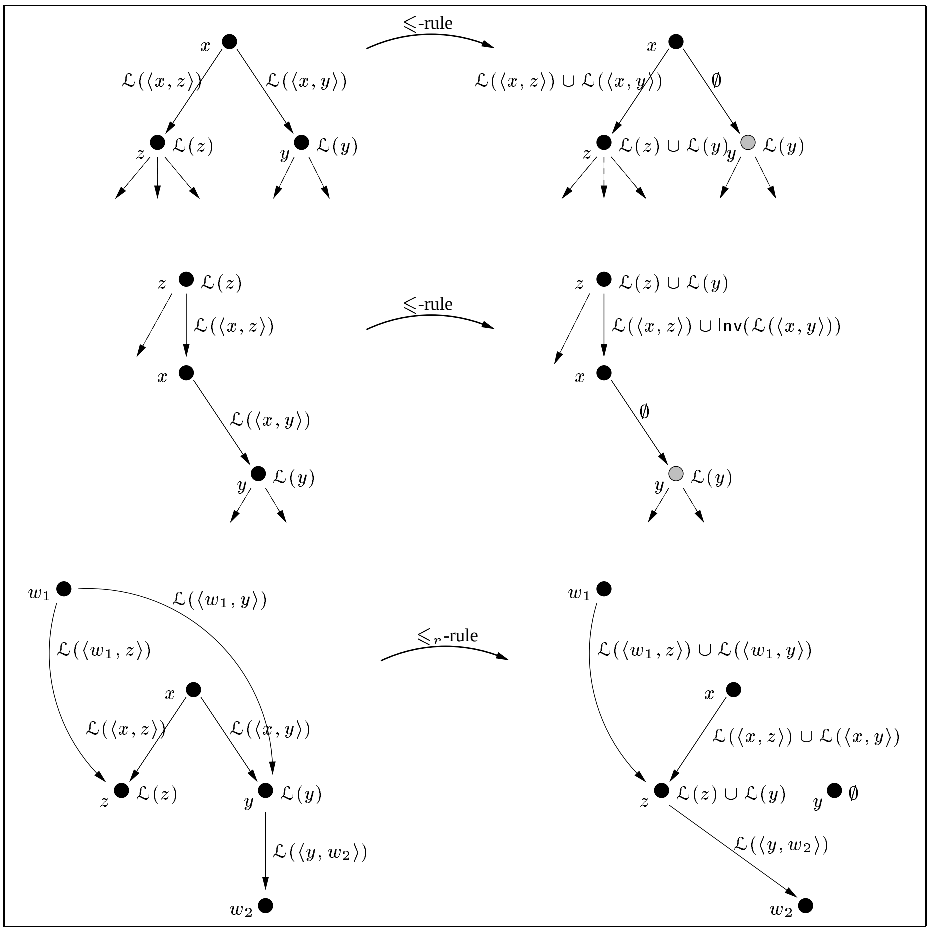

Fig. 2. Effect of the $\leqslant$- and the $\leqslant_r$-rule

the Abox, and to memorise the identification of root nodes (this will be needed in order to construct a tableau from a complete and clash-free completion forest). The $\leqslant_r$ rule includes some additional steps that deal with these issues. Firstly, as well as adding $\mathcal{L}(y)$ to $\mathcal{L}(z)$, the edges (and their respective labels) between y and its neighbours are also added to z. Secondly, $\mathcal{L}(y)$ and all edges going from/to y are removed from the forest. This will not lead to dangling trees, because all neighbours of y became neighbours of z in the previous step. Finally, the identification of y and z is recorded in the $\doteq$ relation.

Lemma 3. *Let $\mathcal{A}$ be a $\mathcal{SHIQ}$-Abox and $\mathcal{R}$ a role hierarchy. The completion algorithm terminates when started for $\mathcal{A}$ and $\mathcal{R}$.*

Proof: Let $m = \sharp\mathsf{clos}(\mathcal{A})$, $n = |\mathbf{R}_\mathcal{A}|$, and $n_{\max} := \max\{n \mid \geqslant nR.C \in \mathsf{clos}(\mathcal{A})\}$. Termination is a consequence of the following properties of the expansion rules:

