The Description Logic $\mathcal{ALCNH}_{R^+}$ Extended with Concrete Domains: A Practically Motivated Approach

Volker Haarslev, Ralf Möller and Michael Wessel

University of Hamburg, Computer Science Department
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany

Abstract. In this paper the description logic $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ is introduced. Prominent language features beyond conjunction, full negation, and quantifiers are number restrictions, role hierarchies, transitively closed roles, generalized concept inclusions, and concrete domains. As in other languages based on concrete domains (e.g. $\mathcal{ALC}(\mathcal{D})$) a so-called existential predicate restriction is provided. However, compared to $\mathcal{ALC}(\mathcal{D})$ only features and no feature chains are allowed in this operator. This results in a limited expressivity w.r.t. concrete domains but is required to ensure the decidability of the language. We show that the results can be exploited for building practical description logic systems for solving e.g. configuration problems.

1 Introduction

In the field of knowledge representation, description logics (DLs) have been proven to be a sound basis for solving application problems. An application domain where DLs have been successfully applied is *configuration* (see [9] for an early publication). The main notions for domain modeling are concepts (unary predicates) and roles (binary predicates). Furthermore, a set of axioms (also called TBox) is used for modeling the terminology of an application. Knowledge about specific individuals and their interrelationships is modeled with a set of additional axioms (so-called ABox).

Experiences with description logics in applications indicate that negation, existential and universal restrictions, transitive roles, role hierarchies, and number restrictions are required to solve practical modeling problems without resorting to ad hoc extensions. A description logic which provides these language constructs is, for instance, $\mathcal{ALCNH}_{R^+}$ [5]. The optimized DL knowledge representation system RACE [4] provides an optimized implementation for ABox reasoning in $\mathcal{ALCNH}_{R^+}$. With the optimized implementation of RACE, practical systems based on description logics can be built. However, it is well-known that, in addition to the language constructs mentioned above, reasoning about objects from other domains (so-called concrete domains, e.g. for the reals) is very important for practical applications as well. In [1] the description logic $\mathcal{ALC}(\mathcal{D})$ is investigated and it is shown that, provided a decision procedure for the concrete domain $\mathcal{D}$ exists, the logic $\mathcal{ALC}(\mathcal{D})$ is decidable. In this paper, an extension of the $\mathcal{ALCNH}_{R^+}$ knowledge representation system RACE with concrete domains is investigated.

Unfortunately, adding concrete domains (as proposed in the original approach) to expressive description logics might lead to undecidable inference problems. For instance, in [2] it is proven that the logic $\mathcal{ALC}(\mathcal{D})$ plus an operator for the transitive closure of roles can be undecidable if expressive concrete domains are considered. $\mathcal{ALCNH}_{R^+}$ offers transitive roles but no operator for the transitive closure of roles. In [8] it is shown that $\mathcal{ALC}(\mathcal{D})$ with generalized inclusion axioms (GCIs) can be undecidable. Even if GCIs were not allowed in $\mathcal{ALCNH}_{R^+}$, $\mathcal{ALCNH}_{R^+}$ with concrete domains would be undecidable (in general) because $\mathcal{ALCNH}_{R^+}$ offers role hierarchies and transitive roles, which provide the same expressivity as GCIs. With role hierarchies it is possible to (implicitly) declare a universal role, which can be used in combination with a value restriction to achieve the same effect as with GCIs. Decidability results can only be obtained for "trivial" concrete domains, which are hardly useful in practical applications. Thus, if termination and soundness of, for instance, a concept consistency algorithm are to be retained, there is no way extending an $\mathcal{ALCNH}_{R^+}$ DL system such as RACE with concrete domains as in $\mathcal{ALC}(\mathcal{D})$ without losing completeness.

Thus, $\mathcal{ALCNH}_{R^+}$ can only be extended with concrete domain operators with limited expressivity. In order to support practical modeling requirements at least to some extent, we pursue a pragmatic approach by supporting only features (and no feature chains as in $\mathcal{ALC}(\mathcal{D})$, for details see [1] and below). The resulting language is called $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$. By proving soundness and completeness (and termination) of a tableaux calculus, the decidability of inference problems w.r.t. the language $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ is proved. As shown in this paper, $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ can be used, for instance, as a basis for building practical application systems for solving configuration problems.

2 The Description Logic $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$

The description logic $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ provides conjunction, full negation, quantifiers, number restrictions, role hierarchies, transitively closed roles and concrete domains. In addition to the operators known from $\mathcal{ALCNH}_{R^+}$, a restricted existential predicate restriction operator for concrete domains is supported. Furthermore, we assume that the unique name assumption holds for the individuals explicitly mentioned in an ABox.

We briefly introduce the syntax and semantics of the DL $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$. We assume five disjoint sets: a set of concept names C, a set of role names R, a set of feature names F, a set of individual names O and a set of names for (concrete) objects O_C. The mutually disjoint subsets P and T of R denote non-transitive and transitive roles, respectively ($R = P \cup T$). The language $\mathcal{ALCNH}_{R^+}$ is introduced in Figure 1 using a standard Tarski-style semantics with an interpretation $\mathcal{I}_\mathcal{D} = (\Delta_\mathcal{I}, \Delta_\mathcal{D}, \cdot^\mathcal{I})$ where $\Delta_\mathcal{I} \cap \Delta_\mathcal{D} = \emptyset$ holds. A variable assignment α maps concrete objects to values in $\Delta_\mathcal{D}$.

Syntax	Semantics
Concepts ($\mathsf{R} \in R$, $\mathsf{S} \in S$, $\mathsf{f} \in F$)	
A	$\mathsf{A}^{\mathcal{I}} \subseteq \Delta_{\mathcal{I}}$
$\neg\mathsf{C}$	$\Delta_{\mathcal{I}} \setminus \mathsf{C}^{\mathcal{I}}$
$\mathsf{C} \sqcap \mathsf{D}$	$\mathsf{C}^{\mathcal{I}} \cap \mathsf{D}^{\mathcal{I}}$
$\mathsf{C} \sqcup \mathsf{D}$	$\mathsf{C}^{\mathcal{I}} \cup \mathsf{D}^{\mathcal{I}}$
$\exists\,\mathsf{R}\,.\,\mathsf{C}$	$\{\mathsf{a} \in \Delta_{\mathcal{I}} \mid \exists\,\mathsf{b} \in \Delta_{\mathcal{I}} : (\mathsf{a},\mathsf{b}) \in \mathsf{R}^{\mathcal{I}}, \mathsf{b} \in \mathsf{C}^{\mathcal{I}}\}$
$\forall\,\mathsf{R}\,.\,\mathsf{C}$	$\{\mathsf{a} \in \Delta_{\mathcal{I}} \mid \forall\,\mathsf{b} \in \Delta_{\mathcal{I}} : (\mathsf{a},\mathsf{b}) \in \mathsf{R}^{\mathcal{I}} \Rightarrow \mathsf{b} \in \mathsf{C}^{\mathcal{I}}\}$
$\exists_{\geq n}\,\mathsf{S}$	$\{\mathsf{a} \in \Delta_{\mathcal{I}} \mid \Vert\{\mathsf{b} \in \Delta_{\mathcal{I}} \mid (\mathsf{a},\mathsf{b}) \in \mathsf{S}^{\mathcal{I}}\}\Vert \geq \mathsf{n}\}$
$\exists_{\leq m}\,\mathsf{S}$	$\{\mathsf{a} \in \Delta_{\mathcal{I}} \mid \Vert\{\mathsf{b} \in \Delta_{\mathcal{I}} \mid (\mathsf{a},\mathsf{b}) \in \mathsf{S}^{\mathcal{I}}\}\Vert \leq \mathsf{m}\}$
$\exists\,\mathsf{f}_1,\ldots,\mathsf{f}_\mathsf{n}\,.\,\mathsf{P}$	$\{\mathsf{a} \in \Delta_{\mathcal{I}} \mid \exists \mathsf{x}_1,\ldots,\mathsf{x}_\mathsf{n} \in \Delta_{\mathcal{D}} : (\mathsf{a},\mathsf{x}_1) \in {\mathsf{f}_1}^{\mathcal{I}},\ldots,(\mathsf{a},\mathsf{x}_\mathsf{n}) \in {\mathsf{f}_\mathsf{n}}^{\mathcal{I}},$ $(\mathsf{x}_1,\ldots,\mathsf{x}_\mathsf{n}) \in \mathsf{P}^{\mathcal{I}}\}$
$\forall\,\mathsf{f}\,.\,\bot_{\mathcal{D}}$	$\{\mathsf{a} \in \Delta_{\mathcal{I}} \mid \neg\exists \mathsf{x}_1 \in \Delta_{\mathcal{D}} : (\mathsf{a},\mathsf{x}_1) \in \mathsf{f}^{\mathcal{I}}\}$
Roles and Features	
R	$\mathsf{R}^{\mathcal{I}} \subseteq \Delta_{\mathcal{I}} \times \Delta_{\mathcal{I}}$
f	$\mathsf{f}^{\mathcal{I}} : \Delta_{\mathcal{I}} \to \Delta_{\mathcal{D}}$ (features are partial functions)

A is a concept name and $\Vert\cdot\Vert$ denotes the cardinality of a set ($n, m \in \mathbb{N}, n > 0$).

