

# A Tableaux-Based Algorithm for $\mathcal{SHIQ}$ with Transitive Closure of Roles in Concept and Role Inclusion Axioms

Chan Le Duc<sup>1</sup>, Myriam Lamolle<sup>1</sup>, and Olivier Curé<sup>2</sup>

<sup>1</sup> LIASD Université Paris 8 - IUT de Montreuil, France  
{chan.leduc, myriam.lamolle}@iut.univ-paris8.fr

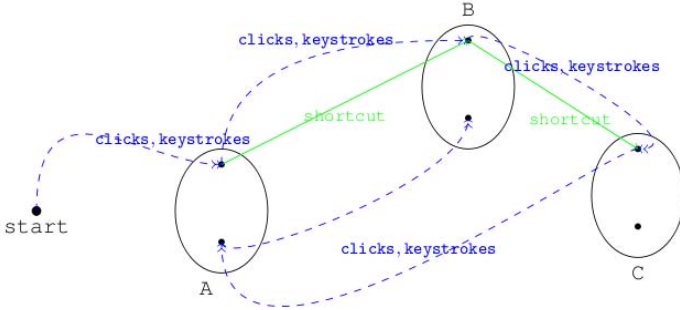
<sup>2</sup> LIGM Université Paris-Est, France  
ocure@univ-mlv.fr

**Abstract.** In this paper, we investigate an extension of the description logic  $\mathcal{SHIQ}$ —a knowledge representation formalism used for the Semantic Web—with transitive closure of roles occurring not only in concept inclusion axioms but also in role inclusion axioms. It was proved that adding transitive closure of roles to  $\mathcal{SHIQ}$  without restriction on role hierarchies may lead to undecidability. We have identified a kind of role inclusion axioms that is responsible for this undecidability and we propose a restriction on these axioms to obtain decidability. Next, we present a tableaux-based algorithm that decides satisfiability of concepts in the new logic.

## 1 Introduction

The ontology language OWL-DL [1] is widely used to formalize semantic resources on the Semantic Web. This language is mainly based on the description logic  $\mathcal{SHOIN}$  which is known to be decidable [2]. Although  $\mathcal{SHOIN}$  is expressive and provides *transitive roles* to model transitivity of relations, we can find several applications in which *the transitive closure of roles*, that is more expressive than transitive roles, is necessary. The difference between transitive roles and the transitive closure of roles is clearer when they are involved in role inclusion axioms. For instance, if we denote by  $R^-$  and  $R^+$  the inverse and transitive closure of a role  $R$  respectively then it is obvious that the concept  $\exists R^+. (C \sqcap \forall R^-. \perp)$  is unsatisfiable w.r.t. an empty TBox and the trivial axiom  $R \sqsubseteq R^+$ . If we now substitute  $R^+$  for a transitive role  $R_t$  such that  $R \sqsubseteq R_t$  (i.e. we substitute each occurrence of  $R^+$  in axioms and concepts for  $R_t$ ) then the concept  $\exists R_t. (C \sqcap \forall R^-. \perp)$  becomes satisfiable. The point is that an instance of  $R^+$  represents a sequence of instances of  $R$  but an instance of  $R_t$  corresponds to a sequence of instances of *itself*.

In several applications, we need to model successive events and relationships between them. An event is something oriented in time, i.e. we can talk about endpoints of an event, or a chronological order of events. When an event of some kind occurs it can trigger an event (or a sequence of events) of another kind. In this situation, it may be suitable to use a role to model an event. If we denote roles *event* and *event'* for two kinds of events then the axiom ( $\text{event} \sqsubseteq \text{event}'$ ) expresses the fact that when an event



**Fig. 1.** Mouse clicks, keystrokes and shortcuts

of the first kind occurs it implies one event or a sequence of events of the second kind. To express “a sequence of events” we can define event' to be transitive. However, the semantics of transitive roles is not sufficient to describe this behaviour since the transitive role event' can represent a sequence of itself but not a sequence of another role. Such behaviours can be found in the following example.

*Example 1.* Let  $S$  be the set of all states of applications running on a computer. We denote by  $A, B, C \subseteq S$  the sets of states of applications  $A, B, C$ , respectively. A user can perform a mouse-click or keystroke to change states. She can type a shortcut (combination of keys) to go from  $A$  to  $B$  or from  $B$  to  $C$ . This action corresponds to a sequence of mouse-clicks or keystrokes. The system's behaviour is depicted in Figure 1. In such a system, users may be interested in the following question: “from the application  $A$ , can one go through the application  $B$  to get directly to the application  $C$  by a mouse-click or keystroke?”.

We now use a description logic with transitive closure of roles to express the constraints as described above. To do this, we use a role *next* to model *mouse clicks* or *keystrokes* and a role *jump* to model *shortcuts* in the following axioms:

- (i)  $\text{start} \sqsubseteq \neg A \sqcap \neg B \sqcap \neg C; X \sqcap Y \sqsubseteq \perp$  with  $X, Y \in \{A, B, C\}$  and  $X \neq Y$ ;
- (ii)  $A \sqsubseteq \exists \text{jump}.B; A \sqsubseteq \exists \text{jump}.C; B \sqsubseteq \exists \text{jump}.C$ ;
- (iii)  $\text{start} \sqsubseteq \forall \text{next}^-. \perp; \text{jump} \sqsubseteq \text{next}^+$ ;

Under some operating systems, users cannot switch directly from an application to a particular one just by one mouse click or keystroke. We can express this constraint with the following axiom:

- (iv)  $C \sqcap \exists \text{next}^-. B \sqsubseteq \perp$ ;

In this case, the concept  $(A \sqcap \exists \text{next}^+. (C \sqcap \exists \text{next}^-. B))$  capturing the question above is unsatisfiable w.r.t. the axioms presented.

Such examples motivate the study of Description Logics (DL) that allow for the transitive closure of roles to occur in both concept and role inclusion axioms. In this work, we introduce a DL that can model systems as described in Example 1 and propose a tableaux-based decision procedure for the concept satisfiability problem in this DL.

To the best of our knowledge, the decidability of  $SHIQ$  with transitive closure of roles, is unknown. [3] and [4] have established decision procedures for concept satisfiability in  $SHI_+$  ( $SHI$  with transitive closure of roles in concept and role inclusion axioms) and  $SHIO_+$  ( $SHI_+$  with nominals). These decision procedures have used *neighborhoods* for representing an individual with its neighbors in a model in order to build completion graphs. In the literature, many decidability results in DLs can be obtained from their counterparts in modal logics ([5], [6]). However, these counterparts do not take into account expressive role inclusion axioms. In particular, [6] has shown decidability of a very expressive DL, so-called  $CATS$ , including  $SHIQ$  with the transitive closure of roles but not allowing it to occur in role inclusion axioms. [6] has pointed out that the complexity of concept subsumption in  $CATS$  is EXPTIME-complete by translating  $CATS$  into the logic Converse PDL in which inference problems are well studied.

Recently, there have been some works in [7] and [8] which have attempted to augment the expressiveness of role inclusion axioms. A decidable logic, namely  $SROIQ$ , resulting from these efforts allows for new role constructors such as composition, disjointness and negation. In addition, [9] has introduced a DL, so-called  $ALCQIb_{reg}^+$ , which can capture  $SRIQ$  ( $SROIQ$  without nominal), and obtained the worst-case complexity (EXPTIME-complete) of the satisfiability problem by using automata-based technique.  $ALCQIb_{reg}^+$  allows for a rich set of operators on roles by which one can simulate role inclusion axioms. However, transitive closures in role inclusion axioms are expressible neither in  $SROIQ$  nor in  $ALCQIb_{reg}^+$ .

Tableaux-based algorithms for expressive DLs such as  $SHIQ$  [10] and  $SROIQ$  [11] result in efficient implementations. This kind of algorithms relies on two structures, the so-called *tableau* and *completion graph*. Roughly speaking, a tableau for a concept represents a model for the concept and it is possibly infinite. A tableau translates satisfiability of all given concept and role inclusion axioms into the satisfiability of constraints imposed *locally* on each individual of the tableau by the semantics of concepts in the individual's label. This feature of tableaux will be called *local satisfiability property*. To check satisfiability of a concept, tableaux-based algorithms try to build a completion graph whose finiteness is ensured by a technique, the so-called *blocking technique*. It provides a termination condition and guarantees soundness and completeness. The underlying idea of the blocking mechanism is to detect "loops" which are repeated pieces of a completion graph. When transitive closure of roles is added to knowledge bases, this blocking technique allows us to lengthen paths through such loops in order to satisfy semantic constraints imposed by transitive closures. The algorithm in [12] for satisfiability in  $ALC_{reg}$  (including the transitive closure of roles and other role operators) introduced a method to deal with loops which can hide unsatisfiable nodes. This method detects on so-called *concept trees*, "good" or "bad" cycles that are similar to those between blocking and blocked nodes on completion trees.

