RWTH
LTCS–Report

Adding Numbers to the $\mathcal{SHIQ}$ Description Logic—First Results

Carsten Lutz

LTCS-Report 01-07

Adding Numbers to the $\mathcal{SHIQ}$ Description Logic—First Results

Carsten Lutz

RWTH Aachen, LuFG Theoretical Computer Science
Ahornstr. 55, 52074 Aachen

April 12, 2002

Abstract

Recently, the Description Logic (DL) $\mathcal{SHIQ}$ has found a large number of applications. This success is due to the fact that $\mathcal{SHIQ}$ combines a rich expressivity with efficient reasoning, as is demonstrated by its implementation in DL systems such as FaCT and RACER. One weakness of $\mathcal{SHIQ}$, however, limits its usability in several application areas: numerical knowledge such as knowledge about the age, weight, or temperature of real-world entities cannot be adequately represented. In this paper, we propose an extension of $\mathcal{SHIQ}$ that aims at closing this gap. The new Description Logic $\mathbb{Q}$-$\mathcal{SHIQ}$, which augments $\mathcal{SHIQ}$ by additional, "concrete domain" style concept constructors, allows to refer to rational numbers in concept descriptions, and also to define concepts based on the comparison of numbers via predicates such as $<$ or $=$. We argue that this kind of expressivity is needed in many application areas such as reasoning about the semantic web. We prove reasoning with $\mathbb{Q}$-$\mathcal{SHIQ}$ to be ExpTime-complete (thus not harder than reasoning with $\mathcal{SHIQ}$) by devising an automata-based decision procedure.

1 Motivation

Description Logics (DLs) are a family of knowledge representation formalisms, which are—apart from their classical application in KR—nowadays used in various application areas such as reasoning about entity relationship (ER) diagrams and providing a formal basis for the so-called semantic web [4; 5]. One of the most influential DLs proposed in the last years is the $\mathcal{SHIQ}$ Description Logic, whose success is based on the following two facts: first, $\mathcal{SHIQ}$ is a very expressive DL providing for, e.g., transitive roles, inverse roles, and number restrictions, but its reasoning problems are nevertheless decidable in ExpTime [12]. Second, $\mathcal{SHIQ}$ has been implemented in efficient DL systems such as FaCT and RACER, which can, despite the high worst-case complexity of reasoning with $\mathcal{SHIQ}$, deal surprisingly well even with huge knowledge bases [9; 6]. Although, as we just argued, $\mathcal{SHIQ}$'s expressive power is one of the main reasons for its success, there is still room for improvement. In particular, $\mathcal{SHIQ}$ cannot adequately represent numerical knowledge such as knowledge about the age, weight, or temperature of real-world entities, which, as we will later discuss in more detail, is

crucial for many important applications [2; 11; 5; 15]. In this paper, we extend $\mathcal{SHIQ}$ with a set of concept constructors that belong to the so-called concrete domain family of constructors and allow a straightforward representation of numerical knowledge. Let us view a concrete example of knowledge representation with the resulting DL, which is called $\mathbb{Q}$-$\mathcal{SHIQ}$. The concept

$$\textsf{Grandfather} \sqcap \exists \textsf{age.}=_{91} \sqcap (\geqslant 20 \textsf{ relatives Human}) \sqcap \forall \textsf{relatives age, age.}<$$

describes a Grandfather who is 91 years old, has at least 20 relatives (such constraints are called "number restrictions"), and is older than all of these relatives. Note that we can refer to rational numbers such as "91" and also compare numbers using predicates such as "$<$". We should like to stress that $\mathbb{Q}$-$\mathcal{SHIQ}$ cannot only be used for toy examples like the one above, but rather is a contribution to several "serious" application areas. Let us briefly review three examples:

(1) As described in [3; 4], reasoning about ER diagrams is an important application area of Description Logics. More specifically, current proposals found in the literature use fragments of $\mathcal{SHIQ}$ to encode ER diagrams. One shortcoming of this approach can be described as follows: ER diagrams make use of so-called attributes to represent non-relational data such as numbers and strings to be stored in the database. If $\mathcal{SHIQ}$ is used for representing ER diagrams, constraints concerning the values of attributes cannot be expressed. To give a simple example, if there exists a relation Employee having two attributes Birthday and Employment-date, then it cannot be expressed that members of this relation should be born before they are employed. If $\mathbb{Q}$-$\mathcal{SHIQ}$ is used for representing ER diagrams, such numerical data constraints on attributes can easily be handled. This topic is discussed in more detail in [17].

(2) In [15; 14], the Description Logic $\mathcal{TDL}$ is motivated as a valuable tool for the representation of temporal conceptual knowledge. $\mathcal{TDL}$ can be obtained from the well-known DL $\mathcal{ALC}$ [18] by adding general TBoxes and concrete domain style concept constructors that allow to represent relations between rational numbers such as "$=$" and "$<$". Indeed, it is not hard to see that $\mathcal{TDL}$ is a fragment of $\mathbb{Q}$-$\mathcal{SHIQ}$, but lacks much of its expressive power such as number restrictions and inverse roles. Thus, $\mathbb{Q}$-$\mathcal{SHIQ}$ is also well-suited for reasoning about temporal conceptual knowledge as described in [15; 14]. Moreover, $\mathbb{Q}$-$\mathcal{SHIQ}$ significantly extends the expressive power provided by $\mathcal{TDL}$, even in the temporal/numerical component of the logic. For example, in $\mathbb{Q}$-$\mathcal{SHIQ}$ one can refer to *concrete* time points and intervals such as 4 or $[1, 12]$ which is not possible in $\mathcal{TDL}$.

(3) A rapidly developing application area of DLs is their use as an ontology language for the semantic web [5]. As noted in [5; 8], the representation of "concrete datatypes" such as numbers is an important task in this context. However, in DLs such as OIL and DAML+OIL, which have been proposed in this application area, appropriate expressivity is either not provided or not taken into account for reasoning, which is done by a translation into $\mathcal{SHIQ}$ or related DLs. In [11], Horrocks and Sattler propose to extend $\mathcal{SHOQ}$, a close relative of $\mathcal{SHIQ}$, with so-called unary concrete domains in order to integrate concrete datatypes. However, this solution is not really satisfying since, as is explained in more detail in [14], unary concrete domains are of

very limited expressivity. If $\mathbb{Q}$-$\mathcal{SHIQ}$ is used as the target logic in translations of OIL and DAML+OIL, a rather powerful means for describing *numerical* concrete datatypes becomes available.

As the main result of this paper, we prove reasoning with $\mathbb{Q}$-$\mathcal{SHIQ}$ to be decidable in ExpTime by devising an automata-based decision procedure. Thus, $\mathbb{Q}$-$\mathcal{SHIQ}$ sensibly enhances the expressive power of $\mathcal{SHIQ}$ without increasing the worst-case complexity of reasoning.

2 Syntax and Semantics

In this section, we introduce the Description Logic $\mathbb{Q}$-$\mathcal{SHIQ}$ in detail. We first give the syntax and semantics of $\mathbb{Q}$-$\mathcal{SHIQ}$-roles, then introduce some useful abbreviations, and finally define syntax and semantics of $\mathbb{Q}$-$\mathcal{SHIQ}$-concepts.

Definition 2.1 ($\mathbb{Q}$-$\mathcal{SHIQ}$-roles). Let N_{rR}, N_{tR}, and N_{aF} be countably infinite and mutually disjoint sets of *regular role names*, *transitive role names*, and *abstract features*, respectively. Moreover, let $N_R = N_{rR} \uplus N_{tR} \uplus N_{aF}$. The set of $\mathbb{Q}$-$\mathcal{SHIQ}$-*roles* ROL is $N_R \cup \{R^- \mid R \in N_R\}$. A *role inclusion* is of the form

$$R \sqsubseteq S,$$

for $R, S \in$ ROL. A *role hierarchy* is a set of role inclusions.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a set $\Delta^{\mathcal{I}}$, called the *domain* of $\mathcal{I}$, and a function $\cdot^{\mathcal{I}}$ which maps every role $R \in$ ROL to a subset $R^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ such that, for $R \in N_R$, $S \in N_{tR}$, and $f \in N_{aF}$,

$$(x, y) \in R^{\mathcal{I}} \text{ iff } (y, x) \in R^{-\mathcal{I}},$$
$$\text{if } (x, y) \in S^{\mathcal{I}} \text{ and } (y, z) \in S^{\mathcal{I}}, \text{ then } (x, z) \in S^{\mathcal{I}}, \text{ and}$$
$$f^{\mathcal{I}} \text{ is functional.}$$

An interpretation $\mathcal{I}$ is a *model* of a role hierarchy $\mathcal{R}$ iff $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$ for each $R \sqsubseteq S \in \mathcal{R}$.
$\Diamond$

We introduce some notation to make the following considerations easier.

1. The function Inv yields the inverse of a role. More precisely, for $R \in$ ROL, we set

$$\mathsf{Inv}(R) := \begin{cases} R^- & \text{if } R \text{ is a role name,} \\ S & \text{if } R = S^- \text{ for a role name } S. \end{cases}$$

2. Since set inclusion is transitive and $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$ implies $\mathsf{Inv}(R)^{\mathcal{I}} \subseteq \mathsf{Inv}(S)^{\mathcal{I}}$, for a role hierarchy $\mathcal{R}$, we introduce $\sqsubseteq^*_{\mathcal{R}}$ as the reflexive-transitive closure of

$$\mathcal{R} \cup \{\mathsf{Inv}(R) \sqsubseteq \mathsf{Inv}(S) \mid R \sqsubseteq S \in \mathcal{R}\}.$$

3

3. We call a role $R \in \mathsf{ROL}$ *transitive* with respect to a role hierarchy $\mathcal{R}$ iff R is interpreted in a transitive relation in every model of $\mathcal{R}$. It is not hard to see that this is the case iff the following predicate evaluates to true:

$$\mathsf{Trans}_{\mathcal{R}}(R) := \begin{cases} \mathsf{true} & \text{if there exists a role } S \in \mathsf{N_{tR}} \text{ such that } S' \mathrel{\underline{\overline{\ast}}}_{\mathcal{R}} R \text{ and} \\ & \quad R \mathrel{\underline{\overline{\ast}}}_{\mathcal{R}} S'' \text{ for some } S', S'' \in \{S, \mathsf{Inv}(S)\} \\ \mathsf{false} & \text{otherwise.} \end{cases}$$

4. A role $R \in \mathsf{ROL}$ is called *simple* with respect to a role hierarchy $\mathcal{R}$ iff $\mathsf{Trans}_{\mathcal{R}}(S)$ does not hold for any $S \in \mathsf{ROL}$ with $S \mathrel{\underline{\overline{\ast}}} R$.

For both "$\mathrel{\underline{\overline{\ast}}}_{\mathcal{R}}$" and $\mathsf{Trans}_{\mathcal{R}}$, we omit the index if clear from the context. Note that no transitive role is simple since $\mathrel{\underline{\overline{\ast}}}$ is defined as the *reflexive*-transitive closure. For the same reason, we have $\mathsf{Trans}(R)$ for all $R \in \mathsf{N_{tR}}$. However, roles must obviously not be in $\mathsf{N_{tR}}$ in order to be transitive. For example, if $R \in \mathsf{N_{tR}}$, then R^- is also transitive. Similarly, if $S \in \mathsf{N_{tR}}$, $R \notin \mathsf{N_{tR}}$, $S^- \sqsubseteq R$, $R \sqsubseteq S^-$, then R is transitive. We are now ready to define $\mathbb{Q}$-$\mathcal{SHIQ}$-concepts and their semantics.

Definition 2.2 ($\mathbb{Q}$-$\mathcal{SHIQ}$-concepts). Let $\mathsf{N_C}$ and $\mathsf{N_{cF}}$ be countably infinite sets of *concept names* and *concrete features*, respectively, such that $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_{cF}}$ are mutually disjoint. A *path* is a sequence $R_1 \cdots R_k g$ consisting of roles $R_1, \ldots, R_k \in \mathsf{ROL}$ and a concrete feature $g \in \mathsf{N_{cF}}$. A path $R_1 \cdots R_k g$ with $\{R_1, \ldots, R_k\} \subseteq \mathsf{N_{aF}}$ is called *feature path*. The set of $\mathbb{Q}$-$\mathcal{SHIQ}$-*concepts* is the smallest set such that

1. every concept name $C \in \mathsf{N_C}$ is a concept,

2. if C and D are concepts and $R \in \mathsf{ROL}$, then $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall R.C$, and $\exists R.C$ are concepts,

3. if C is a concept, $R \in \mathsf{ROL}$ is simple, and $n \in \mathbb{N}$, then $(\leqslant n\ R\ C)$ and $(\geqslant n\ R\ C)$ are concepts,

4. if u_1 and u_2 are feature paths and $P \in \{<, \leq, =, \neq, \geq, >\}$, then $\exists u_1, u_2.P$ and $\forall u_1, u_2.P$ are concepts,

5. if $R \in \mathsf{ROL}$ is simple, g_1 and g_2 are concrete features, and $P \in \{<, \leq, =, \neq, \geq, >\}$, then $\exists R g_1, g_2.P$ and $\forall R g_1, g_2.P$ are concepts, and

6. if g is a concrete feature, $P \in \{<, \leq, =, \neq, \geq, >\}$, and $q \in \mathbb{Q}$, then $\exists g.P_q$ is a concept.

We use $\top$ as an abbreviation for $A \sqcup \neg A$ (for some fixed $A \in \mathsf{N_C}$). The interpretation function $\cdot^{\mathcal{I}}$ of interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ maps, additionally,

- every concept C to a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and

- every concrete feature g to a partial function $g^{\mathcal{I}}$ from $\Delta^{\mathcal{I}}$ to the set of rational numbers $\mathbb{Q}$

4

such that

$$
\begin{aligned}
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}}, \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}}, \\
\neg C^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}, \\
(\exists R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \text{There is some } y \in \Delta^{\mathcal{I}} \text{ with } (x,y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}, \\
(\forall R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \text{For all } y \in \Delta^{\mathcal{I}}, \text{ if } (x,y) \in R^{\mathcal{I}}, \text{ then } y \in C^{\mathcal{I}}\}, \\
(\leqslant n\ R\ C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \mid (x,y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leqslant n\}, \\
(\geqslant n\ R\ C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \mid (x,y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geqslant n\}, \\
(\exists U_1, U_2.P)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \text{There are } q_1 \in U_1^{\mathcal{I}} \text{ and } q_2 \in U_2^{\mathcal{I}} \text{ with } q_1\ P\ q_2\} \\
(\forall U_1, U_2.P)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \text{For all } q_1 \in U_1^{\mathcal{I}} \text{ and } q_2 \in U_2^{\mathcal{I}}, \text{ we have } q_1\ P\ q_2\} \\
(\exists g.P_q)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid g^{\mathcal{I}}(x)\ P\ q\}
\end{aligned}
$$

where U_1 and U_2 denote paths, $\sharp S$ denotes the cardinality of the set S, and, for every path $U = R_1 \cdots R_k g$, $U^{\mathcal{I}}$ is defined as

$$
\begin{aligned}
\{(x,q) \subseteq \Delta_{\mathcal{I}} \times \mathbb{Q} \mid &\exists y_1, \ldots, y_{k+1} : x = y_1, \\
&(y_i, y_{i+1}) \in R_i^{\mathcal{I}} \text{ for } 1 \leq i \leq k, \text{ and } g^{\mathcal{I}}(y_{k+1}) = q\}.
\end{aligned}
$$

An interpretation $\mathcal{I}$ is called a *model* of a concept C iff $C^{\mathcal{I}} \neq \emptyset$. C is called *satisfiable with respect to a role hierarchy* $\mathcal{R}$ iff there exists a model of C and $\mathcal{R}$. A concept D *subsumes* a concept C with respect to $\mathcal{R}$ (written $C \sqsubseteq_{\mathcal{R}} D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for each model $\mathcal{I}$ of $\mathcal{R}$. Two concepts C, D are *equivalent* with respect to $\mathcal{R}$ (written $C \equiv_{\mathcal{R}} D$) iff they are mutually subsuming. $\diamond$

In the following sections, we show that $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concept satisfiability is decidable in deterministic exponential time. This also yields decidability and an ExpTime upper complexity bound for concept subsumption and equivalence: we have (i) $C \sqsubseteq_{\mathcal{R}} D$ iff $C \sqcap \neg D$ is unsatisfiable w.r.t. $\mathcal{R}$ and (ii) $C \equiv_{\mathcal{R}} D$ iff $C \sqsubseteq_{\mathcal{R}} D$ and $D \sqsubseteq_{\mathcal{R}} C$.

Throughout this paper, we denote concept names by A and B, concepts by C, D, and E, roles by P, R, and S, abstract features by f, concrete features by g, paths by U, feature paths by u, and predicates by P.

Let us discuss the $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concept language in some more detail. Since exhaustive information on $\mathcal{SHIQ}$ can be found in, e.g., [12], we concentrate on the additional concept constructors $\exists U_1, U_2.P$, $\forall U_1, U_2.P$, and $\exists g.P_q$, which, as has already been noted, are often called "concrete domain constructors". Concrete domains have been introduced by Baader and Hanschke [1] as a means for representing "concrete knowledge" such as knowledge about numbers, strings, or spatial extensions. More precisely, Baader and Hanschke extend the basic propositionally closed DL $\mathcal{ALC}$ with concrete domains, where a concrete domain $\mathcal{D}$ is comprised of a set called the domain and a set of predicates with a fixed extension on this domain. However, Baader and Hanschke do not commit themselves to a particular concrete domain, but rather view the concrete domain as a parameter to their logic, which they call $\mathcal{ALC}(\mathcal{D})$. From the concrete domain perspective, $\mathbb{Q}\text{-}\mathcal{SHIQ}$ can be viewed as being equipped with one particular concrete domain, whose domain are the rationals and which is equipped

5

with binary predicates $<, \leq, =, \neq, \geq, >$ and unary predicates P_q, where $q \in \mathbb{Q}$ and $P \in \{<, \leq, =, \neq, \geq, >\}$.

The paths U_1 and U_2 that may appear inside $\mathbb{Q}\text{-}\mathcal{SHIQ}$'s binary concrete domain constructors $\exists U_1, U_2.P$ and $\forall U_1, U_2.P$ are of a rather special form: either (i) U_1 and U_2 are feature paths or (ii) U_1 has the form Rg_1 and U_2 has the form g_2. Let us illustrate the expressive power of these two variants of the same constructors: using Variant (i), we can, e.g., describe people whose mother's spouse earns more than their father (we use parentheses for better readability):

$$\exists(\mathsf{mother\ spouse\ wage}), (\mathsf{father\ wage}).>$$

This example illustrates the main advantage of Variant (i): we can talk about *sequences* of features. This variant of $\mathbb{Q}\text{-}\mathcal{SHIQ}$'s binary concrete domain constructors are precisely the concrete domain constructors offered by $\mathcal{ALC(D)}$ and the temporal DL $\mathcal{TDL}$ mentioned in the introduction. The main disadvantage of Variant (i) is that, inside paths, we may only use abstract features but no roles from $\mathsf{N_{rR}}$. For example, if we want to describe people having an older neighbor by the concept

$$\exists(\mathsf{neighbor}, \mathsf{age}), (\mathsf{age}).>,$$

then "neighbor" should clearly be from $\mathsf{N_{rR}}$ rather than from $\mathsf{N_{aF}}$, since otherwise we would enforce that the described persons have at most a single neighbor. Therefore, we need Variant (ii) of the binary concrete domain constructors to define this concept. Note that Variant (ii) is neither provided by $\mathcal{ALC(D)}$ nor by $\mathcal{TDL}$, but rather is a restricted version of the concrete domain constructors defined in [7].