1. The expansion rules never remove nodes from the forest. The only rules that remove elements from the labels of edges or nodes are the $\leqslant$- and $\leqslant_r$-rule, which sets them to $\emptyset$. If an edge label is set to $\emptyset$ by the $\leqslant$-rule, the node below this edge is blocked and will remain blocked forever. The $\leqslant_r$-rule only sets the label of a root node x to $\emptyset$, and after this, x's label is never changed again since all edges to/from x are removed. Since no root nodes are generated, this removal may only happen a finite number of times, and the new edges generated by the $\leqslant_r$-rule guarantees that the resulting structure is still a completion forest.
2. Nodes are labelled with subsets of $\mathsf{clos}(\mathcal{A})$ and edges with subsets of $R_{\mathcal{A}}$, so there are at most 2^{2mn} different possible labellings for a pair of nodes and an edge. Therefore, if a path p is of length at least 2^{2mn}, the pair-wise blocking condition implies the existence of two nodes x, y on p such that y directly blocks y. Since a path on which nodes are blocked cannot become longer, paths are of length at most 2^{2mn}.
3. Only the $\exists$- or the $\geqslant$-rule generate new nodes, and each generation is triggered by a concept of the form $\exists R.C$ or $\geqslant nR.C$ in $\mathsf{clos}(\mathcal{A})$. Each of these concepts triggers the generation of at most $n_{\max}$ successors y_i: note that if the $\leqslant$- or the $\leqslant_r$-rule subsequently causes $\mathcal{L}(\langle x, y_i\rangle)$ to be changed to $\emptyset$, then x will have some R-neighbour z with $\mathcal{L}(z) \supseteq \mathcal{L}(y)$. This, together with the definition of a clash, implies that the rule application which led to the generation of y_i will not be repeated. Since $\mathsf{clos}(\mathcal{A})$ contains a total of at most m $\exists R.C$, the out-degree of the forest is bounded by $mn_{\max}n$. $\square$

Lemma 4. *Let $\mathcal{A}$ be a $\mathcal{SHIQ}$-Abox and $\mathcal{R}$ a role hierarchy. If the expansion rules can be applied to $\mathcal{A}$ and $\mathcal{R}$ such that they yield a complete and clash-free completion forest, then $\mathcal{A}$ has a tableau w.r.t. $\mathcal{R}$.*

Proof: Let $\mathcal{F}$ be a complete and clash-free completion forest. The definition of a tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{I})$ from $\mathcal{F}$ works as follows. Intuitively, an individual in $\mathbf{S}$ corresponds to a *path* in $\mathcal{F}$ from some root node to some node that is not blocked, and which goes only via non-root nodes.

More precisely, a *path* is a sequence of pairs of nodes of $\mathcal{F}$ of the form $p = [\frac{x_0}{x'_0}, \ldots, \frac{x_n}{x'_n}]$. For such a path we define $\mathsf{Tail}(p) := x_n$ and $\mathsf{Tail}'(p) := x'_n$. With $[p|\frac{x_{n+1}}{x'_{n+1}}]$, we denote the path $[\frac{x_0}{x'_0}, \ldots, \frac{x_n}{x'_n}, \frac{x_{n+1}}{x'_{n+1}}]$. The set $\mathsf{Paths}(\mathcal{F})$ is defined inductively as follows:

- For root nodes x^i_0 of $\mathcal{F}$, $[\frac{x^i_0}{x^i_0}] \in \mathsf{Paths}(\mathcal{F})$, and
- For a path $p \in \mathsf{Paths}(\mathcal{F})$ and a node z in $\mathcal{F}$:
  - if z is a successor of $\mathsf{Tail}(p)$ and z is neither blocked nor a root node, then $[p|\frac{z}{z}] \in \mathsf{Paths}(\mathcal{F})$, or
  - if, for some node y in $\mathcal{F}$, y is a successor of $\mathsf{Tail}(p)$ and z blocks y, then $[p|\frac{z}{y}] \in \mathsf{Paths}(\mathcal{F})$.

Please note that, since root nodes are never blocked, nor are they blocking other nodes, the only place where they occur in a path is in the first place. Moreover, by construction of $\mathsf{Paths}(\mathcal{F})$, if $p \in \mathsf{Paths}(\mathcal{F})$, then $\mathsf{Tail}(p)$ is not blocked, $\mathsf{Tail}(p) = \mathsf{Tail}'(p)$ iff $\mathsf{Tail}'(p)$ is not blocked, and $\mathcal{L}(\mathsf{Tail}(p)) = \mathcal{L}(\mathsf{Tail}'(p))$.