Axioms	
Syntax	Satisfied if
$\mathsf{R} \in T$	$\mathsf{R}^{\mathcal{I}} = (\mathsf{R}^{\mathcal{I}})^+$
$\mathsf{R} \sqsubseteq \mathsf{S}$	$\mathsf{R}^{\mathcal{I}} \subseteq \mathsf{S}^{\mathcal{I}}$
$\mathsf{C} \sqsubseteq \mathsf{D}$	$\mathsf{C}^{\mathcal{I}} \subseteq \mathsf{D}^{\mathcal{I}}$

Assertions ($\mathsf{a},\mathsf{b} \in O_O, \mathsf{x}, \mathsf{x}_\mathsf{i} \in O_C$)	
Syntax	Satisfied if
$\mathsf{a} : \mathsf{C}$	$\mathsf{a}^{\mathcal{I}} \in \mathsf{C}^{\mathcal{I}}$
$(\mathsf{a},\mathsf{b}) : \mathsf{R}$	$(\mathsf{a}^{\mathcal{I}},\mathsf{b}^{\mathcal{I}}) \in \mathsf{R}^{\mathcal{I}}$
$(\mathsf{a},\mathsf{x}) : \mathsf{f}$	$(\mathsf{a}^{\mathcal{I}},\alpha(\mathsf{x})) \in \mathsf{f}^{\mathcal{I}}$
$(\mathsf{x}_1,\ldots,\mathsf{x}_\mathsf{n}) : \mathsf{P}$	$(\alpha(\mathsf{x}_1),\ldots,\alpha(\mathsf{x}_\mathsf{n})) \in \mathsf{P}^{\mathcal{I}}$

Fig. 1. Syntax and Semantics of $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$.

If $\mathsf{R}, \mathsf{S} \in R$ are role names, then $\mathsf{R} \sqsubseteq \mathsf{S}$ is called a *role inclusion* axiom. A *role hierarchy* $\mathcal{R}$ is a finite set of role inclusion axioms. Then, we define $\sqsubseteq^*$ as the reflexive transitive closure of $\sqsubseteq$ over such a role hierarchy $\mathcal{R}$. Given $\sqsubseteq^*$, the set of roles $\mathsf{R}^{\downarrow} = \{\mathsf{S} \in R \mid \mathsf{S} \sqsubseteq^* \mathsf{R}\}$ defines the *sub-roles* of a role R and $\mathsf{R}^{\uparrow} = \{\mathsf{S} \in R \mid \mathsf{R} \sqsubseteq^* \mathsf{S}\}$ defines the *super-roles* of a role. We also define the set $S := \{\mathsf{R} \in P \mid \mathsf{R}^{\downarrow} \cap T = \emptyset\}$ of *simple* roles that are neither transitive nor have a transitive role as a sub-role.

The concept language of $\mathcal{ALCNH}_{R^+}$ is syntactically restricted with respect to the combination of number restrictions and transitive roles. Number restrictions are only allowed for simple roles. This restriction is motivated by a known undecidability result in case of an unrestricted syntax [7]. The set of individuals is divided into two subsets, the set of so-called "old" individuals O_O and set the of "new" individuals O_N. Every individual name from O is mapped to a single element of $\Delta_{\mathcal{I}}$ in a way such that for $\mathsf{a}, \mathsf{b} \in O_O$, $\mathsf{a}^{\mathcal{I}} \neq \mathsf{b}^{\mathcal{I}}$ if $\mathsf{a} \neq \mathsf{b}$ *(unique name assumption)*. Only old individuals may be mentioned in an ABox (new individual are generated by the completion rules introduced below).

In accordance with [1] we also define the notion of a concrete domain. A *concrete domain* $\mathcal{D}$ is a pair $(\Delta_{\mathcal{D}}, \Phi_{\mathcal{D}})$, where $\Delta_{\mathcal{D}}$ is a set called the domain, and $\Phi_{\mathcal{D}}$ is a set of predicate names. The interpretation function maps each predicate name P from $\Phi_{\mathcal{D}}$ with arity n to a subset $\mathsf{P}^{\mathcal{I}}$ of $\Delta_{\mathcal{D}}^n$. Concrete objects from O_C

are mapped to an element of $\Delta_{\mathcal{D}}$. We assume that $\bot_{\mathcal{D}}$ is the negation of the predicate $\top_{\mathcal{D}}$.

A concrete domain $\mathcal{D}$ is called *admissible* iff the set of predicate names $\Phi_{\mathcal{D}}$ is closed under negation and $\Phi_{\mathcal{D}}$ contains a name $\top_{\mathcal{D}}$ for $\Delta_{\mathcal{D}}$, and the satisfiability problem $\mathsf{P}_1^{\mathsf{n}_1}(\mathsf{x}_{11}, \ldots, \mathsf{x}_{1\mathsf{n}_1}) \wedge \ldots \wedge \mathsf{P}_{\mathsf{m}}^{\mathsf{n}_{\mathsf{m}}}(\mathsf{x}_{\mathsf{m}1}, \ldots, \mathsf{x}_{\mathsf{m}\mathsf{n}_{\mathsf{m}}})$ is decidable (m is finite, $\mathsf{P}_{\mathsf{i}}^{\mathsf{n}_{\mathsf{i}}} \in \Phi_{\mathcal{D}}$, n_{i} is the arity of P, and x_{jk} is a concrete object).

If C and D are concept terms, then $\mathsf{C} \sqsubseteq \mathsf{D}$ (*generalized concept inclusion* or *GCI*) is a terminological axiom. A finite set of terminological axioms $\mathcal{T}_{\mathcal{R}}$ is called a *terminology* or *TBox* w.r.t. a given role hierarchy $\mathcal{R}$. For brevity, the reference to $\mathcal{R}$ is omitted in the following. An *ABox* $\mathcal{A}$ is a finite set of assertional axioms as defined in Figure 1.

An interpretation $\mathcal{I}$ is a *model* of a concept C (or *satisfies* a concept C) iff $\mathsf{C}^{\mathcal{I}} \neq \emptyset$. An interpretation is a model of a TBox $\mathcal{T}$ iff it satisfies all axioms in $\mathcal{T}$. See Figure 1 for the satisfiability conditions. An interpretation is a model of an ABox $\mathcal{A}$ w.r.t. a TBox iff it is a model of $\mathcal{T}$ and satisfies all assertions in $\mathcal{A}$. Different individuals are mapped to different domain objects (unique name assumption). Note that features are interpreted differently from features in [1].

A concept C is called *consistent* (w.r.t. a TBox $\mathcal{T}$) iff there exists a model of C (that is also a model of $\mathcal{T}$). An ABox $\mathcal{A}$ is consistent (w.r.t. a TBox $\mathcal{T}$) iff $\mathcal{A}$ has model $\mathcal{I}$ (which is also a model of $\mathcal{T}$). A *knowledge base* $(\mathcal{T}, \mathcal{A})$ is called consistent iff there exists a model.

3 Solving an Application Problem with $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$

According to [3] configuration problem solving processes can be formalized as synthesis inference tasks. Following this approach, a solution of a configuration task is defined to be a (logical) model of the given knowledge base consisting of both the conceptual domain model (TBox) as well as the task specification (ABox). The TBox and the role hierarchy describe the configuration space.

For instance, in a technical domain, the concept of a cylinder might be defined as follows. A Cylinder is required to be a Motorpart, to be part_of a Motor, to have a displacement of 1 to 1000ccm, and to have a set of 4 to 6 parts (role has_part) which are all instances of Cylinderpart and it consists of exactly 1 Piston, exactly 1 Piston_Rod, and 2 to 4 Valves. This expression can be transformed to a terminological inclusion axiom of a description logic providing concrete domains. Let the concrete domain $\Re$ be defined as in [1]: $\Re = (\mathbb{R}, \Phi_{\Re})$ where $\Phi_{\Re}$ is a set of predicates which are based on polynomial equations or inequations. The concrete domain $\Re$ is admissible (see also [1]). A TBox $\mathcal{T}$ is defined as follows:

$$\textbf{has\_cylinder\_part} \sqsubseteq \mathsf{has\_part}, \qquad \textbf{has\_piston\_part} \sqsubseteq \mathsf{has\_part}$$
$$\textbf{has\_piston\_rod\_part} \sqsubseteq \mathsf{has\_part}, \qquad \textbf{has\_valve\_part} \sqsubseteq \mathsf{has\_part}$$

$$\top \sqsubseteq \forall\, \mathsf{has\_cylinder\_part} \,.\, \mathsf{Cylinder}, \qquad \top \sqsubseteq \forall\, \mathsf{has\_piston\_part} \,.\, \mathsf{Piston}$$
$$\top \sqsubseteq \forall\, \mathsf{has\_piston\_rod\_part} \,.\, \mathsf{Piston\_Rod}, \qquad \top \sqsubseteq \forall\, \mathsf{has\_valve\_part} \,.\, \mathsf{Valve}$$

In the first block, relationships between roles are declared. Then, in the second block, range restrictions for certain roles are imposed. Below, in the third block for Cylinderpart a so-called cover axiom is given. Moreover, additional axioms ensure the disjointness of more specific subconcepts of Cylinderpart (D is a subconcept of C iff C subsumes D).

$$\textbf{Cylinderpart} \sqsubseteq \mathsf{Piston} \sqcup \mathsf{Piston\_Rod} \sqcup \mathsf{Valve}, \qquad \textbf{Piston} \sqsubseteq \neg\mathsf{Piston\_Rod} \sqcap \neg\mathsf{Valve}$$
$$\textbf{Piston\_Rod} \sqsubseteq \neg\mathsf{Piston} \sqcap \neg\mathsf{Valve}, \qquad \textbf{Valve} \sqsubseteq \neg\mathsf{Piston} \sqcap \neg\mathsf{Piston\_Rod}$$

