# Decidability of $\mathcal{SHI}$ with Transitive Closure of Roles

Chan Le Duc

INRIA Grenoble Rhône-Alpes - LIG
Chan.Leduc@inrialpes.fr

**Abstract.** This paper investigates a Description Logic, namely $\mathcal{SHI}_+$, which extends $\mathcal{SHI}$ by adding transitive closure of roles. The resulting logic $\mathcal{SHI}_+$ allows transitive closure of roles to occur not only in concept inclusion axioms but also in role inclusion axioms. We show that $\mathcal{SHI}_+$ is decidable by devising a terminating, sound and complete algorithm for deciding satisfiability of concepts in $\mathcal{SHI}_+$ with respect to a set of concept and role inclusion axioms.

**Keywords:** OWL, Description Logics, Tableaux, Decidability.

## 1 Introduction

The ontology language OWL-DL [10] is widely used for formalizing resources on Semantic Web. This language is mainly based on the description logic $\mathcal{SHOIN}$ which is known to be decidable [12]. 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. An example in [11] describes two categories of devices as follows: (1) Devices have as their direct part a battery: $\mathsf{Device} \sqcap \exists \mathsf{hasPart}.\mathsf{Battery}$, (2) Devices have at some level of decomposition a battery: $\mathsf{Device} \sqcap \exists \mathsf{hasPart}^+.\mathsf{Battery}$. However, if we now define $\mathsf{hasPart}$ a *transitive role*, the concept $\mathsf{Device} \sqcap \exists \mathsf{hasPart}.\mathsf{Battery}$ does not represent the devices as described above since it does not allow one to distinguish these two categories of devices.

We now consider a more specific example in which we need the transitive closure of roles to occur in role inclusion axioms.

*Example 1.* Let us consider a LOTOS description [9] (producer/consumer):

$$\mathsf{P}[\mathsf{put}, \mathsf{get}] := \mathsf{put}; (\ (\mathsf{get}; \mathbf{stop})\ |||\ \mathsf{P}[\mathsf{put}, \mathsf{get}]\ )$$

Signs ";" and "|||" represent sequential and parallel operators respectively. An execution trace of $\mathsf{P}$ can be generated by executing an action $\mathsf{put}$, and then the parallel block $(\mathsf{get}; \mathbf{stop})\ |||\ \mathsf{P}[\mathsf{put}, \mathsf{get}]$. This block can be started by executing either action $\mathsf{get}$ or a recursive call $\mathsf{P}$. The special action **stop** will be performed if no action of the description can be executed, which is never the case in this example because, according to the semantics of parallel operator, following a $\mathsf{get}$ a recursive call $\mathsf{P}$ can be invoked. This means that execution traces of $\mathsf{P}$

L. Aroyo et al. (Eds.): ESWC 2009, LNCS 5554, pp. 368–383, 2009.

are infinite and their prefixes verify the following property: if $n_1$ and $n_2$ are the number of put and get in a given prefix, $n_1 \geq n_2$ always holds i.e. the intersection of the execution traces with the language $\mathsf{put}^*.\mathsf{get}^*$ is the irregular language $\{\mathsf{put}^{n_1}.\mathsf{get}^{n_2} \mid n_1 \geq n_2\}$.

If one uses a DL to describe these execution traces then roles put, get and disjoint concepts **put**, **put**′, **get** would be needed. A role next subsuming put, get would be used to impose general properties on both of them. In addition, an artificial concept $\mathsf{g}_{\mathsf{start}}$ and role $\overrightarrow{\mathsf{g}_{\mathsf{start}}}$ subsumed by next can be added for the sake of axiom construction.

An expected model of concept $\mathsf{g}_{\mathsf{start}}$ w.r.t. a set of DL axioms is depicted in Fig. 1.

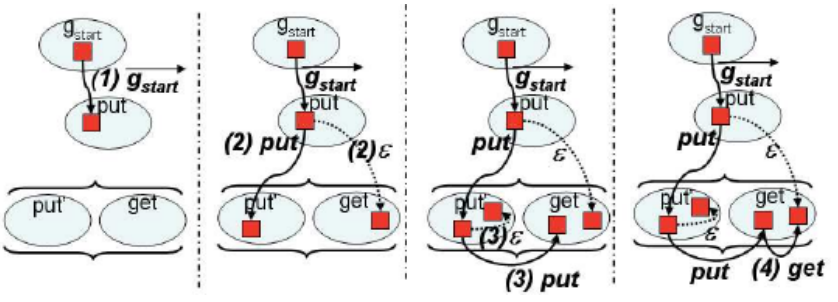


**Fig. 1.** A model of $\mathsf{g}_{\mathsf{start}}$ to represent an execution trace

The essential point is that each execution of the parallel block leads to two executions corresponding to blocks (get; **stop**) and P[put, get]. Since only one action can be executed at one moment, the non-consumed execution must be memorized and will be performed in the future. This characteristic of execution traces is expressed by a role inclusion axiom $\varepsilon \sqsubseteq \mathsf{next}^+$: relation $\varepsilon$ memorizes the non-consumed execution which is initialized by the execution of the parallel block. A sequence of next, i.e. $\mathsf{next}^+$, starts from this point of execution and joints to the memorized non-consumed execution at *some moment in the future*. For instance, the following axioms:

(1) $\varepsilon \sqsubseteq \mathsf{next}^+$; $\mathsf{g}_{\mathsf{start}} \sqsubseteq \exists\overrightarrow{\mathsf{g}_{\mathsf{start}}}.\mathbf{put}$; $\mathsf{next}, \mathsf{next}^-, \varepsilon, \varepsilon^-$: functional
(2) $\mathbf{put} \sqsubseteq (\exists \mathsf{put}.\mathbf{put}' \sqcap \exists\varepsilon.\mathbf{get}) \sqcup (\exists \mathsf{put}.\mathbf{get} \sqcap \exists\varepsilon.\mathbf{put}')$;
(3) $\mathbf{put}' \sqsubseteq (\exists \mathsf{put}.\mathbf{put} \sqcap \exists\varepsilon.\mathbf{get}) \sqcup (\exists \mathsf{put}.\mathbf{get} \sqcap \exists\varepsilon.\mathbf{put})$;
(4) ...

describe a trace as a model of $\mathsf{g}_{\mathsf{start}}$. Since individuals are distinct it holds that the number of instances put from **put** to **put**′ equals to the number of instances $\varepsilon$ from **put** to **get**. Note that if we replace the term $\exists\varepsilon.\mathbf{get}$ in (2) by $\exists\mathsf{next}^+.\mathbf{get}$ then this property no longer holds.

Such examples motivate the study of the logic $\mathcal{SHI}_+$ which is obtained by allowing the transitive closure of roles to occur in both concept and role inclusion axioms in $\mathcal{SHI}$. To the best of our knowledge, the decidability of $\mathcal{SHI}_+$

is unknown. In the literature, many decidability results in Description Logics (DL) can be obtained from their counterparts in modal logics (see [4], [5], [6]). However, these counterparts do not take into account expressive role inclusion axioms. In particular, [5] has shown decidability of a very expressive DL, so-called $\mathcal{CATS}$, including $\mathcal{SHIQ}$ with the transitive closure of roles but not allowing it to occur in role hierarchies.

In addition, tableaux-based algorithms for expressive DLs like $\mathcal{SHIQ}$ [8] and $\mathcal{SHOIQ}$ [7] result in efficient implementations. This kind of algorithms relies on two structures, 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 *local* satisfiability of semantic constraints imposed on each individual of the tableau. This characteristic of tableaux will be called *local satisfiability property*. In turn, a completion graph for a concept is a *finite* representation from which a tableau can be built. To check satisfiability of a concept, tableaux-based algorithms try to build a completion graph whose finiteness is ensured by a technique, so-called *blocking technique*. It provides a termination condition by guaranteeing soundness and completeness. The underlying idea of the blocking mechanism is to detect "loops" which are repeated pieces of a completion graph. The algorithm in [2] for satisfiability in $\mathcal{ALC}_{reg}$ (including the transitive closure of roles and other role operators) introduced a method to deal with loops which can hide unsatisfiable nodes.