We define a tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{I})$ as follows:

$$
\begin{aligned}
\mathbf{S} &= \mathsf{Paths}(\mathcal{F}) \\
\mathcal{L}(p) &= \mathcal{L}(\mathsf{Tail}(p)) \\
\mathcal{E}(R) &= \{\langle p, [p|\tfrac{x}{x'}]\rangle \in \mathbf{S} \times \mathbf{S} \mid x' \text{ is an } R\text{-successor of } \mathsf{Tail}(p)\} \cup \\
&\quad\ \{\langle [q|\tfrac{x}{x'}], q\rangle \in \mathbf{S} \times \mathbf{S} \mid x' \text{ is an } \mathsf{Inv}(R)\text{-successor of } \mathsf{Tail}(q)\} \cup \\
&\quad\ \{\langle [\tfrac{x}{x}], [\tfrac{y}{y}]\rangle \in \mathbf{S} \times \mathbf{S} \mid x, y \text{ are root nodes, and } y \text{ is an } R\text{-neighbour of } x\} \\
\mathcal{I}(a_i) &= \begin{cases} [\tfrac{x_0^i}{x_0^i}] & \text{if } x_0^i \text{ is a root node in } \mathcal{F} \text{ with } \mathcal{L}(x_0^i) \neq \emptyset \\ [\tfrac{x_0^j}{x_0^j}] & \text{if } \mathcal{L}(x_0^i) = \emptyset, x_0^j \text{ a root node in } \mathcal{F} \text{ with } \mathcal{L}(x_0^j) \neq \emptyset \text{ and } x_0^i \doteq x_0^j \end{cases}
\end{aligned}
$$

Please note that $\mathcal{L}(x) = \emptyset$ implies that x is a root node and that there is another root node y with $\mathcal{L}(y) \neq \emptyset$ and $x \doteq y$. We show that T is a tableau for D.

- T satisfies **(P1)** because $\mathcal{F}$ is clash-free.
- **(P2)** and **(P3)** are satisfied by T because $\mathcal{F}$ is complete.
- For **(P4)**, let $p, q \in \mathbf{S}$ with $\forall R.C \in \mathcal{L}(p)$, $\langle p, q\rangle \in \mathcal{E}(R)$. If $q = [p|\frac{x}{x'}]$, then x' is an R-successor of $\mathsf{Tail}(p)$ and, due to completeness of $\mathcal{F}$, $C \in \mathcal{L}(x') = \mathcal{L}(x) = \mathcal{L}(q)$. If $p = [q|\frac{x}{x'}]$, then x' is an $\mathsf{Inv}(R)$-successor of $\mathsf{Tail}(q)$ and, due to completeness of $\mathcal{F}$, $C \in \mathcal{L}(\mathsf{Tail}(q)) = \mathcal{L}(q)$. If $p = [\frac{x}{x}]$ and $q = [\frac{y}{y}]$ for two root nodes x, x, then y is an R-neighbour of x, and completeness of $\mathcal{F}$ yields $C \in \mathcal{L}(y) = \mathcal{L}(q)$. **(P6)** and **(P11)** hold for similar reasons.
- For **(P5)**, let $\exists R.C \in \mathcal{L}(p)$ and $\mathsf{Tail}(p) = x$. Since x is not blocked and $\mathcal{F}$ complete, x has some R-neighbour y with $C \in \mathcal{L}(y)$.
  - If y is a successor of x, then y can either be a root node or not.
    - If y is not a root node: if y is not blocked, then $q := [p|\frac{y}{y}] \in \mathbf{S}$; if y is blocked by some node z, then $q := [p|\frac{z}{y}] \in \mathbf{S}$.
    - If y is a root node: since y is a successor of x, x is also a root node. This implies $p = [\frac{x}{x}]$ and $q = [\frac{y}{y}] \in \mathbf{S}$.
  - x is an $\mathsf{Inv}(R)$-*successor* of y, then either
    - $p = [q|\frac{x}{x'}]$ with $\mathsf{Tail}(q) = y$.
    - $p = [q|\frac{x}{x'}]$ with $\mathsf{Tail}(q) = u \neq y$. Since x only has one predecessor, u is not the predecessor of x. This implies $x \neq x'$, x blocks x', and u is the predecessor of x' due to the construction of Paths. Together with the definition of the blocking condition, this implies $\mathcal{L}(\langle u, x'\rangle) = \mathcal{L}(\langle y, x\rangle)$ as well as $\mathcal{L}(u) = \mathcal{L}(y)$ due to the blocking condition.
    - $p = [\frac{x}{x}]$ with x being a root node. Hence y is also a root node and $q = [\frac{y}{y}]$.