The cylinder example is translated as follows (the term $\lambda_{\mathrm{Vol}}\, c.\,(\dots)$ is a unary predicate of a numeric concrete domain for the dimension *Volume* with unit m^3).

$$\begin{aligned}\textbf{Cylinder} \sqsubseteq\ & \mathsf{Motorpart} \sqcap \exists_{=1}\ \mathsf{part\_of}\ \sqcap \\ & \exists\,\mathsf{displacement}\,.\,\lambda_{\mathrm{Vol}}\, c\,.\,(0.001 \leq c \leq 1) \sqcap \\ & \forall\,\mathsf{has\_part}\,.\,\mathsf{Cylinderpart} \sqcap \\ & \exists_{\geq 4}\ \mathsf{has\_cylinder\_part} \sqcap \exists_{\leq 6}\ \mathsf{has\_cylinder\_part} \sqcap \\ & \exists_{=1}\ \mathsf{has\_piston\_part} \sqcap \exists_{=1}\ \mathsf{has\_piston\_rod\_part} \sqcap \\ & \exists_{\geq 2}\ \mathsf{has\_valve\_part} \sqcap \exists_{\leq 4}\ \mathsf{has\_valve\_part}\end{aligned}$$

We assume that displacement is declared as a feature. Furthermore, let $\exists_{=1}\ \mathsf{R}$ be an abbreviation for $\exists_{\geq 1}\ \mathsf{R} \sqcap \exists_{\leq 1}\ \mathsf{R}$. In our example, the ABox being used is very simple: $\mathcal{A} = \{\mathsf{a} : \mathsf{Cylinder} \sqcap \exists\,\mathsf{displacement}\,.\,\lambda_{\mathrm{Vol}}\, c\,.\,(c \geq 0.5)\}$.
In order to solve the problem to construct a Cylinder, the knowledge base $(\mathcal{T}, \mathcal{A})$ is tested for consistency. If the knowledge base is consistent, there exists a model which can be considered as a solution (see [3]). Note that $(\mathcal{T}, \mathcal{A})$ is only a very simplified example for a representation of a configuration problem. For instance, using an ABox with additional assertions it is possible to explicitly specify some required cylinder parts etc. In order to actually compute a solution to a configuration problem, a sound and complete calculus for the $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ knowledge base consistency problem is required that terminates on any input.

4 A Tableaux Calculus for $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$

In the following a calculus to decide the consistency of an $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ knowledge base $(\mathcal{T}, \mathcal{A})$ is devised. As a first step, the original ABox $\mathcal{A}$ of the knowledge base is transformed w.r.t. the TBox $\mathcal{T}$. The idea is to derive an ABox $\mathcal{A}_{\mathcal{T}}$ that is consistent (w.r.t. an empty TBox) iff $(\mathcal{T}, \mathcal{A})$ is consistent. The calculus introduced below is applied to $\mathcal{A}_{\mathcal{T}}$.

In order to define the transformation steps for deriving $\mathcal{A}_{\mathcal{T}}$, we have to introduce a few technical terms. First, for any concept term we define its negation normal form. A concept is in *negation normal form* iff negation signs may occur only in front of concept names.

Every $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ concept term C can be transformed into negation normal form $\mathit{nnf}(\mathsf{C})$ by recursively applying the following transformation rules to subconcepts from left to right:

$\neg(\mathsf{C} \sqcap \mathsf{D}) \rightarrow \neg\mathsf{C} \sqcup \neg\mathsf{D}$, $\neg(\mathsf{C} \sqcup \mathsf{D}) \rightarrow \neg\mathsf{C} \sqcap \neg\mathsf{D}$, $\neg\forall\,\mathsf{R}\,.\,\mathsf{C} \rightarrow \exists\,\mathsf{R}\,.\,\neg\mathsf{C}$,
$\neg\exists\,\mathsf{R}\,.\,\mathsf{C} \rightarrow \forall\,\mathsf{R}\,.\,\neg\mathsf{C}$, $\neg\neg\mathsf{C} \rightarrow \mathsf{C}$, $\neg\exists_{\geq n}\,\mathsf{S} \rightarrow \exists_{\leq n-1}\,\mathsf{S}$, $\neg\exists_{\leq m}\,\mathsf{S} \rightarrow \exists_{\geq m+1}\,\mathsf{S}$,
$\neg\forall\,\mathsf{f}\,.\,\bot_{\mathcal{D}} \rightarrow \exists\,\mathsf{f}\,.\,\top_{\mathcal{D}}$, $\neg\exists\,\mathsf{f}_1,\ldots,\mathsf{f}_\mathsf{n}\,.\,\mathsf{P} \rightarrow \exists\,\mathsf{f}_1,\ldots,\mathsf{f}_\mathsf{n}\,.\,\overline{\mathsf{P}} \sqcup \forall\,\mathsf{f}_1\,.\,\bot_{\mathcal{D}} \sqcup \ldots \sqcup \forall\,\mathsf{f}_\mathsf{n}\,.\,\bot_{\mathcal{D}}$
where $\overline{\mathsf{P}}$ is the negation of P.

If no rule is applicable, the resulting concept is in negation normal form and all models of C are also models of $\mathit{nnf}(\mathsf{C})$ and vice versa. The transformation is possible in linear time.

Definition 1 (Additional ABox Assertions). *Let* C *be a concept term,* $\mathsf{a},\mathsf{b} \in O$ *be individual names, and* $x \notin O \cup O_C$*, then the following expressions are also assertional axioms:* $\forall\,x\,.\,x\!:\!\mathsf{C}$ (universal concept assertion),[1] $\mathsf{a} \neq \mathsf{b}$ (inequality assertion)*.*

An interpretation $\mathcal{I}_{\mathcal{D}}$ *satisfies an assertional axiom* $\forall\,x\,.\,x\!:\!\mathsf{C}$ *iff* $\mathsf{C}^{\mathcal{I}} = \Delta_{\mathcal{I}}$ *and* $\mathsf{a} \neq \mathsf{b}$ *iff* $\mathsf{a}^{\mathcal{I}} \neq \mathsf{b}^{\mathcal{I}}$*.*

Definition 2 (Fork, Fork Elimination). *If it holds that* $\{(\mathsf{a},\mathsf{x}_1)\!:\!\mathsf{f}, (\mathsf{a},\mathsf{x}_2)\!:\!\mathsf{f}\} \subseteq \mathcal{A}$ *then there exists a* fork *in* $\mathcal{A}$*. In case of a fork w.r.t.* $\mathsf{x}_1,\mathsf{x}_2$*, the replacement of every occurrence of* x_2 *in* $\mathcal{A}$ *by* x_1 *is called* fork elimination.

Definition 3 (Augmented ABox). *For an initial ABox* $\mathcal{A}$ *we define its* augmented *ABox* $\mathcal{A}_{\mathcal{T}}$ *w.r.t a TBox* $\mathcal{T}$ *by applying the following transformation rules to* $\mathcal{A}$*. First of all, all forks in* $\mathcal{A}$ *are eliminated (note that the unique name assumption is not imposed on concrete objects). Then, for every GCI* $\mathsf{C} \sqsubseteq \mathsf{D}$ *in* $\mathcal{T}$ *the assertion* $\forall\,x\,.\,x\!:\!(\neg\mathsf{C} \sqcup \mathsf{D})$ *is added to* $\mathcal{A}$*. Every concept term occurring in* $\mathcal{A}$ *is transformed into its negation normal form. Let* $O_{\mathcal{A}} = \{\mathsf{a}_1,\ldots,\mathsf{a}_\mathsf{n}\}$ *be the set of individuals mentioned in* $\mathcal{A}$*, then the set of inequality assertions* $\{\mathsf{a}_\mathsf{i} \neq \mathsf{a}_\mathsf{j} \mid \mathsf{a}_\mathsf{i},\mathsf{a}_\mathsf{j} \in O_{\mathcal{A}},\ \mathsf{i},\mathsf{j} \in 1..\mathsf{n}, \mathsf{i} \neq \mathsf{j}\}$ *is added to* $\mathcal{A}$*.*

In order to check the consistency of an $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ knowledge base $(\mathcal{T},\mathcal{A})$, the augmented ABox $\mathcal{A}_{\mathcal{T}}$ is computed. Then, a set of so-called completion rules (see below) is applied to the augmented ABox $\mathcal{A}_{\mathcal{T}}$. The rules are applied in accordance with a completion strategy.

Lemma 1. *A knowledge base* $(\mathcal{T},\mathcal{A})$ *is consistent if and only if* $\mathcal{A}_{\mathcal{T}}$ *is consistent (w.r.t. an empty TBox).*

The proof is straightforward, for details see [6].

The tableaux rules require the notion of blocking their applicability. This is based on so-called concept sets, an ordering for new individuals and concrete objects, and the notion of a blocking individual.