The contribution of the present paper is to propose an algorithm for concept satisfiability in $\mathcal{SHI}_+$. Reasoning in $\mathcal{SHI}_+$ is harder than in $\mathcal{SHI}$ because of the two following reasons: (i) adding to a logic transitive closure brings this logic beyond first order logic. This is so because [1] has shown that the transitive closure of roles cannot be expressed in first-order predicate logic, (ii) the presence of the transitive closure of roles in role inclusion axioms, such as $\varepsilon \sqsubseteq a^+$, leads to the loss of the *tree model property*.

## 2 The Logic $\mathcal{SHI}_+$

In this section, we present the syntax and semantics of the logic $\mathcal{SHI}_+$. This includes the definitions of inference problems and how they are interrelated. The definitions reuse notation introduced in [8].

**Definition 1.** *Let* $\mathbf{C}$ *and* $\mathbf{R}$ *be sets of* concept *and* role names*, respectively.*
$*$ *The set of* $\mathcal{SHI}_+$*-roles is* $\mathbf{R}' \cup \{R^- \mid R \in \mathbf{R}'\}$ *with* $\mathbf{R}' = \mathbf{R} \cup \{P^+ \mid P \in \mathbf{R}\}$. *A* role inclusion axiom *is of the form* $R \sqsubseteq S$ *for two* $\mathcal{SHI}_+$*-roles* $R$ *and* $S$. *A* role hierarchy $\mathcal{R}$ *is a finite set of role inclusion axioms. A role* $R \in \mathbf{R}$ *is called* transitive *if* $R^+ \sqsubseteq R \in \mathcal{R}$.
$*$ *Function* Inv *returns the inverse of a role as follows:*

$$\mathsf{Inv}(R) := \begin{cases} R^- & \text{if } R \in \mathbf{R} \cup \{P^+ \mid P \in \mathbf{R}\} \\ S & \text{if } R = S^- \text{ where } S \in \mathbf{R} \cup \{P^+ \mid P \in \mathbf{R}\} \end{cases}$$

∗ *A relation* $\trianglesubseteq$ *is defined as the transitive-reflexive closure of* $\sqsubseteq$ *on* $\mathcal{R} \cup \{\mathsf{Inv}(R) \sqsubseteq \mathsf{Inv}(S) \mid R \sqsubseteq S \in \mathcal{R}\} \cup \{P \sqsubseteq P^+ \mid P \in \mathbf{R}\}$. *We denote* $S \equiv R$ *iff* $R \trianglesubseteq S$ *and* $S \trianglesubseteq R$.

∗ *Function* $\mathsf{Trans}$ *is defined as follows:*

$$\mathsf{Trans}(R) := \begin{cases} true & \text{if there is some } S \text{ such that } S \equiv R \text{ and} \\ & \{S,\ \mathsf{Inv}(S)\} \cap \{\{R \mid R^+ \sqsubseteq R \in \mathcal{R}\} \cup \{P^+ \mid P \in \mathbf{R}\}\} \neq \emptyset \\ false\ , & otherwise \end{cases}$$

∗ *The set of* $\mathcal{SHI}_+$*-concepts is inductively defined as the smallest set containing all* $C$ *in* $\mathbf{C}$, $\top$, $C \sqcap D$, $C \sqcup D$, $\neg C$, $\exists R.C$ *and* $\forall R.C$ *where* $C$ *and* $D$ *are* $\mathcal{SHI}_+$*-concepts and* $R$ *is an* $\mathcal{SHI}_+$*-role. We denote* $\bot$ *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 role* $R$ *to a subset of* $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ *such that, for* $R, P \in \mathbf{R}'$,

$$R^{-\mathcal{I}} = \{\langle x, y\rangle \in (\Delta^{\mathcal{I}})^2 \mid \langle y, x\rangle \in R^{\mathcal{I}}\},$$
$$P^{+\mathcal{I}} = \bigcup_{n>0} (P^n)^{\mathcal{I}} \text{ with } (P^1)^{\mathcal{I}} = P^{\mathcal{I}},\ (P^n)^{\mathcal{I}} = (P^{n-1})^{\mathcal{I}} \circ P^{\mathcal{I}} \text{ and}$$
$$(P^{n-1})^{\mathcal{I}} \circ P^{\mathcal{I}} = \{\langle x, y\rangle \in (\Delta^{\mathcal{I}})^2 \mid \exists z \in \Delta^{\mathcal{I}},\ (\langle x, z\rangle \in (P^{n-1})^{\mathcal{I}}, \langle z, y\rangle \in P^{\mathcal{I}})\}$$

∗ *The function* $\cdot^{\mathcal{I}}$ *maps also each concept to a subset of* $\Delta^{\mathcal{I}}$ *such that*
$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$, $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$, $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$,
$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}},\ (\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$,
$(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}},\ (\langle x, y\rangle \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}})\}$

∗ *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* $\mathcal{R}$, *denoted by* $\mathcal{I} \models \mathcal{R}$.

∗ $C \sqsubseteq D$ *is called a general concept inclusion (GCI) where* $C, D$ *are* $\mathcal{SHI}_+$*-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 an* $\mathcal{SHI}_+$ ontology *and said to be consistent if there is a model of* $(\mathcal{T}, \mathcal{R})$.

∗ *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* $(\mathcal{T}, \mathcal{R})$.

Since negation is allowed in the logic $\mathcal{SHI}_+$, unsatisfiability and subsumption w.r.t. $(\mathcal{T}, \mathcal{R})$ can be reduced each other: $C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable. In addition, we can reduce ontology consistency to concept satisfiability w.r.t. an ontology: $(\mathcal{T}, \mathcal{R})$ is consistent if $A \sqcup \neg A$ is satisfiable w.r.t. $(\mathcal{T}, \mathcal{R})$ for some concept name $A$.

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 $\mathcal{SHI}_+$-concept can be transformed to an equivalent one in NNF by using DeMorgan's

laws and some equivalences as presented in [8]. For a concept $C$, we denote the $\mathsf{nnf}$ of $C$ by $\mathsf{nnf}(C)$ and the $\mathsf{nnf}$ of $\neg C$ by $\neg C$.

Let $D$ be an $\mathcal{SHI}_+$-concept in NNF w.r.t. an ontology $(\mathcal{T}, \mathcal{R})$. We define $\mathsf{sub}(D)$ to be the smallest set that contains all sub-concepts of $D$ including $D$. For an ontology $(\mathcal{T}, \mathcal{R})$, we define the set of all sub-concepts $\mathsf{sub}(\mathcal{T}, \mathcal{R})$ as follows:

$$\begin{aligned}
\mathsf{sub}(\mathcal{T}, \mathcal{R}) &:= \bigcup_{C \sqsubseteq D \in \mathcal{T}} \mathsf{sub}(\mathsf{nnf}(\neg C \sqcup D), \mathcal{R}) \\
\mathsf{sub}(E, \mathcal{R}) &:= \mathsf{sub}(E) \cup \{\neg C \mid \neg C \in \mathsf{sub}(E)\} \ \cup \\
&\quad \{\forall S.C \mid (\forall R.C \in \mathsf{sub}(E), S \sqsubseteq R) \vee (\neg \forall R.C \in \mathsf{sub}(E), S \sqsubseteq R) \\
&\quad \text{and } S \text{ occurs in } \mathcal{T} \text{ or } \mathcal{R}\}
\end{aligned}$$

For the sake of simplicity, for each concept $D$ w.r.t. $(\mathcal{T}, \mathcal{R})$ we denote $\mathsf{sub}(\mathcal{T}, \mathcal{R}, D)$ for $\mathsf{sub}(\mathcal{T}, \mathcal{R}) \cup \mathsf{sub}(D)$, and $\mathbf{R}_{(\mathcal{T}, \mathcal{R}, D)}$ for the set of roles occurring in $\mathcal{T}, \mathcal{R}, D$ with their inverse and transitive closure. If it is clear from the context we use $\mathbf{R}$ instead of $\mathbf{R}_{(\mathcal{T}, \mathcal{R}, D)}$.

## 3 Deciding Satisfiability in $\mathcal{SHI}_+$

In this section, we establish decidability of $\mathcal{SHI}_+$ by devising an algorithm for checking satisfiability of $\mathcal{SHI}_+$ concepts w.r.t. a terminology and role hierarchy.