It is not hard to see that we could also have admitted variants $\exists g_1, Rg_2.P$ and $\forall g_1, Rg_2.P$ of the binary concrete constructors since this variant is just syntactic sugar: $\exists g_1, Rg_2.P$ is equivalent to $\exists Rg_2, g_1.\widetilde{P}$ and $\forall g_1, Rg_2.P$ is equivalent to $\forall Rg_2, g_1.\widetilde{P}$, where $\widetilde{P}$ denotes the inverse of the predicate P—for example, "$\widetilde{<}$" is "$>$" and "$\widetilde{\cong}$" is "$=$". Obviously, the most general approach would be to allow arbitrary paths inside the binary concrete domain constructors.[1] The resulting logic, however, cannot easily be handled by the ExpTime decision procedure for concept satisfiability presented in the remainder of this paper. Indeed, it is currently not even clear whether the resulting logic is decidable.

It may look strange at first sight that $\mathbb{Q}\text{-}\mathcal{SHIQ}$ provides for both abstract features and number restrictions since, as is well-known, number restrictions, transitive roles, and role hierarchies can be used to enforce that a role R_f from $\mathsf{N_{rR}}$ is interpreted functionally: just use the concept $\forall R.(\leqslant 1\ R_f\ \top)$, where $R \in \mathsf{N_{tR}}$, and employ the role hierarchy to ensure that $S \sqsubseteq^* R$ for every "relevant" role S (e.g. for the roles occurring in the concept or role hierarchy whose satisfiability is to be decided). The reason for this redundancy is that number restrictions are, in principle, a more general means of expressivity than abstract features, but having abstract features explicitly available allows for a straightforward definition of Variant (i) of the concrete domain constructors.

---

[1]Admitting arbitrary paths inside the *unary* concrete domain constructor is not an issue since the concept $\exists R_1 \cdots R_k g.P_q$ (with the obvious semantics) can be written as $\exists R_1. \cdots .\exists R_k.\exists g.P_q$.

Note that only simple roles are allowed in Variant (ii) of the binary concrete domain constructor. Similarly, roles used inside number restrictions are also required to be simple. As proved in [12], the latter restriction is crucial since admitting non-simple roles inside number restrictions yields undecidable reasoning problems. Non-simple roles inside the binary concrete domain constructors cannot be handled by the EXPTIME decision procedure in its current form. Again, it is as of now unknown whether admitting them yields undecidability of reasoning.

There exist existential and universal versions of the binary concrete domain constructors but only an existential version of the unary concrete domain constructor. It is not hard to see that we could also have admitted a universal version since $\forall g.P_q$ (with the obvious semantics) is clearly equivalent to $\forall g, g. \neq \sqcup \exists g.P_q$, where $\forall g, g. \neq$ simply expresses that there exists no successor for the concrete feature g. Similarly, the universal version of Variant (i) of the binary concrete domain constructors can be expressed in terms of the existential version of Variant (i) of this constructor. This does, however, not hold for Variant (ii) of the binary concrete domain constructors since it accepts non-functional roles as arguments. For this reason, we have chosen to include universal versions of both Variant (i) and (ii) for uniformity.

Most modern Description Logics do not only consist of a concept language but also provide for a TBox component. Formally, a TBox is a finite set of concept equations $C \doteq D$, where C and D are $\mathbb{Q}$-$\mathcal{SHIQ}$-concepts. An interpretation $\mathcal{I}$ is a model of a TBox $\mathcal{T}$ iff it satisfies $C^{\mathcal{I}} = D^{\mathcal{I}}$ for all $(C \doteq D) \in \mathcal{T}$. In the presence of TBoxes, one is usually interested in the satisfiability of concepts w.r.t. TBoxes and role hierarchies, where a concept C is satisfiable w.r.t. a TBox $\mathcal{T}$ and a role hierarchy $\mathcal{R}$ iff there exists a model $\mathcal{I}$ of C, $\mathcal{T}$, and $\mathcal{R}$. However, as was shown in [10], in the presence of role hierarchies and transitive roles it is possible to polynomially reduce concept satisfiability w.r.t. TBoxes and role hierarchies to concept satisfiability w.r.t. role hierarchies, only. This is done using a rather straightforward technique known as "internalization". Because of this, we will not explicitly consider TBoxes in this paper. They can, however, easily be treated by internalization.

3 Preliminaries

Decidability and the EXPTIME upper complexity bound for $\mathbb{Q}$-$\mathcal{SHIQ}$-concept satisfiability is established by devising an automata-based decision procedure. The general idea behind this procedure is to define, for a given concept C and role hierarchy $\mathcal{R}$, a looping tree-automaton $\mathcal{A}_{C,\mathcal{R}}$ that accepts exactly the so-called Hintikka-trees for C and $\mathcal{R}$. These Hintikka-trees are abstractions of models of C and $\mathcal{R}$, i.e., C and $\mathcal{R}$ have a model if and only if C and $\mathcal{R}$ have a Hintikka-tree. The obvious advantage of Hintikka-trees over models is that they are trees and thus amenable to tree automata techniques. Once the automaton $\mathcal{A}_{C,\mathcal{R}}$ is defined, it remains to apply the standard emptiness test for tree automata: clearly, the language accepted by the constructed automaton is empty iff C is satisfiable w.r.t. $\mathcal{R}$.

In this section, we introduce the basic notions underlying the decision procedure sketched above. We start with developing a useful normal form (called *path normal*

form) for $\mathbb{Q}$-$\mathcal{SHIQ}$-concepts, and then introduce looping tree-automata, whose theory forms the basis for the decision algorithm. Finally, we define constraint graphs, which will play an important role in representing the "numerical part" of $\mathbb{Q}$-$\mathcal{SHIQ}$-interpretations in Hintikka-trees.

3.1 Normal Forms

We start with formulating a property of role hierarchies that we will generally assume to be satisfied in what follows:

A role hierarchy $\mathcal{R}$ is called *admissible* iff all $f \in \mathsf{N}_{\mathsf{aF}}$ are simple w.r.t. $\mathcal{R}$.

Demanding admissibility of role hierarchies is closely related to requiring roles R that appear inside number restrictions $(\leqslant n\ R\ C)$ and $(\geqslant n\ R\ C)$ to be simple: since abstract features are interpreted in functional relations, they are "inherently number restricted", i.e., for each $f \in \mathsf{N}_{\mathsf{aF}}$, $(\leqslant 1\ f\ \top)$ is satisfied by every domain element in every interpretation. However, it seems that, in contrast to admitting arbitrary roles inside number restrictions, dropping admissibility of role hierarchies does not necessarily lead to undecidability of reasoning. Indeed, we claim that the decision procedure presented in this paper can, in principle, be extended to also deal with non-admissible role hierarchies. We nevertheless restrict ourselves to admissible role hierarchies since (i) this eliminates several case distinctions in the proofs, and (ii) we agree with Horrocks and Sattler [10] who argue that non-simple features are rather unnatural: if $f \in \mathsf{N}_{\mathsf{aF}}$ is non-simple, then there exists a role $R \in \mathsf{N}_{\mathsf{R}}$ such that $\mathsf{Trans}(R)$ and $R \sqsubseteq^{\!*} f$. Hence, R is both functional and transitive which produces strange effects: for any interpretation $\mathcal{I}$, $R^{\mathcal{I}}$ may not contain any acyclic paths of length greater 1. Hence, the concept $\exists R.\exists R.\top$ is satisfiable only in models that contain either (i) a domain element a which is its own R-successor or (ii) two domain elements a and b, where b is R-successor of a and of itself (the same holds for the concept $\exists R.\exists R.\exists R.\top$). To avoid such effects, which do not seem to promote writing understandable knowledge bases, we generally require role hierarchies to be admissible.

Let us now turn our attention towards normal forms for $\mathbb{Q}$-$\mathcal{SHIQ}$-concepts. We first introduce the well-known negation normal form.

Definition 3.1 (NNF). A concept C is in *negation normal form* (NNF) if negation occurs only in front of concept names. Exhaustive application of the following rewrite rules translates concepts to equivalent ones in NNF.

$$
\begin{aligned}
\neg\neg C &\implies C \\
\neg(C \sqcap D) &\implies \neg C \sqcup \neg D & \neg(C \sqcup D) &\implies \neg C \sqcap \neg D \\
\neg(\exists R.C) &\implies (\forall R.\neg C) & \neg(\forall R.C) &\implies (\exists R.\neg C) \\
\neg(\leqslant n\ R\ C) &\implies (\geqslant (n+1)\ R\ \neg C) & \neg(\geqslant n\ R\ C) &\implies (\leqslant (n-1)\ R\ \neg C) \\
\neg(\exists U_1, U_2.P) &\implies \forall U_1, U_2.\overline{P} & \neg(\forall U_1, U_2.P) &\implies \exists U_1, U_2.\overline{P} \\
\neg(\exists g.P_q) &\implies \forall g, g.\neq \sqcup \exists g.\overline{P}_q
\end{aligned}
$$

where $\overline{P}$ denotes the negation of predicates, e.g. "$\overline{<}$" is "$\geq$" and "$\overline{\neq}$" is "$=$". By $\mathsf{nnf}(C)$, we denote the result of converting C into NNF using the above rules. Furthermore, we use $\sim C$ as a shorthand for $\mathsf{nnf}(\neg C)$. $\diamond$

Note that, in the $\neg(\exists g.P_q)$ case of the NNF rewrite rules, the concept $\forall g, g.\neq$ is only used to express that there exists no g-successor at all. We now introduce path normal form, which was first described in [15] in the context of the Description Logic $\mathcal{TDL}$ mentioned in the introduction.

Definition 3.2 (Path Normal Form). A $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concept C is in *path normal form* (PNF) iff it is in NNF and, for all subconcepts $\exists U_1, U_2.P$ and $\forall U_1, U_2.P$ of C, we have either

1. $U_1 = g_1$ and $U_2 = g_2$ for some $g_1, g_2 \in \mathsf{N_{cF}}$ or

2. $U_1 = Rg_1$ and $U_2 = g_2$ for some $R \in \mathsf{N_{aF}} \cup \mathsf{N_{rR}}$ and $g_1, g_2 \in \mathsf{N_{cF}}$.

$\diamond$

The following lemma shows that we can w.l.o.g. assume $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concepts to be in PNF.

Lemma 3.3. *Satisfiability of $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concepts can be polynomially reduced to satisfiability of $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concepts in PNF.*

Proof We first define an auxiliary mapping and then use this mapping to translate $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concepts into equivalent ones in PNF. Let C be a $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concept. For every feature path $u = f_1 \cdots f_n g$ used in C, we assume that $[g], [f_n g], \ldots, [f_1 \cdots f_n g]$ are concrete features not used in C. We inductively define a mapping λ from concrete paths u in C to concepts as follows:

$$\begin{aligned}
\lambda(g) &= \top \\
\lambda(fu) &= (\exists[fu], f[u]. =) \sqcap \exists f.\lambda(u)
\end{aligned}$$

For every $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concept C, a corresponding concept $\rho(C)$ is obtained by

- first replacing all subconcepts $\forall u_1, u_2.P$ where $u_i = f_1^{(i)} \cdots a f_{k_i}^{(i)} g_i$ for $i \in \{1, 2\}$ with

$$\forall f_1^{(1)}. \cdots. \forall f_{k_1}^{(1)}.\forall g_1, g_1.\neq \ \sqcup \ \forall f_1^{(2)}. \cdots. \forall f_{k_2}^{(2)}.\forall g_2, g_2.\neq \ \sqcup \ \exists u_1, u_2.P$$

- and then replacing all subconcepts $\exists u_1, u_2.P$ with $\exists[u_1], [u_2].P \sqcap \lambda(u_1) \sqcap \lambda(u_2)$.

Now let C be a $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concept. Using the rewriting rules from Definition 3.1, we can convert C into an equivalent concept C' in NNF. It is then easy to check that C' is satisfiable iff $\rho(C')$ is satisfiable. Moreover, $\rho(C')$ is clearly in PNF and the translation can be done in polynomial time. $\qquad\qquad\square$

Intuitively, Lemma 3.3 states that Variant (i) of the binary concrete domain constructors discussed in the previous section can be reduced to the forms $\exists fg_1, g_2.P$ and $\exists g_1, g_2.P$. Variant (ii) of the binary concrete domain constructors does not need to be manipulated in order to fit into the PNF scheme. Let us remark that our algorithm's need for PNF is the reason why we cannot handle arbitrary paths inside the binary concrete domain constructors: it is an interesting exercise to check that the constructor $\forall U_1, U_2.P$ with $U_1 = R_1 \cdots R_n g$ and $U_2 = S_1 \cdots S_m g'.P$ can be reduced to the forms $\exists Rg_1, g_2.P$ and $\exists g_1, g_2.P$ if $P \in \{<, \leq, =, \geq, >\}$ but *not* if P is "$\neq$".

3.2 Automata and Constraint Graphs

The automata used in this paper are so called looping tree-automata which can roughly be described as Büchi tree-automata where every run is accepting [21; 19]

Definition 3.4 (Looping automaton). Let M be a set and $k \geq 1$. A *k-ary M-tree* is a mapping $T : \{1, \ldots, k\}^* \to M$ that labels each node $\alpha \in \{1, \ldots, k\}^*$ with $T(\alpha) \in M$. Intuitively, the node αi is the i-th child of α. We use ϵ to denote the empty word (corresponding to the root of the tree).

A *looping automaton* $\mathcal{A} = (Q, M, I, \Delta)$ for k-ary M-trees is defined by a finite set Q of *states*, a finite alphabet M, a subset $I \subseteq Q$ of *initial states*, and a *transition relation* $\Delta \subseteq Q \times M \times Q^k$.
A *run* of $\mathcal{A}$ on an M-tree T is a mapping $r : \{1, \ldots, k\}^* \to Q$ with $r(\epsilon) \in I$ and

$$(r(\alpha), T(\alpha), r(\alpha 1), \ldots, r(\alpha k)) \in \Delta$$

for each $\alpha \in \{1, \ldots, k\}^*$. A looping automaton accepts all those M-trees for which there exists a run, i.e., the language $L(\mathcal{A})$ of M-trees accepted by $\mathcal{A}$ is

$$L(\mathcal{A}) := \{T \mid \text{there is a run of } \mathcal{A} \text{ on } T\}. \qquad \diamondsuit$$

Vardi and Wolper [21] show that the emptiness problem for looping automata, i.e., the problem to decide whether the language $L(\mathcal{A})$ accepted by a given looping automaton $\mathcal{A}$ is empty, is decidable in polynomial time.

We now introduce constraint graphs. Such graphs are used to represent the "numerical part" of $\mathbb{Q}$-$\mathcal{SHIQ}$-interpretations in Hintikka-trees. A detailed description of how this is done, however, is delayed until Section 4, where Hintikka trees are defined.

Definition 3.5 (Constraint graph). A *constraint graph* is a directed graph $G = (V, E, \tau)$, where V is a countable set of *nodes*,

$$E \subseteq V \times V \times \{<, \leq, =, \neq, \geq, >\}$$

is a set of *labeled edges*, and

$$\tau \subseteq V \times \{P_q \mid P \in \{<, \leq, =, \neq, \geq, >\} \text{ and } q \in \mathbb{Q}\}$$

is a node labeling function. For simplicity, we generally assume that constraint graphs are *equality closed*, i.e., that we have $(v_1, v_2, =) \in E$ iff $(v_2, v_1, =) \in E$ for all $v_1, v_2 \in V$.

For a set of edges E, we use $\mathsf{cl}_=(E)$ to denote the *equality closure* of E which is defined in the obvious way. In what follows, we sometimes write $\tau(v)$ for $\{P_q \mid (v, P_q) \in \tau\}$.

A constraint graph $G = (V, E, \tau)$ is called *satisfiable over* S—where S is a set equipped with a total ordering $<$—iff there exists a total mapping from V to S such that

1. $(v) \, P \, q$ for all $P_q \in \tau(v)$ and

2. $(v_1) \, P \, (v_2)$ for all $(v_1, v_2, P) \in E$.

Such a mapping is called a *solution* for G. $\hspace{2cm} \diamond$

We will see later that every Hintikka-tree T induces a constraint graph which represents the "numerical part" of the canonical interpretation described by T. As should be intuitively clear, these induced constraint graphs have to be satisfiable in order for Hintikka-trees to be proper abstractions of interpretations. Since, later on, we must define looping automata which accept exactly the Hintikka-trees for a concept C and role hierarchy $\mathcal{R}$, such automata should be able to verify the satisfiability of (induced) constraint graphs. This check is the main problem to be solved when developing an automata-based decision procedure for $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concept satisfiability: the induced constraint graph and its satisfiability are "global" notions while automata work "locally". This problem can be overcome as follows: first, we define Hintikka trees such that their induced constraint graphs have a certain form (we will call such constraint graphs *normal*); second, we formulate an adequate criterion for the satisfiability of normal constraint graphs; and third, we show how this criterion can be verified by "local tests" that can be performed by automata. Let us start with introducing normal constraint graphs and the criterion for their satisfiability, which is called *consistency*.

Definition 3.6 (Normal, $<$-cycle, Consistent). Let $G = (V, E, \tau)$ be a constraint graph. G is called *normal* if it satisfies the following conditions:

1. $(v_1, v_2, P) \in E$ implies $P \in \{<, =\}$,

2. $(v, P_q) \in \tau$ implies $P \in \{<, =, >\}$,

3. for each rational number q appearing in τ and each node $v \in V$, we have $(v, P_q) \in \tau$ for some $P \in \{<, =, >\}$.

A *path* Q in a normal constraint graph G is a finite non-empty sequence of nodes $v_0, \ldots, v_k \in V$ such that, for all $i < k$, there exists a P such that $(v_i, v_{i+1}, P) \in E$. Such a path is also called a path *from* v_0 to v_k. A path $v_0, \ldots, v_k$ is a P-*path* for $P \in \{<, =\}$ iff $(v_i, v_{i+1}, P) \in E$ for some $i < k$. Moreover, $v_0, \ldots, v_k$ is a *strict P-path* for $P \in \{<, =\}$ iff $(v_i, v_{i+1}, P) \in E$ for each $i < k$.

A *cycle* O in G is a path $v_0, \ldots, v_k$ for with there exists a P such that $(v_k, v_0, P) \in E$. For $i \leq k$, we use i_O^+ to denote $(i + 1) \bmod (k + 1)$, i.e., i_O^+ denotes the index following i in the cycle O. The index $\cdot_O$ is omitted if clear from the context. A cycle $O = v_0, \ldots, v_k$ is a $<$-*cycle* iff $(v_i, v_{i^+}, <) \in E$ for some i with $i \leq k$.

A normal constraint graph G is *consistent* iff it satisfies the following conditions:

1. G contains no $<$-cycle,

2. for all $v \in V$, there exists a $q \in \mathbb{Q}$ such that qPq' for all $P_{q'} \in \tau(v)$,

3. for all $(v_1, v_2, P) \in V$, there exist $q_1, q_2 \in \mathbb{Q}$ such that

   - $q_1 \, P \, q_2$,
   - $q_1 \, P' \, q$ for all $P'_q \in \tau(v_1)$, and
   - $q_2 \, P' \, q$ for all $P'_q \in \tau(v_2)$.

   $\diamond$

It may appear that Property 3 of consistency is too weak since it only demands the existence of rationals q_1, q_2 for *each* edge between v_1 and v_2 separately instead of for *all* such edges simultaneously: a normal constraint graph with set of edges $\{(v_1, v_2, <), (v_1, v_2, =)\}$ is clearly unsatisfiable, but does not violate Property 3. This, however, is compensated by Property 1 which is violated in this example.

We now show that consistency is indeed an adequate criterion for the satisfiability of normal constraint graphs.