  In any of these cases, $\langle p, q\rangle \in \mathcal{E}(R)$ and $C \in \mathcal{L}(q)$.
- **(P7)** holds because of the symmetric definition of the mapping $\mathcal{E}$.
- **(P8)** is due to the definition of R-neighbours and R-successor.
- Suppose **(P9)** were not satisfied. Hence there is some $p \in \mathbf{S}$ with $(\leqslant nS.C) \in \mathcal{L}(p)$ and $\sharp S^T(p, C) > n$. We will show that this implies $\sharp S^{\mathcal{F}}(\mathsf{Tail}(p), C) > n$, contradicting either clash-freeness or completeness of $\mathcal{F}$. Let $x := \mathsf{Tail}(p)$ and $P := S^T(p, C)$. We distinguish two cases:

- P contains only paths of the form $[p|\frac{y}{y'}]$ and $[\frac{x_0^{i_\ell}}{x_0^{i_\ell}}]$. Then $\sharp P > n$ is impossible since the function Tail' is injective on P: if we assume that there are two distinct paths $q_1, q_2 \in P$ and $\mathsf{Tail}'(q_1) = \mathsf{Tail}'(q_2) = y'$, then this implies that each q_i is of the form $q_i = [p|\frac{y_i}{y'}]$ or $q_i = [\frac{y'}{y'}]$. From $q_1 \neq q_2$, we have that $q_i = [p|\frac{y_i}{y'}]$ holds for some $i \in \{1, 2\}$. Since root nodes occur only in the beginning of paths and $q_1 \neq q_2$, we have $q_1 = [p|(y_1, y')]$ and $q_2 = [p|(y_2, y')]$. If y' is not blocked, then $y_1 = y' = y_2$, contradicting $q_1 \neq q_2$. If y' is blocked in $\mathcal{F}$, then both y_1 and y_2 block y', which implies $y_1 = y_2$, again a contradiction. Hence Tail' is injective on P and thus $\sharp P = \sharp\, \mathsf{Tail}'(P)$. Moreover, for each $y' \in \mathsf{Tail}'(P)$, y' is an S-successor of x and $C \in \mathcal{L}(y')$. This implies $\sharp S^{\mathcal{F}}(x, C) > n$.
- P contains a path q where $p = [q|\frac{x}{x'}]$. Obviously, P may only contain one such path. As in the previous case, Tail' is an injective function on the set $P' := P \setminus \{q\}$, each $y' \in \mathsf{Tail}'(P')$ is an S-successor of x, and $C \in \mathcal{L}(y')$ for each $y' \in \mathsf{Tail}'(P')$. Let $z := \mathsf{Tail}(q)$. We distinguish two cases:
  * $x = x'$. Hence x is not blocked, and thus x is an $\mathsf{Inv}(S)$-successor of z. Since $\mathsf{Tail}'(P')$ contains only successors of x we have that $z \notin \mathsf{Tail}'(P')$ and, by construction, z is an S-neighbour of x with $C \in \mathcal{L}(z)$.
  * $x \neq x'$. This implies that x' is blocked by x and that x' is an $\mathsf{Inv}(S)$-successor of z. Due to the definition of pairwise-blocking this implies that x is an $\mathsf{Inv}(S)$-successor of some node u with $\mathcal{L}(u) = \mathcal{L}(z)$. Again, $u \notin \mathsf{Tail}'(P')$ and, by construction, u is an S-neighbour of x and $C \in \mathcal{L}(u)$.