In our approach, we first extend the tableau definition presented in [8] by adding a *global* property for expressing the transitive closure of roles. This causes the tableaux to lose the local satisfiability property. Next, we define a substructure of graphs, called *neighborhood*, which consists of a node together with its neighbors. Such a neighborhood captures all semantic constraints imposed by the logic constructors of $\mathcal{SHI}$. A tree-like structure obtained by "tiling" neighborhoods together allows us to represent in some way a model for a concept in $\mathcal{SHI}_+$. In fact, we embed in this tree-like structure another structure, called *cyclic path*, to express the transitive closure of roles. Since all expansion rules for $\mathcal{SHI}$ can be translated into building neighborhoods, the algorithm presented in this paper focuses on defining cyclic paths over such a tree-like structure. By this way, the non-determinism resulting from satisfying the transitive closure of roles can be translated into the search in a space of all possible tree-like structures obtained from tiling neighborhoods.

### 3.1 $\mathcal{SHI}_+$-Tableau

Tableau structure is introduced to describe a model of a concept w.r.t. a terminology and role hierarchy. Properties in the tableau definition express semantic constraints resulting directly from the logic constructors in $\mathcal{SHI}_+$.

Considering the tableau definition for $\mathcal{SHIQ}$ presented in [8], the following definition adopts an additional property, namely $\mathsf{P9}$, that imposes a global constraint on a set of individuals of a tableau.

**Definition 2.** *Let $(\mathcal{T}, \mathcal{R})$ be an $\mathcal{SHI}_+$ ontology. 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,*

*$\mathcal{L}$: $\mathbf{S} \to 2^{\mathsf{sub}(\mathcal{T},\mathcal{R},D)}$ and $\mathcal{E}$: $\mathbf{R} \to 2^{\mathbf{S}\times\mathbf{S}}$, and there is some individual $s \in \mathbf{S}$ such that $D \in \mathcal{L}(s)$. For all $s \in \mathbf{S}$, $C, C_1, C_2 \in \mathsf{sub}(\mathcal{T},\mathcal{R},D)$, and $R, S$, $P^+ \in \mathbf{R}$, $T$ satisfies the following properties:*

P1 *if $C_1 \sqsubseteq C_2 \in \mathcal{T}$ and $s \in \mathbf{S}$ then $\mathsf{nnf}(\neg C_1 \sqcup C_2) \in \mathcal{L}(s)$*
P2 *if $C \in \mathcal{L}(s)$, then $\neg C \notin \mathcal{L}(s)$*
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 $\exists S.C \in \mathcal{L}(s)$, there is $t \in \mathbf{S}$ such that $\langle s,t\rangle \in \mathcal{E}(S)$ and $C \in \mathcal{L}(t)$*
P7 *if $\forall S.C \in \mathcal{L}(s)$ and $\langle s,t\rangle \in \mathcal{E}(R)$ for $R \sqsubseteq S$ and $\mathsf{Trans}(R)$, then $\forall R.C \in \mathcal{L}(t)$*
P8 *$\langle s,t\rangle \in \mathcal{E}(R)$ iff $\langle t,s\rangle \in \mathcal{E}(\mathsf{Inv}(R))$*
P9 *if $\langle s,t\rangle \in \mathcal{E}(P^+)$ then either $\langle s,t\rangle \in \mathcal{E}(P)$, or there exist $t_1, \cdots, t_n \in \mathbf{S}$ such that $\langle s,t_1\rangle, \cdots, \langle t_n,t\rangle \in \mathcal{E}(P)$*
P10 *if $\langle s,t\rangle \in \mathcal{E}(R)$, $R \sqsubseteq S$ then $\langle s,t\rangle \in \mathcal{E}(S)$*

Note that when only *transitive roles* are added without transitive closure, the local satisfiability property of tableaux can be preserved by the property P7 which was already introduced in [8]. The property P9 in Def. 2 expresses explicitly a cycle for each transitive closure occurring in the label of an edge $\langle s,t\rangle$. A tableau for a concept represents exactly a model for the concept, that is affirmed by the following lemma.

**Lemma 1.** *An $\mathcal{SHI}_+$-concept $D$ is satisfiable w.r.t. $(\mathcal{T},\mathcal{R})$ iff $D$ has a tableau.*

Lemma 1 is obvious since the properties of tableaux represent directly semantic constraints imposed by the logic constructors in $\mathcal{SHI}_+$.

### 3.2 Neighborhood

Tableau-based algorithms, as presented in [8], use expansion rules to build a completion tree. Applying expansion rules makes all nodes of a completion tree satisfy semantic constraints imposed by concept definitions in the label associated with each node. This means that *local* satisfiability in such completion trees is sufficient to ensure *global* satisfiability. The notion of *neighborhood* introduced in Def. 3 expresses exactly the expansion rules for $\mathcal{SHI}$, consequently, guarantees local satisfiability. Therefore, a completion tree built by a tableau-based algorithm can be considered as set of neighborhoods which are tiled together. In other terms, building a completion tree by applying expansion rules is equivalent to the search of a tiling of neighborhoods.

**Definition 3 (Neighborhood).** *Let $D$ be an $\mathcal{SHI}_+$ concept w.r.t. $(\mathcal{T},\mathcal{R})$. Let $\mathbf{R}$ be the set of roles occurring in $D$ and $\mathcal{T},\mathcal{R}$ together with their inverse and transitive closure. A* neighborhood*, denoted $(v_B, N_B, l)$, for $D$ w.r.t. $(\mathcal{T},\mathcal{R})$ is formed from a* core *node $v_B$, a set of* neighbor *nodes $N_B$, edges $\langle v_B, v\rangle$ with $v \in N_B$ and a labelling function $l$ such that $l(u) \in 2^{\mathsf{sub}(\mathcal{T},\mathcal{R},D)}$ with $u \in \{v_B\} \cup N_B$ and $l(\langle v_B, v\rangle) \in 2^{\mathbf{R}}$ with $v \in N_B$.*

1. *A node $v$ is* valid *w.r.t. $D$ and $(\mathcal{T}, \mathcal{R})$ iff*
   (a) *tbox-rule: If $C \sqsubseteq D \in \mathcal{T}$ then $\mathsf{nnf}(\neg C \sqcup D) \in l(v)$, and*
   (b) *clash-rule: $\{A, \neg A\} \not\subseteq l(v)$ with any concept name $A$, and*
   (c) *$\sqcap$-rule: If $C_1 \sqcap C_2 \in l(v)$ then $\{C_1, C_2\} \subseteq l(v)$, and*
   (d) *$\sqcup$-rule: If $C_1 \sqcup C_2 \in l(v)$ then $\{C_1, C_2\} \cap l(v) \neq \emptyset$.*
2. *A neighborhood $B = (v_B, N_B, l)$ is* valid *iff all nodes $\{v_B\} \cup N_B$ are valid and the following conditions are satisfied:*
   (a) *$\exists$-rule: If $\exists R.C \in l(v_B)$ then there is a neighbor $v \in N_B$ such that $C \in l_B(v)$ and $R \in l(\langle v_B, v\rangle)$;*
   (b) *rbox-rule: For each $v \in N_B$, if $R \in l(\langle v_B, v\rangle)$ and $R \sqsubseteq S$ then $S \in l(\langle v_B, v\rangle)$;*
   (c) *$\forall$-rule: For each $v \in N_B$, if $R \in l(\langle v_B, v\rangle)$ (resp. $R \in \mathsf{Inv}(l(\langle v_B, v\rangle))$) and $\forall R.C \in l(v_B)$ (resp. $\forall R.C \in l(v)$) then $C \in l(v)$ (resp. $C \in l(v_B)$);*
   (d) *$\forall^+$-rule: For each $v \in N_B$, if $R \in l(\langle v_B, v\rangle)$ (resp. $R \in \mathsf{Inv}(l_B(\langle v_B, v\rangle))$), $S \sqsubseteq R$, $\mathsf{Trans}(S)$, $\forall R.D \in l(v_B)$ (resp. $\forall \mathsf{Inv}(R).D \in l(v_B)$) then $\forall S.D \in l_B(v)$ (resp. $\forall \mathsf{Inv}(S).D \in l(v_B)$);*
   (e) *There are at most two nodes $v, v' \in N_B$ such that $l(v) = l(v')$ and $l(\langle v_B, v\rangle) = l(\langle v_B, v'\rangle)$.*

*We denote $\mathbf{B}_{(\mathcal{T},\mathcal{R},D)}$ for a set of all valid neighborhoods for $D$ w.r.t. $(\mathcal{T}, \mathcal{R})$.*

The condition 2e in Def. 3 ensures that any neighborhood has a finite number of neighbors. Moreover, we could replace "at most two nodes" by "at most one node" since $\mathcal{SHI}_+$ does not allow for qualifying number restrictions but the former can simplify some constructions later on (Def. 5 and Rem. 1).