Definition 4 (Ordering). *We define an* individual ordering *'$\prec$' for new individuals (elements of* O_N*) occurring in an ABox* $\mathcal{A}$*. If* $\mathsf{b} \in O_N$ *is introduced in* $\mathcal{A}$*, then* $\mathsf{a} \prec \mathsf{b}$ *for all new individuals* a *already present in* $\mathcal{A}$*. A* concrete object ordering *'$\prec_C$' for elements of* O_C *occurring in an ABox* $\mathcal{A}$ *is defined as follows. If* $\mathsf{y} \in O_C$ *is introduced in* $\mathcal{A}$*, then* $\mathsf{x} \prec_C \mathsf{y}$ *for all concrete objects* x *already present in* $\mathcal{A}$*.*

---
[1] $\forall\,x\,.\,x\!:\!\mathsf{C}$ is to be read as $\forall\,x\,.\,(x\!:\!\mathsf{C})$.

Definition 5 (Concept Set, Blocking Individual, Blocked by). *Given an ABox* $\mathcal{A}$ *and an individual* a *occurring in* $\mathcal{A}$*, we define the* concept set *of* a *as* $\sigma(\mathcal{A}, \mathsf{a}) := \{\mathsf{C} \mid \mathsf{a}:\mathsf{C} \in \mathcal{A}\}$*. Let* $\mathcal{A}$ *be an ABox and* $\mathsf{a}, \mathsf{b} \in O_N$ *be individuals in* $\mathcal{A}$*. We call* a *the* blocking individual *of* b *if the following conditions hold:* $\sigma(\mathcal{A}, \mathsf{a}) \supseteq \sigma(\mathcal{A}, \mathsf{b})$ *and* $\mathsf{a} \prec \mathsf{b}$*. If* a *is a blocking individual for* b*, then* b *is said to be* blocked by a*. An individual* b *mentioned in an ABox* $\mathcal{A}$ *is said to be blocked (in* $\mathcal{A}$*) iff there exists a blocking individual for* b *in* $\mathcal{A}$*.*

4.1 Completion Rules

We are now ready to define the *completion rules* that are intended to generate a so-called completion (see also below) of an ABox $\mathcal{A}_{\mathcal{T}}$. From this point on, if we refer to an ABox $\mathcal{A}$, we always consider ABoxes derived from $\mathcal{A}_{\mathcal{T}}$.

Definition 6 (Completion Rules).

R$\sqcap$ *The conjunction rule.*
if *1.* $\mathsf{a}:\mathsf{C} \sqcap \mathsf{D} \in \mathcal{A}$*, and*
2. $\{\mathsf{a}:\mathsf{C},\ \mathsf{a}:\mathsf{D}\} \not\subseteq \mathcal{A}$
then $\mathcal{A}' = \mathcal{A} \cup \{\mathsf{a}:\mathsf{C},\ \mathsf{a}:\mathsf{D}\}$

R$\sqcup$ *The disjunction rule (nondeterministic).*
if *1.* $\mathsf{a}:\mathsf{C} \sqcup \mathsf{D} \in \mathcal{A}$*, and*
2. $\{\mathsf{a}:\mathsf{C},\ \mathsf{a}:\mathsf{D}\} \cap \mathcal{A} = \emptyset$
then $\mathcal{A}' = \mathcal{A} \cup \{\mathsf{a}:\mathsf{C}\}$ ***or*** $\mathcal{A}' = \mathcal{A} \cup \{\mathsf{a}:\mathsf{D}\}$

R$\forall$***C*** *The role value restriction rule.*
if *1.* $\mathsf{a}:\forall\,\mathsf{R}\,.\,\mathsf{C} \in \mathcal{A}$*, and*
2. $\exists\,\mathsf{b} \in O, \mathsf{S} \in \mathsf{R}^{\downarrow} : (\mathsf{a},\mathsf{b}):\mathsf{S} \in \mathcal{A}$*, and*
3. $\mathsf{b}:\mathsf{C} \notin \mathcal{A}$
then $\mathcal{A}' = \mathcal{A} \cup \{\mathsf{b}:\mathsf{C}\}$

R$\forall_+$***C*** *The transitive role value restriction rule.*
if *1.* $\mathsf{a}:\forall\,\mathsf{R}\,.\,\mathsf{C} \in \mathcal{A}$*, and*
2. $\exists\,\mathsf{b} \in O, \mathsf{T} \in \mathsf{R}^{\downarrow}, \mathsf{T} \in T, \mathsf{S} \in \mathsf{T}^{\downarrow} : (\mathsf{a},\mathsf{b}):\mathsf{S} \in \mathcal{A}$*, and*
3. $\mathsf{b}:\forall\,\mathsf{T}\,.\,\mathsf{C} \notin \mathcal{A}$
then $\mathcal{A}' = \mathcal{A} \cup \{\mathsf{b}:\forall\,\mathsf{T}\,.\,\mathsf{C}\}$

R$\forall_x$ *The universal concept restriction rule.*
if *1.* $\forall\,x\,.\,x:\mathsf{C} \in \mathcal{A}$*, and*
2. $\exists\,\mathsf{a} \in O$*:* a *mentioned in* $\mathcal{A}$*, and*
3. $\mathsf{a}:\mathsf{C} \notin \mathcal{A}$
then $\mathcal{A}' = \mathcal{A} \cup \{\mathsf{a}:\mathsf{C}\}$

R$\exists$***C*** *The role exists restriction rule (generating).*
if *1.* $\mathsf{a}:\exists\,\mathsf{R}\,.\,\mathsf{C} \in \mathcal{A}$*, and*
2. a *is not blocked, and*
3. $\neg\exists\,\mathsf{b} \in O, \mathsf{S} \in \mathsf{R}^{\downarrow} : \{(\mathsf{a},\mathsf{b}):\mathsf{S},\ \mathsf{b}:\mathsf{C}\} \subseteq \mathcal{A}$
then $\mathcal{A}' = \mathcal{A} \cup \{(\mathsf{a},\mathsf{b}):\mathsf{R},\ \mathsf{b}:\mathsf{C}\}$ *where* $\mathsf{b} \in O_N$ *is not used in* $\mathcal{A}$

R$\exists_{\geq n}$ *The number restriction exists rule (generating).*
if *1.* $\mathsf{a} : \exists_{\geq n} \mathsf{R} \in \mathcal{A}$*, and*
2. a *is not blocked, and*
3. $\neg\exists\, \mathsf{b_1}, \ldots, \mathsf{b_n} \in O_N, \mathsf{S_1}, \ldots, \mathsf{S_n} \in \mathsf{R}^{\downarrow}$:
$\{(\mathsf{a}, \mathsf{b_k}) : \mathsf{S_k} \mid \mathsf{k} \in 1..\mathsf{n}\} \cup \{\mathsf{b_i} \neq \mathsf{b_j} \mid \mathsf{i}, \mathsf{j} \in 1..\mathsf{n}, \mathsf{i} \neq \mathsf{j}\} \subseteq \mathcal{A}$
then $\mathcal{A}' = \mathcal{A} \cup \{(\mathsf{a}, \mathsf{b_k}) : \mathsf{R} \mid \mathsf{k} \in 1..\mathsf{n}\} \cup \{\mathsf{b_i} \neq \mathsf{b_j} \mid \mathsf{i}, \mathsf{j} \in 1..\mathsf{n}, \mathsf{i} \neq \mathsf{j}\}$
where $\mathsf{b_1}, \ldots, \mathsf{b_n} \in O_N$ *are not used in* $\mathcal{A}$

R$\exists_{\leq n}$ *The number restriction merge rule (nondeterministic).*
if *1.* $\mathsf{a} : \exists_{\leq n} \mathsf{R} \in \mathcal{A}$*, and*
2. $\exists\, \mathsf{b_1}, \ldots, \mathsf{b_m} \in O, \mathsf{S_1}, \ldots, \mathsf{S_m} \in \mathsf{R}^{\downarrow}$: $\{(\mathsf{a}, \mathsf{b_1}) : \mathsf{S_1}, \ldots, (\mathsf{a}, \mathsf{b_m}) : \mathsf{S_m}\} \subseteq \mathcal{A}$
with $\mathsf{m} > \mathsf{n}$*, and*
3. $\exists\, \mathsf{b_i}, \mathsf{b_j} \in \{\mathsf{b_1}, \ldots, \mathsf{b_m}\} : \mathsf{i} \neq \mathsf{j},\ \mathsf{b_i} \neq \mathsf{b_j} \notin \mathcal{A}$
then $\mathcal{A}' = \mathcal{A}[\mathsf{b_i}/\mathsf{b_j}]$*, i.e. replace every occurrence of* $\mathsf{b_i}$ *in* $\mathcal{A}$ *by* $\mathsf{b_j}$

R$\exists$***P*** *The predicate exists rule (generating).*
if *1.* $\mathsf{a} : \exists\, \mathsf{f_1}, \ldots, \mathsf{f_n} \,.\, \mathsf{P} \in \mathcal{A}$*, and*
2. $\neg\exists \mathsf{x_1}, \ldots, \mathsf{x_n} \in \mathsf{O_C} : \{(\mathsf{a}, \mathsf{x_1}) : \mathsf{f_1}, \ldots (\mathsf{a}, \mathsf{x_n}) : \mathsf{f_n}, (\mathsf{x_1}, \ldots, \mathsf{x_n}) : \mathsf{P}\} \subseteq \mathcal{A}$
then $\mathcal{A}' = \mathcal{A} \cup \{(\mathsf{a}, \mathsf{x_1}) : \mathsf{f_1}, \ldots (\mathsf{a}, \mathsf{x_n}) : \mathsf{f_n}, (\mathsf{x_1}, \ldots, \mathsf{x_n}) : \mathsf{P}\}$
where $\mathsf{x_1}, \ldots, \mathsf{x_n} \in O_C$ *are not used in* $\mathcal{A}$*,*
eliminate all forks $\{(\mathsf{a}, \mathsf{x}) : \mathsf{f_i}, (\mathsf{a}, \mathsf{x_i}) : \mathsf{f_i}\} \subseteq \mathcal{A}$
such that $(\mathsf{a}, \mathsf{x}) : \mathsf{f_i}$ *remains in* $\mathcal{A}$ *if* $\mathsf{x} \prec_C \mathsf{x_i}, \mathsf{i} \in 1..\mathsf{n}$

We call the rules R$\sqcup$ and R$\exists_{\leq n}$ *nondeterministic* rules since they can be applied in different ways to the same ABox. The remaining rules are called *deterministic* rules. Moreover, we call the rules R$\exists$C, R$\exists_{\geq n}$ and R$\exists$P *generating* rules since they can introduce new individuals or concrete objects.