Theorem 3.7. *A normal constraint graph G is satisfiable over $\mathbb{Q}$ iff G is consistent.*

Proof Since the "only if" direction is trivial, we concentrate on the "if" direction. Let G be a normal constraint graph that is consistent. We define a relation $\sim$ on V by setting $v_1 \sim v_2$ iff $v_1 = v_2$ or there exists a strict $=$-path between v_1 and v_2. Since constraint graphs are assumed to be equality closed, $\sim$ is an equivalence relation. For $v \in V$, we denote the equivalence class of v w.r.t. $\sim$ by $[v]$. Define a new constraint graph $G' = (V', E', \tau')$ as follows:

$$
\begin{aligned}
V' \quad &:= \quad \{[v] \mid v \in V\} \\
E' \quad &:= \quad \{([v_1], [v_2], <) \mid \exists v'_1, v'_2 \in V \text{ such that} \\
&\qquad\qquad v'_1 \in [v_1], v'_2 \in [v_2], \text{ and } (v'_1, v'_2, <) \in E\} \\
\tau' \quad &:= \quad \bigcup_{v' \in [v]} \tau(v')
\end{aligned}
$$

It is not hard to check that G' is normal. Moreover, G' is consistent, i.e., it satisfies Properties 1 to 3 from Definition 3.6:

1. Assume that there is a $<$-cycle $[v_0], \ldots, [v_k]$ in G'. Since G' contains no "$=$-edges", we have $([v_i], [v_{i+}], <) \in E'$ for $i \leq k$. By construction of G', there exist $x_0, \ldots, x_k, y_0, \ldots, y_k \in V$ such that, for $i \leq k$, we have $x_i, y_i \in [v_i]$ and $(x_i, y_{i+}, <) \in E$. Moreover, for $i \leq k$, it holds that either $y_i = x_i$ or there exists a strict $=$-path between y_i and x_i in G. This obviously implies that G contains a $<$-cycle which contradicts the assumption that G satisfies Property 1.

2. Let $[v] \in V'$. Since G satisfies Properties 2 and 3 of normality and Property 3 of consistency, the existence of a strict $=$-path from a node $v_1 \in V$ to a node

$v_2 \in V$ implies that $\tau(v_1) = \tau(v_2)$. This, in turn, implies that $\tau(v') = \tau'([v])$ for all $v' \in [v]$. Thus, since G satisfies Property 2 of consistency, it is clear that G' also satisfies Property 2.

3. Let $([v_1], [v_2], P) \in E'$. By definition of G', this implies that P is "$<$". By definition of G' there exist $v'_1 \in [v_1]$ and $v'_2 \in [v_2]$ such that $(v'_1, v'_2, <) \in E$. As in the previous case, we have $\tau(v') = \tau'([v_1])$ for all $v' \in [v_1]$ and similar for v_2. Thus, $\tau(v'_1) = \tau'([v_1])$ and $\tau(v'_2) = \tau'([v_2])$. Since G satisfies Property 3 of consistency, it is hence clear that there exists a $q \in \mathbb{Q}$ as required.

We now define a solution for G'. This su ces to prove the theorem since, obviously, solutions for G' can straightforwardly be converted into solutions for G.

Since G' does not contain a $<$-cycle, E' induces a partial order $\prec$ with domain V' such that $v_1 \prec v_2$ iff there exists a $<$-path from v_1 to v_2 in G'. Let $\ll$ be an enumeration of all nodes $v \in V'$ such that, for all nodes v, v', the following property is satisfied: if

- there exists a $q \in \mathbb{Q}$ such that $=_q \in \tau'(v)$ and

- for all $q \in \mathbb{Q}$, we have $=_q \notin \tau'(v')$,

then $v \ll v'$. Such an enumeration exists since V and thus also V' is countable. The mapping from V' to $\mathbb{Q}$ is constructed by induction on $\ll$ such that the following conditions are satisfied at every time during the construction: for all $v, v_1, v_2 \in V'$, we have that

(I) $v_1 \prec v_2$ and (v_1), (v_2) defined implies $(v_1) < (v_2)$ and

(II) (v) defined implies that $(v) \, P \, q'$ for all $P_q \in \tau'(v)$.

For the induction start, we set

$$(v) := q \text{ iff } =_q \in \tau'(v) \qquad \text{for all } v \in V'.$$

This operation is well-defined since G' satisfies Property 2 of consistency which implies that, for every $v \in V'$, there exist no $q_1, q_2 \in \mathbb{Q}$ such that $q_1 \neq q_2$ and $\{=_{q_1}, =_{q_2}\} \subseteq \tau'(v)$. Moreover, (I) and (II) are satisfied:

- if $v_1 \prec v_2$, then there exists a $<$-path $v'_0, \ldots, v'_k$ from v_1 to v_2, i.e. $v'_0 = v_1$, $v'_k = v_2$, and $(v_i, v_{i+1}, <) \in E'$ for $i < k$ (recall that G' contains no "$=$-edges"). Let $(v_1) = q$ and $(v_2) = q'$. By induction on k, it is straightforward to show that $>_q \in \tau'(v'_i)$ for $0 < i \leq k$: for the induction start, $=_q \in \tau'(v'_0)$ and the fact that G' satisfies Properties 2 and 3 of normality and Property 3 of consistency yields $>_q \in \tau'(v'_1)$. The induction step is analogous. Hence, $>_q \in \tau'(v_2)$. Since G' satisfies Property 2 of consistency, $=_{q'} \in \tau'(v_2)$ and $>_q \in \tau'(v_2)$ imply $q < q'$ what was to be shown.

- We make a case distinction according to the predicate P. If $=_q \in \tau'(v)$, then $(v) = q$ by definition of . Now let $<_q \in \tau'(v)$. If (v) is defined, then there exists some $q' \in \mathbb{Q}$ such that $=_{q'} \in \tau'(v)$. Since G' satisfies Property 2 of consistency, we have $(v) = q' < q$. The case $>_q \in \tau'(v)$ is analogous.

Now for the induction step. Fix a $v \in V'$ such that $\sigma(v)$ is undefined and $\sigma(v')$ is defined for all v' with $v' \ll v$. Define four sets as follows:

$$
\begin{aligned}
\Gamma_1 &:= \{\, \sigma(v') \mid v' \in V, v' \ll v \text{ and } v' \prec v\} \\
\Gamma_2 &:= \{q \mid {>_q} \in \tau'(v)\} \\
\Gamma_3 &:= \{\, \sigma(v') \mid v' \in V, v' \ll v \text{ and } v \prec v'\} \\
\Gamma_4 &:= \{q \mid {<_q} \in \tau'(v)\}
\end{aligned}
$$

Moreover, set $\Gamma_< := \Gamma_1 \cup \Gamma_2$ and $\Gamma_> := \Gamma_3 \cup \Gamma_4$. We distinguish four cases:

1. $\Gamma_>$ and $\Gamma_<$ are both empty. Then set $\sigma(v)$ to some arbitrary $q \in \mathbb{Q}$.

2. $\Gamma_> = \emptyset$ and $\Gamma_< \neq \emptyset$. Since $\mathbb{Q}$ has no maximum and $\Gamma_<$ is finite, there exists a $q \in \mathbb{Q}$ such that $q > \max(\Gamma_<)$. Set $\sigma(v) := q$.

3. $\Gamma_> \neq \emptyset$ and $\Gamma_< = \emptyset$. Since $\mathbb{Q}$ has no minimum and $\Gamma_>$ is finite, there exists a $q \in \mathbb{Q}$ such that $q < \min(\Gamma_>)$. Set $\sigma(v) := q$.

4. $\Gamma_> \neq \emptyset$ and $\Gamma_< \neq \emptyset$. We will show that $\max(\Gamma_<) < \min(\Gamma_>)$. By density of $\mathbb{Q}$, it then follows that there exists a $q \in \mathbb{Q}$ such that $\max(\Gamma_<) < q < \min(\Gamma_>)$. We set $\sigma(v) := q$. To show that $\max(\Gamma_<) < \min(\Gamma_>)$, we need to prove that $q_1 \in \Gamma_i$ and $q_2 \in \Gamma_j$ implies $q_1 < q_2$ for $(i,j) \in \{(1,3), (1,4), (2,3), (2,4)\}$:

   - $i = 1$ and $j = 3$. Then there exist $v_1, v_2 \in V'$ such that $v_1 \ll v$, $v_2 \ll v$, $v_1 \prec v$, $v \prec v_2$, $q_1 = \sigma(v_1)$, and $q_2 = \sigma(v_2)$. Clearly, $v_1 \prec v$ and $v \prec v_2$ implies $v_1 \prec v_2$. Since Property (I) is satisfied, we have $\sigma(v_1) < \sigma(v_2)$.

   - $i = 1$ and $j = 4$. Then there exists a v' such that $v' \ll v$, $v' \prec v$, $q_1 = \sigma(v')$, and ${<_{q_2}} \in \tau'(v)$. Since $v' \prec v$, there exists a $<$-path $v_0, \ldots, v_k$ from v' to v, i.e., $v_0 = v'$, $v_k = v$, and $(v_i, v_{i+1}, <) \in E'$ for $i < k$. By induction on k, it is straightforward to prove that ${<_{q_2}} \in \tau'(v_0)$: for the induction start, ${<_{q_2}} \in \tau'(v_k)$ and $(v_{k-1}, v_k, <) \in E'$ implies ${<_{q_2}} \in \tau'(v_{k-1})$ since G' satisfies Properties 2 and 3 of normality and Property 3 of consistency. The induction step is identical. Hence, ${<_{q_2}} \in \tau'(v')$. Since G' satisfies Property (II) and $q_1 = \sigma(v')$, we thus have $q_1 < q_2$.

   - $i = 2$ and $j = 3$. Analogous to the previous case.

   - $i = 2$ and $j = 4$. Then we have $q_1 < q_2$ by Property (II).

It is straightforward to check that the induction step preserves Properties (I) and (II). Moreover, the fact that σ satisfies Properties (I) and (II) clearly implies that σ is a solution for G' (again note that G' contains no "=-edges"). ❑

It is interesting to note that Theorem 3.7 also holds if satisfiability over $\mathbb{R}$ is considered instead of satisfiability over $\mathbb{Q}$ (the same proof works). However, as noted in [15], Theorem 3.7 does *not* hold if satisfiability over non-dense structures such as $\mathbb{N}$ is considered. We will return to this issue in Section 7.

Intuitively, every constraint graph $G = (V, E, \tau)$ can be converted into a normal one (called a *normalization* of G) by first specializing the relations in E and τ such that Conditions 1 and 2 of normality are satisfied and then augmenting τ such that Condition 3 holds.

Definition 3.8 (Normalization). A constraint graph $G = (V, E, \tau)$ is a *normalization* of the constraint graph $G' = (V, E', \tau')$ iff it is normal and the following conditions are satisfied:

1. $(v_1, v_2, P) \in E'$ with $P \in \{<, =\}$ implies $(v_1, v_2, P) \in E$,

2. $(v_1, v_2, >) \in E'$ implies $(v_2, v_1, <) \in E$,

3. $(v_1, v_2, \leq) \in E'$ implies $\{(v_1, v_2, <), (v_1, v_2, =)\} \cap E \neq \emptyset$,

4. $(v_1, v_2, \geq) \in E'$ implies $\{(v_2, v_1, <), (v_1, v_2, =)\} \cap E \neq \emptyset$,

5. $(v_1, v_2, \neq) \in E'$ implies $\{(v_1, v_2, <), (v_2, v_1, <)\} \cap E \neq \emptyset$,

6. if $(v, P_q) \in \tau$, then there exists a $v' \in V$ and a P' such that $(v', P'_q) \in \tau'$,

7. $(v, P_q) \in \tau'$ with $P \in \{<, =, >\}$ implies $(v, P_q) \in \tau$,

8. $(v, \leq_q) \in \tau'$ implies $\{(v, <_q), (v, =_q)\} \cap \tau \neq \emptyset$,

9. $(v, \geq_q) \in \tau'$ implies $\{(v, >_q), (v, =_q)\} \cap \tau \neq \emptyset$, and

10. $(v, \neq_q) \in \tau'$ implies $\{(v, <_q), (v, >_q)\} \cap \tau \neq \emptyset$.

$\Diamond$

4 Defining Hintikka-trees

In this section, we define Hintikka-trees, which are, as has already been noted, abstractions of canonical tree-models. Let us start with defining, for each concept C and role hierarchy $\mathcal{R}$, the set of concepts $\mathsf{cl}(C, \mathcal{R})$ that are "relevant" for deciding whether a given interpretation is a model of C and $\mathcal{R}$: for a given concept C and role hierarchy $\mathcal{R}$, we use $\mathsf{cl}(C, \mathcal{R})$ to denote the smallest set such that

1. $C \in \mathsf{cl}(C, \mathcal{R})$,

2. $\top \in \mathsf{cl}(C, \mathcal{R})$,

3. if $\forall R.D \in \mathsf{cl}(C, \mathcal{R})$, $\mathsf{Trans}(S)$, and $S \sqsubseteq^* R$, then $\forall S.D \in \mathsf{cl}(C, \mathcal{R})$, and

4. $\mathsf{cl}(C, \mathcal{R})$ is closed under subformulas and $\sim$ (c.f. Definition 3.1).

Note that $\sharp\mathsf{cl}(C, \mathcal{R})$ is polynomial in the length of C and the number of role inclusions in $\mathcal{R}$.

Hintikka-trees are defined in several steps. We start with introducing Hintikka-sets, which form the basis for the definition of so-called Hintikka-labels. As the name indicates, Hintikka-labels are used as node labels in Hintikka-trees. We then define Hintikka-tuples, which are tuples of Hintikka-labels that describe a valid label configuration for a node and its direct successors in a Hintikka-tree (Hintikka-tuples will also be rather convenient for defining looping automata that accept Hintikka-trees). Eventually, we use Hintikka-labels and Hintikka-tuples to define Hintikka-trees.

Intuitively, each node α of a Hintikka-tree T describes a domain element x of the corresponding canonical model $\mathcal{I}$. The node label of α consists of several parts, one of them a Hintikka-set. This Hintikka-set contains all concepts D from $\mathsf{cl}(C, \mathcal{R})$ such that $x \in D^{\mathcal{I}}$.

Definition 4.1 (Hintikka-set). Let C be a concept in PNF and $\mathcal{R}$ a role hierarchy. A set $\Psi \subseteq \mathsf{cl}(C, \mathcal{R})$ is a *Hintikka-set for C and $\mathcal{R}$* iff it satisfies the following conditions:

(S1) if $C_1 \sqcap C_2 \in \Psi$, then $\{C_1, C_2\} \subseteq \Psi$,

(S2) if $C_1 \sqcup C_2 \in \Psi$, then $\{C_1, C_2\} \cap \Psi \neq \emptyset$,

(S3) $\{A, \neg A\} \not\subseteq \Psi$ for all concept names $A \in \mathsf{cl}(C, \mathcal{R})$,

(S4) if $f \in \mathsf{N_{aF}}$ is used in C or $\mathcal{R}$, then $(\leqslant 1\ f\ \top) \in \Psi$,

(S5) if $(\leqslant n\ R\ D) \in \mathsf{cl}(C, \mathcal{R})$, then $\{D, {\sim}D\} \cap \Psi \neq \emptyset$,

(S6) $\top \in \Psi$

Concepts of the form $\exists R.D$, $(\geqslant n\ R\ D)$, $(\leqslant n\ R\ D)$, and $\exists Rg_1, g_2.P$ may appear either *marked* or *unmarked* in Ψ. $\Diamond$

The marking of concepts is a technical trick that allows us to deal with the inverse role constructor. Intuitively, edges of a Hintikka-tree T describe role successor relationships of the corresponding canonical model $\mathcal{I}$. If $\exists R.D$ occurs in the Hintikka-set of a node β, then there has to exist a "witness" for this concept: either (i) there exists a successor γ of β such that the edge from β to γ represents an R role relationship and D is in the Hintikka-set of γ, or (ii) β has a predecessor α, the edge from α to β represents an $\mathsf{Inv}(R)$ role relationship, and D occurs in the Hintikka-set of α. The marking of concepts is used for bookkeeping of these two possibilities: if $\exists R.D$ occurs *marked* in the Hintikka-set of β, then α is a "witness" for $\exists R.D$ and we do not need to enforce the existence of a witness among β's successors. The marking of $(\geqslant n\ R\ D)$, $(\leqslant n\ R\ D)$, and $\exists Rg_1, g_2.P$ concepts can be explained similarly. Hintikka-sets are one of the components of Hintikka-labels:

Definition 4.2 (Hintikka-label). Let C be a concept in PNF and $\mathcal{R}$ a role hierarchy. A *Hintikka-label $(\Psi, \omega, V, E, \tau)$ for C and $\mathcal{R}$* consists of

1. a Hintikka-set Ψ for C and $\mathcal{R}$,

2. a set $\omega \subseteq \mathsf{ROL}$ of roles occurring in C or $\mathcal{R}$, and

3. a constraint graph (V, E, τ) where $V \subseteq \mathsf{N_{cF}}$, every $g \in V$ occurs in C, and every q appearing in τ occurs in C.

such that

(N1) if $\exists g_1, g_2.P \in \Psi$, then $g_1, g_2 \in V$ and $(g_1, g_2, P) \in E$,

(N2) if $\forall g_1, g_2.P \in \Psi$ and $g_1, g_2 \in V$, then $(g_1, g_2, P) \in E$,

(N3) if $\exists g.P_q \in \Psi$, then $g \in V$ and $(g, P_q) \in \tau$,

(N4) $R \in \omega$ and $R \sqsubseteq^{*} S$ implies $S \in \omega$,

(N5) if $g_1, g_2 \in V$, then $\{(g_1, g_2, <), (g_1, g_2, =), (g_2, g_1, <)\} \cap E \neq \emptyset$, and

(N6) if q appears in C, then, for each $g' \in V$, there exists a $P' \in \{<, =, >\}$ such that $(g', P'_q) \in \tau$.

The set of all Hintikka-labels for C and $\mathcal{R}$ is denoted by $\Gamma_{C,\mathcal{R}}$. $\qquad \diamond$

Let us explain the intuition behind Hintikka-labels. If α is a node in a Hintikka-tree T, $\mathcal{I}$ the canonical model corresponding to T, and $x \in \Delta_\mathcal{I}$ the domain element associated with α, then the Hintikka-label $L = (\Psi, \omega, V, E, \tau)$ of α is a description of x in $\mathcal{I}$. More precisely, (i) the Hintikka-set Ψ is the set of concepts $D \in \mathsf{cl}(C, \mathcal{R})$ such that $x \in D^\mathcal{I}$ (ii) ω is the set of roles $R \in \mathsf{ROL}$ such that $(y, x) \in R^\mathcal{I}$, where y is the domain element corresponding to the precessor β of α in T; and (iii) the constraint graph (V, E, τ) describes the numerical successors of x and their relationships: if, for some $g \in \mathsf{N_{cF}}$, we have $g \in V$, then $g^\mathcal{I}(x)$ is defined. By **(N5)** and **(N6)**, (V, E, τ) fixes the relationship between any two numerical successors of x as well as the relationship between any numerical successor of x and any rational number q appearing in the input concept. By **(N1)**, **(N2)**, and **(N3)**, the relationships stated by E and τ are "consistent" with the Hintikka-set Ψ.

It is rather important that the constraint graph (V, E, τ) fixes the relationship between *any* two nodes of V: as already noted in Section 3, every Hintikka-tree T induces a (normal) constraint graph $G(T)$ that describes the "numerical part" of the canonical interpretation corresponding to T, and should thus be satisfiable. Since $G(T)$ is normal, by Theorem 3.7 it su ces to demand that $G(T)$ should be consistent. The *complete* determination of the relationships between nodes of the constraint graphs (V, E, τ) in Hintikka-labels will allow us to ensure the consistency of $G(T)$ using a *local* condition which can be verified by looping automata. This condition is part of the definition of Hintikka-tuples, which are introduced next.