A valid neighborhood as presented in Def. 3 captures all necessary information related to an individual of a tableau. When a global property like $\mathsf{P9}$ is added to tableaux, it imposes just a relationship on a set of individuals of the tableau but each individual is still characterized by a neighborhood which is built from its neighbors and itself. For this reason, neighborhoods can be still used to tile a completion tree for $\mathcal{SHI}_+$ without taking care of expansion rules for $\mathcal{SHI}$. In other terms, the neighbourhood notion expresses the local satisfiability in a sufficient way for being used in a global context.

**Lemma 2.** *Let $D$ be an $\mathcal{SHI}_+$ concept w.r.t. $(\mathcal{T}, \mathcal{R})$. Let $(v_B, N_B, l), (v_{B'}, N_{B'}, l)$ be two valid neighborhoods with $l(v_B) = l(v_{B'})$. If there is $v \in N_B$ such that there does not exist any $v' \in N_{B'}$ satisfying $l(v') = l(v)$ and $l(\langle v_B, v\rangle) = l(\langle v_{B'}, v'\rangle)$ then the neighborhood $(v_{B'}, N_{B'} \cup \{u\}, l)$ is valid where $l(u) = l(v)$ and $l(\langle v_B, u\rangle) = l(\langle v_B, v\rangle)$.*

This lemma holds due to the facts that (i) a valid neighbor in a valid neighborhood $B$ is also a valid neighbor in another valid neighborhood $B'$ if the labels of two core nodes of $B$ and $B'$ are identical, (ii) since $\mathcal{SHI}_+$ does not allow for qualifying number restrictions hence Def. 3 has no restriction on the number of neighbors of a core node.

### 3.3 Completion Tree with Cyclic Paths

As discussed in [3], the blocking technique fails in treating DLs with the transitive closure of roles. It works correctly only if satisfiability of a node in completion tree can be decided from its neighbors and itself i.e. *local* satisfiability must be sufficient for such completion graphs. However, the presence of the transitive closure of roles makes satisfiability of a node depend on further nodes which can be arbitrarily far from it.

More precisely, satisfying the transitive closure $P^+$ in an edge $\langle x, y\rangle$ (i.e. $P^+ \in L(\langle x, y\rangle)$) is related to a set of nodes on a path rather than a node with its neighbors i.e. it imposes a semantic constraint on a set of nodes $x, x_1, \cdots, x_n, y$ such that they are connected together by $P$-edges. In general, satisfying the transitive closure is quite non-deterministic since the semantic constraint can lead to apply to an *arbitrary* number of nodes. In addition, the presence of the transitive closure of roles in a role hierarchy makes this difficulty worse. For instance, if $P \sqsubseteq Q^+, Q \sqsubseteq S^+$ are axioms in a role hierarchy then each $Q$-edge generated for satisfying $Q^+$ may lead to generate an arbitrary number of $S$-edges for satisfying $S^+$.

The most common way for dealing with a new logic constructor is to add a new expansion rule for satisfying the semantic constraint imposed by the new constructor. Such an expansion rule for the transitive closure of roles must:

1. find or create a set of $P$-edges forming a path for each occurrence of $P^+$ in the label of edges,
2. deal with non-deterministic behaviours of the expansion rule resulting from the semantics of the transitive closure of roles,
3. enable to control the expansion of completion trees by a new blocking technique which has to take into account the fact that satisfying the transitive closure of a role may add an arbitrary number of new transitive closures to be satisfied.

To avoid these difficulties, our approach does not aim to directly extend the construction of completion trees by using a new expansion rule, but to translate this construction into selecting a "good" normalization tree, namely *completion tree with cyclic paths*, from a finite set of completion trees without taking into account the semantic constraint imposed by the transitive closure of roles. The process of selecting a "good" normalization tree is guided by finding in this tree a *cyclic path* for each occurrence of the transitive closure of a role.

Summing up, a completion tree with cyclic paths will be built in two stages. The first one consists of tiling valid neighborhoods together such that two neighborhoods are tiled if they have *compatible* neighbors. This stage yields a *normalization tree* as described in Def. 4. The second stage given in Def. 5 deals with the transitive closure of roles by defining cyclic paths over normalization trees.

**Definition 4 (Normalization tree).** *Let $D$ be an $\mathcal{SHI}_+$ concept w.r.t. $(\mathcal{T}, \mathcal{R})$. Let $\mathbf{B}_{(\mathcal{T},\mathcal{R},D)}$ be the set of all valid neighborhoods for $D$ w.r.t. $(\mathcal{T}, \mathcal{R})$. A* normalization tree $\mathbf{T} = (V, E, L)$ *for $D$ w.r.t. $(\mathcal{T}, \mathcal{R})$ is built from $\mathbf{B}_{(\mathcal{T},\mathcal{R},D)}$ as follows:*

- *A root node $x_0$ of* $\mathbf{T}$ *is built from a valid neighborhood* $(v_0, N_0, l) \in \mathbf{B}_{(\mathcal{T},\mathcal{R},D)}$ *such that* $L(x_0) = l(v_0)$ *with* $D \in l(v_0)$*. Additionally, for each* $v \in N_0$ *a successor* $x$ *of* $x_0$ *is added to* $V$ *with* $L(x) = l(v)$ *and* $L(\langle x_0, x\rangle) = l(\langle v_0, v\rangle)$*,*
- *For each node* $x \in V$ *with its predecessor* $x'$*,*
  - *If there are ancestors* $y, y'$ *of* $x'$ *such that* $y'$ *is the predecessor of* $y$ *and* $L(y) = L(x), L(y') = L(x')$*,* $L(\langle y', y\rangle) = L(\langle x', x\rangle)$*, then* $x$ *is* blocked *by* $y$*. In this case,* $x$ *is a leaf node;*
  - *Otherwise, we find a neighborhood* $B_x = (v_{B_x}, N_{B_x}, l)$ *from* $\mathbf{B}_{(\mathcal{T},\mathcal{R},D)}$ *such that* $l(v_{B_x}) = L(x)$*,* $l(v) = L(x')$*,* $\mathsf{Inv}(\ l(\langle v_{B_x}, v\rangle)\ ) = L(\langle x', x\rangle)$ *for some* $v \in N_{B_x}$*, and build a successor* $y$ *for each* $u \in N_{B_x} \setminus \{v\}$ *such that* $L(y) = l(u)$ *and* $L(\langle x, y\rangle) = l(\langle v_{B_x}, u\rangle)$*.*

*We say a node* $x$ *is a* $R$*-successor of* $x' \in V$ *if* $R \in L(\langle x', x\rangle)$*. A node* $x$ *is called a* $R$*-neighbor of* $x'$ *if* $x$ *is a* $R$*-successor of* $x'$ *or* $x'$ *is a* $\mathsf{Inv}(R)$*-successor of* $x$*.*

Note that the construction of normalization trees uses the blocking technique for termination condition. The following definition embeds cyclic paths into normalization trees for satisfying the semantic constraint imposed by the transitive closure of roles.

**Definition 5 (Completion tree with cyclic paths).** *Let* $D$ *be an* $\mathcal{SHI}_+$ *concept w.r.t.* $(\mathcal{T}, \mathcal{R})$*. Let* $\mathbf{T} = (V, E, L)$ *be a normalization tree for* $D$ *w.r.t.* $(\mathcal{T}, \mathcal{R})$*.*

- *A path, denoted* $\varphi = \langle x_0, \cdots, x_k, \cdots, x_{n+1}\rangle$*, is formed from nodes* $x_i \in V$ *if*
  - $x_i$ *is not blocked for all* $i \in \{0, \cdots, n+1\}$*;*
  - $x_{i+1}$ *is a successor of* $x_i$ *or blocks a successor of* $x_i$ *for all* $k \le i \le n$*;*
  - $x_{i-1}$ *is a successor of* $x_i$ *or blocks a successor of* $x_i$ *for all* $1 \le i \le k$*.*
- *A path* $\langle x_0, \cdots, x_{n+1}\rangle$ *is called* non-duplicated *if there do not exist* $i, j \in \{0, \cdots, n\}$ *with* $j > i$ *such that* $L'(\langle x_i, x_{i+1}\rangle) = L'(\langle x_j, x_{j+1}\rangle)$ *and* $L(x_i) = L(x_j), L(x_{i+1}) = L(x_{j+1})$*;*
- *A path* $\langle x_0, \cdots, x_k, \cdots, x_{n+1}\rangle$ *is* cyclic *if* $L(\langle x_1, x_0\rangle) = \mathsf{Inv}(\ L(\langle x_n, x_{n+1}\rangle)\ )$ *and* $L(x_0) = L(x_n), L(x_1) = L(x_{n+1})$*.*