– For **(P10)**, let $(\geqslant n S.C) \in \mathcal{L}(p)$. Hence there are n S-neighbours $y_1, \ldots, y_n$ of $x = \mathsf{Tail}(p)$ in $\mathcal{F}$ with $C \in \mathcal{L}(y_i)$. For each y_i there are three possibilities:
  - y_i is an S-successor of x and y_i is not blocked in $\mathcal{F}$. Then $q_i := [p|\frac{y_i}{y_i}]$ or y_i is a root node and $q_i := [\frac{y_i}{y_i}]$ is in $\mathbf{S}$.
  - y_i is an S-successor of x and y_i is blocked in $\mathcal{F}$ by some node z. Then $q_i = [p|\frac{z}{y_i}]$ is in $\mathbf{S}$. Since the same z may block several of the y_js, it is indeed necessary to include y_i explicitly into the path to make them distinct.
  - x is an $\mathsf{Inv}(S)$-successor of y_i. There may be at most one such y_i if x is not a root node. Hence either $p = [q_i|\frac{x}{x'}]$ with $\mathsf{Tail}(q_i) = y_i$, or $p = [\frac{x}{x}]$ and $q_i = [\frac{y_i}{y_i}]$.

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

– **(P12)** is due to the fact that, when the completion algorithm is started for an Abox $\mathcal{A}$, the initial completion forest $\mathcal{F}_\mathcal{A}$ contains, for each individual name a_i occurring in $\mathcal{A}$, a root node x_0^i with $\mathcal{L}(x_0^i) = \{C \in \mathsf{clos}(\mathcal{A}) \mid a_i : C \in \mathcal{A}\}$. The algorithm never blocks root individuals, and, for each root node x_0^i whose label and edges are removed by the $\leqslant_r$-rule, there is another root node x_0^j with $x_0^i \doteq x_0^j$ and $\{C \in \mathsf{clos}(\mathcal{A}) \mid a_i : C \in \mathcal{A}\} \subseteq \mathcal{L}(x_0^j)$. Together with the definition of $\mathcal{I}$, this yields (P12). **(P13)** is satisfied for similar reasons.

– **(P14)** is satisfied because the $\leqslant_r$-rule does not identify two root nodes x_0^i, y_0^i when $x_0^i \neq y_0^i$ holds. □

Lemma 5. *Let $\mathcal{A}$ be a $\mathcal{SHIQ}$-Abox and $\mathcal{R}$ a role hierarchy. If $\mathcal{A}$ has a tableau w.r.t. $\mathcal{R}$, then the expansion rules can be applied to $\mathcal{A}$ and $\mathcal{R}$ such that they yield a complete and clash-free completion forest.*

Proof: Let $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{I})$ be a tableau for $\mathcal{A}$ and $\mathcal{R}$. We use T to trigger the application of the expansion rules such that they yield a completion forest $\mathcal{F}$ that is both complete and clash-free. To this purpose, a function π is used which maps the nodes of $\mathcal{F}$ to elements of $\mathbf{S}$. The mapping π is defined as follows:

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

Obviously, the mapping for the initial completion forest for $\mathcal{A}$ and $\mathcal{R}$ satisfies the following conditions:

$$\left.\begin{array}{l} \mathcal{L}(x) \subseteq \mathcal{L}(\pi(x)), \\ \text{if } y \text{ is an } S\text{-neighbour of } x, \text{ then } \langle \pi(x), \pi(y) \rangle \in \mathcal{E}(S), \text{ and} \\ x \neq y \text{ implies } \pi(x) \neq \pi(y). \end{array}\right\} \qquad (*)$$

It can be shown that the following claim holds:
CLAIM: Let $\mathcal{F}$ be generated by the completion algorithm for $\mathcal{A}$ and $\mathcal{R}$ and let π satisfy $(*)$. If an expansion rule is applicable to $\mathcal{F}$, then this rule can be applied such that it yields a completion forest $\mathcal{F}'$ and a (possibly extended) π that satisfy $(*)$.

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

From Theorem 1, Lemma 2, 3 4, and 5, we thus have the following theorem:

Theorem 2. *The completion algorithm is a decision procedure for the consistency of $\mathcal{SHIQ}$-Aboxes and the satisfiability and subumption of concepts with respect to role hierarchies and terminologies.*

4 Conclusion

We have presented an algorithm for deciding the satisfiability of $\mathcal{SHIQ}$ KBs where the Abox may be non-empty and where the uniqueness of individual names is not assumed but can be asserted in the Abox. This algorithm is of particular interest as it can be used to decide the problem of conjunctive query containment w.r.t. a schema [17].