Definition 4.3 (Tuple-graph, Hintikka-tuple). Let C be a concept in PNF and $\mathcal{R}$ a role hierarchy. With $\mathsf{b}_{C,\mathcal{R}}$, we denote

$$\sharp\{D \in \mathsf{cl}(C, \mathcal{R}) \mid D = \exists R.E \text{ or } D = \exists R g_1, g_2.P\} + \sum_{(\geqslant n \; R \; C) \in \mathsf{cl}(C,\mathcal{R})} n.$$

Let $\chi = (L_0, \ldots, L_{\mathsf{b}_{C,\mathcal{R}}})$ be an $\mathsf{b}_{C,\mathcal{R}}$+1-tuple of Hintikka-labels with $L_i = (\Psi_i, \omega_i, V_i, E_i, \tau_i)$ for $i \leq \mathsf{b}_{C,\mathcal{R}}$. A constraint graph $G = (V, E, \tau)$ is a *tuple-graph* for χ if

$$\begin{aligned} V &= V_0 \cup \{ig \mid 1 \leq i \leq \mathsf{b}_{C,\mathcal{R}} \text{ and } g \in V_i\} \\ E &\supseteq E_0 \cup \{(ig_1, ig_2, P) \mid 1 \leq i \leq \mathsf{b}_{C,\mathcal{R}} \text{ and } (g_1, g_2, P) \in E_i\} \\ \tau &= \tau_0 \cup \{(ig, P_q) \mid 1 \leq i \leq \mathsf{b}_{C,\mathcal{R}} \text{ and } (g, P_q) \in \tau_i\} \end{aligned}$$

such that

(G1) if $\exists Rg, g'.P$ is unmarked in Ψ_0, then there exists an i with $1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}$ such that $ig, g' \in V$, $R \in \omega_i$, and $(ig, g', P) \in E$,

(G2) if $\exists Rg, g'.P$ is marked in Ψ_i with $1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}$, then $g, ig' \in V$, $\mathsf{Inv}(R) \in \omega_i$, and $(g, ig', P) \in E$,

(G3) if $\forall Rg, g'.P \in \Psi_0$, $R \in \omega_i$, $g \in V_i$, and $g' \in V_0$ for some i with $1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}$, then $(ig, g', P) \in E$, and

(G4) if $\forall Rg, g'.P \in \Psi_i$, $\mathsf{Inv}(R) \in \omega_i$, $g \in V_0$, and $g' \in V_i$ for some i with $1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}$, then $(g, ig', P) \in E$.

The tuple χ is a *Hintikka-tuple* iff the following conditions are satisfied:

(M1) if $\exists R.D$ is unmarked in Ψ_0, then there exists an i with $1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}$ such that $R \in \omega_i$ and $D \in \Psi_i$,

(M2) if $(\geqslant n\ R\ D) \in \Psi_0$, then either

- $(\geqslant n\ R\ D)$ is unmarked in Ψ_0 and there exists a set $I \subseteq \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}$ of cardinality n such that, for each $i \in I$, we have $R \in \omega_i$ and $D \in \Psi_i$ or
- $(\geqslant n\ R\ D)$ is marked in Ψ_0 and there exists a set $I \subseteq \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}$ of cardinality $n - 1$ such that, for each $i \in I$, we have $R \in \omega_i$ and $D \in \Psi_i$,

(M3) if $\exists R.D$ or $(\geqslant n\ R\ D)$ is marked in Ψ_i with $1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}$, then $\mathsf{Inv}(R) \in \omega_i$ and $D \in \Psi_0$,

(M4) if $\forall R.D \in \Psi_0$ and $R \in \omega_i$ with $1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}$, then $D \in \Psi_i$,

(M5) if $\forall R.D \in \Psi_i$ and $\mathsf{Inv}(R) \in \omega_i$ with $1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}$, then $D \in \Psi_0$,

(M6) if $\forall R.D \in \Psi_0$, $S \in \omega_i$ with $1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}$, $\mathsf{Trans}(S)$, and $S \overset{*}{\sqsubseteq} R$, then $\forall S.D \in \Psi_i$,

(M7) if $\forall R.D \in \Psi_i$ and $\mathsf{Inv}(S) \in \omega_i$ with $1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}$, $\mathsf{Trans}(S)$, and $S \overset{*}{\sqsubseteq} R$, then $\forall S.D \in \Psi_0$,

(M8) if $(\leqslant n\ R\ D) \in \Psi_0$, then either

- $(\leqslant n\ R\ D)$ is unmarked in Ψ_0 and the cardinality of the set $\{i \mid 1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}, R \in \omega_i$ and $D \in \Psi_i\}$ is at most n or
- $(\leqslant n\ R\ D)$ is marked in Ψ_0 and the cardinality of the set $\{i \mid 1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}, R \in \omega_i$ and $D \in \Psi_i\}$ is at most $n - 1$,

(M9) if $D \in \Psi_0$, $\mathsf{Inv}(R) \in \omega_i$, and $(\leqslant n\ R\ D) \in \Psi_i$ for $1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}$, then $(\leqslant n\ R\ D)$ is marked in Ψ_i,

(M10) there exists a tuple-graph for χ that has a consistent normalization.

$\diamond$

Except for **(M10)**, which refers to tuple-graphs and is the aforementioned local condition enforcing consistency of induced constraint graphs, the properties of Hintikka-tuples should be quite easy to understand. Before we discuss tuple graphs and **(M10)** in more detail, let us introduce Hintikka-trees.

Definition 4.4 (Hintikka-tree). An $\mathsf{b}_{C,\mathcal{R}}$-ary $\Gamma_{C,\mathcal{R}}$-tree T with $T(\epsilon) = (\Psi_\epsilon, \omega_\epsilon, V_\epsilon, E_\epsilon, \tau_\epsilon)$ is a *Hintikka-tree for C and $\mathcal{R}$* iff it satisfies the following conditions:

(T1) $C \in \Psi_\epsilon$,

(T2) all concepts in Ψ_ϵ are unmarked, and

(T3) for all $\alpha \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$, the tuple $(T(\alpha), T(\alpha 1), \ldots, T(\alpha \mathsf{b}_{C,\mathcal{R}}))$ is a Hintikka-tuple.

Let T be a Hintikka-tree, $\alpha \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$ a node in T, and $T(\alpha) = (\Psi, \omega, V, E, \tau)$. We use $\Psi_T(\alpha)$ to denote Ψ and ω_T to denote ω.

$\diamond$

We can now return to the discussion of Property **(M10)**. As is apparent from their definition, tuple-graphs are built by taking the union of all the constraint graphs that appear as a part of the Hintikka-labels in a Hintikka-tuple. The constraint graph $G(T)$ induced by a Hintikka-tree T, in turn, is constructed from tuple-graphs: by **(T3)**, for each node α of T, the tuple

$$\chi_T(\alpha) := (T(\alpha), T(\alpha 1), \ldots, T(\alpha \mathsf{b}_{C,\mathcal{R}}))$$

is a Hintikka-tuple. By **(M10)**, there exists a tuple-graph $G_T(\alpha)$ for $\chi_T(\alpha)$ which has a consistent normalization $G_T^n(\alpha)$. Modulo some technical details, the constraint graph $G(T)$ induced by T can be viewed as the union of the constraint graphs $G_T^n(\alpha)$ for all nodes α of T. Figure 1 illustrates the relationship between the various constraint graphs involved. In the following section, we will prove that the consistency of the normalizations $G_T^n(\alpha)$ enforced by **(M10)** implies consistency of the constraint graph $G(T)$. The hardest part of this proof is to show that $G(T)$ satisfies Property 1 of consistency, i.e., that it contains no $<$-cycle: for this proof, it is crucial that

1. the tuple-graph $G_T(\alpha)$ overlaps with the tuple-graph $G_T(\beta)$ if β is a successor of α in T, and

2. the constraint graphs (V, E, τ), which are part of Hintikka-tuples and thus used to build of tuple-graphs, fix the relationship between *any* two elements of V as discussed above.

Using the fact that the constraint graphs induced by Hintikka-trees are consistent, we can then show that Hintikka-trees are indeed proper abstractions of $\mathbb{Q}\text{-}\mathcal{SHIQ}$-interpretations.
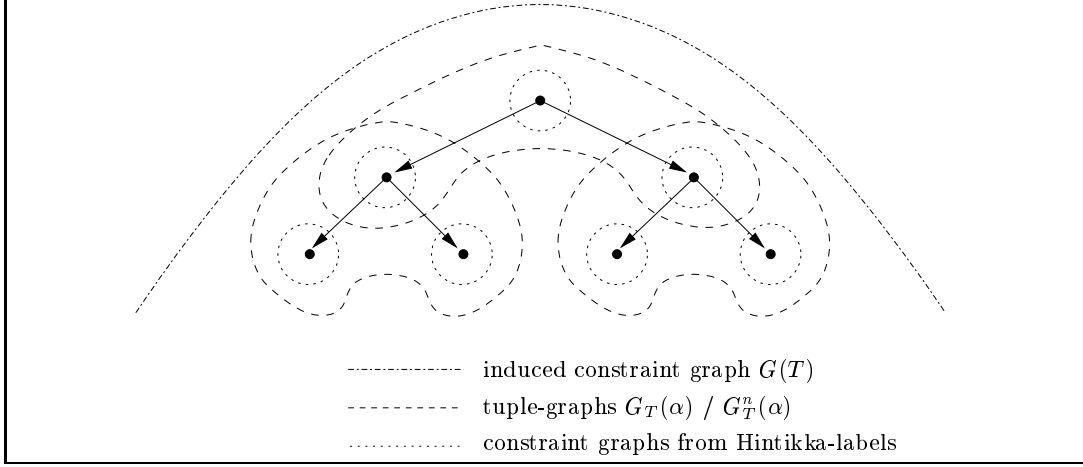
Figure 1: Hintikka-trees and constraint graphs.

5 Correctness and Complexity

In this section, we show that Hintikka-trees are proper abstractions of models, i.e., that a concept C and a role hierarchy $\mathcal{R}$ have a model iff C and $\mathcal{R}$ have a Hintikka-tree. We start with formally defining the constraint graph $G(T)$ induced by the Hintikka-tree T.

Definition 5.1 (Corresponding Constraint Graph). Let T be a Hintikka-tree. By Properties **(T3)** and **(M10)**, for each node α in T, there exists a tuple-graph

$$G_T(\alpha) = (V_T(\alpha), E_T(\alpha), \tau_T(\alpha))$$

for the Hintikka-tuple $(\alpha, \alpha 1, \ldots, \alpha \mathsf{b}_{C,\mathcal{R}})$, and $G_T(\alpha)$ has a consistent normalization

$$G_T^n(\alpha) = (V_T^n(\alpha), E_T^n(\alpha), \tau_T^n(\alpha)).$$

We define a constraint graph $G(T)$ *corresponding to* T. Its nodes have the form $\alpha | v$, where α is a node in T and $v \in V_T(\alpha)$. More precisely, $G(T)$ is defined as (V, E, τ), where

$$
\begin{aligned}
V &:= \bigcup_{\alpha \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*} \{\alpha | v \mid v \in V_T(\alpha)\} \\
E &:= \bigcup_{\alpha \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*} \{(\alpha | v, \alpha | v', P) \mid (v, v', P) \in E_T^n(\alpha)\} \\
&\quad \cup \mathsf{cl}_= \big(\{(\alpha | ig, \alpha i | g, =) \mid ig \in V_T(\alpha)\} \big) \\
\tau &:= \bigcup_{\alpha \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*} \{(\alpha | v, P_q) \mid (v, P_q) \in \tau_T^n(\alpha)\}
\end{aligned}
$$

It is not hard to see that $G(T)$ is well defined: if $ig \in V_T(\alpha)$, then, by definition of tuple-graphs, we have $g \in V_T(\alpha i)$.   ◇

We will use the naming conventions introduced in Definition 5.1 (i.e., $G_T(\alpha)$, $V_T(\alpha)$, $G_T^n(\alpha)$, $V_T^n(\alpha)$, etc.) throughout this section. Let us now establish a lemma that will be central for showing soundness, i.e., for showing that the existence of a Hintikka-tree for C and $\mathcal{R}$ implies the existence of a model of C and $\mathcal{R}$.

Lemma 5.2. *For every Hintikka-tree T, the corresponding constraint graph $G(T)$ is satisfiable.*

Proof By Theorem 3.7, it su ces to show that $G(T)$ is normal and consistent. We start with normality, i.e., we show that the the corresponding Properties 1 to 3 from Definition 3.6 are satisfied:

1. Property 1 is satisfied by construction of $G(T)$ since all constraint graphs $G_T^n(\alpha)$ are normal.

2. Property 2 is satisfied for the same reasons.

3. Let $(\alpha|v, P_q) \in \tau$ and $\alpha'|v' \in V$. We need to show that there exists a predicate $P^* \in \{<, =, >\}$ such that $(\alpha'|v', P_q^*) \in \tau$. By definition of $G(T)$, $v \in V_T(\alpha)$, $v' \in V_T(\alpha')$, and $(v, P_q) \in \tau_T^n(\alpha)$. By Property 6 of normalizations, there exists a $v'' \in V_T(\alpha)$ and a P'' such that $(v'', P_q'') \in \tau_T(\alpha)$. By definition of tuple-graphs and Hintikka-labels, this implies that q occurs in C. Since $v' \in V_T(\alpha')$, we can distinguish two cases by definition of tuple-graphs:

    - $v' = g$ for some $g \in \mathsf{N_{cF}}$. Then $v' \in V_T(\alpha')$, **(N6)**, and the fact that q occurs in C imply that there exists a $P^* \in \{<, =, >\}$ such that $(v', P_q^*) \in \tau(\alpha')$. By Property 7 of normalizations, this implies $(v', P_q^*) \in \tau_T^n(\alpha')$. By definition of $G(T)$, we have $(\alpha'|v', P_q^*) \in \tau$.
    - $v' = ig$ for some $i \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}$ and $g \in \mathsf{N_{cF}}$. Then $v' \in V_T(\alpha')$ and the definition of tuple-graphs implies $g \in V_T(\alpha'i)$. By **(N6)**, and the fact that q occurs in C, this yields the existence of a $P^* \in \{<, =, >\}$ such that $(g, P_q^*) \in \tau(\alpha'i)$. By definition of tuple-graphs, this implies $(ig, P_q^*) \in \tau(\alpha')$. Property 7 of normalizations yields $(ig, P_q^*) \in \tau_T^n(\alpha')$. By definition of $G(T)$ and since $v' = ig$, we obtain $(\alpha'|v', P_q^*) \in \tau$.

We now show that $G(T)$ is consistent, i.e., that it satisfies the corresponding Properties 1 to 3 from Definition 3.6. We start with the simpler Properties 2 and 3:

2. Let $\alpha|v \in V$. We need to show that there exists a $q \in \mathbb{Q}$ such that qPq' for all $P_{q'} \in \tau(\alpha|v)$. Since $G_T^n(\alpha)$ is normal and consistent, Theorem 3.7 implies that it is also satisfiable. Let be a solution for $G_T^n(\alpha)$ and set $q := (v)$. We show that q is as required. Let $P_{q'} \in \tau(\alpha|v)$. By definition of $G(T)$, we have $v \in V_T(\alpha)$ and $(v, P_{q'}) \in \tau_T^n(\alpha)$. Since is a solution for $G_T^n(\alpha)$, we clearly have $(v)\, P\, q'$.

3. Let $\alpha_1|v_1, \alpha_2|v_2 \in V$ and $(\alpha_1|v_1, \alpha_2|v_2, P) \in E$. We need to show that there exist $q_1, q_2 \in \mathbb{Q}$ such that $q_1\, P\, q_2$, $q_1\, P'\, q$ for all $P_q' \in \tau(\alpha_1|v_1)$, and $q_2\, P'\, q$ for all $P_q' \in \tau(\alpha_2|v_2)$. By definition of $G(T)$, we can make a case distinction as follows:

    - $\alpha_1 = \alpha_2$, $v_1, v_2 \in V_T(\alpha_1)$, and $(v_1, v_2, P) \in E_T^n(\alpha_1)$. Since $G_T^n(\alpha_1)$ is normal and consistent, Theorem 3.7 implies that it is also satisfiable. Let be a solution for $G_T^n(\alpha_1)$ and set $q_1 := (v_1)$ and $q_2 := (v_2)$. We show that q_1 and q_2 are as required. Since $(v_1, v_2, P) \in E_T^n(\alpha_1)$ and is a solution for $G_T^n(\alpha)$, we clearly have $q_1\, P\, q_2$. Now let $P_q \in \tau(\alpha_i|v_i)$ with $i \in \{1, 2\}$.

21

By definition of $G(T)$, we have $(v_i, P_q) \in \tau_T^n(\alpha_1)$. Since $\quad$ is a solution for $G_T^m(\alpha_1)$, this clearly implies $q_i \, P \, q$.

- $\alpha_2 = \alpha_1 i$, $v_1 = ig$, $v_2 = g$, P is "=", and $ig \in V_T(\alpha_1)$. Since $G_T^n(\alpha_1)$ is normal and consistent, Theorem 3.7 implies that it is also satisfiable. Let $\quad$ be a solution for $G_T^m(\alpha_1)$ and set $q_1 := q_2 := \quad(ig)$. We show that q_1 and q_2 are as required. Clearly, $q_1 \, P \, q_2$ is satisfied. Let $P_q' \in \tau(\alpha_1|v) = \tau(\alpha_1|ig)$. By definition of $G(T)$, we have $(ig, P_q') \in \tau_T^n(\alpha_1)$. Since $\quad$ is a solution for $G_T^m(\alpha_1)$, we have $q_1 \, P' \, q$. Now let $P_q' \in \tau(\alpha_2|v_2) = \tau(\alpha_1 i|g)$. By definition of $G(T)$, we have $(g, P_q') \in \tau_T^n(\alpha_1 i)$. In the following, we show that this implies $(ig, P_q') \in \tau_T^n(\alpha_1)$, which is clearly sufficient to prove that $q_2 \, P' \, q$ since $q_2 = \quad(ig)$ and $\quad$ is a solution for $G_T^m(\alpha_1)$. By Property 6 of normalizations and definition of tuple-graphs and Hintikka-tuples, $(g, P_q') \in \tau_T^n(\alpha_1 i)$ implies that q occurs in C. Since $g \in V_T(\alpha_1 i)$, this together with **(N6)** implies that there exists a $P^* \in \{<, =, >\}$ such that $(g, P_q^*) \in \tau_T(\alpha_1 i)$. By Property 7 of normalizations, we obtain $(g, P_q^*) \in \tau_T^n(\alpha_1 i)$. Since $G_T^n(\alpha_1 i)$ is normal and consistent, $(g, P_q') \in \tau_T^n(\alpha_1 i)$ and $(g, P_q^*) \in \tau_T^n(\alpha_1 i)$ implies $P' = P^*$. Thus, $(g, P_q') \in \tau_T(\alpha_1 i)$, which implies $(ig, P_q') \in \tau_T^n(\alpha_1)$ by definition of tuple-graphs and Hintikka-tuples and Property 7 of normalizations.

- $\alpha_1 = \alpha_2 i$, $v_2 = ig$, $v_1 = g$, P is "=", and $ig \in V_T(\alpha_2)$. Analogous to the previous case.

To show that $G(T)$ is consistent and thus satisfiable over $\mathbb{Q}$, it remains to prove that Property 1 is satisfied. Assume to the contrary that $G(T)$ contains a $<$-cycle and that $O = \alpha_0|v_0, \ldots, \alpha_n|v_n$ is the $<$-cycle in $G(T)$ with minimal length. Fix a $t \leq n$ such that

$$\text{for each } i \text{ with } i \leq n \text{ and each } \beta \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^+, \text{ we have } \alpha_i \neq \alpha_t \beta, \qquad (*)$$

i.e., there exist no α_i in O such that α_t is a true prefix of α_i (such a t exists since O is of finite length). Since O is a $<$-cycle, there exists an $s \leq n$ such that we have $(\alpha_s|v_s, \alpha_{s+}|v_{s+}, <) \in E$. We make a case distinction and derive a contradiction in either case.

- $\alpha_s \neq \alpha_t$. Define a sequence of nodes O' from O by deleting all nodes $\alpha_i|v_i$ with $\alpha_i = \alpha_t$. O' is non-empty since $\alpha_s \neq \alpha_t$. We show that O' is a $<$-cycle in $G(T)$ which is a contradiction to the minimality of O. Let $O' = \alpha_0'|v_0', \ldots, \alpha_m'|v_m'$. By definition of $G(T)$, the fact that $(\alpha_s|v_s, \alpha_{s+}|v_{s+}, <) \in E$ implies $\alpha_{s+} = \alpha_s$. Since $\alpha_s \neq \alpha_t$, $\alpha_s|v_s$ and $\alpha_{s+}|v_{s+}$ are in O' and it remains to show that O' is a cycle in $G(T)$, i.e., for all $i \leq m$, we have $(\alpha_i'|v_i', \alpha_{i+}'|v_{i+}', P) \in E$ for some $P \in \{<, =\}$.