$\mathbf{T} = (V, E, L)$ *is called a* completion tree with cyclic paths *if for each* $\langle u, v\rangle \in E$ *such that* $Q^+ \in L(\langle u, v\rangle)$ *and* $Q \notin L(\langle u, v\rangle)$ *with* $Q \in \mathbf{R} \cup \{\mathsf{Inv}(P) \mid P \in \mathbf{R}\}$ *there exists a cyclic non-duplicated path* $\varphi = \langle x_0, \cdots, x_k, \cdots, x_{n+1}\rangle$ *which satisfies:*

- $u = x_k$*, and* $x_{k-1} = v$ *if* $v$ *is not blocked or* $x_{k-1} = z$ *if* $z$ *blocks* $v$*;*
- $\mathsf{Inv}(Q) \in L'(\langle x_{k-i}, x_{k-i-1}\rangle)$ *for all* $1 \le i \le k-1$*;*
- $Q \in L'(\langle x_i, x_{i+1}\rangle)$ *for all* $k \le i \le n$*.*

*where* $L'(\langle x, y\rangle) = L(\langle x, y\rangle)$ *if* $y$ *is a successor of* $x$*,* $L'(\langle x, y\rangle) = \mathsf{Inv}(\ L(\langle y, x\rangle)\ )$ *if* $x$ *is a successor of* $y$*, and* $L'(\langle x, y\rangle) = L(\langle x, z\rangle)$ *if* $y$ *blocks a successor* $z$ *of* $x$*.*
*In this case,* $\varphi$ *is called a cyclic* $Q$*-path and denoted by* $\varphi_{Q\langle u,v\rangle}$*.*

By its name, we mean that each cyclic path $\langle x_0, \cdots, x_{n+1}\rangle$ becomes a cycle if $x_1$ and $x_0$ are respectively pasted to $x_{n+1}$ and $x_n$.

*Remark 1.* From Def. 5, each node $x_i$ of a cyclic non-duplicated path $\varphi_{Q\langle u,v\rangle} = \langle x_0, \cdots, x_k, \cdots, x_{n+1}\rangle$ with $x_k = u$ can be reached through a path which only goes *down* either from $u$ to a blocked node, or from a blocking to blocked node. This ensures that for any occurrence of $\langle u, v\rangle$ with $Q^+ \in L(\langle u, v\rangle)$ in a possibly infinite path crossing blocked, blocking nodes, we can define a cyclic $Q$-path in a tableau for satisfying this occurrence. This property guarantees the soundness of Alg. 1. For the completeness, when building a cyclic $Q$-path as down-going one from a $Q$-path in a tableau, we need to add a successor which is identical to the predecessor of a node. This is allowed by Def. 3 which accepts a neighborhood with two identical neighbors.

A completion tree with cyclic paths encapsulates the following notions: neighborhood, blocking condition and cyclic path. The first one captures the semantics of all logic constructors except for the transitive closure of roles. The second one which was introduced in [8] is crucial for obtaining a finite representation of a possibly infinite model. The third one represents the transitive closure of roles.

**Lemma 3 (Soundness and completeness).** *Let $D$ be an $\mathcal{SHI}_+$-concept. Let $\mathcal{T}$ and $\mathcal{R}$ be a terminology and role hierarchy.*

1. *If there exists a completion tree with cyclic paths for $D$ w.r.t. $(\mathcal{T}, \mathcal{R})$ then $D$ has a tableau.*
2. *If $D$ has a tableau w.r.t. $(\mathcal{T}, \mathcal{R})$ then there exists a completion tree with cyclic paths.*

To prove the soundness (1.), we define a "quasi-tableau" for $D$ from a set of Paths in a given completion tree with cyclic paths. This technique is already used in [8]. Next, in order to introduce cycles to the quasi-tableau from cyclic paths of the completion tree, we use a function which performs a contraction of the set of paths by unifying the ending points of cyclic paths Paths. Since qualifying number restrictions are not allowed in $\mathcal{SHI}_+$, such a contraction function preserves all properties of the quasi-tableau. For the completeness (2.), we define directly neighborhoods from individuals of a given tableau and build a normalization tree from them. Next, cyclic paths are embedded into the normalization tree by devising non-duplicated paths from finite cycles for the transitive closure of roles in the tableau. A more complete proof of Lem. 3 can be found in Appendix.

From the construction of completion trees with cyclic paths according to Def. 5 and Lem. 3 we can devise immediately Alg. 1 for concept satisfiability in $\mathcal{SHI}_+$.

**Lemma 4 (Termination).** *Algorithm 1 terminates.*

Termination of Alg. 1 is a consequence of the following facts: (i) the number of neighborhoods is bounded, (ii) the size of normalization trees which are tiled from neighbourhoods is bounded.

Alg. 1 is highly complex since it is not a goal-directed decision procedure at all. Such an exhaustive behaviour is very different from that of tableaux-based algorithms in which the construction of a completion tree is inherited from step to step. In Alg. 1, when a normalization tree cannot satisfy an occurrence of the

**Input** : Concept $D$, terminology $\mathcal{T}$ and role hierarchy $\mathcal{R}$
**Output**: IsSatisfiable($D$)

```
foreach Normalization tree T = (V, E, L) do
    if For each ⟨x, y⟩ ∈ E with Q⁺ ∈ L(⟨x, y⟩), Q ∉ L(⟨x, y⟩), T has a φ_{Q⟨x,y⟩}
    then
        return true;
return false;
```

**Algorithm 1.** Deciding concept satisfiability in $\mathcal{SHI}_+$

transitive closure of a role (after satisfying others), a new normalization tree will be picked and embedding cyclic paths into this one has to restart. The following theorem is a direct consequence of Lem. 3 and 4.

**Theorem 1.** *Algorithm 1 decides the satisfiability of $\mathcal{SHI}_+$-concepts w.r.t. a terminology and role hierarchy.*

## 4 Conclusion

We have proved decidability of the concept satisfiability in $\mathcal{SHI}_+$ by providing a sound and complete algorithm. The establishment of the algorithm relies on the neighborhood notion which is an abstraction of the local satisfiability property of tableaux. This abstraction enables us to encapsulate all semantic constraints imposed by the logic constructors in $\mathcal{SHI}$, and thus to deal with the transitive closure of roles independently from the other constructors.

This work is a first step toward an empirical algorithm whose behaviour is more goal-directed i.e. the construction of a completion tree would be refined along with satisfying the transitive closure of roles, e.g., the non-impacted parts of the tree when rebuilding it would be reused.

**Acknowledgements.** This work has been partially supported by the European integrated project NeOn (IST-2005-027595). Thanks to Sophie Coudert and Ludovic Apvrille for the motivating example, to Jérôme Euzenat, Marie-Christine Rousset and the anonymous reviewers for their comments.