An implementation of the $\mathcal{SHIQ}$ Tbox satisfiability algorithm is already available in the FaCT system [14], and is able to reason efficiently with Tboxes derived from realistic ER schemas. This suggests that the algorithm presented here could form the basis of a practical decision procedure for the query containment problem. Work is already underway to test this conjecture by extending the FaCT system with an implementation of the new algorithm.

References

1. C. Areces, P. Blackburn, and M. Marx. A road-map on complexity for hybrid logics. In *Proc. of CSL'99*, number 1683 in *LNCS*, pages 307–321 Springer-Verlag, 1999.
2. F. Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In *Proc. of IJCAI-91*, 1991.
3. F. Baader, H.-J. Bürckert, B. Nebel, W. Nutt, and G. Smolka. On the expressivity of feature logics with negation, functional uncertainty, and sort equations. *Journal of Logic, Language and Information*, 2:1–18, 1993.
4. F. Baader, H.-J. Heinsohn, B. Hollunder, J. Muller, B. Nebel, W. Nutt, and H.-J. Profitlich. Terminological knowledge representation: A proposal for a terminological logic. Technical Memo TM-90-04, DFKI, Saarbrücken, Germany, 1991.
5. P. Blackburn and J. Seligman. What are hybrid languages? In *Advances in Modal Logic*, volume 1, pages 41–62. CSLI Publications, Stanford University, 1998.
6. M. Buchheit, F. M. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. *J. of Artificial Intelligence Research*, 1:109–138, 1993.
7. D. Calvanese. Reasoning with inclusion axioms in description logics: Algorithms and complexity. In *Proc. of ECAI'96*, pages 303–307. John Wiley & Sons Ltd., 1996.
8. D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of PODS'98*, pages 149–158. 1998.
9. D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Source integration in data warehousing. In *Proc. of DEXA-98*. IEEE Computer Society Press, 1998.
10. Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Daniele Nardi, and Riccardo Rosati. Description logic framework for information integration. In *Proc. of KR-98*, 1998.
11. G. De Giacomo and F. Massacci. Combining deduction and model checking into tableaux and algorithms for converse-PDL. *Information and Computation*, 1998. To appear.
12. Giuseppe De Giacomo and Maurizio Lenzerini. What's in an aggregate: Foundations for description logics with tuples and sets. In *Proc. of IJCAI-95*, 1995.
13. V. Haarslev and R. Möller. An empirical evaluation of optimization strategies for abox reasoning in expressive description logics. In Lambrix et al. [19], pages 115–119..
14. I. Horrocks. FaCT and iFaCT. In Lambrix et al. [19], pages 133–135.
15. I. Horrocks, A. Rector, and C. Goble. A description logic based schema for the classification of medical data. In *Proc. of the 3rd Workshop KRDB'96*. CEUR, June 1996.
16. I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *Journal of Logic and Computation*, 9(3):385–410, 1999.
17. I. Horrocks, U. Sattler, S. Tessaris, and S. Tobies. Query containment using a DLR ABox. LTCS-Report 99-15, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 1999.
18. 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.
19. P. Lambrix, A. Borgida, M. Lenzerini, R. Möller, and P. Patel-Schneider, editors. *Proc. of the International Workshop on Description Logics (DL'99)*, 1999.
20. E. Mays, R. Weida, R. Dionne, M. Laker, B. White, C. Liang, and F. J. Oles. Scalable and expressive medical terminologies. In *Proc. of the 1996 AMAI Annual Fall Symposium*, 1996.
21. U. Sattler. A concept language extended with different kinds of transitive roles. In *20. Deutsche Jahrestagung für KI*, volume 1137 in *LNAI*. Springer-Verlag, 1996.
22. A. Schaerf. Reasoning with individuals in concept languages. *Data and Knowledge Engineering*, 13(2):141–176, 1994.
23. K. Schild. A correspondence theory for terminological logics: Preliminary report. In J. Mylopoulos, R. Reiter, editors, *Proc. of IJCAI-91*, Sydney, 1991.
24. M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.