  Let $\alpha_i'|v_i'$ and $\alpha_{i+}'|v_{i+}'$ be nodes in O'. If these two nodes are already neighbor nodes in O, we are obviously done. Hence, assume that this is not the case. By construction of O', this implies the existence of a path

  $$\alpha_i'|v_i', \alpha_t|v_1^*, \ldots, \alpha_t|v_x^*, \alpha_{i+}'|v_{i+}'$$

  in $G(T)$ of length at most n. Since $\alpha_i' \neq \alpha_t$ and $\alpha_{i+}' \neq \alpha_t$, by construction of $G(T)$ and by $(*)$, we have that

22

1. there exists a $\beta \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$ and a $j \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}$ such that $\alpha_t = \beta j$ and $\alpha'_i = \alpha'_{i+} = \beta$,

2. $v'_i = jg$, $v^*_1 = g$, $v^*_x = g'$, and $v'_{i+} = jg'$ for some $g, g' \in \mathsf{N}_{\mathsf{cF}}$, and

3. $(\beta|jg, \beta j|g, =) \in E$ and $(\beta|jg', \beta j|g', =) \in E$.

By definition of $G(T)$, Point 3 implies $jg, jg' \in V_T(\beta)$. By **(N5)**, and Property 1 of normalizations, this implies that either

(a) $(jg, jg', P) \in E_T(\beta) \cap E^n_T(\beta)$ for some $P \in \{<, =\}$ or

(b) $(jg', jg, <) \in E_T(\beta) \cap E^n_T(\beta)$.

First assume that (a) holds. Together with Point 1 and 2 and the definition of $G(T)$, $(jg, jg', P) \in E^n_T(\beta)$ obviously implies $(\alpha'_i|v'_i, \alpha'_{i+}|v'_{i+}, P) \in E$ and we are done. Moreover, case (b) leads to a contradiction, i.e., it cannot occur: by definition of tuple-graphs, $(jg', jg, <) \in E_T(\beta)$ implies $(g', g, <) \in E_T(\beta j)$. By Property 1 of normalizations, we thus have $(g', g, <) \in E^n_T(\beta j)$. By definition of $G(T)$ and Point 1 and 2, this implies that $(\alpha_t|v^*_x, \alpha_t|v^*_1, <) \in E$. Hence, the path $\alpha_t|v^*_1, \ldots, \alpha_t|v^*_x$ is a $<$-cycle in $G(T)$ of length at most $n - 2$ which contradicts the minimality of O.

- $\alpha_s = \alpha_t$. We first show that there exists a node $\alpha_z|v_z$ in O such that $\alpha_z \neq \alpha_t$. For suppose that no such node exists. Then, by definition of $G(T)$, $v_0, \ldots, v_n$ is a $<$-cycle in $G^n_T(\alpha_t)$. This, however, contradicts the fact that $G^n_T(\alpha_t)$ is consistent. Hence, we may conclude the existence of an α_z as above. Define a sequence of nodes O' from O by deleting all nodes $\alpha_i|v_i$ with $\alpha_i \neq \alpha_t$. O' is non-empty since $\alpha_s = \alpha_t$. Moreover, O' is shorter than O due to the existence of α_z. We show that O' is a $<$-cycle in $G(T)$ which is a contradiction to the minimality of O. Let $O' = \alpha_t|v'_0, \ldots, \alpha_t|v'_m$. By definition of $G(T)$, the fact that $(\alpha_s|v_s, \alpha_{s+}|v_{s+}, <) \in E$ implies $\alpha_{s+} = \alpha_s = \alpha_t$. Hence, it remains to show that O' is a cycle in $G(T)$, i.e., that, for all $i \leq m$, we have $(\alpha_t|v'_i, \alpha_t|v'_{i+}, P) \in E$ for some $P \in \{<, =\}$.

Let $\alpha_t|v'_i$ and $\alpha_t|v'_{i+}$ be nodes in O'. If these two nodes are already neighbor nodes in O, we are obviously done. Hence, assume that this is not the case. By construction of O', this implies the existence of a path

$$\alpha_t|v'_i, \alpha^*_1|v^*_1, \ldots, \alpha^*_x|v^*_x, \alpha_t|v'_{i+}$$

in $G(T)$ of length at most n such that $\alpha^*_i \neq \alpha_t$ for $1 \leq i \leq x$. By construction of $G(T)$ and by $(*)$, we have that

1. there exists a $\beta \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$ and a $j \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}$ such that $\alpha_t = \beta j$ and $\alpha^*_1 = \alpha^*_x = \beta$,

2. $v'_i = g$, $v^*_1 = jg$, $v^*_x = jg'$, and $v'_{i+} = g'$ for some $g, g' \in \mathsf{N}_{\mathsf{cF}}$, and

3. $(\beta|jg, \beta j|g, =) \in E$ and $(\beta|jg', \beta j|g', =) \in E$.

By definition of $G(T)$ and by Point 3, both jg and jg' are nodes in $V_T(\beta)$. By **(N5)** and Property 1 of normalizations, this implies that either

23

(a) $(jg', jg, <) \in E_T(\beta) \cap E_T^n(\beta)$ or

(b) $(jg, jg', P) \in E_T(\beta) \cap E_T^n(\beta)$ for some $P \in \{<, =\}$.

Case (a) cannot occur: together with Point 1 and 2 and the definition of $G(T)$, $(jg', jg, <) \in E_T^n(\beta)$ implies $(\alpha_x^* | v_x^*, \alpha_1^* | v_1^*, <) \in E$. Hence, the path $\alpha_1^* | v_1^*, \ldots, \alpha_x^* | v_x^*$ is a $<$-cycle in $G(T)$ of length at most $n - 2$ which contradicts the minimality of O. Hence, assume that (b) holds. By definition of tuple-graphs, $(jg, jg', P) \in E_T(\beta)$ implies $(g, g', P) \in E_T(\beta j)$. By Property 7 of normalizations, this, in turn, implies $(g, g', P) \in E_T^n(\beta j)$. By construction of $G(T)$ and Point 1 and 2, we thus obtain $(\alpha_t | v_i', \alpha_t | v_{i+}', P) \in E$ what was to be shown.

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ ❑

We can now prove soundness.

Lemma 5.3. *If there exists a Hintikka-tree for a concept C in PNF and a role hierarchy $\mathcal{R}$, then C is satisfiable w.r.t. $\mathcal{R}$.*

Proof Let C be a concept, $\mathcal{R}$ a role hierarchy, and $\mathsf{b}_{C,\mathcal{R}}$ as in Definition 4.3. Moreover, let T be a Hintikka-tree for C and $\mathcal{R}$. By Lemma 5.2, the corresponding constraint graph $G(T) = (V, E, \tau)$ is satisfiable and thus has a solution. For each $R \in \mathsf{ROL}$, set

$$\mathcal{E}(R) = \{(\alpha, \beta) \mid \beta = \alpha i \text{ and } R \in \omega_T(\beta)\} \cup \{(\alpha, \beta) \mid \alpha = \beta i \text{ and } \mathsf{Inv}(R) \in \omega_T(\alpha)\}.$$

We define an interpretation $\mathcal{I} = (\Delta_\mathcal{I}, \cdot^\mathcal{I})$ as follows:

$$
\begin{aligned}
\Delta_\mathcal{I} &= \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^* \\
A^\mathcal{I} &= \{\alpha \mid A \in \Psi_T(\alpha)\} \text{ for all } A \in C_N \\
R^\mathcal{I} &= \mathcal{E}(R) \cup \bigcup_{P \, \underline{\mathbb{E}} \, R, \mathsf{Trans}(P)} \mathcal{E}(P)^+ \text{ for all } R \in \mathsf{N_R} \\
g^\mathcal{I} &= \{(\alpha, x) \mid \alpha | g \in V \text{ and } (\alpha | g) = x\} \text{ for all } g \in \mathsf{N_{cF}}.
\end{aligned}
$$

Before we show that $\mathcal{I}$ is well-defined, we establish a claim:

Claim 1: If $R \in \mathsf{ROL}$ is simple, then $R^\mathcal{I} = \mathcal{E}(R)$.

Proof: If $R \in \mathsf{N_R}$ is simple, then $R^\mathcal{I} = \mathcal{E}(R)$ is an immediate consequence of the definition of $R^\mathcal{I}$. Now let $R = S^-$ be simple with $S \in \mathsf{N_R}$. Then it is readily checked that $(S^-)^\mathcal{I} = (S^\mathcal{I})^- = \mathcal{E}(S)^- = \mathcal{E}(S^-)$.

We now show that $\mathcal{I}$ is well-defined:

- For all $R \in \mathsf{N_{tR}}$, $R^\mathcal{I}$ is transitive. Let $\{(\alpha_1, \beta_1), (\alpha_2, \beta_2)\} \subseteq R^\mathcal{I}$ where $\beta_1 = \alpha_2$. Then, for each $i \in \{1, 2\}$, either (i) $(\alpha_i, \beta_i) \in \mathcal{E}(R)$ or (ii) there exists a role P_i such that $P_i \, \underline{\mathbb{E}} \, R$, $\mathsf{Trans}(P_i)$, and $(\alpha_i, \beta_i) \in \mathcal{E}(P_i)^+$. In Case (ii), $(\alpha_i, \beta_i) \in \mathcal{E}(P_i)^+$ implies $(\alpha_i, \beta_i) \in \mathcal{E}(R)^+$ by **(N4)** and definition of $\mathcal{E}$. Hence, $\{(\alpha_1, \beta_1), (\alpha_2, \beta_2)\} \subseteq \mathcal{E}(R)^+$ which obviously implies $(\alpha_1, \beta_2) \in \mathcal{E}(R)^+$. Since $R \, \underline{\mathbb{E}} \, R$ and $\mathsf{Trans}(R)$, we have $\mathcal{E}(R)^+ \subseteq R^\mathcal{I}$ and thus $(\alpha_1, \beta_2) \in R^\mathcal{I}$.

- For all $f \in \mathsf{N}_{\mathsf{aF}}$, $f^{\mathcal{I}}$ is functional. For assume to the contrary that, for some $f \in \mathsf{N}_{\mathsf{aF}}$, we have $\{(\alpha, \beta_1), (\alpha, \beta_2)\} \subseteq f^{\mathcal{I}}$ with $\beta_1 \neq \beta_2$. By assumption (c.f. Section 3.1), f is simple. Thus, Claim 1, the non-emptyness of $f^{\mathcal{I}}$, and the definition of $\mathcal{E}$ and of Hintikka-tuples imply that f occurs in C or $\mathcal{R}$. Moreover, Claim 1 clearly implies that $\{(\alpha, \beta_1), (\alpha, \beta_2)\} \subseteq \mathcal{E}(f)$. By **(S4)** and since f occurs in C or $\mathcal{R}$, we have $(\leqslant 1\ f\ \top) \in \Psi_T(\alpha)$. We distinguish two cases: either there exists a $j \in \{1, 2\}$ such that $\alpha = \beta_j i$ and $\mathsf{Inv}(f) \in \omega_T(\alpha)$ (for some i) or (ii) no such j exists. In Case (i), $(\leqslant 1\ f\ \top)$ is marked in $\Psi_T(\alpha)$ by **(S6)** and **(M9)**. Thus, by **(M8)** and **(S6)** there exists no i such that $f \in \omega_T(\alpha i)$. Since T is a tree, this is a contradiction to the fact that $(\alpha, \beta_\ell) \in \mathcal{E}(f)$, where $\ell \in \{1, 2\}$ such that $\ell \neq j$. In Case (ii), there clearly exist i_1 and i_2 such that $\beta_j = \alpha i_j$ and $f \in \omega_T(\beta_j)$ for $j \in \{1, 2\}$. Since $\beta_1 \neq \beta_2$, we have $i_1 \neq i_2$. This, however, is a contradiction to **(M8)**, **(S6)**, and the fact that $(\leqslant 1\ f\ \top) \in \Psi_T(\alpha)$.

- For all $g \in \mathsf{N}_{\mathsf{cF}}$, $g^{\mathcal{I}}$ is obviously functional.

We now prove that $\mathcal{I}$ is a model of $\mathcal{R}$. Hence, let $R \sqsubseteq S$ and $(\alpha, \beta) \in R^{\mathcal{I}}$. Then either (i) $(\alpha, \beta) \in \mathcal{E}(R)$ or (ii) there exists a role P such that $P \mathbin{\underline{\underline{*}}} R$, $\mathsf{Trans}(P)$, and $(\alpha, \beta) \in \mathcal{E}(P)^+$. In Case (i), the definition of $\mathcal{E}$, **(N4)**, and $R \mathbin{\underline{\underline{*}}} S$ yields $(\alpha, \beta) \in \mathcal{E}(S) \subseteq S^{\mathcal{I}}$. In Case (ii), we clearly have $P \mathbin{\underline{\underline{*}}} S$ and thus $(\alpha, \beta) \in S^{\mathcal{I}}$ by definition of $S^{\mathcal{I}}$.

It remains to prove that there exists an $\alpha \in \Delta_{\mathcal{I}}$ such that $\alpha \in C^{\mathcal{I}}$. To do this, we prove the following claim:

Claim 2: $D \in \Psi_T(\alpha)$ implies $\alpha \in D^{\mathcal{I}}$ for all $\alpha \in \Delta_{\mathcal{I}}$ and $D \in \mathsf{cl}(C, \mathcal{R})$.

Proof: The claim is proved by induction on the *norm* $||\cdot||$ of concepts D which is defined inductively as follows:

$$
\begin{aligned}
||A|| &:= ||\neg A|| &&:= 0 \text{ for } A \in \mathsf{N}_{\mathsf{C}} \\
||C_1 \sqcap C_2|| &:= ||C_1 \sqcup C_2|| &&:= 1 + ||C_1|| + ||C_2|| \\
||\exists R.D|| &:= ||\forall R.D|| &&:= 1 + ||D|| \\
||(\geqslant n\ R\ D)|| &:= ||(\leqslant n\ R\ D)|| &&:= 1 + ||D|| \\
||\exists u_1, u_2.P|| &:= ||\forall u_1, u_2.P|| &&:= 0 \\
||\exists R g_1, g_2.P|| &:= ||\forall R g_1, g_2.P|| &&:= 0 \\
||\exists g.P_q|| &:= 0
\end{aligned}
$$

First for the induction start, which splits into several subcases:

- D is a concept name. Immediate by definition of $\mathcal{I}$.

- $D = \neg E$. Since C is in NNF and by definition of $\mathsf{cl}()$, D is in NNF. Hence, E is a concept name. By definition of $\mathcal{I}$ and since $\Psi_T(\alpha)$ is a Hintikka-set and thus satisfies **(S3)**, we have $\alpha \in (\neg E)^{\mathcal{I}}$.

- $D = \exists U_1, U_2.P$. We distinguish three subcases:

  1. $U_1 = g_1$ and $U_2 = g_2$. By **(N1)** and definition of tuple-graphs, we have $g_1, g_2 \in V_T(\alpha)$ and $(g_1, g_2, P) \in E_T(\alpha)$. By Properties 1 to 5 of normalizations, this implies that there exists a $P' \in \{<, =\}$ such that $q_1 P' q_2$ implies

25

$q_1 P q_2$ for all $q_1, q_2 \in \mathbb{Q}$ and $(g_1, g_2, P') \in E^n_T(\alpha)$. By definition of $G(T)$, $g_1, g_2 \in V_T(\alpha)$ and $(g_1, g_2, P') \in E^n_T(\alpha)$ implies that $\alpha|g_1, \alpha|g_2 \in V$ and $(\alpha|g_1, \alpha|g_2, P') \in E$. By definition of $\mathcal{I}$, we have $g^{\mathcal{I}}_1(\alpha) = (\alpha|g_1)$ and $g^{\mathcal{I}}_2(\alpha) = (\alpha|g_2)$. Since is a solution for $G(T)$ and $(\alpha|g_1) P' (\alpha|g_2)$ implies $(\alpha|g_1) P (\alpha|g_2)$, this yields $\alpha \in (\exists g_1, g_2.P)^{\mathcal{I}}$.

2. $U_1 = Rg_1$, $U_2 = g_2$, and $\exists Rg_1, g_2.P$ is unmarked in $\Psi_T(\alpha)$. By **(G1)** and since $\exists Rg_1, g_2.P$ is unmarked, there exists an i such that $ig_1, g_2 \in V_T(\alpha)$, $R \in \omega_T(\alpha i)$, and $(ig_1, g_2, P) \in E_T(\alpha)$. By Properties 1 to 5 of normalizations, this implies that there exists a $P' \in \{<, =\}$ such that $q_1 P' q_2$ implies $q_1 P q_2$ for all $q_1, q_2 \in \mathbb{Q}$ and $(ig_1, g_2, P') \in E^n_T(\alpha)$. By definition of $G(T)$, $ig_1, g_2 \in V_T(\alpha)$ and $(ig_1, g_2, P') \in E^n_T(\alpha)$ yields $\alpha|ig_1, \alpha|g_2 \in V$ and $(\alpha|ig_1, \alpha|g_2, P') \in E$. Moreover, $ig_1 \in V_T(\alpha)$ implies $(\alpha|ig_1, \alpha i|g_1, =) \in E$. By definition of $\mathcal{I}$ and since $R \in \omega_T(\alpha i)$, we have $(\alpha, \alpha i) \in R^{\mathcal{I}}$, $g^{\mathcal{I}}_1(\alpha i) = (\alpha i|g_1)$, and $g^{\mathcal{I}}_2(\alpha) = (\alpha|g_2)$. Since $(\alpha|ig_1, \alpha i|g_1, =) \in E$ and is a solution for $G(T)$, we have $(\alpha i|g_1) = (\alpha|ig_1)$. Together with $(\alpha, \alpha i) \in R^{\mathcal{I}}$ and $g^{\mathcal{I}}_1(\alpha i) = (\alpha i|g_1)$, this yields $(\alpha|ig_1) \in (Rg_1)^{\mathcal{I}}(\alpha)$. Moreover, since is a solution for $G(T)$ and $(\alpha|ig_1, \alpha|g_2, P') \in E$, we have $(\alpha|ig_1) P' (\alpha|g_2)$, which implies $(\alpha|ig_1) P (\alpha|g_2)$. Summing up, $(\alpha|ig_1) \in (Rg_1)^{\mathcal{I}}(\alpha)$, $g^{\mathcal{I}}_2(\alpha) = (\alpha|g_2)$, and $(\alpha|ig_1) P (\alpha|g_2)$ yields $\alpha \in (\exists Rg_1, g_2.P)^{\mathcal{I}}$.

3. $U_1 = Rg_1$, $U_2 = g_2$, and $\exists Rg_1, g_2.P$ is marked in $\Psi_T(\alpha)$. By **(T2)**, there exists a β such that $\alpha = \beta i$. By **(G2)**, we have $g_1, ig_2 \in V_T(\beta)$, $\mathsf{Inv}(R) \in \omega_T(\alpha)$, and $(g_1, ig_2, P) \in V_T(\beta)$. By Properties 1 to 5 of normalizations, this implies that there exists a $P' \in \{<, =\}$ such that $q_1 P' q_2$ implies $q_1 P q_2$ for all $q_1, q_2 \in \mathbb{Q}$ and $(g_1, ig_2, P') \in E^n_T(\beta)$. By definition of $G(T)$, $g_1, ig_2 \in V_T(\beta)$ and $(g_1, ig_2, P') \in E^n_T(\beta)$ implies that $\beta|g_1, \beta|ig_2 \in V$ and $(\beta|g_1, \beta|ig_2, P') \in E$. By definition of $\mathcal{I}$ and since $\mathsf{Inv}(R) \in \omega_T(\alpha)$, we have $(\alpha, \beta) \in R^{\mathcal{I}}$, $g^{\mathcal{I}}_1(\beta) = (\beta|g_1)$, and $g^{\mathcal{I}}_2(\alpha) = (\alpha|g_2)$. This implies $(\beta|g_1) \in (Rg_1)^{\mathcal{I}}(\alpha)$. By definition of $G(T)$, $ig_2 \in V_T(\beta)$ implies $(\beta|ig_2, \alpha|g_2, =) \in E$. Since is a solution for $G(T)$, we thus have $g^{\mathcal{I}}_2(\alpha) = (\beta|ig_2)$. Moreover, since $(\beta|g_1, \beta|ig_2, P') \in E$ and is a solution for $G(T)$, we have $(\beta|g_1) P' (\beta|ig_2)$ which implies $(\beta|g_1) P (\beta|ig_2)$. Summing up, $(\beta|g_1) \in (Rg_1)^{\mathcal{I}}(\alpha)$, $g^{\mathcal{I}}_2(\alpha) = (\beta|ig_2)$, and $(\beta|g_1) P (\beta|ig_2)$ yields $\alpha \in (\exists Rg_1, g_2.P)^{\mathcal{I}}$.