## References

1. Aho, A.V., Ullman, J.D.: Universality of data retrieval languages. In: Proceedings of the 6th of ACM on Principles of Programming Language (1979)
2. 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)
3. Baader, F., Sattler, U.: Tableau algorithms for description logics. In: Dyckhoff, R. (ed.) TABLEAUX 2000. LNCS(LNAI), vol. 1847, p. 118. Springer, Heidelberg (2000)

4. 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 (1994)
5. 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)
6. de Giacomo, G., Massacci, F.: Combining deduction and model checking into tableaux and algorithms for converse-PDL. Information and Computation (1) (1998)
7. Horrocks, I., Sattler, U.: A tableau decision procedure for $\mathcal{SHOIQ}$. Journal Of Automated Reasoning 39(3), 249–276 (2007)
8. 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)
9. ISO. Standard ISO 8807: LOTOS, a formal description technique based on temporal ordering of observational behaviour (1988)
10. Patel-Schneider, P., Hayes, P., Horrocks, I.: OWL web ontology language semantics and abstract syntax. W3C Recommendation (2004)
11. Sattler, U.: A concept language extended with different kinds of transitive roles. In: Proceedings of the 20th German Annual Conf. on Artificial Intelligence (KI 2001), vol. 1137, pp. 199–204. Springer, Heidelberg (2001)
12. Tobies, S.: The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. Journal of Artificial Intelligence Research 12, 199–217 (2000)

## Appendix

**Proof of soundness** (Lem. 3)**.** Let $\mathbf{T} = (V, E, L)$ be a completion tree with cyclic paths for $D$ w.r.t. $\mathcal{T}$ and $\mathcal{R}$. A path is of the form $p = [(x_0, x'_0), \cdots, (x_n, x'_n)]$ where $x_i, x'_i \in V$. For such a path we define $\mathsf{Tail}(p) = x_n$, $\mathsf{Tail}'(p) = x'_n$. We denote $[p|(x_{n+1}, x'_{n+1})]$ for the path $[(x_0, x'_0), \cdots, (x_n, x'_n), (x_{n+1}, x'_{n+1})]$. The set $\mathsf{Paths}(\mathbf{T})$ is defined as follows:

1. For the root $v_0$ we define $[(v_0, v_0)] \in \mathsf{Paths}(\mathbf{T})$, and
2. For a path $p \in \mathsf{Paths}(\mathbf{T})$ and a node $v' \in V$,
   (a) If $\langle \mathsf{Tail}(p), v' \rangle \in E$, $v'$ is not a blocked node then $[p|(v', v')] \in \mathsf{Paths}(\mathbf{T})$,
   (b) If $\langle \mathsf{Tail}(p), v' \rangle \in E$ and $v'$ is blocked by $z$ then $[p|(z, v')] \in \mathsf{Paths}(\mathbf{T})$.

From the definition of $\mathsf{Paths}$, for $p \in \mathsf{Paths}(\mathbf{T})$ and $p = [q|(v, v')]$ we have $\mathcal{L}(v) = \mathcal{L}(v')$, and $v \neq v'$ iff $v'$ is blocked by $v$.

We define a quasi-tableau $T' = (\mathbf{S}', \mathcal{L}', \mathcal{E}')$ as follows:

$$
\begin{aligned}
\mathbf{S}' &= \mathsf{Paths}(\mathbf{T}) \\
\mathcal{L}'(p) &= \mathcal{L}'(\mathsf{Tail}(p)) \\
\mathcal{E}'(R) &= \{ \langle p, q \rangle \in \mathsf{Paths}^2(\mathbf{T}) \mid \\
&\qquad 1.\ q = [p|(v, v')],\ R \in L(\langle \mathsf{Tail}(p), v' \rangle), \text{ or} \\
&\qquad 2.\ p = [q|(v, v')] \text{ and } \mathsf{Inv}(R) \in L(\langle \mathsf{Tail}(q), v' \rangle) \ \}
\end{aligned}
$$

From $T' = (\mathbf{S}', \mathcal{L}', \mathcal{E}')$, we denote $\mathcal{L}'(\langle s,t\rangle) = \{R \mid \langle s,t\rangle \in \mathcal{E}(R)\}$. We now build a tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ from this quasi-tableau $T' = (\mathbf{S}', \mathcal{L}', \mathcal{E}')$ with help of function $\pi : \mathbf{S}' \to \mathbf{S}$ that is inductively defined on the length of paths from the root $v_0$. For $p_0, \cdots, p_{n+1} \in \mathbf{S}'$ such that there is a cyclic path $\langle x_0, \cdots, x_{n+1}\rangle$ with $\mathsf{Tail}(p_i) = x_i$, the function $\pi$ merges $p_{n+1}$ with $p_1$, and $p_n$ with $p_0$.

1. $[(v_0, v_0)] \in \mathbf{S}'$ and $\pi([(v_0, v_0)]) = [(v_0, v_0)]$ where $v_0$ is the root of $\mathbf{T}$. Assume that $\pi(p)$ are define for all $p \in \mathbf{S}'$ such that $|p| \leq K$ with $|[(v_0, v_0)]| = 0$ and $|p| = |q| + 1$, $p = [q|(v, v')]$.
2. For each $p \in \mathbf{S}'$ with $\pi(p) = q$, $|p| = K$ and for each $p' = [p|(v, v')] \in \mathbf{S}'$,
   (a) If there is a cyclic $Q$-path $\langle x_0, \cdots, x_k, \cdots, x_{n+1}\rangle$ where $p_0, \cdots, p_k, \cdots, p_{n+1} \in \mathbf{S}'$, $\mathsf{Tail}(p_k) = x_k$ and $p_{i-1} = [p_i|(x_i, x'_i)]$ for all $1 \leq i \leq k$, $p_{j+1} = [p_j|(x_j, x'_j)]$ for all $k \leq j \leq n$, such that $p = p_1, p' = p_0$ or $p = p_n, p' = p_{n+1}$ then
      – If $p = p_1, p' = p_0$ we define $\pi(p_{n+1}) = \pi(p_1) = q$, $\pi(p') = \pi(p_n)$ and $\mathcal{L}(\langle q, \pi(p')\rangle) = \mathcal{L}'(\langle p, p_0\rangle)$, $\mathcal{L}(\langle \pi(p_n), \pi(p_{n+1})\rangle) = \mathcal{L}'(\langle p_n, p_{n+1}\rangle)$,
      – If $p = p_n, p' = p_{n+1}$ we define $\pi(p_0) = \pi(p_n) = q$, $\pi(p') = \pi(p_1)$ and $\mathcal{L}(\langle q, \pi(p_{n+1})\rangle) = \mathcal{L}'(\langle p, p_{n+1}\rangle)$, $\mathcal{L}(\langle \pi(p_1), \pi(p_0)\rangle) = \mathcal{L}'(\langle p_1, p_0\rangle)$.
   (b) Otherwise, we define $q' \in \mathbf{S}$ such that $\pi(p') = q'$, $\mathcal{L}(q') = \mathcal{L}'(p')$, $\mathcal{L}(\langle q, q'\rangle) = \mathcal{L}'(\langle p, p'\rangle)$.

We can define $\mathcal{E}$ from $\mathcal{L}$ as follows: $\mathcal{E}(R) = \{\langle s,t\rangle \in \mathbf{S} \times \mathbf{S} \mid R \in \mathcal{L}(\langle s,t\rangle)\}$ for all $R \in \mathbf{R}$.

The function $\pi$ is well defined since (i) $\mathcal{L}'(p_0) = \mathcal{L}'(p_n) = \mathcal{L}(\pi(p_0)) = \mathcal{L}(\pi(p_n))$, $\mathcal{L}'(p_1) = \mathcal{L}'(p_{n+1}) = \mathcal{L}(\pi(p_1)) = \mathcal{L}(\pi(p_{n+1}))$ by the definition of cyclic $Q$-paths, and (ii) $\mathcal{L}(\langle \pi(p_1), \pi(p_0)\rangle) = \mathcal{L}'(\langle p_1, p_0\rangle)$ with $p_0 = [p_1|(x, x')]$ and $\mathcal{L}(\langle \pi(p_n), \pi(p_{n+1})\rangle) = \mathcal{L}'(\langle p_n, p_{n+1}\rangle)$ with $p_{n+1} = [p_n|(y, y')]$ i.e. the construction of $\pi$ never changes the label of nodes and edges defined previously. However, it may merge some nodes.

*Claim.* 4. For all $p, q \in \mathbf{S}'$ with $q = [p|(x, x')]$ it holds that $\mathcal{L}(\pi(p)) = \mathcal{L}'(p), \mathcal{L}(\pi(q)) = \mathcal{L}'(q)$ and $\mathcal{L}(\langle \pi(p), \pi(q)\rangle) = \mathcal{L}'(\langle p, q\rangle)$.

Claim 4. asserts that $\pi$ preserves all label of nodes and edges.

*Proof of Claim 4.* We prove the claim by induction on the length of $p \in \mathbf{S}'$. If $|p| = 0$ then Claim 4. is verified. Assume that Claim 4. is verified for all $p$ with $|p| \leq m$. Let $q = [p|(x, x')]$ with $\langle \mathsf{Tail}(p), x\rangle \in E$. By the induction hypothesis, we have $\mathcal{L}(\pi(p)) = \mathcal{L}'(p)$.