To deal with transitive closure of roles occurring in terms such as  $\exists Q^+.C$ , we have to introduce a new expansion rule to build completion trees such that it can generate a path formed from nodes that are connected by edges whose label contains role  $Q$ . In addition, this rule propagates terms  $\exists Q^+.C$  to each node along with the path before reaching a node whose label includes concept  $C$ . Such a path may go through blocked

and blocking nodes and has an arbitrary length. To handle transitive closures of roles occurring in role inclusion axioms such as  $R \sqsubseteq Q^+$ , we use another new expansion rule that translates satisfaction of such axioms into satisfaction of a term  $\exists Q^+.\Phi$ . From the path generated from  $\exists Q^+.\Phi$ , a cycle can be formed to satisfy the semantic constraint imposed by  $R \sqsubseteq Q^+$ . Since the role  $Q$ , which will be defined to be *simple*, does not occur in number restrictions, the cycle obtained from this method does not violate other semantic constraints.

The contribution of the present paper consists of (i) designing a decidable logic, namely  $\mathcal{SHIQ}_+$ , with a new definition for simple roles and (ii) proposing a tableaux-based algorithm for satisfiability of concepts in  $\mathcal{SHIQ}_+$ .

## 2 The Description Logic $\mathcal{SHIQ}_+$

The logic  $\mathcal{SHIQ}_+$  is an extension of  $\mathcal{SHIQ}$  introduced in [11] by allowing transitive closure of roles to occur in concept and role inclusion axioms. In this section, we present the syntax and semantics of the logic  $\mathcal{SHIQ}_+$ . This includes an extension of the definition of simple roles to  $\mathcal{SHIQ}_+$  and the definition of inference problems that we are interested in. The definitions reuse some notation introduced in [11].

**Definition 1.** Let  $\mathbf{R}$  be a non-empty set of role names. We denote  $\mathbf{R}_1 = \{P^- \mid P \in \mathbf{R}\}$  and  $\mathbf{R}_+ = \{Q^+ \mid Q \in \mathbf{R} \cup \mathbf{R}_1\}$ .

\* The set of  $\mathcal{SHIQ}_+$ -roles is  $\mathbf{R} \cup \mathbf{R}_1 \cup \mathbf{R}_+$ . A role inclusion axiom is of the form  $R \sqsubseteq S$  for two  $\mathcal{SHIQ}_+$ -roles  $R$  and  $S$ . A role hierarchy  $\mathcal{R}$  is a finite set of role inclusion axioms.

\* An interpretation  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  consists of a non-empty set  $\Delta^{\mathcal{I}}$  (domain) and a function  $\cdot^{\mathcal{I}}$  which maps each role name to a subset of  $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$  such that, for  $R \in \mathbf{R}$ ,  $Q^+ \in \mathbf{R}_+$ ,

$$R^{-\mathcal{I}} = \{\langle x, y \rangle \in (\Delta^{\mathcal{I}})^2 \mid \langle y, x \rangle \in R^{\mathcal{I}}\}, (Q^+)^{\mathcal{I}} = \bigcup (Q^n)^{\mathcal{I}} \text{ with } (Q^1)^{\mathcal{I}} = Q^{\mathcal{I}} \text{ and } \\ (Q^n)^{\mathcal{I}} = \{\langle x, y \rangle \in (\Delta^{\mathcal{I}})^2 \mid \exists z \in \Delta^{\mathcal{I}}, \langle x, z \rangle \in (Q^{n-1})^{\mathcal{I}}, \langle z, y \rangle \in Q^{\mathcal{I}}\}.$$

An interpretation  $\mathcal{I}$  satisfies a role hierarchy  $\mathcal{R}$  if  $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$  for each  $R \sqsubseteq S \in \mathcal{R}$ . Such an interpretation is called a model of  $\mathcal{R}$ , denoted by  $\mathcal{I} \models \mathcal{R}$ .

\* To simplify notations for nested inverse roles and transitive closures of roles, we define two functions  $\cdot^{\ominus}$  and  $\cdot^{\oplus}$  as follows:

$$R^{\ominus} = \begin{cases} R^- & \text{if } R \in \mathbf{R}; \\ S & \text{if } R = S^- \text{ and } S \in \mathbf{R}; \\ (S^-)^+ & \text{if } R = S^+, S \in \mathbf{R}; \\ S^+ & \text{if } R = (S^-)^+, S \in \mathbf{R} \end{cases} \quad R^{\oplus} = \begin{cases} R^+ & \text{if } R \in \mathbf{R}; \\ S^+ & \text{if } R = (S^+)^+ \text{ and } S \in \mathbf{R}; \\ (S^-)^+ & \text{if } R = S^- \text{ and } S \in \mathbf{R}; \\ (S^-)^+ & \text{if } R = (S^+)^- \text{ and } S \in \mathbf{R} \end{cases}$$

\* A relation  $\underline{\boxtimes}$  is defined as the transitive-reflexive closure  $\mathcal{R}^+$  of  $\sqsubseteq$  on  $\mathcal{R} \cup \{R^{\ominus} \sqsubseteq S^{\ominus} \mid R \sqsubseteq S \in \mathcal{R}\} \cup \{R^{\oplus} \sqsubseteq S^{\oplus} \mid R \sqsubseteq S \in \mathcal{R}\} \cup \{Q \sqsubseteq Q^{\oplus} \mid Q \in \mathbf{R} \cup \mathbf{R}_1\}$ . We denote  $S \equiv R$  iff  $R \underline{\boxtimes} S$  and  $S \underline{\boxtimes} R$ .

\* A role  $R$  is called *simple* w.r.t.  $\mathcal{R}$  iff (i)  $Q^{\oplus} \underline{\boxtimes} R \notin \mathcal{R}^+$  for each  $Q \in \mathbf{R} \cup \mathbf{R}_1$ , and (ii)  $R' \underline{\boxtimes} R$ ,  $P \underline{\boxtimes} R'^{\oplus} \in \mathcal{R}^+$  implies  $P \underline{\boxtimes} R' \in \mathcal{R}^+$ .

The reason for the introduction of two functions  $\cdot^{\ominus}$  and  $\cdot^{\oplus}$  in Definition 1 is that they avoid using  $R^{--}$  and  $R^{++}$ , moreover it remains a unique nested case  $(R^-)^+$ .

Notice that a transitive role  $S$  (i.e.  $\langle x, y \rangle \in S^{\mathcal{I}}, \langle y, z \rangle \in S^{\mathcal{I}}$  implies  $\langle x, z \rangle \in S^{\mathcal{I}}$  where  $\mathcal{I}$  is an interpretation) can be expressed by using a role axiom  $S^{\oplus} \sqsubseteq S$ . In addition, a role  $R$  which is *simple* according to Definition 1 is simple according to [10] as well. In fact, if  $Q^{\oplus} \sqsubseteq R \notin \mathcal{R}^+$  for each  $Q \in \mathbf{R} \cup \mathbf{R}_I$  then there is no transitive role  $S$  such that  $S \sqsubseteq R \in \mathcal{R}^+$ . Finally, if  $R \sqsubseteq S \in \mathcal{R}^+$  and  $R$  is not simple according to Definition 1 then  $S$  is not simple according to Definition 1.

**Definition 2.** Let  $\mathbf{C}$  be a non-empty set of concept names.

\* The set of  $\mathcal{SHIQ}_+$ -concepts is inductively defined as the smallest set containing all  $C$  in  $\mathbf{C}$ ,  $\top$ ,  $C \sqcap D$ ,  $C \sqcup D$ ,  $\neg C$ ,  $\exists R.C$ ,  $\forall R.C$ ,  $(\leq n S.C)$  and  $(\geq n S.C)$  where  $C$  and  $D$  are  $\mathcal{SHIQ}_+$ -concepts,  $R$  is an  $\mathcal{SHIQ}_+$ -role and  $S$  is a simple role. We denote  $\perp$  for  $\neg \top$ .

\* An interpretation  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  consists of a non-empty set  $\Delta^{\mathcal{I}}$  (domain) and a function  $\cdot^{\mathcal{I}}$  which maps each concept name to a subset of  $\Delta^{\mathcal{I}}$  such that  $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ ,  $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$ ,  $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$ ,  $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$ ,  $(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}, \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$ ,  $(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}, \langle x, y \rangle \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$ ,  $(\geq n S.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \text{card}\{y \in C^{\mathcal{I}} \mid \langle x, y \rangle \in S^{\mathcal{I}}\} \geq n\}$ ,  $(\leq n S.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \text{card}\{y \in C^{\mathcal{I}} \mid \langle x, y \rangle \in S^{\mathcal{I}}\} \leq n\}$  where  $\text{card}\{S\}$  denotes the cardinality of a set  $S$ .