- $D = \forall U_1, U_2.P$. Two subcases can be distinguished:

  1. $U_1 = g_1$ and $U_2 = g_2$. If $g^{\mathcal{I}}_1(\alpha)$ or $g^{\mathcal{I}}_2(\alpha)$ is undefined, then $\alpha \in (\forall g_1, g_2.P)^{\mathcal{I}}$ is trivially satisfied. Hence assume that $g^{\mathcal{I}}_1(\alpha)$ and $g^{\mathcal{I}}_2(\alpha)$ are defined. By definition of $\mathcal{I}$, this implies that $\alpha|g_1 \in V$, $g^{\mathcal{I}}_1(\alpha) = (\alpha|g_1)$, $\alpha|g_2 \in V$, and $g^{\mathcal{I}}_2(\alpha) = (\alpha|g_2)$. By definition of $G(T)$, $\alpha|g_1, \alpha|g_2 \in V$ implies $g_1, g_2 \in V_T(\alpha)$. By **(N2)** and construction of tuple-graphs, we thus have $(g_1, g_2, P) \in E_T(\alpha)$. By Properties 1 to 5 of normalizations, there exists a $P' \in \{<, =\}$ such that $q_1 P' q_2$ implies $q_1 P q_2$ for all $q_1, q_2 \in \mathbb{Q}$ and $(g_1, g_2, P') \in E^n_T(\alpha)$. The latter implies that $(\alpha|g_1, \alpha|g_2, P') \in E$. Since is a solution for $G(T)$, we thus have $(\alpha|g_1) P' (\alpha|g_2)$ which yields

$(\alpha|g_1)P\ (\alpha|g_2)$. Together $g_1^{\mathcal{I}}(\alpha) = (\alpha|g_1)$ and $g_2^{\mathcal{I}}(\alpha) = (\alpha|g_2)$, we obtain $\alpha \in (\forall g_1, g_2.P)^{\mathcal{I}}$.

2. $U_1 = Rg_1$ and $U_2 = g_2$. Let $x_1 \in (Rg_1)^{\mathcal{I}}(\alpha)$ and $x_2 \in g_2^{\mathcal{I}}(\alpha)$. By definition of $\mathcal{I}$, we have $\alpha|g_2 \in V$ and $x_2 = (\alpha|g_2)$. Since, by definition of $\mathbb{Q}$-$\mathcal{SHIQ}$-concepts, R is simple, Claim 1 and the definition of $\mathcal{E}$ implies that one of the following two cases applies:

   - There exists an $i \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}$ such that $\alpha i|g_1 \in V$, $R \in \omega_T(\alpha i)$ and $x_1 = (\alpha i|g_1)$. By definition of $G(T)$, $\alpha i|g_1, \alpha|g_2 \in V$ implies $g_1 \in V_T(\alpha i)$ and $g_2 \in V_T(\alpha)$. By **(G3)**, we thus have $(ig_1, g_2, P) \in E_T(\alpha)$. By Properties 1 to 5 of normalizations, there exists a $P' \in \{<, =\}$ such that $q_1 P' q_2$ implies $q_1 P q_2$ for all $q_1, q_2 \in \mathbb{Q}$ and $(ig_1, g_2, P') \in E_T^n(\alpha)$. The latter implies $(\alpha|ig_1, \alpha|g_2, P') \in E$. By definition of $G(T)$, $ig_1 \in V_T(\alpha)$ implies $(\alpha|ig_1, \alpha i|g_1, =) \in E$. Since is a solution for $G(T)$ and $x_1 = (\alpha i|g_1)$, this yields $x_1 = (\alpha|ig_1)$. Moreover, since $(\alpha|ig_1, \alpha|g_2, P') \in E$, we have $(\alpha|ig_1)P'\ (\alpha|g_2)$ implying $(\alpha|ig_1)P\ (\alpha|g_2)$. Hence, $x_1 P x_2$.

   - There exists a β such that $\alpha = \beta i$, $\beta|g_1 \in V$, $\mathsf{Inv}(R) \in \omega_T(\alpha)$, and $x_1 = (\beta|g_1)$. By definition of $G(T)$, $\beta|g_1, \alpha|g_2 \in V$ implies $g_1 \in V_T(\beta)$ and $g_2 \in V_T(\alpha)$. By **(G4)**, we thus have $(g_1, ig_2) \in E_T(\beta)$. By Properties 1 to 5 of normalizations, there exists a $P' \in \{<, =\}$ such that $q_1 P' q_2$ implies $q_1 P q_2$ for all $q_1, q_2 \in \mathbb{Q}$ and $(g_1, ig_2, P') \in E_T^n(\beta)$. The latter implies $(\beta|g_1, \beta|ig_2, P') \in E$. By definition of $G(T)$, $ig_2 \in V_T(\beta)$ implies $(\beta|ig_2, \alpha|g_2, =) \in E$. Since is a solution for $G(T)$ and $x_2 = (\alpha|g_2)$, this yields $x_2 = (\beta|ig_2)$. Moreover, since $(\beta|g_1, \beta|ig_2, P') \in E$, we have $(\beta|g_1)P'\ (\beta|ig_2)$ implying $(\beta|g_1)P\ (\beta|ig_2)$. Hence, $x_1 P x_2$.

   Since this holds independently of the choice of x_1 and x_2, we have $\alpha \in (\forall Rg_1, g_2.P)^{\mathcal{I}}$.

- $D = \exists g.P_q$. By **(N3)**, this implies $g \in V_T(\alpha)$ and $(g, P_q) \in \tau_T(\alpha)$. By Properties 7 to 10 of normalizations, there exists a $P' \in \{<, =, >\}$ such that $q' P' q$ implies $q' P q$ for all $q' \in \mathbb{Q}$, and $(g, P_q') \in \tau_T^n(\alpha)$. By definition of $G(T)$, we have $\alpha|g \in V$ and $(\alpha|g, P_q') \in \tau$. By definition of $\mathcal{I}$, $\alpha|g \in V$ implies $g^{\mathcal{I}}(\alpha) = (\alpha|g)$. Since is a solution for $G(T)$ and $(\alpha|g, P_q') \in \tau$, we have $(\alpha|g) P' q$, which implies $(\alpha|g) P q$. Summing up, we obtain $\alpha \in (\exists g.P_q)^{\mathcal{I}}$.

For the induction step, we make a case distinction according to the topmost operator in D. Assume $D \in \Psi_T(\alpha)$.

- $D = C_1 \sqcap C_2$ or $D = C_1 \sqcup C_2$. Straightforward by **(S1)** and **(S2)** of Hintikka-sets and by induction hypothesis.

- $D = \exists R.E$. We can distinguish two cases:

  1. D is unmarked in $\Psi_T(\alpha)$. By **(M1)**, there exists an i such that $R \in \omega_T(\alpha i)$ and $E \in \Psi_T(\alpha i)$. By definition of $R^{\mathcal{I}}$, we have $(\alpha, \alpha i) \in R^{\mathcal{I}}$. By induction, we obtain $\alpha i \in E^{\mathcal{I}}$. Thus, $\alpha \in D^{\mathcal{I}}$.

27

2. D is marked in $\Psi_T(\alpha)$. By **(T2)**, there exists a β such that $\alpha = \beta i$. By **(M3)**, we have $\mathsf{Inv}(R) \in \omega_t(\alpha)$ and $E \in \Psi_T(\beta)$. By definition of $R^{\mathcal{I}}$, we have $(\alpha, \beta) \in R^{\mathcal{I}}$. By induction, we obtain $\beta \in E^{\mathcal{I}}$. Thus, $\alpha \in D^{\mathcal{I}}$.

- $D = \forall R.E$. Let $(\alpha, \beta) \in R^{\mathcal{I}}$. By definition of $R^{\mathcal{I}}$, one of the following to cases applies:

  1. $(\alpha, \beta) \in \mathcal{E}(R)$. By definition of $\mathcal{E}$, this implies that there exists an i such that either (i) $\beta = \alpha i$ and $R \in \omega_T(\beta)$ or (ii) $\alpha = \beta i$ and $\mathsf{Inv}(R) \in \omega_T(\alpha)$. In Subcase (i), **(M4)** yields $E \in \Psi_T(\beta)$. In Subcase (ii), **(M5)** yields $E \in \Psi_T(\beta)$.

  2. there exists a role P such that $P \mathrel{\underline{\overline{\ast}}} R$, $\mathsf{Trans}(P)$, and $(\alpha, \beta) \in \mathcal{E}(P)^+$. Then there exist $\alpha_0, \ldots, \alpha_k$ such that $\alpha = \alpha_0$, $\beta = \alpha_k$, and $(\alpha_i, \alpha_{i+1}) \in \mathcal{E}(P)$ for $i < k$. If $k = 1$, then the definition of $\mathcal{E}$ and **(N4)** yields $(\alpha, \beta) \in \mathcal{E}(R)$ and we can continue as in Case 1. Hence assume $k > 1$. Using the fact that $\forall R.E \in \Psi_T(\alpha_0)$, the definition of $\mathcal{E}$, **(M6)**, and **(M7)**, it is readily checked that $\forall P.E \in \Psi_T(\alpha_i)$ for all i with $0 < i \leq k$. As in Case 1, we may now use the fact that $\forall P.E \in \Psi_T(\alpha_{k-1})$, the definition of $\mathcal{E}$, **(M4)**, and **(M5)** to show that $E \in \Psi_T(\alpha_k) = \Psi_T(\beta)$. By induction, we obtain $\beta \in E^{\mathcal{I}}$. Since this holds independently of the choice of β, we conclude $\alpha \in D^{\mathcal{I}}$.

- $D = (\geqslant n\ R\ E)$. First assume that D is unmarked in $\Psi_T(\alpha)$. By **(M2)**, there exists a set $I \subseteq \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}$ of cardinality n such that, for each $i \in I$, we have $R \in \omega_T(\alpha i)$ and $E \in \Psi_T(\alpha i)$. By definition of $R^{\mathcal{I}}$, we have $(\alpha, \alpha i) \in R^{\mathcal{I}}$ for every $i \in I$. By induction, we obtain $\alpha i \in E^{\mathcal{I}}$ for every $i \in I$. Thus, $\sharp\{\beta \mid (\alpha, \beta) \in R^{\mathcal{I}} \text{ and } \beta \in E^{\mathcal{I}}\} \geq n$ and $\alpha \in D^{\mathcal{I}}$.

  Now assume that D is marked in $\Psi_T(\alpha)$. By **(M2)**, there exists a set $I \subseteq \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}$ of cardinality $n - 1$ such that, for each $i \in I$, we have $R \in \omega_T(\alpha i)$ and $E \in \Psi_T(\alpha i)$. By **(T2)**, there exists a β such that $\alpha = \beta i$. By **(M3)**, we have $\mathsf{Inv}(R) \in \omega_T(\alpha)$ and $E \in \Psi_T(\beta)$. The definition of $R^{\mathcal{I}}$ implies that $(\alpha, \alpha i) \in R^{\mathcal{I}}$ for every $i \in I$ and $(\alpha, \beta) \in R^{\mathcal{I}}$. By induction, we obtain $\alpha i \in E^{\mathcal{I}}$ for every $i \in I$ and $\beta \in E^{\mathcal{I}}$. Since, clearly, $\beta \neq \alpha i$ for all $i \in I$, we thus have $\sharp\{\beta \mid (\alpha, \beta) \in R^{\mathcal{I}} \text{ and } \beta \in E^{\mathcal{I}}\} \geq n$ and conclude $\alpha \in D^{\mathcal{I}}$.

- $D = (\leqslant n\ R\ E)$. Since R is simple by definition of $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concepts, Claim 1 yields $R^{\mathcal{I}} = \mathcal{E}(R)$, i.e., the set $\{\beta \mid (\alpha, \beta) \in R^{\mathcal{I}} \text{ and } \beta \in E^{\mathcal{I}}\}$, whose cardinality we must show to be at most n, can be written as the union of the following two sets:
$$\begin{aligned} S_1 &= \{\alpha i \mid R \in \omega_T(\alpha i) \text{ and } \alpha i \in E^{\mathcal{I}}\} \\ S_2 &= \{\beta \mid \alpha = \beta i,\ \mathsf{Inv}(R) \in \omega_T(\alpha),\ \text{and } \beta \in E^{\mathcal{I}}\}. \end{aligned}$$

First assume that D is unmarked in $\Psi_T(\alpha)$. By **(M9)**, this implies that either (i) $\alpha = \epsilon$, (ii) $\mathsf{Inv}(R) \notin \omega_T(\alpha)$ or (iii) $E \notin \Psi_T(\beta)$ where $\alpha = \beta i$. In all three cases, S_2 is empty: this is obviously true for (i) and (ii); in Case (iii), **(S5)** yields $\sim E \in \Psi_T(\beta)$, and by induction, we obtain $\beta \in (\sim E)^{\mathcal{I}}$ implying $\beta \notin E^{\mathcal{I}}$. By **(M8)**, $\sharp S_1$ is bounded by n: **(M8)** yields

$$\sharp\{\alpha i \mid R \in \omega_T(\alpha i) \text{ and } E \in \Psi_T(\alpha i)\} \leq n. \tag{$*$}$$

28

Now assume that $\sharp S_1$ is greater than the cardinality of the set in the inequality $(*)$ which implies that there exists an i such that $E \notin \Psi_T(\alpha i)$ but $\alpha i \in E^{\mathcal{I}}$. By **(S5)**, $E \notin \Psi_T(\alpha i)$ yields $\sim E \in \Psi_T(\alpha i)$. From the induction hypothesis, we obtain $\alpha i \in (\sim E)^{\mathcal{I}}$, a contradiction. We thus conclude that $\sharp S_1$ is bounded by n which yields $\alpha \in D^{\mathcal{I}}$.

Now assume that D is marked in $\Psi_T(\alpha)$. Since T is a tree, we have $\sharp S_2 \leq 1$. Similar to the unmarked case, we can use **(M8)** and **(S5)** to deduce that $\sharp S_1 \leq n - 1$. We thus conclude $\sharp\{\beta \mid (\alpha, \beta) \in R^{\mathcal{I}} \text{ and } \beta \in E^{\mathcal{I}}\} \leq n$ and $\alpha \in D^{\mathcal{I}}$.

This completes the proof of the claim. Since $C \in \Psi_T(\epsilon)$, it is an immediate consequence of Claim 2 that $\mathcal{I}$ is a model of C. ❑

We now prove completeness.

Lemma 5.4. *If a concept C in PNF is satisfiable w.r.t. a role hierarchy $\mathcal{R}$, then there exists a Hintikka-tree for C and $\mathcal{R}$.*

Proof Let C be a concept, $\mathcal{R}$ a role hierarchy, and $\mathsf{b}_{C,\mathcal{R}}$ as in Definition 4.3. Moreover, let $\mathcal{I}$ be a model of C and $\mathcal{R}$, i.e., $\mathcal{I}$ is an interpretation such that there exists an $x_{\mathsf{root}} \in \Delta_{\mathcal{I}}$ with $x_{\mathsf{root}} \in C^{\mathcal{I}}$ and we have $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$ for all $R \sqsubseteq S \in \mathcal{R}$.

Before we define a Hintikka-tree T for C and $\mathcal{R}$, we introduce two auxiliary functions: (i) for every $x \in \Delta_{\mathcal{I}} \cup \{@\}$ and $y \in \Delta_{\mathcal{I}}$, we set $\mathcal{I}(x, y) := (\Psi, \omega, V, E, \tau)$, where

$$
\begin{aligned}
\Psi &:= \{D \in \mathsf{cl}(C, \mathcal{R}) \mid y \in D^{\mathcal{I}}\} \\
V &:= \{g \in \mathsf{N_{cF}} \mid g \text{ occurrs in } C \text{ and } g^{\mathcal{I}}(y) \text{ defined}\} \\
\omega &:= \begin{cases} \{R \mid R \text{ occurs in } C \text{ or } \mathcal{R} \text{ and } (x, y) \in R^{\mathcal{I}}\} & \text{if } x \neq @ \\ \emptyset & \text{otherwise} \end{cases} \\
E &:= \{(g_1, g_2, P) \mid g_1, g_2 \in V \text{ and } g_1^{\mathcal{I}}(y) \, P \, g_2^{\mathcal{I}}(y)\} \\
\tau &:= \{(g, P_q) \mid g \in V, \ q \text{ occurs in } C, \text{ and } g^{\mathcal{I}}(y) \, P \, q\}.
\end{aligned}
$$

(ii) for every $x \in \Delta_{\mathcal{I}}$, we fix a set $W(x) \subseteq \Delta_{\mathcal{I}}$ of witnesses such that

1. if $x \in (\exists R.D)^{\mathcal{I}}$, then there exists a $y \in W(x)$ such that $(x, y) \in R^{\mathcal{I}}$ and $y \in D^{\mathcal{I}}$,

2. if $x \in (\geqslant n \, R \, D)^{\mathcal{I}}$, then there exists a subset $\{y_0, \ldots, y_{n-1}\} \subseteq W(x)$ such that $(x, y_i) \in R^{\mathcal{I}}$ and $y_i \in D^{\mathcal{I}}$ for $i < k$,

3. if $x \in (\leqslant n \, R \, D)^{\mathcal{I}}$, then $\sharp\{y \in W(x) \mid (x, y) \in R^{\mathcal{I}} \text{ and } y \in D^{\mathcal{I}}\} \leq n$,

4. if $x \in (\exists R g_1, g_2.P)^{\mathcal{I}}$, then there exists a $y \in W(x)$ such that $(x, y) \in R^{\mathcal{I}}$ and $g_1^{\mathcal{I}}(y) \, P \, g_2^{\mathcal{I}}(x)$, and

5. the cardinality of $W(x)$ is at most $\mathsf{b}_{C,\mathcal{R}}$.

Using the semantics of $\mathbb{Q}\text{-}\mathcal{SHIQ}$-concepts and the definition of $\mathsf{b}_{C,\mathcal{R}}$, it is not hard to check that, for every $x \in \Delta_{\mathcal{I}}$, such a set of witnesses does indeed exist.

We now inductively define a Hintikka-tree T for C and $\mathcal{R}$, i.e., an $\mathsf{b}_{C,\mathcal{R}}$-ary $\Gamma_{C,\mathcal{R}}$-tree that satisfies **(T1)** to **(T3)**. To simplify the construction of T, along with T we

define a mapping π from $\{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$ to $\Delta_{\mathcal{I}}$ that keeps track of the correspondence between nodes in T and elements of $\Delta_{\mathcal{I}}$.