Given an ABox $\mathcal{A}$, more than one rule might be applicable to $\mathcal{A}$. This is controlled by a completion strategy in accordance to the ordering for new individuals (see Definition 4).

Definition 7 (Completion Strategy). *We define a* completion strategy *that must observe the following restrictions:*

- *Meta rules:*
  - *Apply a rule to an individual* $\mathsf{b} \in O_N$ *only if no rule is applicable to an individual* $\mathsf{a} \in O_O$*.*
  - *Apply a rule to an individual* $\mathsf{b} \in O_N$ *only if no rule is applicable to another individual* $\mathsf{a} \in O_N$ *such that* $\mathsf{a} \prec \mathsf{b}$*.*
- *The completion rules are always applied in the following order. A step is skipped in case the corresponding set of applicable rules is empty.*
  1. *Apply all nongenerating rules (R$\sqcap$, R$\sqcup$, R$\forall C$, R$\forall_+ C$, R$\forall_x$, R$\exists_{\leq n}$) as long as possible.*
  2. *Apply a generating rule (R$\exists C$, R$\exists_{\geq n}$, R$\exists P$) and restart with step 1 as long as possible.*

In the following we always assume that rules are applied in accordance to this strategy. It ensures that the rules are applied to new individuals w.r.t. the ordering '$\prec$' which guarantees a breadth-first order. No rules are applied if a so-called clash is discovered.

Definition 8 (Clash, Clash Triggers, Completion). *We assume the same naming conventions as used above. An ABox $\mathcal{A}$ contains a* clash *if one of the following* clash triggers *is applicable. If none of the clash triggers is applicable to $\mathcal{A}$, then $\mathcal{A}$ is called* clash-free.

- Primitive clash*:* $\{\mathsf{a}:\mathsf{C}, \mathsf{a}:\neg\mathsf{C}\} \subseteq \mathcal{A}$
- Number restriction merging clash*:*
  $\exists \mathsf{S}_1, \ldots, \mathsf{S}_\mathsf{m} \in \mathsf{R}^{\downarrow} : \{\mathsf{a}:\exists_{\leq n} \mathsf{R}\} \cup \{(\mathsf{a}, \mathsf{b}_\mathsf{i}):\mathsf{S}_\mathsf{i} \mid \mathsf{i} \in 1..\mathsf{m}\} \cup$
  $\{\mathsf{b}_\mathsf{i} \neq \mathsf{b}_\mathsf{j} \mid \mathsf{i}, \mathsf{j} \in 1..\mathsf{m}, \mathsf{i} \neq \mathsf{j}\} \subseteq \mathcal{A}$ *with* $\mathsf{m} > \mathsf{n}$
- No concrete domain feature clash*:* $\{(\mathsf{a}, \mathsf{x}):\mathsf{f}, \mathsf{a}:\forall \mathsf{f} . \bot_\mathcal{D}\} \subseteq \mathcal{A}$.
- Concrete domain predicate clash*:* $(\mathsf{x}_1^{(1)}, \ldots, \mathsf{x}_{\mathsf{n}_1}^{(1)}):\mathsf{P}_1 \in \mathcal{A}, \ldots,$
  $(\mathsf{x}_1^{(\mathsf{k})}, \ldots, \mathsf{x}_{\mathsf{n}_\mathsf{k}}^{(\mathsf{k})}):\mathsf{P}_\mathsf{k} \in \mathcal{A}$ *and the conjunction* $\bigwedge_{\mathsf{i}=1}^{\mathsf{k}} \mathsf{P}_\mathsf{i}(\mathsf{x}_1^{(\mathsf{i})}, \ldots, \mathsf{x}_{\mathsf{n}_\mathsf{i}}^{(\mathsf{i})})$ *is not satisfiable in $\mathcal{D}$. Note that this can be decided since $\mathcal{D}$ is required to be admissible.*

A clash-free ABox $\mathcal{A}$ is called complete *if no completion rule is applicable to $\mathcal{A}$. A complete ABox $\mathcal{A}'$ derived from an ABox $\mathcal{A}$ is also called a* completion *of $\mathcal{A}$.*

Any ABox containing a clash is obviously unsatisfiable. The purpose of the calculus is to generate a completion for an initial ABox $\mathcal{A}_\mathcal{T}$ that proves the consistency of $\mathcal{A}_\mathcal{T}$ or its inconsistency if no completion can be found.

4.2 Decidability of the $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ ABox Consistency Problem

In order to show that the calculus introduced above is correct, first the local correctness of the rules is proven.

Proposition 1 (Invariance). *Let $\mathcal{A}$ and $\mathcal{A}'$ be ABoxes. Then:*

1. *If $\mathcal{A}'$ is derived from $\mathcal{A}$ by applying a deterministic rule, then $\mathcal{A}$ is consistent iff $\mathcal{A}'$ is consistent.*
2. *If $\mathcal{A}'$ is derived from $\mathcal{A}$ by applying a nondeterministic rule, then $\mathcal{A}$ is consistent if $\mathcal{A}'$ is consistent. Conversely, if $\mathcal{A}$ is consistent and a nondeterministic rule is applicable to $\mathcal{A}$, then it can be applied in such a way that it yields an ABox $\mathcal{A}'$ which is consistent.*

Proof. **1.** "$\Leftarrow$" Due to the structure of the deterministic rules one can immediately verify that $\mathcal{A}$ is a subset of $\mathcal{A}'$. Therefore, $\mathcal{A}$ is consistent if $\mathcal{A}'$ is consistent.

"$\Rightarrow$" In order to show that $\mathcal{A}'$ is consistent after applying a deterministic rule to the consistent ABox $\mathcal{A}$, we examine each applicable rule separately. We assume that $\mathcal{I}_\mathcal{D} = (\Delta_\mathcal{I}, \Delta_\mathcal{D}, \cdot^\mathcal{I})$ satisfies $\mathcal{A}$. Then, by definition of $\sqsubseteq^*$ it holds that $\mathsf{R}^\mathcal{I} \subseteq \mathsf{S}^\mathcal{I}$ if $(\mathsf{R}, \mathsf{S}) \in \sqsubseteq^*$.

If the conjunction rule is applied to $\mathsf{a}:\mathsf{C} \sqcap \mathsf{D} \in \mathcal{A}$, then we get a new Abox $\mathcal{A}' = \mathcal{A} \cup \{\mathsf{a}:\mathsf{C},\ \mathsf{a}:\mathsf{D}\}$. Since $\mathcal{I}_\mathcal{D}$ satisfies $\mathsf{a}:\mathsf{C} \sqcap \mathsf{D}$, $\mathcal{I}_\mathcal{D}$ satisfies $\mathsf{a}:\mathsf{C}$ and $\mathsf{a}:\mathsf{D}$ and therefore $\mathcal{A}'$.

If the role value restriction rule is applied to $\mathsf{a}:\forall \mathsf{R} . \mathsf{C} \in \mathcal{A}$, then there must be a role assertion $(\mathsf{a}, \mathsf{b}):\mathsf{S} \in \mathcal{A}$ with $\mathsf{S} \in \mathsf{R}^{\downarrow}$ and $\mathcal{A}' = \mathcal{A} \cup \{\mathsf{b}:\mathsf{C}\}$. $\mathcal{I}_\mathcal{D}$ satisfies $\mathcal{A}$, hence it holds that $(\mathsf{a}^\mathcal{I}, \mathsf{b}^\mathcal{I}) \in \mathsf{S}^\mathcal{I}, \mathsf{S}^\mathcal{I} \subseteq \mathsf{R}^\mathcal{I}$. Since $\mathcal{I}_\mathcal{D}$ satisfies $\mathsf{a}:\forall \mathsf{R} . \mathsf{C}$, $\mathsf{b}^\mathcal{I} \in \mathsf{C}^\mathcal{I}$ must hold. Thus, $\mathcal{I}_\mathcal{D}$ satisfies $\mathsf{b}:\mathsf{C}$ and therefore $\mathcal{A}'$.