\*  $C \sqsubseteq D$  is called a general concept inclusion (GCI) where  $C, D$  are  $\mathcal{SHIQ}_+$ -concepts (possibly complex), and a finite set of GCIs is called a terminology  $\mathcal{T}$ . An interpretation  $\mathcal{I}$  satisfies a GCI  $C \sqsubseteq D$  if  $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$  and  $\mathcal{I}$  satisfies a terminology  $\mathcal{T}$  if  $\mathcal{I}$  satisfies each GCI in  $\mathcal{T}$ . Such an interpretation is called a model of  $\mathcal{T}$ , denoted by  $\mathcal{I} \models \mathcal{T}$ .

\* A concept  $C$  is called satisfiable w.r.t. a role hierarchy  $\mathcal{R}$  and a terminology  $\mathcal{T}$  iff there is some interpretation  $\mathcal{I}$  such that  $\mathcal{I} \models \mathcal{R}$ ,  $\mathcal{I} \models \mathcal{T}$  and  $C^{\mathcal{I}} \neq \emptyset$ . Such an interpretation is called a model of  $C$  w.r.t.  $\mathcal{R}$  and  $\mathcal{T}$ . A pair  $(\mathcal{T}, \mathcal{R})$  is called a  $\mathcal{SHIQ}_+$  knowledge base and said to be consistent if there is a model  $\mathcal{I}$  of both  $\mathcal{T}$  and  $\mathcal{R}$ , i.e.,  $\mathcal{I} \models \mathcal{T}$  and  $\mathcal{I} \models \mathcal{R}$ .

\* A concept  $D$  subsumes a concept  $C$  w.r.t.  $\mathcal{R}$  and  $\mathcal{T}$ , denoted by  $C \sqsubseteq D$ , if  $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$  holds in each model  $\mathcal{I}$  of  $(\mathcal{T}, \mathcal{R})$ .

Notice that we can reduce subsumption and consistency problems in  $\mathcal{SHIQ}_+$  to concept satisfiability w.r.t. a knowledge base  $(\mathcal{T}, \mathcal{R})$ . Thanks to these reductions, it suffices to study the concept satisfiability problem in  $\mathcal{SHIQ}_+$ .

For the ease of construction, we assume all concepts to be in *negation normal form* (NNF) i.e. negation occurs only in front of concept names. Any  $\mathcal{SHIQ}_+$ -concept can be transformed to an equivalent one in NNF by using DeMorgan's laws and some equivalences as presented in [10]. For a concept  $C$ , we denote the nnf of  $C$  by  $\text{nnf}(C)$  and the nnf of  $\neg C$  by  $\neg C$ .

Let  $D$  be an  $\mathcal{SHIQ}_+$ -concept in NNF. We define  $\text{sub}(D)$  to be the smallest set that contains all sub-concepts of  $D$  including  $D$ . For a knowledge base  $(\mathcal{T}, \mathcal{R})$ ,  $\mathbf{R}_{(\mathcal{T}, \mathcal{R})}$  is used to denote the set of all role names occurring in  $\mathcal{T}, \mathcal{R}$  with their transitive closure and inverses. We denote by  $\mathbf{R}_{(\mathcal{T}, \mathcal{R})}^+$  the set of transitive closure of roles occurring in

$\mathbf{R}_{(\mathcal{T}, \mathcal{R})}$ . Finally, we define sets  $\text{sub}(\mathcal{T}, \mathcal{R})$  and  $\widehat{\text{sub}}(\mathcal{T}, \mathcal{R})$  as follows:

$$\text{sub}(\mathcal{T}, \mathcal{R}) = \bigcup_{C \sqsubseteq D \in \mathcal{T}} \text{sub}(\text{nnf}(\neg C \sqcup D), \mathcal{R}) \text{ where} \quad (1)$$

$$\begin{aligned} \text{sub}(E, \mathcal{R}) = & \text{sub}(E) \cup \{\dot{\neg}C \mid C \in \text{sub}(E)\} \cup \\ & \{\forall S.C \mid (\forall R.C \in \text{sub}(E), S \sqsubseteq R) \text{ or } (\dot{\neg}\forall R.C \in \text{sub}(E), S \sqsubseteq R) \\ & \text{where } S \text{ occurs in } \mathcal{T} \text{ or } \mathcal{R}\} \cup \\ & \{\exists P.\beta \mid \beta \in \{C, \exists P^\oplus.C\}, \exists P^\oplus.C \in \text{sub}(E)\} \end{aligned} \quad (2)$$

$$\Phi_\sigma = \prod_{C \in \sigma \cup \{\dot{\neg}D \mid D \in \text{sub}(\mathcal{T}, \mathcal{R}) \setminus \sigma\}} C \text{ for each } \sigma \subseteq \text{sub}(\mathcal{T}, \mathcal{R}) \quad (3)$$

$$\Omega = \{\Phi_\sigma \mid \sigma \subseteq \text{sub}(\mathcal{T}, \mathcal{R})\} \quad (4)$$

$$\widehat{\text{sub}}(\mathcal{T}, \mathcal{R}) = \Omega \cup \{\alpha.\beta \mid \alpha \in \{\exists P.\exists P^\oplus, \exists P^\oplus, \exists P\}, P^\oplus \in \mathbf{R}_{(\mathcal{T}, \mathcal{R})}^+, \beta \in \Omega\} \quad (5)$$

### 3 Tableaux for $\mathcal{SHIQ}_+$

Basically, a tableau structure is used to represent a model of a  $\mathcal{SHIQ}_+$  knowledge base. Properties in such a tableau definition express semantic constraints resulting directly from the logic constructors in  $\mathcal{SHIQ}_+$ . Considering the tableau definition for  $\mathcal{SHIQ}$  presented in [10], Definition 3 for  $\mathcal{SHIQ}_+$  adopts two additional properties, namely **P8** and **P9**. In particular, **P8** imposes a global constraint on a set of individuals of a tableau. This causes the tableaux to lose *the local satisfiability property*. A tableau has the local satisfiability property if each property of the tableau is related to only one node and its neighbors. This means that, for a graph with a labelling function, checking each node of the graph and its neighbors for each property is sufficient to prove whether this graph is a tableau. The tableau definition for  $\mathcal{SHIQ}$  in [10] has the local satisfiability property although  $\mathcal{SHIQ}$  includes transitive roles. The propagation of value restrictions on transitive roles by  $\forall^+$ -rule (i.e. the rule for  $\forall R.C$  if  $R$  is transitive or includes a transitive role) and the absence of number restrictions on transitive roles help to avoid global properties that impose a constraint on an arbitrary set of individuals in a tableau.

**Definition 3.** Let  $(\mathcal{T}, \mathcal{R})$  be a  $\mathcal{SHIQ}_+$  knowledge base. A tableau  $T$  for a concept  $D$  w.r.t  $(\mathcal{T}, \mathcal{R})$  is defined to be a triplet  $(\mathbf{S}, \mathcal{L}, \mathcal{E})$  such that  $\mathbf{S}$  is a set of individuals,  $\mathcal{L}: \mathbf{S} \rightarrow 2^{\text{sub}(\mathcal{T}, \mathcal{R}) \cup \widehat{\text{sub}}(\mathcal{T}, \mathcal{R})}$  and  $\mathcal{E}: \mathbf{R}_{(\mathcal{T}, \mathcal{R})} \rightarrow 2^{\mathbf{S} \times \mathbf{S}}$ , and there is some individual  $s \in \mathbf{S}$  such that  $D \in \mathcal{L}(s)$ . For all  $s \in \mathbf{S}$ ,  $C, C_1, C_2 \in \text{sub}(\mathcal{T}, \mathcal{R}) \cup \widehat{\text{sub}}(\mathcal{T}, \mathcal{R})$ ,  $R, S \in \mathbf{R}_{(\mathcal{T}, \mathcal{R})}$  and  $Q^\oplus \in \mathbf{R}_{(\mathcal{T}, \mathcal{R})}^+$ ,  $T$  satisfies the following properties:

**P1** If  $C_1 \sqsubseteq C_2 \in \mathcal{T}$  then  $\text{nnf}(\neg C_1 \sqcup C_2) \in \mathcal{L}(s)$ ,

**P2** If  $C \in \mathcal{L}(s)$  then  $\dot{\neg}C \notin \mathcal{L}(s)$ ,

**P3** If  $C_1 \sqcap C_2 \in \mathcal{L}(s)$  then  $C_1 \in \mathcal{L}(s)$  and  $C_2 \in \mathcal{L}(s)$ ,

**P4** If  $C_1 \sqcup C_2 \in \mathcal{L}(s)$  then  $C_1 \in \mathcal{L}(s)$  or  $C_2 \in \mathcal{L}(s)$ ,

**P5** If  $\forall S.C \in \mathcal{L}(s)$  and  $\langle s, t \rangle \in \mathcal{E}(S)$  then  $C \in \mathcal{L}(t)$ ,

**P6** If  $\forall S.C \in \mathcal{L}(s)$ ,  $Q^\oplus \sqsubseteq S$  and  $\langle s, t \rangle \in \mathcal{E}(Q)$  then  $\forall Q^\oplus.C \in \mathcal{L}(t)$ ,

**P7** If  $\exists P.C \in \mathcal{L}(s)$  with  $P \in \mathbf{R}_{(\mathcal{T}, \mathcal{R})} \setminus \mathbf{R}_{(\mathcal{T}, \mathcal{R})}^+$  then there is some  $t \in \mathbf{S}$  such that  $\langle s, t \rangle \in \mathcal{E}(P)$  and  $C \in \mathcal{L}(t)$ ,

**P8** If  $\exists Q^\oplus.C \in \mathcal{L}(s)$  then  $(\exists Q.C \cup \exists Q.\exists Q^\oplus.C) \in \mathcal{L}(s)$ , and there are  $s_1, \dots, s_n \in \mathbf{S}$  such that  $\exists Q.C \in \mathcal{L}(s_0) \cup \mathcal{L}(s_{n-1})$  and  $\langle s_i, s_{i+1} \rangle \in \mathcal{E}(Q)$  with  $0 \leq i < n$ ,  $s_0 = s$  and  $\exists Q^\oplus.C \in \mathcal{L}(s_j)$  for all  $0 \leq j < n$ .