According to the definition of $\pi$, we consider the two following cases:

– There does not exist any cyclic $Q$-path $\varphi = \langle x_0, \cdots, x_k, \cdots, x_{n+1}\rangle$ such that $p = p_1, q = p_0$ with $\mathsf{Tail}(p_1) = x_1, \mathsf{Tail}(q) = x_0$ or $p = p_n, q = p_{n+1}$ with $\mathsf{Tail}(p_n) = x_n, \mathsf{Tail}(p_{n+1}) = x_{n+1}$. From the definition of $\pi$, we have $\mathcal{L}(\pi(q)) = \mathcal{L}'(q)$ and $\mathcal{L}(\langle \pi(p), \pi(q)\rangle) = \mathcal{L}'(\langle p, q\rangle)$.
– Otherwise, assume that $p = p_1, q = p_0$ with $\mathsf{Tail}(p_1) = x_1, \mathsf{Tail}(q) = x_0$ (if $p = p_n, q = p_{n+1}$ with $\mathsf{Tail}(p_n) = x_n, \mathsf{Tail}(p_{n+1}) = x_{n+1}$ it will be similarly treated). By the definition of $\pi$, we have $\pi(q) = \pi(p_n)$, $\mathcal{L}(\langle \pi(p), \pi(q)\rangle) = \mathcal{L}'(\langle p, q\rangle)$, and thus, $\mathcal{L}(\pi(q)) = \mathcal{L}'(\pi(p_n))$ and $\mathcal{L}(\langle \pi(p), \pi(q)\rangle) = \mathcal{L}'(\langle p, q\rangle)$. ■

*Claim.* 5

*1. For all cyclic $Q$-path $\varphi = \langle x_0, \cdots, x_k, \cdots, x_{n+1}\rangle$, there are $p_i \in \mathbf{S}'$ with $\mathsf{Tail}(p_i) = x_i$ for all $i \in \{0, \cdots, n+1\}$ such that $\mathcal{L}(\langle \pi(p_i), \pi(p_{i+1})\rangle) = \mathcal{L}'(\langle p_i, p_{i+1}\rangle)$ for all $i \in \{0, \cdots, n\}$, and $\pi(p_0) = \pi(p_n), \pi(p_{n+1}) = \pi(p_1)$.*

*2. For all $\pi(p), \pi(q)$ with $p, q \in \mathbf{S}'$ such that $Q^+ \in \mathcal{L}(\langle \pi(p), \pi(q)\rangle)$ and $Q \notin \mathcal{L}(\langle \pi(p), \pi(q)\rangle)$, there are $p_0, \cdots, p_k, \cdots, p_{n+1} \in \mathbf{S}'$ with $\pi(p_0) = \pi(p_n), \pi(p_{n+1}) = \pi(p_1)$ such that $p_k = p, p_{k-1} = q$ and $\mathsf{Inv}(Q) \in \mathcal{L}(\langle \pi(p_i), \pi(p_{i-1})\rangle)$ for all $i \in \{k-1, \cdots, 1\}$, $Q \in \mathcal{L}(\langle \pi(p_i), \pi(p_{i+1})\rangle)$ for all $j \in \{k, \cdots, n\}$.*

Item (1.) asserts that $\pi$ transforms all cyclic paths in $\mathbf{S}'$ into cycles in $\pi(\mathbf{S}')$. Item (2.) asserts that each occurrence $Q^+$ in the label of an edge has a cycle formed of $Q$-edges.

*Proof of Claim 5.*

1. This is directly implied from the definition of $\pi$ and Claim 4.
2. Let $\pi(p), \pi(q) \in \pi(\mathbf{S})$ such that $Q^+ \in \mathcal{L}(\langle \pi(p), \pi(q)\rangle)$ and $Q \notin \mathcal{L}(\langle \pi(p), \pi(q)\rangle)$. From Claim 4. we have $Q^+ \in \mathcal{L}(\langle p, q\rangle)$ and $Q \notin \mathcal{L}(\langle p, q\rangle)$, and thus, $Q^+ \in L(\langle \mathsf{Tail}(p), \mathsf{Tail}(q)\rangle)$ and $Q \notin L(\langle \mathsf{Tail}(p), \mathsf{Tail}(q)\rangle)$.
   By the definition of cyclic $Q$-paths, there is a cyclic $Q$-path $\langle x_0, \cdots, x_k, \cdots, x_{n+1}\rangle$ such that $\mathsf{Tail}(p) = x_k, \mathsf{Tail}(q) = x_{k-1}$. In addition, by the definition of cyclic paths in Def. 5 we have $x_{k-1}, x_{k+1}$ are successors of $x_k$ or blocking nodes of a successor of of $x_k$. Thus, from the definition of the quasi-tableau we have $p_0, \cdots, p_{n+1} \in \mathbf{S}'$ with $p = p_k, q = p_{k-1}$, $\mathsf{Tail}(p_i) = x_i$ for all $i \in \{0, \cdots, n+1\}$ and $\mathsf{Inv}(Q) \in \mathcal{L}'(\langle p_i, p_{i-1}\rangle)$ for all $i \in \{1, \cdots, k-1\}$, and $Q \in \mathcal{L}'(\langle p_j, p_{j+1}\rangle)$ for all $j \in \{k+1, \cdots, n\}$.
   From the item .1 it holds that $\pi(x_0) = \pi(x_n), \pi(x_{n+1}) = \pi(x_1)$. Moreover, by Claim 4. we obtain $\mathsf{Inv}(Q) \in \mathcal{L}(\langle \pi(p_i), \pi(p_{i-1})\rangle)$ for all $i \in \{1, \cdots, k-1\}$, $Q \in \mathcal{L}(\langle \pi(p_i), \pi(p_{i+1})\rangle)$ for all $i \in \{k, \cdots, n\}$. ■

To show that $T$ is a tableau, we have to prove that $T$ satisfies all the properties from Def. 2. The property P9 is a direct consequence of the claims. For instance, we check the property P6.

Assume $\exists R.C \in \mathcal{L}(\pi(p))$ and $v = \mathsf{Tail}(p)$. We show that there exists $\pi(q) \in \mathbf{S}$ such that $C \in \mathcal{L}(\pi(q))$ and $\langle \pi(p), \pi(q)\rangle \in \mathcal{E}(R)$. According to Claim 4. it suffices to show that there exists $q \in \mathbf{S}'$ such that $C \in \mathcal{L}'(q)$ and $\langle p, q\rangle \in \mathcal{E}(R)$. By the construction of neighborhoods and Def. 3 there exists some neighbor $v' \in N$ in the corresponding neighborhood $(v, N, l)$ such that $C \in l(v')$ and $R \in l(\langle v, v'\rangle)$. The definition of $\mathcal{E}'$ implies $R \in L(\langle v, v'\rangle) = l(\langle v, v'\rangle)$ or $\mathsf{Inv}(R) \in L(\langle v', v\rangle) = l(\langle v, v'\rangle)$ i.e. $v'$ can be a $R$-successor of $v$ or $v$ is a $R$-successor of $v'$ in the completion tree. We consider the following cases.

(i) Assume that $R \in L(\langle v, v'\rangle)$. If $v'$ is not blocked, this is trivial since there is $q = [p|(v', v')]$, $\langle p, q\rangle \in \mathcal{E}(R)$ and $C \in \mathcal{L}(q)$. Assume that $q = [p|(z, v')] \in \mathbf{S}$ and $v'$ is blocked by $z$. Since $v'$ is a $R$-successor of $v$, by the definition of $\mathcal{E}$, we have $\langle p, q\rangle \in \mathcal{E}(R)$. Furthermore, we have $C \in \mathcal{L}(q) = L(z) = L(v')$.

(ii) Assume $\mathsf{Inv}(R) \in L(\langle v', v\rangle)$. If $p = [q|(v, v)] \in \mathbf{S}$ i.e. $v$ is not blocked, this is trivial since $\langle q, p\rangle \in \mathcal{E}(R)$ and $C \in \mathcal{L}(q) = L(v')$. Note that $v$ is never

blocked since $v = \mathsf{Tail}(p)$. Therefore, assume that $p = [q|(v, x)] \in \mathbf{S}$ such that $v$ blocks $x$. By the blocking condition we have $L(v') = L(\mathsf{Tail}(q))$ and $L(\langle \mathsf{Tail}(q), x\rangle) = L(\langle v', v\rangle)$. This implies that $\mathsf{Inv}(R) \in L(\langle \mathsf{Tail}(q), x\rangle)$. By the definition of $\mathcal{E}$, we have $\langle q, p\rangle \in \mathcal{E}(R)$ and $C \in \mathcal{L}(q) = L(v')$.