For the induction start, set

$$
\begin{array}{rcl}
\pi(\epsilon) & := & x_{\mathsf{root}} \\
T(\epsilon) & := & \mathcal{I}(@, x_{\mathsf{root}})
\end{array}
$$

where all concepts are unmarked in $\Psi_T(\epsilon)$. Now for the induction step. Let $\alpha \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$ be a word of minimal length such that $\pi(\alpha)$ is defined and $\pi(\alpha i)$ is undefined for all $i \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}$. Set $S := W(\pi(\alpha))$ if $\pi(\alpha) = x_{\mathsf{root}}$ and $S := W(\pi(\alpha)) \setminus \{\pi(\beta)\}$ if $\alpha = \beta i$. Moreover, let $S = \{x_1, \ldots, x_n\}$. Then do the following:

1. For all $i \leq n$, set $\pi(\alpha i) := x_i$ and $T(\alpha i) := \mathcal{I}(\pi(\alpha), x_i)$. A concept $\exists R.E$, $(\leqslant n\ R\ E)$, or $(\geqslant n\ R\ E)$ is marked in $\Psi_T(\alpha i)$ iff $\mathsf{Inv}(R) \in \omega_T(\alpha i)$ and $E \in \Psi_T(\alpha)$. A concept $\exists R g_1, g_2.P$ is marked in $\Psi_T(\alpha i)$ iff $\mathsf{Inv}(R) \in \omega_T(\alpha i)$ and $g_1^{\mathcal{I}}(\pi(\alpha))\, P\, g_2^{\mathcal{I}}(x_i)$.

2. For all i with $n < i \leq \mathsf{b}_{C,\mathcal{R}}$, set $\pi(\alpha i) := x_{\mathsf{root}}$ and $T(\alpha i) := \mathcal{I}(@, x_{\mathsf{root}})$. All concepts are unmarked in $\Psi_T(\alpha i)$.

Clearly, T is a $\mathsf{b}_{C,\mathcal{R}}$-ary tree. To show that T is a Hintikka-tree for C and $\mathcal{R}$, we need to prove that T is a $\Gamma_{C,\mathcal{R}}$-tree and that **(T1)** to **(T3)** are satisfied. To show that T is a $\Gamma_{C,\mathcal{R}}$-tree, in turn, we must prove that every node label is a Hintikka-label: clearly, if $T(\alpha) = (\Psi, \omega, V, E, \tau)$ for some node α, then ω is a set of roles occurring in C or $\mathcal{R}$ and (V, E, τ) is a constraint graph as required by Definition 4.3. It thus remains to show that Ψ is a Hintikka-set for C and $\mathcal{R}$, i.e., that it satisfies **(S1)** to **(S6)**, and that $T(\alpha)$ satisfies **(N1)** to **(N6)**. Both tasks, however, are trivial using the definition of $\mathcal{I}(x, y)$.

It thus remains to show that **(T1)** to **(T3)** are satisfied. This is easy for **(T1)** and **(T2)** using the definition of $T(\epsilon)$. In order to show that **(T3)** is satisfied, we have to prove that, for all $\alpha \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$, the tuple $(T(\alpha), T(\alpha 1), \ldots, T(\alpha \mathsf{b}_{C,\mathcal{R}}))$ is a Hintikka-tuple, i.e., that it satisfies conditions **(M1)** to **(M10)**. Before we do this, let us state two properties that can easily be seen to hold by considering the definitions of $\mathcal{I}(\cdot, \cdot)$, T, and π:

(a) for all $\alpha \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$, we have $\Psi_T(\alpha) = \{D \mid D \in \mathsf{cl}(C, \mathcal{R}) \text{ and } \pi(\alpha) \in D^{\mathcal{I}}\}$;

(b) for all $\alpha i \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$, we have $\omega_T(\alpha i) = \{R \text{ occurrs in } C \text{ or } \mathcal{R} \text{ and}$
$$(\pi(\alpha), \pi(\alpha i)) \in R^{\mathcal{I}}\}.$$

We now show that conditions **(M1)** to **(M10)** are satisfied.

(M1) Let $\exists R.D \in \Psi_T(\alpha)$. By (a), we have $\pi(\alpha) \in (\exists R.D)^{\mathcal{I}}$. Thus, by Property 1 of $W(\cdot)$, there exists a $y \in W(\pi(\alpha))$ such that $(\pi(\alpha), y) \in R^{\mathcal{I}}$ and $y \in D^{\mathcal{I}}$. If $\exists R.D$ is unmarked in $\Psi_T(\alpha)$, then we either have (i) $\pi(\alpha) = x_{\mathsf{root}}$ (induction start or Case 2 of the induction step) or (ii) there exists a β such that $\alpha = \beta i$ and we have either $\mathsf{Inv}(R) \notin \omega_T(\alpha)$ or $D \notin \Psi_T(\beta)$. In Case (i), we clearly

30

have $y \in S = W(\pi(\alpha))$ during the induction step generating successors for α. In Case (ii), we have $(\pi(\alpha), \pi(\beta)) \notin R^{\mathcal{I}}$ or $\pi(\beta) \notin D^{\mathcal{I}}$ by (b) and (a). This implies $y \neq \pi(\beta)$ and thus we have $y \in S$ during the induction step generating successors for α in both Case (i) and (ii). Hence, by construction of T there exists an $i \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}$ such that $\pi(\alpha i) = y$, $R \in \omega_T(\alpha i)$, and $D \in \Psi_T(\alpha i)$.

(M2) Let $(\geqslant n\ R\ D) \in \Psi_T(\alpha)$. By (a), we have $\pi(\alpha) \in (\geqslant n\ R\ D)^{\mathcal{I}}$. Thus, by Property 2 of $W(\cdot)$, there exists a subset $\{y_0, \ldots, y_{n-1}\} \subseteq W(\pi(\alpha))$ such that $(\pi(\alpha), y_i) \in R^{\mathcal{I}}$ and $y_i \in D^{\mathcal{I}}$ for $i < k$. We can distinguish two cases:

1. $(\geqslant n\ R\ D)$ is unmarked in $\Psi_T(\alpha)$. Then we have either (i) $\pi(\alpha) = x_{\mathsf{root}}$ or (ii) there exists a β such that $\alpha = \beta i$ and we have either $R \notin \omega_T(\alpha)$ or $D \notin \Psi_T(\beta)$. Similar to the **(M1)** case, in both cases we have $\{y_0, \ldots, y_{n-1}\} \subseteq S$ during the induction step generating successors for α. By construction of T, this clearly implies

$$\sharp\{i \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^* \mid R \in \omega_T(\alpha i) \text{ and } D \in \Psi_T(\alpha i)\} \geq n$$

as required.

2. $(\geqslant n\ R\ D)$ is marked in $\Psi_T(\alpha)$. Then there exists a $\beta \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$ such that $\mathsf{Inv}(R) \in \omega_T(\alpha)$ and $D \in \Psi_T(\beta)$. Since we may or may not have $\pi(\beta) \in W(\pi(\alpha))$, it is easy to see that

$$\sharp\{y \in S \mid (\pi(\alpha), y) \in R^{\mathcal{I}} \text{ and } y_i \in D^{\mathcal{I}}\} \in \{n-1, n\}$$

during the induction step generating successors for α. By construction of T, we thus have

$$\sharp\{i \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^* \mid R \in \omega_T(\alpha i) \text{ and } D \in \Psi_T(\alpha i)\} \geq n-1$$

as required.

(M3) Immediate from the construction of T.

(M4) Let $\forall R.D \in \Psi_T(\alpha)$ and $R \in \omega_T(\alpha i)$ for some $i \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$. By (a) and (b), this implies $\pi(\alpha) \in (\forall R.D)^{\mathcal{I}}$ and $(\pi(\alpha), \pi(\alpha i)) \in R^{\mathcal{I}}$. By $\mathbb{Q}\text{-}\mathcal{SHIQ}$ semantics, we have $\pi(\alpha i) \in D^{\mathcal{I}}$. Again by (a), we obtain $D \in \Psi_T(\alpha i)$.

(M5) Let $\forall R.D \in \Psi_T(\alpha)$ and $\mathsf{Inv}(R) \in \omega_T(\alpha)$. Since $\omega_T(\alpha) \neq \emptyset$, there exist β and i such that $\alpha = \beta i$. By (a) and (b), $\forall R.D \in \Psi_T(\alpha)$ and $\mathsf{Inv}(R) \in \omega_T(\alpha)$ implies $\pi(\alpha) \in (\forall R.D)^{\mathcal{I}}$ and $(\pi(\alpha), \pi(\beta)) \in R^{\mathcal{I}}$. By $\mathbb{Q}\text{-}\mathcal{SHIQ}$ semantics, we have $\pi(\beta) \in D^{\mathcal{I}}$. Again by (a), we obtain $D \in \Psi_T(\beta)$.

(M6) Let $\forall R.D \in \Psi_T(\alpha)$, $S \in \omega_T(\alpha i)$ with $1 \leq i \leq \mathsf{b}_{C,\mathcal{R}}$, $\mathsf{Trans}(S)$, and $S \sqsubseteq^* R$. We show that $\pi(\alpha i) \in (\forall S.D)^{\mathcal{I}}$ which, by (a), implies $\forall S.D \in \Psi_T(\alpha i)$ what needs to be shown. To this end, assume $(\pi(\alpha i), x) \in S^{\mathcal{I}}$. By (b), $S \in \omega_T(\alpha i)$ implies $(\pi(\alpha), \pi(\alpha i)) \in S^{\mathcal{I}}$. Since $\mathsf{Trans}(S)$ holds, together with $(\pi(\alpha i), x) \in S^{\mathcal{I}}$ this implies $(\pi(\alpha), x) \in S^{\mathcal{I}} \subseteq R^{\mathcal{I}}$. By (a), $\forall R.D \in \Psi_T(\alpha)$ implies $\pi(\alpha) \in (\forall R.D)^{\mathcal{I}}$. By $\mathbb{Q}\text{-}\mathcal{SHIQ}$ semantics, we obtain $x \in D^{\mathcal{I}}$. Since this holds independently of the choice of x, we conclude $\pi(\alpha i) \in (\forall S.D)^{\mathcal{I}}$.

(M7) Let $\forall R.D \in \Psi_T(\alpha)$ and $\mathsf{Inv}(S) \in \omega_T(\alpha)$, $\mathsf{Trans}(S)$, and $S \sqsubseteq^* R$. Since $\omega_T(\alpha) \neq \emptyset$, there exist β and i such that $\alpha = \beta i$. We show that $\pi(\beta) \in \forall S.D$ which, by (a), implies $\forall S.D \in \Psi_T(\beta)$ what needs to be shown. Hence, assume $(\pi(\beta), x) \in S^{\mathcal{I}}$. By (b), $\mathsf{Inv}(S) \in \omega_T(\alpha)$ implies $(\pi(\alpha), \pi(\beta)) \in S^{\mathcal{I}}$. Since $\mathsf{Trans}(S)$ holds, together with $(\pi(\beta), x) \in S^{\mathcal{I}}$ this implies $(\pi(\alpha), x) \in S^{\mathcal{I}} \subseteq R^{\mathcal{I}}$. By (a), $\forall R.D \in \Psi_T(\alpha)$ implies $\pi(\alpha) \in (\forall R.D)^{\mathcal{I}}$. By $\mathbb{Q}\text{-}\mathcal{SHIQ}$ semantics, we obtain $x \in D^{\mathcal{I}}$. Since this holds independently of the choice of x, we conclude $\pi(\beta) \in (\forall S.D)^{\mathcal{I}}$.

(M8) Let $(\leqslant n\ R\ D) \in \Psi_T(\alpha)$. By (a), we have $\pi(\alpha) \in (\leqslant n\ R\ D)^{\mathcal{I}}$. Thus, Property 3 of $W(\pi(\alpha))$ yields $\sharp K \leq n$, where

$$K := \{y \in W(\pi(\alpha)) \mid (\pi(\alpha), y) \in R^{\mathcal{I}} \text{ and } y \in D^{\mathcal{I}}\} \leq n.$$

We make a case distinction as follows:

1. $(\leqslant n\ R\ D)$ is unmarked in $\Psi_T(\alpha)$. We clearly have $S \subseteq W(\pi(\alpha))$ during the induction step generating successors for α. This obviously implies

$$\{y \in S \mid (\pi(\alpha), y) \in R^{\mathcal{I}} \text{ and } y \in D^{\mathcal{I}}\} \subseteq K.$$

   It is easy to check that this, together with $\sharp K \leq n$, implies

$$\sharp\{i \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}} \mid R \in \omega_T(\alpha i) \text{ and } D \in \Psi_T(\alpha i)\} \leq n$$

   as required.

2. $(\leqslant n\ R\ D)$ is marked in $\Psi_T(\alpha)$. Then there exists a $\beta \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$ such that $\mathsf{Inv}(R) \in \omega_T(\alpha i)$ and $D \in \Psi_T(\beta)$. First assume $\pi(\beta) \in K$. This implies that we have

$$\{y \in S \mid (\pi(\alpha), y) \in R^{\mathcal{I}} \text{ and } y \in D^{\mathcal{I}}\} = K \setminus \{\pi(\beta)\}$$

   during the induction step generating successors for α. Together with $\sharp K \leq n$, we obtain

$$\sharp\{i \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}} \mid R \in \omega_T(\alpha i) \text{ and } D \in \Psi_T(\alpha i)\} < n$$

   as required. Now assume $\pi(\beta) \notin K$. Since $\mathsf{Inv}(R) \in \omega_T(\alpha i)$ and $D \in \Psi_T(\beta)$, we have $(\pi(\alpha), \pi(\beta)) \in R^{\mathcal{I}}$ and $\pi(\beta) \in D^{\mathcal{I}}$ by (a) and (b). Since $\pi(\alpha) \in (\leqslant n\ R\ D)^{\mathcal{I}}$ and $\pi(\beta) \notin K$, $\mathbb{Q}\text{-}\mathcal{SHIQ}$ semantics implies that $\sharp K < n$. Moreover, $\pi(\beta) \notin K$ implies

$$\{y \in S \mid (\pi(\alpha), y) \in R^{\mathcal{I}} \text{ and } y \in D^{\mathcal{I}}\} = K$$

   during the induction step generating successors for α. Together with $\sharp K < n$, this yields

$$\sharp\{i \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}} \mid R \in \omega_T(\alpha i) \text{ and } D \in \Psi_T(\alpha i)\} < n$$

   as required.

(M9) Immediate from the construction of T.

(M10) Let $T(\alpha) = (\Psi_0, \omega_0, V_0, E_0, \tau_0)$ and let $T(\alpha i) = (\Psi_i, \omega_i, V_i, E_i, \tau_i)$ for $1 \le i \le \mathsf{b}_{C,\mathcal{R}}$. We first establish the following claim:

Claim: if $\exists R g_1, g_2.P$ appears unmarked in Ψ_0, then there exists an $i \in \{1, \dots, \mathsf{b}_{C,\mathcal{R}}\}$ such that $g_1 \in V_i$, $g_2 \in V_0$, $R \in \omega_i$ and $g_1^{\mathcal{T}}(\pi(\alpha i)) \, P \, g_2^{\mathcal{T}}(\pi(\alpha))$.

Proof: Assume that $\exists R g_1, g_2.P$ appears unmarked in Ψ_0. By (a), we have $\pi(\alpha) \in (\exists R g_1, g_2.P)^{\mathcal{I}}$. Thus, by Property 4 of $W(\cdot)$, there exists a $y \in W(\pi(\alpha))$ such that $(\pi(\alpha), y) \in R^{\mathcal{I}}$ and $g_1^{\mathcal{T}}(y) \, P \, g_2^{\mathcal{T}}(\pi(\alpha))$ (implying that $g_1^{\mathcal{T}}(y)$ and $g_2^{\mathcal{T}}(\pi(\alpha))$ are defined). By definition of T, $\exists R g_1, g_2.P$ appearing unmarked in Ψ_0 implies that either (i) $\pi(\alpha) = x_{\mathsf{root}}$ or (ii) there exists a β such that $\alpha = \beta i$ and we have either $\mathsf{Inv}(R) \notin \omega_0$ or $g_1^{\mathcal{T}}(\pi(\beta)) \, P \, g_2^{\mathcal{T}}(\pi(\alpha))$ does not hold. In Case (i), we clearly have $y \in S = W(\pi(\alpha))$ during the induction step generating successors for α. In Case (ii), (b) implies that $(\pi(\alpha), \pi(\beta)) \notin R^{\mathcal{I}}$ or $g_1^{\mathcal{T}}(\pi(\beta)) \, P \, g_2^{\mathcal{T}}(\pi(\alpha))$ does not hold. This implies $y \ne \pi(\beta)$ and thus we have $y \in S$ during the induction step generating successors for α in both Case (i) and (ii). Hence, there exists an $i \in \{1, \dots, \mathsf{b}_{C,\mathcal{R}}\}$ such that $\pi(\alpha i) = y$, $R \in \omega_i$, and $g_1^{\mathcal{T}}(\pi(\alpha i)) \, P \, g_2^{\mathcal{T}}(\pi(\alpha))$. Moreover, since $g_1^{\mathcal{T}}(y)$ and $g_2^{\mathcal{T}}(\pi(\alpha))$ are defined, the construction of T and definition of $\mathcal{I}(\cdot, \cdot)$ implies that $g_1 \in V_i$ and $g_2 \in V_0$.

For each $\exists R g_1, g_2.P$ appearing unmarked in Ψ_0, the $i \in \{1, \dots, \mathsf{b}_{C,\mathcal{R}}\}$ provided by the claim is denoted by $\lambda(\exists R g_1, g_2.P)$.

We define a constraint graph $G = (V, E, \tau)$ as follows:

$$
\begin{aligned}
V &:= V_0 \cup \{ig \mid 1 \le i \le \mathsf{b}_{C,\mathcal{R}} \text{ and } g \in V_i\} \\
E &:= F \cup E_0 \cup \{(ig_1, ig_2, P) \mid 1 \le i \le \mathsf{b}_{C,\mathcal{R}} \text{ and } (g_1, g_2, P) \in E_i\} \\
\tau &:= \tau_0 \cup \{(ig, P_q) \mid 1 \le i \le \mathsf{b}_{C,\mathcal{R}} \text{ and } (g, P_q) \in \tau_i\}
\end{aligned}
$$

where F is defined in the following way:

$$
\begin{aligned}
F := \ & \{(ig_1, g_2, P) \mid \exists R g_1, g_2.P \text{ appears unmarked in } \Psi_0 \text{ and } \lambda(\exists R g_1, g_2.P) = i\} \\
& \cup \{(g_1, ig_2, P) \mid \exists R g_1, g_2.P \text{ appears marked in } \Psi_i\} \\
& \cup \{(ig_1, g_2, P) \mid \forall R g_1, g_2.P \in \Psi_0, R \in \omega_i, g_1 \in V_i, \text{ and } g_2 \in V_0\} \\
& \cup \{(g_1, ig_2, P) \mid \forall R g_1, g_2.P \in \Psi_i, \mathsf{Inv}(R) \in \omega_i, g_1 \in V_0, \text{ and } g_2 \in V_i\}
\end{aligned}
$$

It is not hard to verify that G is well-defined, i.e., that (i) if $\exists R g_1, g_2.P$ appears unmarked in Ψ_0 and $\lambda(\exists R g_1, g_2.P) = i$, then $ig_1, g_2 \in V$, and (ii) if $\exists R g_1, g_2.P$ appears marked in Ψ_i, then $g_1, ig_2 \in V$: for (i), we just need to use the above claim, and for (ii), it suffices to refer to the way in which concepts inside node labels are marked during the construction of T. Moreover, it is readily checked that G is a tuple-graph for $(T(\alpha), T(\alpha 1), \dots, T(\alpha \mathsf{b}_{C,\mathcal{R}}))$: V, E, and τ are clearly of the form described in Definition 4.3 and it is not hard to verify that **(G1)** to **(G4)** are satisfied (use the above Claim for **(G1)** and the way in which concepts inside node labels are marked for **(G2)**). To show that **(M10)** is satisfied, it thus

remains to show that G has a consistent normalization. However, since it is easily seen that every solution for a constraint graph induces a consistent normalization of this constraint graph, it suffices to show that G itself is consistent.