**P9** If  $\langle s, t \rangle \in \mathcal{E}(Q^\oplus)$  then  $\exists Q^\oplus.\Phi_\sigma \in \mathcal{L}(s)$  with  $\sigma = \mathcal{L}(t) \cap \text{sub}(\mathcal{T}, \mathcal{R})$  and

$$\Phi_\sigma = \bigcap_{C \in \sigma \cup \{\neg D \mid D \in \text{sub}(\mathcal{T}, \mathcal{R}) \setminus \sigma\}} C,$$

**P10**  $\langle s, t \rangle \in \mathcal{E}(R)$  iff  $\langle t, s \rangle \in \mathcal{E}(R^\ominus)$ ,

**P11** If  $\langle s, t \rangle \in \mathcal{E}(R)$  and  $R \sqsubseteq S$  then  $\langle s, t \rangle \in \mathcal{E}(S)$ ,

**P12** If  $(\leq nS.C) \in \mathcal{L}(s)$  then  $\text{card}\{S^T(s, C)\} \leq n$  where

$$S^T(s, C) := \{t \in \mathbf{S} \mid \langle s, t \rangle \in \mathcal{E}(S) \wedge C \in \mathcal{L}(t)\},$$

**P13** If  $(\geq nS.C) \in \mathcal{L}(s)$  then  $\text{card}\{S^T(s, C)\} \geq n$ ,

**P14** If  $(\leq nS.C) \in \mathcal{L}(s)$  and  $\langle s, t \rangle \in \mathcal{E}(S)$  then  $C \in \mathcal{L}(t)$  or  $\neg C \in \mathcal{L}(t)$ .

**P8** in Definition 3 expresses not only the semantic constraint imposed by the transitive closure of roles occurring in concepts such as  $\exists Q^\oplus.C$  (i.e. a path including nodes are connected by edges containing  $Q$  and the label of the last node contains  $C$ ) but also the non-determinism of transitive closure of roles (i.e. the term  $\exists Q.C$  may be chosen at any node of such a path to satisfy  $\exists Q^\oplus.C$ ). Additionally, **P8** and **P9** in Definition 3 enable to satisfy each transitive closure  $Q^\oplus$  occurring in the label of an edge  $\langle s, t \rangle$  with simple role  $Q$ . In fact, **P9** makes  $\Phi_\sigma$  belong to the label of a node  $t'$  and  $s$  connected to  $t'$  by edges containing  $Q$  due to **P8**. The definition of  $\Phi_\sigma$  allows  $t'$  to be combined with  $t$  without causing contradiction. Moreover, this combination does not violate number restrictions since  $Q$  is simple. For this reason, the new definition for simple roles presented in Definition 1 is crucial to decidability of  $\mathcal{SHIQ}_+$ .

In addition, **P8** and **P9** defined in this way do not require explicitly cycles to satisfy role inclusion axioms such as  $R \sqsubseteq Q^\oplus$ . This makes it possible to design of tableaux-based algorithm for  $\mathcal{SHIQ}_+$  that aims to build tree-like structure i.e. no cycle is explicitly required to be embedded within this structure. The following lemma affirms that a tableau represents exactly a model for the concept.

**Lemma 1.** *Let  $(\mathcal{T}, \mathcal{R})$  be a  $\mathcal{SHIQ}_+$  knowledge base. Let  $D$  be a  $\mathcal{SHIQ}_+$  concept.  $D$  is satisfiable w.r.t.  $(\mathcal{T}, \mathcal{R})$  iff there is a tableau for  $D$  w.r.t.  $(\mathcal{T}, \mathcal{R})$ .*

For a proof of Lemma 1, we refer the reader to [13].

## 4 A Tableaux-Based Decision Procedure for $\mathcal{SHIQ}_+$

As mentioned, a tableau for a concept represents a model that is possibly infinite. However, the goal of a tableaux-based algorithm is to find a finite structure that must imply a tableau. Conversely, the existence of a tableau can guide us to build such a structure. We introduce in Definition 4 such a finite structure, namely, completion tree.

**Definition 4.** *Let  $(\mathcal{T}, \mathcal{R})$  be a  $\mathcal{SHIQ}_+$  knowledge base. Let  $D$  be a  $\mathcal{SHIQ}_+$  concept. A completion tree for  $D$  and  $(\mathcal{T}, \mathcal{R})$  is a tree  $\mathbf{T} = (V, E, \mathcal{L}, x_{\mathbf{T}}, \neq)$  where*

\*  $V$  is a set of nodes containing a root node  $x_{\mathbf{T}} \in V$ . Each node  $x \in V$  is labelled with a function  $\mathcal{L}$  such that  $\mathcal{L}(x) \subseteq \text{sub}(\mathcal{T}, \mathcal{R}) \cup \widehat{\text{sub}}(\mathcal{T}, \mathcal{R})$ . In addition,  $\neq$  is a symmetric binary relation over  $V$ .

\*  $E$  is a set of edges. Each edge  $\langle x, y \rangle \in E$  is labelled with a function  $\mathcal{L}$  such that  $\mathcal{L}(\langle x, y \rangle) \subseteq \mathbf{R}_{(\mathcal{T}, \mathcal{R})}$ .

\* If  $\langle x, y \rangle \in E$  then  $y$  is called a successor of  $x$ , denoted by  $y \in \text{succ}^1(x)$ , or  $x$  is called the predecessor of  $y$ , denoted by  $x = \text{pred}^1(y)$ . In this case, we say that  $x$  is a neighbor of  $y$  or  $y$  is a neighbor of  $x$ . If  $z \in \text{succ}^n(x)$  (resp.  $z = \text{pred}^n(x)$ ) and  $y$  is a successor of  $z$  (resp.  $y$  is the predecessor of  $z$ ) then  $y \in \text{succ}^{(n+1)}(x)$  (resp.  $y = \text{pred}^{(n+1)}(x)$ ) for all  $n \geq 0$  where  $\text{succ}^0(x) = \{x\}$  and  $\text{pred}^0(x) = x$ .

\* A node  $y$  is called a  $R$ -successor of  $x$ , denoted by  $y \in \text{succ}_R^1(x)$  (resp.  $y$  is called the  $R$ -predecessor of  $x$ , denoted by  $y = \text{pred}_R^1(x)$ ) if there is some role  $R'$  such that  $R' \in \mathcal{L}(\langle x, y \rangle)$  (resp.  $R' \in \mathcal{L}(\langle y, x \rangle)$ ) and  $R' \sqsubseteq R$ . A node  $y$  is called a  $R$ -neighbor of  $x$  if  $y$  is either a  $R$ -successor or  $R$ -predecessor of  $x$ . If  $z$  is a  $R$ -successor of  $y$  (resp.  $z$  is the  $R$ -predecessor of  $y$ ) and  $y \in \text{succ}_R^n(x)$  (resp.  $y = \text{pred}_R^n(x)$ ) then  $z \in \text{succ}_R^{(n+1)}(x)$  (resp.  $z = \text{pred}_R^{(n+1)}(x)$ ) for  $n \geq 0$  with  $\text{succ}_R^0(x) = \{x\}$  and  $x = \text{pred}_R^0(x)$ .