**Proof of completeness**(Lem. 3)**.** Assume that $\mathcal{SHI}_+$-concept $D$ has a tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$. A normalization tree $\mathbf{T} = (V, E, L)$ can be inductively built from $T$ together with a function $\pi$ from $V$ to $\mathbf{S}$. This construction is quite intuitive since we can define a valid neighborhood from each individual $s \in \mathbf{S}$ as follows:

- We define $l(v) := \mathcal{L}(s)$. $v$ is valid since any node whose label is included in the label of a node in the tableau $T$ is always valid.
- Let $S'(s) \subseteq \mathbf{S}$ such that $s' \in S'(s)$ iff $\mathcal{L}(\langle s, s'\rangle) \neq \emptyset$ where $\mathcal{L}(\langle s, s'\rangle) := \{R \mid \langle s, s'\rangle \in \mathcal{E}(R)$ for some $R \in \mathbf{R}_{(\mathcal{T},\mathcal{R},D)}\}$.
  Let $S(s) \subseteq S'(s)$ such that for each $\mathscr{C} \in 2^{\mathsf{sub}(\mathcal{T},\mathcal{R},D)}$ and $\mathscr{R} \in 2^{\mathbf{R}}$ if there is a $t \in S'(s)$ with $\mathcal{L}(t) = \mathscr{C}$ and $\mathcal{L}(\langle s, t\rangle) = \mathscr{R}$ then there is a unique node $t' \in S(s)$ with $\mathcal{L}(t') = \mathcal{L}(t)$ and $\mathcal{L}(\langle s, t\rangle) = \mathcal{L}(\langle s, t'\rangle)$. This implies that $S(s)$ is finite.
- For each $t \in S(s)$ we define a node $u \in N_0$ such that $l(u) = \mathcal{L}(t)$ and $l(\langle v, u\rangle) = \mathcal{L}(\langle s, s'\rangle)$. From the construction, $(v, N_0, l)$ is valid.

A normalization tree $\mathbf{T} = (V, E, L)$ will be obtained by tiling neighborhoods built from connected individuals started at $s_0 \in \mathbf{S}$ with $D \in \mathcal{L}(s_0)$ and a function $\pi$ from $V$ to $\mathbf{S}$. Note that if $u, v$ are neighbors in $\mathbf{T}$ then $\pi(u), \pi(v)$ are also neighbors in $T$. The blocking condition ensures this construction terminates.

We now build cyclic paths for the transitive closure of roles. By the construction of $\mathbf{T}$, for each $x, y \in V$ such that $Q^+ \in L(\langle x, y\rangle)$, $Q \notin L(\langle x, y\rangle)$ with $Q \in \mathbf{R} \cup \{\mathsf{Inv}(P) \mid P \in \mathbf{R}\}$ we have $\langle \pi(x), \pi(y)\rangle \in \mathcal{E}(Q^+)$.

According to $\mathsf{T9}$ there are $t_1, \cdots, t_n \in \mathbf{S}$ such that $\langle s, t_1\rangle, \cdots, \langle t_n, t\rangle \in \mathcal{E}(Q)$ and $\pi(x) = s, \pi(y) = t$. From this set of edges, we can pick $\langle s, t_1\rangle, \cdots, \langle t_k, t_{k+1}\rangle$ and $\langle t_l, t_{l+1}\rangle, \cdots, \langle t_n, t\rangle \in \mathcal{E}(Q)$ such that

1. $\mathcal{L}(t_k) = \mathcal{L}(t_l), \mathcal{L}(t_{k+1}) = \mathcal{L}(t_{l+1})$, $\mathcal{L}(\langle t_k, t_{k+1}\rangle) = \mathcal{L}(\langle t_l, t_{l+1}\rangle)$,
2. there are not $i, j \in \{0, \cdots, k, l+1, \cdots, n\}, i < j$ with $t_0 = s, t_{n+1} = t$ such that $\mathcal{L}(t_i) = \mathcal{L}(t_j), \mathcal{L}(t_{i+1}) = \mathcal{L}(t_{j+1})$, $\mathcal{L}(\langle t_i, t_{i+1}\rangle) = \mathcal{L}(\langle t_j, t_{j+1}\rangle)$.

We now build a cyclic non-duplicated $Q$-path from $\{s, t_1, \cdots, t_k, t_{k+1}, t_l, t_{l+1}, \cdots, t_n, t\}$. Since $x$ is not blocked ($x$ has a successor $y$), by the construction of $\mathbf{T}$ with $\pi(x) = s$, $\langle s, t_1\rangle \in \mathcal{E}(Q)$, $L(x) = \mathcal{L}(s)$, there exists a neighbor $w$ of $x$ such that $L(w) = \mathcal{L}(t_1)$ and $L'(\langle x, w\rangle) = \mathcal{L}(\langle s, t_1\rangle)$ where $L'(\langle x, w\rangle) = L(\langle x, w\rangle)$. If $w$ is a not successor of $x$, by Lem.2, we can add a new successor $w'$ of $x$ (and the subtree whose root is $w'$ can be built as above) such that $L(w) = \mathcal{L}(t_1)$ and $L(\langle x, w\rangle) = \mathcal{L}(\langle s, t_1\rangle)$. Thus, we can assume that $w$ is a $Q$-successor of $x$ since $\langle s, t_1\rangle \in \mathcal{E}(Q)$, and define $x_1 = w$ and $\pi(w) = t_1$.

Assume that there is $x_k \in V$ ( $x_k$ is not blocked by construction ) with $\pi(x_k) = t_k$ such that $L(x_k) = \mathcal{L}(t_k), L(x_{k-1}) = \mathcal{L}(t_{k-1})$, $L(\langle x_{k-1}, x_k\rangle) = \mathcal{L}(\langle t_{k-1}, t_k\rangle)$. We consider the following cases:

1. $x_k$ has a successor $w'$ such that $L(w') = \mathcal{L}(t_{k+1})$ and $L'(\langle x_k, w' \rangle) = \mathcal{L}(\langle t_k, t_{k+1} \rangle)$. If $w'$ is not blocked then define $x_{k+1} = w'$. If $w'$ is blocked by $z$ then define $x_{k+1} = z$. Since $\langle t_k, t_{k+1} \rangle \in \mathcal{E}(Q)$ hence $w'$ is a $Q$-successor of $x_k$.
2. $x_k$ has no successor $w'$ such that $L(w') = \mathcal{L}(t_{k+1})$ and $L'(\langle x_k, w' \rangle) = \mathcal{L}(\langle t_k, t_{k+1} \rangle)$. Since $L(x_k) = \mathcal{L}(t_k)$, by Lem.2, we can add a successor $w'$ of $x_k$ such that $L(w') = \mathcal{L}(t_{k+1})$ and $L'(\langle x_k, w' \rangle) = \mathcal{L}(\langle t_k, t_{k+1} \rangle)$. If $w'$ is not blocked then define $x_{k+1} = w'$. If $w'$ is blocked by $z$ then define $x_{k+1} = z$. In addition, we define $\pi(w') = t_{k+1}$. Since $\langle t_k, t_{k+1} \rangle \in \mathcal{E}(Q)$ hence $w'$ is a $Q$-successor of $x_k$.

Consequently, we obtain $x_1, \cdots, x_{k+1}$ such that $L(x_i) = \mathcal{L}(t_i)$ for all $i \in \{1, \cdots, k+1\}$ and $Q \in L(\langle x_i, x_{i+1} \rangle) = \mathcal{L}(\langle t_i, t_{i+1} \rangle)$, for all $i \in \{1, \cdots, k\}$ with $x_0 = x$ and $t_0 = t$.

By the same way, we can obtain $x_l, \cdots, x_n$ such that $L(x_i) = \mathcal{L}(t_i)$ and $\mathsf{Inv}(Q) \in L(\langle x_i, x_{i+1} \rangle) = \mathcal{L}(\langle t_i, t_{i+1} \rangle)$ for all $i \in \{l, \cdots, n\}$ where $t_{n+1} = t$ and $x_{n+1} = y$ if $y$ is not blocked or $x_{n+1} = z$ if $z$ blocks $y$.

According to Def. 5, $\langle x_{l+1}, x_l, \cdots, x_{n+1}, x, x_1, \cdots, x_{k+1} \rangle$ form a cyclic non-duplicated $Q$-path. Thus, $\mathbf{T}$ is a completion tree with cyclic paths for $D$. □