Define a mapping σ from V to $\mathbb{Q}$ as follows:

$$\sigma(g) := g^{\mathcal{I}}(\pi(\alpha)) \text{ and } \sigma(ig) := g^{\mathcal{I}}(\pi(\alpha i)).$$

We show that σ is a solution for G:

- Let $(v, P_q) \in \tau$. Then either (i) $v = g$ for some $g \in \mathsf{N}_{\mathsf{cF}}$ or (ii) $v = ig$ for some $i \in \{1, \dots, \mathsf{b}_{C,\mathcal{R}}\}$ and $g \in \mathsf{N}_{\mathsf{cF}}$. In Case (i), we have $(g, P_q) \in \tau_0$ by definition of G. By construction of T and definition of $\mathcal{I}(\cdot, \cdot)$, this implies $g^{\mathcal{I}}(\pi(\alpha)) P q$. By definition of σ, we thus have $\sigma(g) P q$. Similarly, in Case (ii) we have $(g, P_q) \in \tau_i$ which implies $g^{\mathcal{I}}(\pi(\alpha i)) P q$ and thus $\sigma(ig) P q$.

- Let $(v_1, v_2, P) \in E$. By definition of G, we can distinguish the following cases:

  1. $v_1 = g_1$ and $v_2 = g_2$ for some $g_1, g_2 \in \mathsf{N}_{\mathsf{cF}}$. By definition of G, we have $(g_1, g_2, P) \in V_0$. By construction of T and definition of $\mathcal{I}(\cdot, \cdot)$, this implies $g_1^{\mathcal{I}}(\pi(\alpha)) P g_2^{\mathcal{I}}(\pi(\alpha))$. Hence, $\sigma(g_1) P \sigma(g_2)$.

  2. $v_1 = ig_1$ and $v_2 = ig_2$ for some $i \in \{1, \dots, \mathsf{b}_{C,\mathcal{R}}\}$ and $g_1, g_2 \in \mathsf{N}_{\mathsf{cF}}$. Then we have $(g_1, g_2, P) \in V_i$ and can continue as in the previous case.

  3. $v_1 = ig_1$ and $v_2 = g_2$ for some $i \in \{1, \dots, \mathsf{b}_{C,\mathcal{R}}\}$ and $g_1, g_2 \in \mathsf{N}_{\mathsf{cF}}$. By construction of F, we can distinguish two subcases:
     - there exists some concept $\exists R g_1, g_2.P$ such that $\exists R g_1, g_2.P$ appears unmarked in Ψ_0 and $\lambda(\exists R g_1, g_2.P) = i$. By the above Claim, we have $g_1^{\mathcal{I}}(\pi(\alpha i)) P g_2^{\mathcal{I}}(\pi(\alpha))$ which clearly implies $\sigma(ig_1) P \sigma(g_2)$.
     - $\forall R g_1, g_2.P \in \Psi_0$, $R \in \omega_i$, $g_1 \in V_i$, and $g_2 \in V_0$. By (a), $\forall R g_1, g_2.P \in \Psi_0$ implies $\pi(\alpha) \in (\forall R g_1, g_2.P)^{\mathcal{I}}$. By construction of T and definition of $\mathcal{I}(\cdot, \cdot)$, $R \in \omega_i$ implies $(\pi(\alpha), \pi(\alpha i)) \in R^{\mathcal{I}}$, $g_1 \in V_i$ implies that $g_1^{\mathcal{I}}(\pi(\alpha i))$ is defined, and $g_2 \in V_0$ implies that $g_2^{\mathcal{I}}(\pi(\alpha))$ is defined. Thus $g_1^{\mathcal{I}}(\pi(\alpha i)) P g_2^{\mathcal{I}}(\pi(\alpha))$ by $\mathbb{Q}\text{-}\mathcal{SHIQ}$ semantics and $\sigma(ig_1) P \sigma(g_2)$ by definition of σ.

  4. $v_1 = g_1$ and $v_2 = ig_2$ for some $i \in \{1, \dots, \mathsf{b}_{C,\mathcal{R}}\}$ and $g_1, g_2 \in \mathsf{N}_{\mathsf{cF}}$. By construction of F, we can distinguish two subcases:
     - there exists some concept $\exists R g_1, g_2.P$ that appears marked in Ψ_i. By construction of T (in particular the marking of concepts in node labels), we have $\mathsf{Inv}(R) \in \omega_i$ and $g_1^{\mathcal{I}}(\pi(\alpha)) P g_2^{\mathcal{I}}(\pi(\alpha i))$. It remains to refer to the definition of σ.
     - $\forall R g_1, g_2.P \in \Psi_i$, $\mathsf{Inv}(R) \in \omega_i$, $g_1 \in V_0$, and $g_2 \in V_i$. By (a), $\forall R g_1, g_2.P \in \Psi_i$ implies $\pi(\alpha i) \in (\forall R g_1, g_2.P)^{\mathcal{I}}$. By construction of T and definition of $\mathcal{I}(\cdot, \cdot)$, $\mathsf{Inv}(R) \in \omega_i$ implies $(\pi(\alpha i), \pi(\alpha)) \in R^{\mathcal{I}}$, $g_1 \in V_0$ implies that $g_1^{\mathcal{I}}(\pi(\alpha))$ is defined, and $g_2 \in V_i$ implies that $g_2^{\mathcal{I}}(\pi(\alpha i))$ is defined. Thus $g_1^{\mathcal{I}}(\pi(\alpha)) P g_2^{\mathcal{I}}(\pi(\alpha i))$ by $\mathbb{Q}\text{-}\mathcal{SHIQ}$ semantics and $\sigma(g_1) P \sigma(ig_2)$ by definition of σ.

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\square$

6 Defining Looping Automata

To prove decidability of $\mathbb{Q}$-$\mathcal{SHIQ}$-concept satisfiability, it remains to define a looping automaton $\mathcal{A}_{C,\mathcal{R}}$ for each concept C and role hierarchy $\mathcal{R}$ such that $\mathcal{A}_{C,\mathcal{R}}$ accepts exactly the Hintikka-trees for C and $\mathcal{R}$. Using the notion of Hintikka-tuples from Definition 4.3, this is rather straightforward.

Definition 6.1. Let C be a concept in PNF, $\mathcal{R}$ be a role hierarchy, and $\mathsf{b}_{C,\mathcal{R}}$ be as in Definition 4.3. The looping automaton $\mathcal{A}_{C,\mathcal{R}} = (Q, M, \Delta, I)$ is defined as follows:

- $Q := M := \Gamma_{C,\mathcal{R}}$

- $I := \{(\Psi, \omega, V, E, \tau) \in Q \mid C \in \Psi \text{ and all concepts in } \Psi \text{ are unmarked}\}$.

- $\Delta \subseteq Q^{\mathsf{b}_{C,\mathcal{R}}+2}$ such that $(L, L', L_1, \ldots, L_{\mathsf{b}_{C,\mathcal{R}}}) \in \Delta$ iff

  $L = L'$ and

  $(L, L_1, \ldots, L_{\mathsf{b}_{C,\mathcal{R}}})$ is a Hintikka-tuple.

  $\diamond$

As a consequence of the following lemma and Lemmas 5.3 and 5.4, we can reduce satisfiability of concepts (in PNF) w.r.t. role hierarchies to the emptiness of the language accepted by looping automata.

Lemma 6.2. *T is a Hintikka-tree for C and $\mathcal{R}$ iff $T \in L(\mathcal{A}_{C,\mathcal{R}})$.*

Proof Let C be a concept, $\mathcal{R}$ a role hierarchy, and $\mathsf{b}_{C,\mathcal{R}}$ as in Definition 4.3.

For the "if" direction, let r be a run of $\mathcal{A}_{C,\mathcal{R}}$ on T. By definition of runs and of Δ, we have

$$r(\alpha) = T(\alpha) \text{ for all } \alpha \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*.$$

Hence, it remains to be shown that r is a Hintikka-tree for C and $\mathcal{R}$, which is straightforward: (i) by definition of Q, r is a $\Gamma_{C,\mathcal{R}}$-tree; (ii) since, by definition of runs, $r(\epsilon) \in I$, **(T1)** and **(T2)** are satisfied; and (iii) by definition of runs and of Δ, **(T3)** is satisfied.

Now for the "only if" direction. It is straightforward to check that T itself is a run of $\mathcal{A}_{C,\mathcal{R}}$ on T: (i) by definition of Hintikka-trees and $\mathcal{A}_{C,\mathcal{R}}$, $T(\alpha) \in Q$ for all $\alpha \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$; (ii) by **(T1)**, **(T2)**, and definition of I, we have $T(\epsilon) \in I$; (iii) by **(T3)** and by definition of Δ, we have $(T(\alpha), T(\alpha), T(\alpha 1), \ldots, T(\alpha k)) \in \Delta$ for all $\alpha \in \{1, \ldots, \mathsf{b}_{C,\mathcal{R}}\}^*$. ❑

It is an immediate consequence of Lemmas 3.3, 5.3, 5.4, and 6.2 and the decidability of the emptiness problem of looping automata [21] that satisfiability of $\mathbb{Q}$-$\mathcal{SHIQ}$-concepts w.r.t. role hierarchies is decidable. However, the presented automata-based algorithm additionally provides us with a tight complexity bound if the numbers inside number restrictions are assumed to be encoded unarily. We use $|C|$ to denote the length of the concept C and $|\mathcal{R}|$ to denote the number of role inclusions in the role hierarchy $\mathcal{R}$.

Theorem 6.3. *If numbers inside number restrictions are encoded unarily, then satisfiability of $\mathbb{Q}$-$\mathcal{SHIQ}$-concepts w.r.t. role hierarchies is* EXPTIME-*complete.*

Proof   The lower bound is an immediate consequence of the fact that $\mathcal{SHIQ}$-concept satisfiability is EXPTIME-hard [20]. For the upper bound, we need to show that the size of $\mathcal{A}_{C,\mathcal{R}}$ is exponential in $|C| + |\mathcal{R}|$, which clearly implies that $\mathcal{A}_{C,\mathcal{R}}$ can be computed in exponential time. Indeed, if this is established, we can use Lemmas 3.3, 5.3, 5.4, and 6.2 together with the fact that the emptiness problem for looping automata $\mathcal{A}_{C,\mathcal{R}}$ is in PTIME [21] to conclude that satisfiability of $\mathbb{Q}$-$\mathcal{SHIQ}$-concepts w.r.t. role hierarchies can be decided in deterministic exponential time. Hence, let us investigate the size of $\mathcal{A}_{C,\mathcal{R}} = (Q, M, \Delta, I)$. It is straightfoward to verify that the cardinality of $\mathsf{cl}(C, \mathcal{R})$ is quadratic in $|C| + |\mathcal{R}|$. Hence, using the definition of $\mathcal{A}_{C,\mathcal{R}}$, Hintikka-labels, and Hintikka-sets, it is not hard to check that the cardinality of Q and M are exponential in $|C| + |\mathcal{R}|$. Again by definition of $\mathcal{A}_{C,\mathcal{R}}$, this implies that the cardinalities of I and Δ are also exponential in $|C| + |\mathcal{R}|$. Hence, the size of $\mathcal{A}_{C,\mathcal{R}}$ is exponential in $|C| + |\mathcal{R}|$.   ❑

Since subsumption can be reduced to (un)satisfiability, $\mathbb{Q}$-$\mathcal{SHIQ}$-concept subsumption w.r.t. role hierarchies is also EXPTIME-complete.

7   Future Work

In this paper, we have presented the Description Logic $\mathbb{Q}$-$\mathcal{SHIQ}$, which extends the well-known DL $\mathcal{SHIQ}$ with several concrete domain concept constructors that allow to represent numerical knowledge, such as knowledge about the age, weight, or height of real-world entities. As argued in the introduction, $\mathbb{Q}$-$\mathcal{SHIQ}$ is a contribution to several interesting application areas. However, we regard the results presented in this paper only as a first step: first, it is currently unknown whether some of the restricting assumptions made in the definition of $\mathbb{Q}$-$\mathcal{SHIQ}$ are really necessary to ensure decidability of reasoning; second, to make $\mathbb{Q}$-$\mathcal{SHIQ}$ usable for applications, there still remains work to be done such as finding a tableau-style algorithm or devising an algorithm for finite model reasoning. More precisely, we would like to complete the research on $\mathbb{Q}$-$\mathcal{SHIQ}$ by the following investigations:

(1) To make $\mathbb{Q}$-$\mathcal{SHIQ}$ available for use in applications, modern DL systems like FaCT and RACER, which are implementations of the $\mathcal{SHIQ}$ Description Logic, need to be extended to $\mathbb{Q}$-$\mathcal{SHIQ}$. Unfortunately, the results presented in this paper cannot immediately be used to do this: the afore mentioned DL systems are based on tableau-style algorithms while the decision procedure described in this paper is automata-based. Hence, it would be interesting to devise a tableau-based algorithm for $\mathbb{Q}$-$\mathcal{SHIQ}$-concept satisfiability. As discussed in [13] in the context of $\mathcal{TDL}$, the automata-based algorithm presented in this paper can provide important information (i.e., a "regular model property") for this task.

(2) If $\mathbb{Q}$-$\mathcal{SHIQ}$ is to be used for reasoning about ER diagrams as sketched in the introduction, one is usually not interested in the satisfiability of concepts in arbitrary models, but rather in the satisfiability in finite models [3]. These two problems do not

coincide since $\mathcal{SHIQ}$, and hence also $\mathbb{Q}$-$\mathcal{SHIQ}$, lacks the finite model property [10]. Thus, it is worthwile to investigate the decidability and complexity of finite model reasoning with $\mathbb{Q}$-$\mathcal{SHIQ}$.

(3) For some applications, it is desirable to refer to natural numbers instead of rational numbers. As a simple example, consider the concept

$$\exists(\text{left-neighbor numchild}).=_2 \sqcap \exists(\text{left-neighbor numchild}), (\text{numchild}).<$$
$$\sqcap \exists(\text{right-neighbor numchild}).=_3 \sqcap \exists(\text{numchild}), (\text{right-neighbor numchild}).<,$$

where numchild is a concrete feature representing the number of children. Clearly, the above concept should be unsatisfiable. In $\mathbb{Q}$-$\mathcal{SHIQ}$, however, this concept is satisfiable since, in a model, the number of children of the described person may e.g. be 2.5. It would thus be interesting to add a concept constuctor $\exists g.\text{nat}$ to $\mathbb{Q}$-$\mathcal{SHIQ}$ expressing that the filler of the concrete feature g is a natural number. If the extended logic is decidable, then at least it seems to require some serious modifications of the presented decision procedure: as noted in [15] in the context of $\mathcal{TDL}$, Theorem 3.7 does *not* hold if the satisfiability of constraint graphs over non-dense structures such as $\mathbb{N}$ is considered. However, if $\mathbb{Q}$-$\mathcal{SHIQ}$ is extended with an $\exists g.\text{nat}$ constructor, then concepts of the resulting logic can clearly be used to describe constraint graphs all of whose nodes are labeled with the nat predicates. This means that, effectively, we have to decide satisfiability of these constraint graphs over $\mathbb{N}$.

(2) In the current version of $\mathbb{Q}$-$\mathcal{SHIQ}$, the syntactic form of paths U_1 and U_2 inside the constructors $\exists U_1, U_2.P$ and $\forall U_1, U_2.P$ is restricted. Can we allow arbitrary paths here without losing decidability? Our feeling is that we can, and that the resulting logic is still in EXPTIME, but a proof is yet to be found.

(3) In this paper, we have disallowed the use of transitive roles inside the constructors $\exists Rg_1, g_2.P$ and $\forall R_1 g_1, g_2.P$. What happens w.r.t. decidability and complexity if transitive roles are admitted?

(4) The EXPTIME upper bound established in this paper requires numbers inside number restrictions to be coded unarily. Does the upper bound still apply if binary coding of numbers is assumed? Note that, in [20], Tobies describes an automata-based decision procedure to show EXPTIME-completeness of reasoning with the Description Logic $\mathcal{ALCQIb}$ with binary coding of numbers. Since $\mathcal{ALCQIb}$ is rather closely related to $\mathcal{SHIQ}$, one could think that a straightforward combination of Tobies' techniques with the ones presented in the current paper yields an EXPTIME upper bound for $\mathbb{Q}$-$\mathcal{SHIQ}$ with binary coding of numbers. However, it seems that there are some intricate interactions between Tobies' approach to handle number restricions and our way to deal with the "numerical part" of $\mathbb{Q}$-$\mathcal{SHIQ}$.

It is interesting to note that the expressive power provided by $\mathbb{Q}$-$\mathcal{SHIQ}$ is in many aspects already on the "border" to undecidability. This concerns the $\mathcal{SHIQ}$-part of the logic (recall that admitting non-simple roles inside number restrictions destroys decidability) as well as the numerical part. For example, it seems rather unlikely that any kind of arithmetics can be added to $\mathbb{Q}$-$\mathcal{SHIQ}$ without losing decidability. More precisely, it follows from results in [16; 14] that the addition of a concept constructor expressing the addition of two numbers already yields undecidability of reasoning.

References

[1] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 452–457, Sydney, Australia, 1991.

[2] F. Baader and P. Hanschke. Extensions of concept languages for a mechanical engineering application. In *Proceedings of the 16th German AI-Conference (GWAI-92)*, volume 671 of *Lecture Notes in Computer Science*, pages 132–143. Springer-Verlag, 1992.

[3] D. Calvanese. *Unrestricted and Finite Model Reasoning in Class-Based Representation Formalisms*. Dottorato di ricerca in informatica, Università degli Studi di Roma "La Sapienza", Italia, 1996.

[4] D. Calvanese, M. Lenzerini, and D. Nardi. Description logics for conceptual data modeling. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, pages 229–263. Kluwer Academic Publisher, 1998.

[5] D. Fensel, I. Horrocks, F. van Harmelen, S. Decker, M. Erdmann, and M. Klein. OIL in a nutshell. In *Proceedings of the European Knowledge Acquisition Conference (EKAW 2000)*, volume 1937 of *Lecture Notes In Artificial Intelligence*. Springer-Verlag, 2000.

[6] V. Haarslev and R. Möller. High performance reasoning with very large knowledge bases: A practical case study. In B. Nebel, editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 161–166. Morgan-Kaufmann, 2001.

[7] P. Hanschke. Specifying role interaction in concept languages. In W. Nebel, Bernhard; Rich, Charles; Swartout, editor, *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, pages 318–329. Morgan Kaufmann, 1992.

[8] I. Horrocks and P. Patel-Schneider. The generation of DAML+OIL. In C. Goble, D. L. McGuinness, R. Möller, and P. F. Patel-Schneider, editors, *Proceedings of the International Workshop in Description Logics 2001 (DL2001)*, number 49 in CEUR-WS (http://ceur-ws.org/), pages 30–35, 2001.

[9] I. Horrocks and P. F. Patel-Schneider. Optimising description logic subsumption. *Journal of Logic and Computation*, 9(3):267–293, 1999.

[10] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *Journal of Logic and Computation*, 9(3), 1999.

[11] I. Horrocks and U. Sattler. Ontology reasoning in the $\mathcal{SHOQ}(D)$ description logic. In B. Nebel, editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 199–204. Morgan-Kaufmann, 2001.

[12] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–264, 2000.

[13] C. Lutz. Interval-based temporal reasoning with general TBoxes. LTCS-Report 00-06, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 2000. See http://www-lti.informatik.rwth-aachen.de/Forschung/Reports.html.

[14] C. Lutz. *The Complexity of Reasoning with Concrete Domains*. PhD thesis, Teaching and Research Area for Theoretical Computer Science, RWTH Aachen, 2001.

[15] C. Lutz. Interval-based temporal reasoning with general TBoxes. In B. Nebel, editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 89–94. Morgan-Kaufmann, 2001.

[16] C. Lutz. NExpTime-complete description logics with concrete domains. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proceedings of the First International Joint Conference on Automated Reasoning (IJCAR'01)*, number 2083 in Lecture Notes in Artifical Intelligence, pages 45–60. Springer-Verlag, 2001.

[17] C. Lutz. Reasoning about entity relationship diagrams with constraints on attributes. Submitted to the 2002 Workshop on Description Logics, 2002.

[18] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[19] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, chapter 4, pages 133–191. Elsevier Science Publishers B. V., 1990.

[20] S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, 2001.

[21] M. Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logic of programs. *Journal of Computer and System Sciences*, 32:183–221, 1986.