\* For a node  $x$  and a role  $S$ , we define the set  $S^{\mathbf{T}}(x, C)$  of  $x$ 's  $S$ -neighbors as follows:

$$S^{\mathbf{T}}(x, C) = \{y \in V \mid y \text{ is a } S\text{-neighbor of } x \text{ and } C \in \mathcal{L}(x)\}$$

\* A node  $x$  is called blocked by  $y$ , denoted by  $y = \mathbf{b}(x)$ , if there are numbers  $n, m > 0$  and nodes  $x', y, y'$  such that

1.  $x_{\mathbf{T}} = \text{pred}^n(y)$ ,  $y = \text{pred}^m(x)$ , and
2.  $x' = \text{pred}^1(x)$ ,  $y' = \text{pred}^1(y)$ , and
3.  $\mathcal{L}(x) = \mathcal{L}(y)$ ,  $\mathcal{L}(x') = \mathcal{L}(y')$ , and
4.  $\mathcal{L}(\langle x', x \rangle) = \mathcal{L}(\langle y', y \rangle)$ , and
5. if there are  $z, z'$  such that  $z' = \text{pred}^1(z)$ ,  $\text{pred}^i(z') = x_{\mathbf{T}}$ ,  $\mathcal{L}(z) = \mathcal{L}(y)$ ,  $\mathcal{L}(z') = \mathcal{L}(y')$  and  $\mathcal{L}(\langle z', z \rangle) = \mathcal{L}(\langle y', y \rangle)$  then  $n \leq i$ .

\* We define an extended function  $\widehat{\text{succ}}$  from  $\text{succ}$  over  $\mathbf{T}$  as follows:

- if  $x$  has a successor  $y$  (resp.  $x$  has a  $R$ -successor  $y$ ) that is not blocked then  $y \in \widehat{\text{succ}}^1(x)$  (resp.  $y \in \widehat{\text{succ}}_R^1(x)$ ),
- if  $x$  has a successor  $z$  (resp.  $x$  has a  $R$ -successor  $z$ ) that is blocked by  $\mathbf{b}(z)$  then  $\mathbf{b}(z) \in \widehat{\text{succ}}^1(x)$  (resp.  $\mathbf{b}(z) \in \widehat{\text{succ}}_R^1(x)$ ).
- if  $y \in \widehat{\text{succ}}_R^n(x)$  and  $z \in \widehat{\text{succ}}_R^1(y)$  then  $z \in \widehat{\text{succ}}_R^{(n+1)}(x)$  for  $n \geq 0$ .

\* A node  $z$  is called a  $\exists R^{\oplus}.C$ -reachable of  $x$  with  $\exists R^{\oplus}.C \in \mathcal{L}(x)$  if there are  $x_1, \dots, x_{k+n} \in V$  with  $x_{k+n} = z, x_0 = x$  and  $k+n \geq 0$  such that  $x_i = \text{pred}_R^i(x_0)$ ,  $\exists R^{\oplus}.C \in \mathcal{L}(x_i)$  with  $i \in \{0, \dots, k\}$ , and  $x_{j+k} \in \widehat{\text{succ}}_R^j(x_k)$ ,  $\exists R^{\oplus}.C \in \mathcal{L}(x_{j+k})$ ,  $\exists R.C \in \mathcal{L}(x_{(k+n)})$  with  $j \in \{0, \dots, n\}$ .

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

1. There is some node  $x \in V$  such that  $\{A, \neg A\} \subseteq \mathcal{L}(x)$  for some concept name  $A \in \mathbf{C}$ ,



2. *There is some node  $x \in V$  with  $(\leq nS.C) \in \mathcal{L}(x)$  and there are  $(n + 1)$   $S$ -neighbors  $y_1, \dots, y_{n+1}$  of  $x$  such that  $y_i \neq y_j$  and  $C \in \mathcal{L}(x_i)$  for all  $1 \leq i < j \leq (n + 1)$ ,*
3. *There is some node  $x \in V$  with  $\exists R^\oplus.C \in \mathcal{L}(x)$  such that there does not exist any  $\exists R^\oplus.C$ -reachable node  $y$  of  $x$ ,*

Algorithm 2 builds a completion tree for a  $\mathcal{SHIQ}_+$  concept by applying the expansion rules in Figure 2 and 3. The expansion rules in Figure 2 were given in [10]. We introduce two new expansion rules that correspond to **P8** and **P9** in Definition 3.

In comparison with  $\mathcal{SHIQ}$ , there is a new source of non-determinisms that could augment the complexity of an algorithm for satisfiability of concepts in  $\mathcal{SHIQ}_+$ . This source comes from the presence of transitive closure of role in concepts. This means that for each occurrence of a term such as  $\exists Q^\oplus.C$  in the label of a node of a completion tree we have to check the existence of a sequence of edges such that the label of each edge contains  $Q$  and the label of the last node contains  $C$ . The process for checking the existence of paths whose length is arbitrary must be translated into a process that works for a finite structure. To do this, we reuse the blocking condition introduced in [10] and introduce a function  $\widehat{\text{succ}}(x)$  that returns the set of  $x$ 's successors in a completion tree. An infinite path over a completion tree can be defined thanks to this function. The  $\exists_+$ -rule in Figure 3 generates all possible paths. The clash-freeness of the third kind in Definition 4 ensures that a “good” path has to be picked from this set of all possible paths.

The function  $\text{checkReachability}_C^Q(x, d, \mathcal{B})$  depicted in Algorithm 1 represents an algorithm for checking the clash-freeness of the third kind for a completion tree. It returns true iff there exists a  $\exists Q^\oplus.C$ -reachable node of  $x$ . In this function, the parameter  $x$  represents a node of the tree to be checked i.e. there is a term such as  $\exists Q^\oplus.C \in \mathcal{L}(x)$ . The parameter  $d$  indicates the direction to search from  $x$ . Depending on  $d = 1$  or  $d = 0$ , the algorithm goes up to ancestors of  $x$  or goes down to descendants of  $x$  respectively. When the algorithm goes down, it never goes up again. The subset  $\mathcal{B} \subseteq V$  represents the set of all blocked nodes among the nodes that the algorithm has visited. The function  $\text{checkReachability}_C^Q(x, 1, \emptyset)$  would be called for each non-blocked node  $x$  of a completion tree and for each term of the form  $\exists Q^\oplus.C \in \mathcal{L}(x)$ .

**Lemma 2 (Termination).** *Let  $(\mathcal{T}, \mathcal{R})$  be a  $\mathcal{SHIQ}_+$  knowledge base. Let  $D$  be a  $\mathcal{SHIQ}_+$ -concept w.r.t.  $(\mathcal{T}, \mathcal{R})$ . Algorithm 2 terminates.*

*Proof.* The termination of Algorithm 2 is a consequence of the following claims:

1. Applications of rules in Figure 2 and 3 do not remove concepts from the label of nodes. Moreover, applications of rules in Figure 2 and 3 do not remove roles from the label of edges except that they may set the label of edges to an empty set. However, when the label of an edge becomes empty it remains to be empty forever. Therefore, we can compute an upper bound of the completion tree's height from the blocking condition. This upper bound equals  $K = 2^{2m+k}$  where  $m = \text{card}\{\text{sub}(\mathcal{T}, \mathcal{R}) \cup \widehat{\text{sub}}(\mathcal{T}, \mathcal{R})\}$  and  $k$  is the number of roles occurring in  $\mathcal{T}$  and  $\mathcal{R}$  plus their inverse and transitive closure. Moreover, the number of neighbors of any node is bounded by  $M = \sum m_i$  where  $m_i$  occurs in a number restriction term  $(\geq m_i R.C)$  that appears in  $\mathcal{T}$ .