If the transitive role value restriction rule is applied to $\mathsf{a}:\forall\,\mathsf{R}\,.\,\mathsf{C}\in\mathcal{A}$, there must be an assertion $(\mathsf{a},\mathsf{b}):\mathsf{S}\in\mathcal{A}$ with $\mathsf{S}\in\mathsf{T}^{\downarrow}$ for some $\mathsf{T}\in T$ and $\mathsf{T}\in\mathsf{R}^{\downarrow}$ such that we get $\mathcal{A}'=\mathcal{A}\cup\{\mathsf{b}:\forall\,\mathsf{T}\,.\,\mathsf{C}\}$. Since $\mathcal{I}_{\mathcal{D}}$ satisfies $\mathcal{A}$, we have $\mathsf{a}^{\mathcal{I}}\in(\forall\,\mathsf{R}\,.\,\mathsf{C})^{\mathcal{I}}$ and $(\mathsf{a}^{\mathcal{I}},\mathsf{b}^{\mathcal{I}})\in\mathsf{S}^{\mathcal{I}},\mathsf{S}^{\mathcal{I}}\subseteq\mathsf{T}^{\mathcal{I}}\subseteq\mathsf{R}^{\mathcal{I}}$. It holds that $\mathsf{b}^{\mathcal{I}}\in(\forall\,\mathsf{T}\,.\,\mathsf{C})^{\mathcal{I}}$ unless there is some $z\in\Delta_{\mathcal{I}}$ with $(\mathsf{b}^{\mathcal{I}},z)\in\mathsf{T}^{\mathcal{I}}$ and $z\notin\mathsf{C}^{\mathcal{I}}$. Since T is transitive, $(\mathsf{a}^{\mathcal{I}},z)\in\mathsf{T}^{\mathcal{I}}$ and $\mathsf{a}^{\mathcal{I}}\notin(\forall\,\mathsf{R}\,.\,\mathsf{C})^{\mathcal{I}}$ in contradiction to the assumption that $\mathcal{I}$ satisfies $\mathcal{A}$. Hence, $\mathcal{I}$ must satisfy $\mathsf{b}:\forall\,\mathsf{T}\,.\,\mathsf{C}$ and therefore $\mathcal{I}_{\mathcal{D}}$ is a model for $\mathcal{A}'$.

If the universal concept restriction rule is applied to an individual a in $\mathcal{A}$ because of $\forall\,x\,.\,x:\mathsf{C}\in\mathcal{A}$, then $\mathcal{A}'=\mathcal{A}\cup\{\mathsf{a}:\mathsf{C}\}$. Since $\mathcal{I}_{\mathcal{D}}$ satisfies $\mathcal{A}$, it holds that $\mathsf{C}^{\mathcal{I}}=\Delta_{\mathcal{I}}$. Thus, it holds that $\mathsf{a}^{\mathcal{I}}\in\mathsf{C}^{\mathcal{I}}$ and $\mathcal{I}_{\mathcal{D}}$ satisfies $\mathcal{A}'$.

If the role exists restriction rule is applied to $\mathsf{a}:\exists\,\mathsf{R}\,.\,\mathsf{C}\in\mathcal{A}$, then we get the ABox $\mathcal{A}'=\mathcal{A}\cup\{(\mathsf{a},\mathsf{b}):\mathsf{R},\ \mathsf{b}:\mathsf{C}\}$. Since $\mathcal{I}_{\mathcal{D}}$ satisfies $\mathcal{A}$, there exists a $y\in\Delta_{\mathcal{I}}$ such that $(\mathsf{a}^{\mathcal{I}},y)\in\mathsf{R}^{\mathcal{I}}$ and $y\in\mathsf{C}^{\mathcal{I}}$. We define the interpretation function $\cdot^{\mathcal{I}'}$ such that $\mathsf{b}^{\mathcal{I}'}:=y$ and $x^{\mathcal{I}'}:=x^{\mathcal{I}}$ for $x\neq\mathsf{b}$. Hence, $\mathcal{I}'_{\mathcal{D}}=(\Delta_{\mathcal{I}},\Delta_{\mathcal{D}},\cdot^{\mathcal{I}'})$ satisfies $\mathcal{A}'$.

If the number restriction exists rule is applied to $\mathsf{a}:\exists_{\geq n}\,\mathsf{R}\in\mathcal{A}$, then we get $\mathcal{A}'=\mathcal{A}\cup\{(\mathsf{a},\mathsf{b_k}):\mathsf{R}\mid\mathsf{k}\in 1..\mathsf{n}\}\cup\{\mathsf{b_i}\neq\mathsf{b_j}\mid\mathsf{i},\mathsf{j}\in 1..\mathsf{n},\mathsf{i}\neq\mathsf{j}\}$. Since $\mathcal{I}_{\mathcal{D}}$ satisfies $\mathcal{A}$, there must exist n distinct individuals $y_i\in\Delta_{\mathcal{I}}$, $i\in 1..n$ such that $(\mathsf{a}^{\mathcal{I}},y_i)\in\mathsf{R}^{\mathcal{I}}$. We define the interpretation function $\cdot^{\mathcal{I}'}$ such that $\mathsf{b_i}^{\mathcal{I}'}:=y_i$ and $x^{\mathcal{I}'}:=x^{\mathcal{I}}$ for $x\notin\{\mathsf{b}_1,\ldots,\mathsf{b_n}\}$. Hence, $\mathcal{I}'_{\mathcal{D}}=(\Delta_{\mathcal{I}},\Delta_{\mathcal{D}},\cdot^{\mathcal{I}'})$ satisfies $\mathcal{A}'$.

If the predicate exists rule is applied to $\mathsf{a}:\exists\,\mathsf{f}_1,\ldots,\mathsf{f_n}\,.\,\mathsf{P}\in\mathcal{A}$, then we get the ABox $\mathcal{A}'=\mathcal{A}\cup\{(\mathsf{x}_1,\ldots,\mathsf{x_n}):\mathsf{P},(\mathsf{a},\mathsf{x}_1):\mathsf{f}_1,\ldots,(\mathsf{a},\mathsf{x_n}):\mathsf{f_n}\}$. After fork elimination, some $\mathsf{x_i}$ may be replaced by $\mathsf{z_i}$ with $\mathsf{z_i}\prec_C\mathsf{x_i}$. Since $\mathcal{I}_{\mathcal{D}}$ satisfies $\mathcal{A}$, there exist $y_1,\ldots,y_n\in\Delta_{\mathcal{D}}$ such that $\forall i\in\{1,\ldots,n\}:(\mathsf{a}^{\mathcal{I}},y_i)\in\mathsf{f_i}^{\mathcal{I}}$ and $(y_1,\ldots,y_n)\in\mathsf{P}^{\mathcal{I}}$. We define the interpretation function $\cdot^{\mathcal{I}'}$ such that $\mathsf{x_i}^{\mathcal{I}'}:=y_i$ for all $\mathsf{x_i}$ not replaced by $\mathsf{z_i}$ and $(y_1,\ldots,y_n)\in\mathsf{P}^{\mathcal{I}'}$. The fork elimination strategy used in the R∃P rule guarantees that concrete objects introduced in previous steps are not eliminated. Thus, it is ensured that the interpretation of $\mathsf{x_i}$ is not changed in $\mathcal{I}'_{\mathcal{D}}$. It is easy to see that $\mathcal{I}'_{\mathcal{D}}=(\Delta_{\mathcal{I}},\Delta_{\mathcal{D}},\cdot^{\mathcal{I}'})$ satisfies $\mathcal{A}'$.

2. "$\Leftarrow$" Assume that $\mathcal{A}'$ is satisfied by $\mathcal{I}'_{\mathcal{D}}=(\Delta_{\mathcal{I}},\Delta_{\mathcal{D}},\cdot^{\mathcal{I}'})$. By examining the nondeterministic rules we show that $\mathcal{A}$ is also consistent.

If $\mathcal{A}'$ is obtained from $\mathcal{A}$ by applying the disjunction rule, then $\mathcal{A}$ is a subset of $\mathcal{A}'$ and therefore satisfied by $\mathcal{I}'_{\mathcal{D}}$.

If $\mathcal{A}'$ is obtained from $\mathcal{A}$ by applying the number restriction merge rule to $\mathsf{a}:\exists_{\leq n}\,\mathsf{R}\in\mathcal{A}$, then there exist $\mathsf{b_i},\mathsf{b_j}$ in $\mathcal{A}$ such that $\mathcal{A}'=\mathcal{A}[\mathsf{b_i}/\mathsf{b_j}]$. We define the interpretation function $\cdot^{\mathcal{I}}$ such that $\mathsf{b_i}^{\mathcal{I}}:=\mathsf{b_j}^{\mathcal{I}'}$ and $\mathsf{x}^{\mathcal{I}}:=\mathsf{x}^{\mathcal{I}'}$ for every $\mathsf{x}\neq\mathsf{b_i}$. Obviously, $\mathcal{I}_{\mathcal{D}}=(\Delta_{\mathcal{I}},\Delta_{\mathcal{D}},\cdot^{\mathcal{I}})$ satisfies $\mathcal{A}$.

"$\Rightarrow$" We suppose that $\mathcal{I}_{\mathcal{D}}=(\Delta_{\mathcal{I}},\Delta_{\mathcal{D}},\cdot^{\mathcal{I}})$ satisfies $\mathcal{A}$ and a nondeterministic rule is applicable to an individual a in $\mathcal{A}$.

If the disjunction rule is applicable to $\mathsf{a}:\mathsf{C}\sqcup\mathsf{D}\in\mathcal{A}$ and $\mathcal{A}$ is consistent, it holds $\mathsf{a}^{\mathcal{I}}\in(\mathsf{C}\sqcup\mathsf{D})^{\mathcal{I}}$. It follows that either $\mathsf{a}^{\mathcal{I}}\in\mathsf{C}^{\mathcal{I}}$ or $\mathsf{a}^{\mathcal{I}}\in\mathsf{D}^{\mathcal{I}}$ (or both). Hence, the disjunction rule can be applied in a way that $\mathcal{I}_{\mathcal{D}}$ also satisfies the ABox $\mathcal{A}'$.