$\sqsubseteq$ -rule:	if $C \sqsubseteq D \in \mathcal{T}$ and $\text{nnf}(\neg C \sqcup D) \notin \mathcal{L}(x)$ then $\mathcal{L}(x) \leftarrow \mathcal{L}(x) \cup \{\text{nnf}(\neg C \sqcup D)\}$
$\sqcap$ -rule:	if $C_1 \sqcap C_2 \in \mathcal{L}(x)$ and $\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$ then $\mathcal{L}(x) \leftarrow \mathcal{L}(x) \cup \{C_1, C_2\}$
$\sqcup$ -rule:	if $C_1 \sqcup C_2 \in \mathcal{L}(x)$ and $\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$ then $\mathcal{L}(x) \leftarrow \mathcal{L}(x) \cup \{C\}$ for some $C \in \{C_1, C_2\}$
$\exists$ -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. $\forall S.C \in \mathcal{L}(x)$ , and 2. there is a $S$ -neighbour $y$ of $x$ such that $C \notin \mathcal{L}(y)$ then $\mathcal{L}(y) \leftarrow \mathcal{L}(y) \cup \{C\}$
$\forall_+$ -rule:	if 1. $\forall S.C \in \mathcal{L}(x)$ , and 2. there is some $Q$ with $Q^{\oplus} \sqsubseteq S$ , and 3. there is an $Q$ -neighbour $y$ of $x$ such that $\forall Q^{\oplus}.C \notin \mathcal{L}(y)$ then $\mathcal{L}(y) \leftarrow \mathcal{L}(y) \cup \{\forall Q^{\oplus}.C\}$
$ch$ -rule:	if 1. $(\leq_n S.C) \in \mathcal{L}(x)$ , and 2. there is an $S$ -neighbour $y$ of $x$ with $\{C, \dot{\neg}C\} \cap \mathcal{L}(y) = \emptyset$ then $\mathcal{L}(y) \leftarrow \mathcal{L}(y) \cup \{E\}$ for some $E \in \{C, \dot{\neg}C\}$
$\geq$ -rule:	if 1. $(\geq_n S.C) \in \mathcal{L}(x)$ and $x$ is not blocked, and 2. there are no $n$ $S$ -neighbors $y_1, \dots, y_n$ 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 $y_1, \dots, y_n$ with $\mathcal{L}(\langle x, y_i \rangle) = \{S\}$ , $\mathcal{L}(y_i) = \{C\}$ , and $y_i \neq y_j$ for $1 \leq i < j \leq n$ .
$\leq$ -rule:	if 1. $(\leq_n S.C) \in \mathcal{L}(x)$ , and 2. $\text{card}\{S^T(x, C)\} > n$ and there are two $S$ -neighbors $y, z$ of $x$ with $C \in \mathcal{L}(y) \cap \mathcal{L}(z)$ , $y$ is not an ancestor of $z$ , and not $y \neq z$ then 1. $\mathcal{L}(z) \leftarrow \mathcal{L}(z) \cup \mathcal{L}(y)$ and $\mathcal{L}(\langle x, y \rangle) \leftarrow \emptyset$ 2. If $z$ is an ancestor of $x$ then $\mathcal{L}(\langle z, x \rangle) \leftarrow \mathcal{L}(\langle z, x \rangle) \cup \{R^{\ominus} \mid R \in \mathcal{L}(\langle x, y \rangle)\}$ else $\mathcal{L}(\langle x, z \rangle) \leftarrow \mathcal{L}(\langle x, z \rangle) \cup \mathcal{L}(\langle x, y \rangle)$ 4. Add $u \neq z$ for all $u$ such that $u \neq y$

**Fig. 2.** Expansion rules for  $\mathcal{SHIQ}$  presented in [10]

$\exists_+$ -rule:	if $\exists S^{\oplus}.C \in \mathcal{L}(x)$ and $(\exists S.C \sqcup \exists S.\exists S^{\oplus}.C) \notin \mathcal{L}(x)$ then $\mathcal{L}(x) \leftarrow \mathcal{L}(x) \cup \{\exists S.C \sqcup \exists S.\exists S^{\oplus}.C\}$
$\oplus$ -rule:	if $x$ has a $P^{\oplus}$ -neighbor $y$ and $\exists P^{\oplus}.\Phi_{\sigma} \notin \mathcal{L}(x)$ with $\sigma = \mathcal{L}(y) \cap \text{sub}(\mathcal{T}, \mathcal{R})$ then $\mathcal{L}(x) = \mathcal{L}(x) \cup \{\exists P^{\oplus}.\Phi_{\sigma}\}$

**Fig. 3.** New expansion rules for  $\mathcal{SHIQ}_+$

```

1 checkReachabilityCQ(x, d, B)
2 if ∃Q.C ∈ L(x) then
3   └ return true;
4 if d = 1 then
5   └ if there is predQ1(x) with ∃Q⊕.C ∈ L(predQ1(x)) then
6     └ checkReachabilityCQ(predQ1(x), 1, B);
7 foreach x' ∈ succQ1(x) such that ∃Q⊕.C ∈ L(x') do
8   └ if ∃Q.C ∈ L(x') then
9     └ return true;
10  └ if x' is not blocked then
11    └ checkReachabilityCQ(x', 0, B);
12  └ else
13    └ if x' ∉ B then
14      └ B = B ∪ {x'};
15      └ checkReachabilityCQ(b(x'), 0, B);
16 return false;

```

**Algorithm 1.**  $\text{checkReachability}_C^Q(x, d, \mathcal{B})$  for checking the existence of a  $\exists Q^\oplus.C$ -reachable node of  $x \in V$  where  $d \in \{1, 0\}$ ,  $\mathcal{B} \subseteq V$ ,  $\exists Q^\oplus.C \in \mathcal{L}(x)$  and  $\mathbf{T} = (V, E, \mathcal{L}, x_{\mathbf{T}}, \neq)$  is a completion tree. As shown in Lemma 2, the complexity of Algorithm 1 is bounded by an exponential function in size of a completion tree. This implies that the complexity of the tableaux algorithm for  $\mathcal{SHIQ}_+$  (Algorithm 2) is bounded by a double exponential function in size of inputs.

**Input** : A  $\mathcal{SHIQ}_+$  knowledge base  $(\mathcal{T}, \mathcal{R})$  and a  $\mathcal{SHIQ}_+$ -concept  $D$

**Output**: Is  $D$  satisfiable w.r.t.  $(\mathcal{T}, \mathcal{R})$ ?

```

1 Let  $\mathbf{T} = (V, E, \mathcal{L}, x_{\mathbf{T}}, \neq)$  be an initial tree such that  $V = \{x_{\mathbf{T}}\}$ ,  $\mathcal{L}(x_{\mathbf{T}}) = \{D\}$ , and
  there is no  $x, y \in V$  such that  $x \neq y$ ;
2 while there is a non-empty set  $S$  of expansion rules in Figure 2 and 3 such that each  $r \in S$ 
  can be applied to a node  $x \in V$  do
3   └ Apply  $r$ ;
4 if there is a clash-free tree  $\mathbf{T}'$  which is built by Line 2 to 2 then
5   └ YES;
6 else
7   └ NO;

```

**Algorithm 2.** Algorithm for building a completion tree in  $\mathcal{SHIQ}_+$

2. Algorithm 1 checks the clash-freeness of the third kind for each  $x \in V$  with  $\exists Q^\oplus.C \in \mathcal{L}(x)$ . To do this, it starts from  $x$  and go up to an ancestor  $x'$  of  $x$ , and go down to a descendant of  $x'$  through the function  $\text{succ}(x')$ . The length of such a path is bounded by  $K \times L$  where  $K$  is given above and  $L$  is the number of blocked nodes of the completion tree. Algorithm 1 may consider all paths which go through all possible blocked nodes. The cardinality of this set is bounded by the number of all permutations of the blocked nodes. Therefore, the complexity of Algorithm 1 is bounded by  $(K \times L) \times L!$ . Algorithm 1 would be called for each occurrence of each term such as  $\exists Q^\oplus.C$  that occurs in each node  $v \in V$ .

**Lemma 3 (Soundness).** *Let  $(\mathcal{T}, \mathcal{R})$  be a  $\text{SHIQ}_+$  knowledge base. Let  $D$  be a  $\text{SHIQ}_+$ -concept w.r.t.  $(\mathcal{T}, \mathcal{R})$ . If Algorithm 2 can build a clash-free completion tree for  $D$  w.r.t.  $(\mathcal{T}, \mathcal{R})$  then there is a tableau for  $D$  w.r.t.  $(\mathcal{T}, \mathcal{R})$ .*

**Proof sketch.** Assume that  $\mathbf{T} = (V, E, \mathcal{L}, x_{\mathbf{T}}, \neq)$  is a clash-free completion tree for  $D$  w.r.t.  $(\mathcal{T}, \mathcal{R})$ . First, we build an extended tree  $\widehat{\mathbf{T}} = (\widehat{V}, \widehat{E}, \mathcal{L}, x_{\widehat{\mathbf{T}}}, \neq)$  from  $\mathbf{T}$  with help of functions  $\widehat{\text{succ}}$  and  $\text{b}(x)$  as follows:

We define  $x_{\widehat{\mathbf{T}}} = x_{\mathbf{T}}$ . If  $x \in \widehat{V}$  and  $x' \in \widehat{\text{succ}}(x)$  then we add to  $\widehat{V}$  a successor  $x'$  of  $x$ . In particular, if  $z, z'$  are two distinct successors of  $x$  such that  $\text{b}(z) = \text{b}(z')$  then there are two distinct nodes that are added to  $\widehat{V}$ . We define a tableau  $T = (\mathbf{S}, \mathcal{L}', \mathcal{E})$  for  $D$  as follows:

- We define  $\mathbf{S} = \widehat{V} = \bigcup_{n \geq 0} \widehat{\text{succ}}^n(x_{\mathbf{T}})$ ,
- For each  $s \in \widehat{V}$  there is a unique  $x_s \in V$  such that  $x_s \in \text{succ}^k(x_{\mathbf{T}})$  and  $s \in \widehat{\text{succ}}^l(x_s)$  with  $n = k + l$ . We define  $\mathcal{L}'(s) = \mathcal{L}(x_s)$ .
- $\mathcal{E}(R) = \mathcal{E}_1(R) \cup \mathcal{E}_2(R)$  where
  - $\mathcal{E}_1(R) = \{\langle s, t \rangle \in \mathbf{S}^2 \mid R \in \mathcal{L}(\langle x_s, x_t \rangle) \vee R^\ominus \in \mathcal{L}(\langle x_t, x_s \rangle)\}$ , and
  - $\mathcal{E}_2(R) = \{\langle s, t \rangle \in \mathbf{S}^2 \mid (R \in \mathcal{L}(\langle x_s, z \rangle) \wedge (\text{b}(z) = x_t)) \vee (R^\ominus \in \mathcal{L}(\langle x_t, z' \rangle) \wedge (\text{b}(z') = x_s))\}$

We now show that  $T$  satisfies P8 in Definition 3, which is the most problematical property. For the other properties, we refer the reader to [13].

Assume that  $s \in \mathbf{S}$  with  $\exists Q^\oplus.C \in \mathcal{L}'(s)$ . Since  $\mathbf{T}$  is clash-free (third kind),  $x_s$  has a  $\exists Q^\oplus.C$ -reachable  $x_n$  i.e. there are  $x_1, \dots, x_n$  such that  $x_{i+1}$  is a  $Q$ -neighbor of  $x_i$  or  $x_{i+1}$  blocks a  $Q$ -successor of  $x_i$  with  $x_s = x_0$  and  $\exists Q.C \in \mathcal{L}(x_n)$ ,  $\exists Q^\oplus.C \in \mathcal{L}(x_i)$  for all  $i \in \{0, \dots, n-1\}$ .

Assume that  $\exists Q.C \in \mathcal{L}'(s)$ . This implies that  $x_s$  has a  $Q$ -neighbor  $y$  such that  $C \in \mathcal{L}(y)$  due to the non-applicability of  $\exists$ -rule. By the definition of  $T$ , there is some  $t \in \mathbf{S}$  with  $t \in \widehat{\text{succ}}^1(s)$  or  $s \in \widehat{\text{succ}}^1(t)$  such that  $\langle s, t \rangle \in \mathcal{E}(Q)$ . Thus, P8 holds.

Assume that  $\exists Q.C \notin \mathcal{L}'(s)$ . According to the definition of  $\exists Q^\oplus.C$ -reachable nodes, there is some  $0 \leq k < n$  such that  $x_k$  is an ancestor of  $x_0$  and  $x_{k+1}$  is a (extended) successor of  $x_k$ . If  $k = 0$  then there are  $s_1, \dots, s_n$  with  $x_{s_i} = x_i$ ,  $s_0 = s$  and  $\langle s_i, s_{i+1} \rangle \in \mathcal{E}(Q)$ ,  $\exists Q.C \in \mathcal{L}'(s_n)$ ,  $\exists Q^\oplus.C \in \mathcal{L}'(s_i)$  for all  $i \in \{0, \dots, n\}$ . Thus, P8 holds. Assume that  $k > 0$ . We define a function  $\widehat{\text{pred}}^j(t)$  as follows:  $\widehat{\text{pred}}^j(t) = x_{\mathbf{T}}$  iff  $t \in \widehat{\text{succ}}^j(x_{\mathbf{T}})$  for all  $t \in \mathbf{S}$ . This implies that for each  $t \in \mathbf{S}$  there is a unique

$j$  such that  $\widehat{\text{pred}}^j(t) = x_{\mathbf{T}}$ . Let  $x_{\mathbf{T}} = \widehat{\text{pred}}^l(s)$ ,  $x_{\mathbf{T}} = \widehat{\text{pred}}^m(x_0) = \text{pred}^m(x_0)$  and  $x_{\mathbf{T}} = \widehat{\text{pred}}^p(x_k) = \text{pred}^p(x_k)$ . We consider the following cases :

Assume  $m = l$ . By the definition of  $T$  there are  $s_0, \dots, s_n \in \mathbf{S}$  such that  $x_{s_i} = x_i$  and  $\langle s_i, s_{i+1} \rangle \in \mathcal{E}(Q)$ ,  $\exists Q^\oplus.C \in \mathcal{L}'(s_i)$  for all  $i \in \{0, \dots, n-1\}$  with  $s_0 = s$  and  $\exists Q.C \in \mathcal{L}'(s_n)$ . Thus, P8 holds.

Assume  $m < l$ . Let  $0 \leq K \leq l$  be the least number such that  $x_{\widehat{\text{pred}}^K(s)}$  has a  $\exists Q^\oplus.C$ -reachable  $y$  with  $y \in \widehat{\text{succ}}^{K'}(x_{\widehat{\text{pred}}^K(s)})$ . We can pick  $K = l - p$  with  $x_{\mathbf{T}} = \widehat{\text{pred}}^p(x_k)$  if there is no such  $K$  with  $K < l - p$ . If  $K = 0$  then  $k = 0$ , which was considered. For  $K > 0$ , we show that  $\langle \widehat{\text{pred}}^{j+1}(s), \widehat{\text{pred}}^j(s) \rangle \in \mathcal{E}(Q)$  and  $\exists Q^\oplus.C \in \mathcal{L}'(\widehat{\text{pred}}^{j+1}(s))$  for all  $j \in \{0, \dots, K-1\}$  (\*\*\*) .

For  $j = 0$ , we have  $\langle s, \widehat{\text{pred}}^1(s) \rangle \in \mathcal{E}(Q)$  and  $\exists Q^\oplus.C \in \mathcal{L}'(\widehat{\text{pred}}^1(s))$ , since  $\langle s, \widehat{\text{pred}}^1(s) \rangle \notin \mathcal{E}(Q)$  or  $\exists Q^\oplus.C \notin \mathcal{L}'(\widehat{\text{pred}}^1(s))$  implies  $K = 0$ .

Assume that  $\exists Q^\oplus.C \in \mathcal{L}'(\widehat{\text{pred}}^j(s))$  with  $j < K$ . Due to the clash-freeness (third kind) of  $\mathbf{T}$ ,  $x_{\widehat{\text{pred}}^j(s)}$  has a  $\exists Q^\oplus.C$ -reachable node  $w$  i.e. there are nodes  $w_1, \dots, w_{n'}$  and some  $k' \geq 0$  such that  $w_{k'}$  is an ancestor of  $x_{\widehat{\text{pred}}^j(s)}$ ,  $w_{k'+1}$  is a (extended) successor of  $w_{k'}$ ,  $w_i$  is a  $Q$ -neighbor of  $w_{i-1}$  and  $\exists Q^\oplus.C \in \mathcal{L}(w_i)$ ,  $\exists Q.C \in \mathcal{L}(w_{n'})$  for all  $i \in \{1, \dots, n'\}$  with  $w_0 = x_{\widehat{\text{pred}}^j(s)}$ . Due to  $j < K$  and  $\mathcal{L}'(\widehat{\text{pred}}^{j+1}(s)) = \mathcal{L}(x_{\widehat{\text{pred}}^{j+1}(s)})$ , we have  $k' > 0$ ,  $\langle \widehat{\text{pred}}^{j+1}(s), \widehat{\text{pred}}^j(s) \rangle \in \mathcal{E}(Q)$  and  $\exists Q^\oplus.C \in \mathcal{L}'(\widehat{\text{pred}}^{j+1}(s))$ . Thus, (\*\*\*) holds.

From (\*\*\*) , it follows that there are  $s_i = \widehat{\text{pred}}^i(s)$  for all  $i \in \{0, \dots, K\}$  and  $s_{K+j} = \widehat{\text{succ}}^j(\widehat{\text{pred}}^K(s))$  for all  $j \in \{1, \dots, K'\}$  such that  $\langle s_h, s_{h+1} \rangle \in \mathcal{E}(Q)$  and  $\exists Q^\oplus.C \in \mathcal{L}'(s_h)$ ,  $\exists Q.C \in \mathcal{L}'(s_{K+K'})$  for all  $h \in \{0, \dots, K+K'\}$  with  $s_0 = s$ . Thus, P8 holds.  $\triangleleft$

**Lemma 4 (Completeness).** *Let  $(\mathcal{T}, \mathcal{R})$  be a  $\mathcal{SHIQ}_+$  knowledge base. Let  $D$  be a  $\mathcal{SHIQ}_+$ -concept w.r.t.  $(\mathcal{T}, \mathcal{R})$ . If there is a tableau for  $D$  w.r.t.  $(\mathcal{T}, \mathcal{R})$  then Algorithm 2 can build a clash-free completion tree for  $D$  w.r.t.  $(\mathcal{T}, \mathcal{R})$ .*

**Proof sketch.** Let  $T = (\mathbf{S}, \mathcal{L}', \mathcal{E})$  be a tableau for  $D$  w.r.t.  $(\mathcal{T}, \mathcal{R})$ . We show that there exists a sequence of expansion rule applications such that it generates a clash-free completion tree  $\mathbf{T} = (V, E, \mathcal{L}, x_{\mathbf{T}}, \neq)$  (\*\*). We define a function  $\pi$  from  $V$  to  $\mathbf{S}$  progressively over the construction of  $\mathbf{T}$  such that it satisfies the following conditions, denoted by (\*):