If the number restriction merge rule is applicable to $\mathsf{a}:\exists_{\leq n}\,\mathsf{R}\in\mathcal{A}$ and $\mathcal{A}$ is consistent, it holds $\mathsf{a}^{\mathcal{I}}\in(\exists_{\leq n}\,\mathsf{R})^{\mathcal{I}}$ and $\|\{\mathsf{b}\mid(\mathsf{a}^{\mathcal{I}},\mathsf{b}^{\mathcal{I}})\in R^{\mathcal{I}}\}\|\leq n$. However, it also holds $\|\{\mathsf{b}\mid(\mathsf{a}^{\mathcal{I}},\mathsf{b}^{\mathcal{I}})\in\mathsf{R}^{\mathcal{I}}\}\|\geq m$ with $m>n$. Without loss of generality we only need to consider the case that $m = n + 1$. Thus, we can conclude by the Pigeonhole Principle that there exist at least two R-successors $\mathsf{b_i}, \mathsf{b_j}$ of a such that $\mathsf{b_i}^{\mathcal{I}} = \mathsf{b_j}^{\mathcal{I}}$. Since $\mathcal{I}_{\mathcal{D}}$ satisfies $\mathcal{A}$, it must have been possible to map $\mathsf{b_i}$ and $\mathsf{b_j}$ to the same domain object, i.e. at least one of the two individuals must be a new individual. Let us assume $\mathsf{b_i} \in O_N$, then $\mathcal{I}_{\mathcal{D}}$ obviously satisfies $\mathcal{A}[\mathsf{b_i}/\mathsf{b_j}]$.

In order to define a canonical interpretation from a completion $\mathcal{A}$, the notion of a specific blocking individual is introduced. We call a the *witness* of b iff b is blocked by a and $\neg\exists\, \mathsf{c}$ in $\mathcal{A} : \mathsf{c} \in O_N, \mathsf{c} \prec \mathsf{a}, \sigma(\mathcal{A}, \mathsf{c}) \supseteq \sigma(\mathcal{A}, \mathsf{b})$. The witness for a blocked individual is unique (see [6]). Note that the canonical interpretation is constructed differently from the one describe in [7].

Definition 9. *Let $\mathcal{A}$ be a complete ABox that has been derived by the calculus from an augmented ABox $\mathcal{A}_{\mathcal{T}}$. Since $\mathcal{A}$ is clash-free, there exists a variable assignment α that satisfies (the conjunction of) all occurring assertions $(\mathsf{x_1}, \ldots, \mathsf{x_n}) : \mathsf{P} \in \mathcal{A}$. We define the* canonical interpretation $\mathcal{I}_{\mathcal{C}} = (\Delta_{\mathcal{I}_{\mathcal{C}}}, \Delta_{\mathcal{D}}, \cdot^{\mathcal{I}_{\mathcal{C}}})$ *w.r.t. $\mathcal{A}$ as follows:*

1. $\Delta_{\mathcal{I}_{\mathcal{C}}} := \{\mathsf{a} \mid \mathsf{a}$ *is mentioned in* $\mathcal{A}\}$
2. $\mathsf{a}^{\mathcal{I}_{\mathcal{C}}} := \mathsf{a}$ *iff* a *is mentioned in* $\mathcal{A}$
3. $\mathsf{x}^{\mathcal{I}_{\mathcal{C}}} := \alpha(\mathsf{x})$ *iff* x *is mentioned in* $\mathcal{A}$
4. $\mathsf{a} \in \mathsf{A}^{\mathcal{I}_{\mathcal{C}}}$ *iff* $\mathsf{a} : \mathsf{A} \in \mathcal{A}$ *and* A *is a concept name*
5. $(\mathsf{a}, \alpha(\mathsf{x})) \in \mathsf{f}^{\mathcal{I}_{\mathcal{C}}}$ *iff* $(\mathsf{a}, \mathsf{x}) : \mathsf{f} \in \mathcal{A}$
6. $(\mathsf{a}, \mathsf{b}) \in \mathsf{R}^{\mathcal{I}_{\mathcal{C}}}$ *iff* $\exists\, \mathsf{c_0}, \ldots, \mathsf{c_n}, \mathsf{d_0}, \ldots, \mathsf{d_{n-1}}$ *mentioned in* $\mathcal{A}$:[2],
   (a) $\mathsf{n} \geq 1, \mathsf{c_0} = \mathsf{a}, \mathsf{c_n} = \mathsf{b}$, *and*

   (b) $(\mathsf{a}, \mathsf{c_1}) : \mathsf{S_1}$, $(\mathsf{d_1}, \mathsf{c_2}) : \mathsf{S_2}$, $\ldots$ $(\mathsf{d_{n-2}}, \mathsf{c_{n-1}}) : \mathsf{S_{n-1}}$, $(\mathsf{d_{n-1}}, \mathsf{b}) : \mathsf{S_n} \in \mathcal{A}$, *and*

   (c) $\forall \mathsf{i} \in 1..\mathsf{n} - 1$:
   $\mathsf{d_i} = \mathsf{c_i}$ *or*
   $\mathsf{d_i}$ *is a witness for* $\mathsf{c_i}$, *and* $(\mathsf{d_i}, \mathsf{c_{i+1}}) : \mathsf{S_{i+1}} \in \mathcal{A}$, *and*

   (d) *if* $\mathsf{n} > 1$
   $\forall\, \mathsf{i} \in 1..\mathsf{n} : \exists \mathsf{R}' \in \mathcal{T}$, $\mathsf{R}' \in \mathsf{R}^{\downarrow}$, $\mathsf{S_i} \in \mathsf{R}'^{\downarrow}$
   else
   $\mathsf{S_1} \in \mathsf{R}^{\downarrow}$.

The construction of the canonical interpretation for the case 6 is illustrated with an example in Figure 2. The following cases can be seen as special cases of case 6 introduced above ($\mathsf{n} = 1, \mathsf{c_0} = \mathsf{a}, \mathsf{c_1} = \mathsf{b}$):

- $\mathsf{c_0} = \mathsf{d_0}$: $(\mathsf{a}, \mathsf{b}) \in \mathsf{R}^{\mathcal{I}_{\mathcal{C}}}$ iff $(\mathsf{c_0}, \mathsf{c_1}) : \mathsf{S_1} \in \mathcal{A}$ for a role $\mathsf{S_1} \in \mathsf{R}^{\downarrow}$.
- $\mathsf{c_0} \neq \mathsf{d_0}$: $(\mathsf{a}, \mathsf{b}) \in \mathsf{R}^{\mathcal{I}_{\mathcal{C}}}$ iff $\mathsf{d_0}$ is a witness for $\mathsf{c_0}$, and $(\mathsf{d_0}, \mathsf{c_1}) : \mathsf{S_1} \in \mathcal{A}$, for a role $\mathsf{S_1} \in \mathsf{R}^{\downarrow}$.

Since the witness of an individual is unique, the canonical interpretation is well-defined because there exists a unique blocking individual (witness) for each individual that is blocked.

[2] Note that the variables $\mathsf{c_0}, \ldots, \mathsf{c_n}, \mathsf{d_0}, \ldots, \mathsf{d_{n-1}}$ not necessarily denote different individual names.

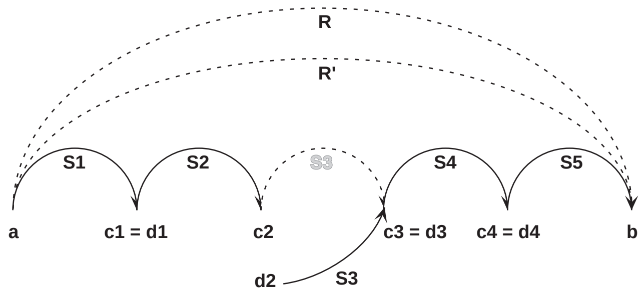

Fig. 2. Construction of the canonical interpretation. In the lower example we assume that the individual $\mathsf{d2}$ is a witness for $\mathsf{c2}$ (see text).

Lemma 2 (Soundness). *Let $\mathcal{A}$ be a complete ABox that has been derived by the calculus from an augmented ABox $\mathcal{A}_\mathcal{T}$, then $\mathcal{A}_\mathcal{T}$ has a model.*

Proof. Let $\mathcal{I}_\mathcal{C} = (\Delta_{\mathcal{I}_\mathcal{C}}, \Delta_\mathcal{D}, \cdot^{\mathcal{I}_\mathcal{C}})$ be the canonical interpretation for the ABox $\mathcal{A}$ constructed w.r.t. the TBox $\mathcal{T}$. $\mathcal{A}$ is clash-free.

Features are interpreted in the correct way: There can be no forks in $\mathcal{A}$ because (i) there are no forks in the augmented ABox $\mathcal{A}_\mathcal{T}$ and (ii) forks are immediately eliminated after an application of the R$\exists$P rule. This rule is the only rule that introduces new assertions of the form $(\mathsf{a}, \mathsf{x}) : \mathsf{f} \in \mathcal{A}$. Note that forks cannot be introduced by the R$\exists_{\leq n}$ rule due to the completion strategy. Thus, $\mathcal{I}_\mathcal{C}$ maps features to (partial) functions because the variable assignment α is a function.