1.  $\mathcal{L}(x) \subseteq \mathcal{L}'(\pi(x))$  for  $x \in V$ ,
2. if  $y$  is a  $S$ -neighbor of  $x$  in  $\mathbf{T}$  then  $\langle \pi(x), \pi(y) \rangle \in \mathcal{E}(S)$ ,
3.  $x \neq y$  implies  $\pi(x) \neq \pi(y)$ ,
4. if  $\exists Q^\oplus.C \in \mathcal{L}(x)$  and  $\exists Q.C \in \mathcal{L}'(\pi(x))$  then  $\exists Q.C \in \mathcal{L}(x)$  for  $x \in V$ ,

To prove (\*\*), we have to show that (i) we can apply expansion rules such that the conditions in (\*) are preserved, and (ii) if the conditions (\*) are satisfied when constructing a completion tree by expansion rules then the obtained completion tree is

clash-free. Since  $T$  is a tableau there is a node  $s \in \mathbf{S}$  such that  $D \in \mathcal{L}'(s)$ . A node  $x \in V$  is created with  $\pi(x) = s$  and  $\mathcal{L}(x) = \{D\}$ . Applications of  $\sqsubseteq$ -rule,  $\sqcap$ -rule,  $\exists$ -rule,  $\sqcup$ -rule,  $\forall$ -rule,  $\forall_+$ -rule,  $\leq$ -rule,  $\geq$ -rule and *ch*-rule preserve the conditions in (\*). The proof is similar to that in [10]. It is not hard to check that applications of  $\exists_+$ -rule,  $\oplus$ -rule preserve the conditions in (\*) as well.

We now show that if a completion tree  $\mathbf{T}$  can be built with a function  $\pi$  satisfying (\*) then  $\mathbf{T}$  is clash-free.

1. If the condition 1 in (\*) is satisfied then there is no node  $x$  in  $\mathbf{T}$  such that  $A, \dot{\neg}A \in \mathcal{L}(x)$  due to **P2** and the condition 1. That means that  $T$  does not contain a clash of the first kind as described in Definition 4.
2. There is no clash of the second kind in  $\mathbf{T}$  if the conditions 1 to 3 in (\*) are satisfied with **P12**.
3. Assume that  $\exists Q^\oplus.C \in \mathcal{L}(x)$ . Due to the condition 1 in (\*), we have  $\exists Q^\oplus.C \in \mathcal{L}'(\pi(x))$ . According to **P8** and **P4**, there are  $s_1, \dots, s_n \in \mathbf{S}$  such that  $\langle s_i, s_{i+1} \rangle \in \mathcal{E}(Q)$ ,  $\exists Q^\oplus.C \in \mathcal{L}'(s_i)$  and  $\{\exists Q.\exists Q^\oplus.C, \exists Q.C\} \cap \mathcal{L}'(s_i) \neq \emptyset$  for  $i \in \{0, \dots, n-1\}$  with  $s_0 = \pi(x)$ , and  $\exists Q.C \in \mathcal{L}'(s) \cup \mathcal{L}'(s_{n-1})$ .

Assume  $\exists Q.C \in \mathcal{L}'(s)$ . Due to the condition 4 in (\*), we have  $\exists Q.C \in \mathcal{L}(x)$ . This implies that  $\mathbf{T}$  does not have a clash of the third kind.

Assume  $\exists Q.C \notin \mathcal{L}'(s)$  and  $n > 1$ . Without loss of the generality, assume that  $\exists Q.C \notin \mathcal{L}'(s_i)$  for all  $i \in \{0, \dots, n-2\}$  and  $\exists Q.C \in \mathcal{L}'(s_{n-1})$  (otherwise, if there is some  $0 \leq k < n-1$  such that  $\exists Q.C \in \mathcal{L}'(s_k)$  then we pick  $n = k+1$ ).

By applying successively  $\exists$ -rule,  $\exists_+$ -rule and  $\sqcup$ -rule, there are nodes  $x_1, \dots, x_l \in V$  such that  $\pi(x_i) = s_i$ ,  $Q \in \mathcal{L}(\langle x_{i-1}, x_i \rangle)$  and  $\{\exists Q^\oplus.C, \exists Q.\exists Q^\oplus.C\} \subseteq \mathcal{L}(x_i)$  for all  $i \leq l$  with some  $l \leq n-1$ . If  $l = n-1$  then  $x$  has a  $\exists Q^\oplus.C$ -reachable node  $x_l$  such that  $\exists Q.C \in \mathcal{L}(x_l)$  due to  $\exists Q^\oplus.C \in \mathcal{L}(x_l)$ ,  $\exists Q.C \in \mathcal{L}'(\pi(x_l))$  and the condition 4 in (\*). If  $l < n-1$  and  $x_l$  is blocked by  $z$  then we restart from the node  $z$  with  $\exists Q^\oplus.C \in \mathcal{L}(z)$  (since  $\mathcal{L}(z) = \mathcal{L}(x_l)$ ) finding  $x'_1, \dots, x'_l \in V$  which have the same properties as those of  $x_1, \dots, x_l$ . This process can be repeated until finding a node  $w \in V$  such that  $w$  is a  $\exists Q^\oplus.C$ -reachable node of  $x$ .

Therefore,  $\mathbf{T}$  does not have a clash of the third kind.  $\triangleleft$

The following theorem is a consequence of Lemmas 2, 3 and 4.

**Theorem 1.** *Algorithm 2 is a decision procedure for satisfiability of  $\mathcal{SHIQ}_+$ -concepts w.r.t. a  $\mathcal{SHIQ}_+$  knowledge base.*

## 5 Conclusion and Discussion

We have presented a tableaux-based decision procedure for  $\mathcal{SHIQ}_+$  concept satisfiability. In order to define tableaux for  $\mathcal{SHIQ}_+$  we have introduced new properties that allow to represent semantic constraints imposed by transitive closure of roles and to avoid expressing explicitly cycles for role inclusion axioms with transitive closure. These new tableaux properties are translated into new non-deterministic expansion rules which cause the complexity of the tableaux-based algorithm presented in this paper to jump up to double exponential. An open issue consists in investigating whether this

complexity is worst-case optimal. To the best of our knowledge, this problem has not addressed yet. Another future work concerns the extension of our tableaux-based algorithm to  $\mathcal{SHIQ}_+$  with nominals.

**Acknowledgements.** Thanks to Ulrike Sattler for helpful discussions and to the anonymous reviewers for their comments.

## References

1. Patel-Schneider, P., Hayes, P., Horrocks, I.: Owl web ontology language semantics and abstract syntax. In: W3C Recommendation (2004)
2. Tobies, S.: The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *Journal of Artificial Intelligence Research* 12, 199–217 (2000)
3. Le Duc, C.: Decidability of  $\mathcal{SHI}$  with transitive closure of roles. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) *ESWC 2009. LNCS*, vol. 5554, pp. 368–383. Springer, Heidelberg (2009)
4. Le Duc, C., Lamolle, M.: Decidability of description logics with transitive closure of roles. In: *Proceedings of the 23rd International Workshop on Description Logics (DL 2010)*, CEUR-WS.org (2010)
5. De Giacomo, G., Lenzerini, M.: Boosting the correspondence between description logics and propositional dynamic logics. In: *Proceedings of the 12th National Conference on Artificial Intelligence*, pp. 205–212. The MIT Press, Cambridge (1994)
6. De Giacomo, G., Lenzerini, M.: What's in an aggregate: Foundations for description logics with tuples and sets. In: *Proceedings of the Fourteenth International Joint Conference On Intelligence Artificial 1995, IJCAI 1995* (1995)
7. Horrocks, I., Sattler, U.: Decidability of  $\mathcal{SHIQ}$  with complex role inclusion axioms. *Artificial Intelligence* 160, 79–104 (2004)
8. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible  $\mathcal{SROIQ}$ . In: *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*. Springer, Heidelberg (2006)
9. Ortiz, M.: An automata-based algorithm for description logics around  $\mathcal{SRIQ}$ . In: *Proceedings of the fourth Latin American Workshop on Non-Monotonic Reasoning 2008*, CEUR-WS.org (2008)
10. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for expressive description logics. In: Ganzinger, H., McAllester, D., Voronkov, A. (eds.) *LPAR 1999. LNCS*, vol. 1705. Springer, Heidelberg (1999)
11. Horrocks, I., Sattler, U.: A tableau decision procedure for  $\mathcal{SHOIQ}$ . *Journal Of Automated Reasoning* 39(3), 249–276 (2007)
12. Baader, F.: Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In: *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence* (1991)
13. Le Duc, C., Lamolle, M., Curé, O.: A tableaux-based algorithm for  $\mathcal{SHIQ}$  with transitive closure of roles in concept and role inclusion axioms. In: *Technical Report* (2010), <http://www.iut.univ-paris8.fr/files/webfm/recherche/linc/RR201012A.pdf>