All role inclusions in the role hierarchy are satisfied: For every $\mathsf{S} \sqsubseteq \mathsf{R}$ it holds that $\mathsf{S}^{\mathcal{I}_\mathcal{C}} \subseteq \mathsf{R}^{\mathcal{I}_\mathcal{C}}$ This can be shown as follows. If $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}^{\mathcal{I}_\mathcal{C}}) \in S^{\mathcal{I}_\mathcal{C}}$, case 6 of Definition 9 must be applicable. Hence, there exists a chain of sub-roles possibly with gaps and witnesses (see Definition 9, case 6). Thus, the corresponding construction for $\mathcal{I}_\mathcal{C}$ adding $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}^{\mathcal{I}_\mathcal{C}})$ to $\mathsf{S}^{\mathcal{I}_\mathcal{C}}$ is also applicable to R since $\mathsf{S} \in \mathsf{R}^{\downarrow}$ (see 6d). Therefore, there is also a tuple $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}^{\mathcal{I}_\mathcal{C}}) \in R^{\mathcal{I}_\mathcal{C}}$.

All (implicit) transitivity axioms are satisfied, i.e. transitive roles are interpreted in the correct way: $\forall\, \mathsf{R} \in T : \mathsf{R}^{\mathcal{I}_\mathcal{C}} = (\mathsf{R}^{\mathcal{I}_\mathcal{C}})^+$. If there exist $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}^{\mathcal{I}_\mathcal{C}}) \in R^{\mathcal{I}_\mathcal{C}}$ and $(\mathsf{b}^{\mathcal{I}_\mathcal{C}}, \mathsf{c}^{\mathcal{I}_\mathcal{C}}) \in R^{\mathcal{I}_\mathcal{C}}$ then case 6 in Definition 9 must have been applied for each tuple. But then, a chain of roles from a to c exists as well (possibly with gaps and witnesses) such that $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{c}^{\mathcal{I}_\mathcal{C}})$ is added to $\mathsf{R}^{\mathcal{I}_\mathcal{C}}$ as well.

In the following we prove that $\mathcal{I}_\mathcal{C}$ satisfies every assertion in $\mathcal{A}$.

For any $\mathsf{a} \neq \mathsf{b} \in \mathcal{A}$ or $(\mathsf{a}, \mathsf{b}) : \mathsf{R} \in \mathcal{A}$, $\mathcal{I}_\mathcal{C}$ satisfies them by definition.

For any $(\mathsf{a}, \mathsf{x}) : \mathsf{f} \in \mathcal{A}$, $\mathcal{I}_\mathcal{C}$ satisfies them by definition.

For any $(\mathsf{x}_1, \ldots, \mathsf{x}_n) : \mathsf{P} \in \mathcal{A}$, $\mathcal{I}_\mathcal{C}$ satisfies them by definition. Since $\mathcal{A}$ is clash-free there exists a variable assignment such that the conjunction of all predicate assertions is satisfied. The variable assignment can be computed because the concrete domain is required to be admissible.

Next we consider assertions of the form $\mathsf{a} : \mathsf{C}$. We show by induction on the structure of concepts that $\mathsf{a} : \mathsf{C} \in \mathcal{A}$ implies $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in \mathsf{C}^{\mathcal{I}_\mathcal{C}}$.

If C is a concept name, then $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in \mathsf{C}^{\mathcal{I}_\mathcal{C}}$ by definition of $\mathcal{I}_\mathcal{C}$.

If $\mathsf{C} = \neg\mathsf{D}$, then D is a concept name since all concepts are in negation normal form (see Definition 3). $\mathcal{A}$ is clash-free and cannot contain $\mathsf{a}:\mathsf{D}$. Thus, $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \notin \mathsf{D}^{\mathcal{I}_\mathcal{C}}$, i.e. $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in \Delta_{\mathcal{I}_\mathcal{C}} \setminus \mathsf{D}^{\mathcal{I}_\mathcal{C}}$. Hence $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in (\neg\mathsf{D})^{\mathcal{I}_\mathcal{C}}$.

If $\mathsf{C} = \mathsf{C}_1 \sqcap \mathsf{C}_2$ then (since $\mathcal{A}$ is complete) $\mathsf{a}:\mathsf{C}_1 \in \mathcal{A}$ and $\mathsf{a}:\mathsf{C}_2 \in \mathcal{A}$. By induction hypothesis, $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in \mathsf{C}_1{}^{\mathcal{I}_\mathcal{C}}$ and $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in \mathsf{C}_2{}^{\mathcal{I}_\mathcal{C}}$. Hence $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in (\mathsf{C}_1 \sqcap \mathsf{C}_2)^{\mathcal{I}_\mathcal{C}}$.

If $\mathsf{C} = \mathsf{C}_1 \sqcup \mathsf{C}_2$ then (since $\mathcal{A}$ is complete) either $\mathsf{a}:\mathsf{C}_1 \in \mathcal{A}$ or $\mathsf{a}:\mathsf{C}_2 \in \mathcal{A}$. By induction hypothesis, $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in \mathsf{C}_1{}^{\mathcal{I}_\mathcal{C}}$ or $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in \mathsf{C}_2{}^{\mathcal{I}_\mathcal{C}}$. Hence $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in (\mathsf{C}_1 \sqcup \mathsf{C}_2)^{\mathcal{I}_\mathcal{C}}$.

If $\mathsf{C} = \forall\,\mathsf{R}\,.\,\mathsf{D}$, then it must be shown that for all $\mathsf{b}^{\mathcal{I}_\mathcal{C}}$ with $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{R}^{\mathcal{I}_\mathcal{C}}$ it holds that $\mathsf{b}^{\mathcal{I}_\mathcal{C}} \in \mathsf{D}^{\mathcal{I}_\mathcal{C}}$. If $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{R}^{\mathcal{I}_\mathcal{C}}$, then according to Definition 9, b is a successor of a via a chain of roles $\mathsf{S_i} \in \mathsf{R}^{\downarrow}$ or there exists corresponding witnesses as domain elements of $\mathsf{S_i} \in \mathsf{R}^{\downarrow}$, i.e. the chain might contain "gaps" with associated witnesses (see Figure 2). Since $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{R}^{\mathcal{I}_\mathcal{C}}$ and $\mathsf{S_i}^{\mathcal{I}_\mathcal{C}} \subseteq \mathsf{R}^{\mathcal{I}_\mathcal{C}}$ there exists tuples $(\mathsf{c_i}^{\mathcal{I}_\mathcal{C}}, \mathsf{c_{i+1}}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{S_i}^{\mathcal{I}_\mathcal{C}}$. Due to Definition 9 it holds that $\forall \mathsf{i} \in 1..\mathsf{n}: \exists \mathsf{R}' \in T, \mathsf{R}' \in \mathsf{R}^{\downarrow}, \mathsf{S_i} \in \mathsf{R}'^{\downarrow}$. Therefore $\mathsf{c_k}:\forall\,\mathsf{R}'\,.\,\mathsf{D} \in \mathcal{A}, (\mathsf{k} \in 1..\mathsf{n}-1)$ because $\mathcal{A}$ is complete. For the same reason $\mathsf{b}:\mathsf{D} \in \mathcal{A}$. By induction hypothesis it holds that $\mathsf{b}^{\mathcal{I}_\mathcal{C}} \in \mathsf{D}^{\mathcal{I}_\mathcal{C}}$. As mentioned before, the chain of roles can have one or more "gaps" (see Figure 2). However, due to Definition 9 in case of a "gap" there exists a witness such that a similar argument as in case 6 can be applied, i.e. in case of a gap between $\mathsf{c_i}$ and $\mathsf{c_{i+1}}$ with witness $\mathsf{d_i}$ for $\mathsf{c_i}$, the blocking condition ensures that the concept set of the witness is a superset of the concept set of the blocked individual. Since it is assumed that $(\mathsf{d_i}, \mathsf{c_{i+1}}):\mathsf{S_{i+1}} \in \mathcal{A}$ and $\mathcal{A}$ is complete it holds that $\mathsf{c_{i+1}}:\forall\,\mathsf{R}'\,.\,\mathsf{D} \in \mathcal{A}$. Applying the same argument inductively, we can conclude that $\mathsf{c_{n-1}}:\forall\,\mathsf{R}'\,.\,\mathsf{D} \in \mathcal{A}$ and again, we have $\mathsf{b}^{\mathcal{I}_\mathcal{C}} \in \mathsf{D}^{\mathcal{I}_\mathcal{C}}$ by induction hypothesis.

If $\mathsf{C} = \exists\,\mathsf{R}\,.\,\mathsf{D}$, then it must be shown that there exists an individual $\mathsf{b}^{\mathcal{I}_\mathcal{C}} \in \Delta_{\mathcal{I}_\mathcal{C}}$ with $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{R}^{\mathcal{I}_\mathcal{C}}$ and $\mathsf{b}^{\mathcal{I}_\mathcal{C}} \in \mathsf{D}^{\mathcal{I}_\mathcal{C}}$. Since ABox $\mathcal{A}$ is complete, we have either $(\mathsf{a}, \mathsf{b}):\mathsf{S} \in \mathcal{A}$ with $\mathsf{S} \in \mathsf{R}^{\downarrow}$ and $\mathsf{b}:\mathsf{D} \in \mathcal{A}$ or a is blocked by an individual c and $(\mathsf{c}, \mathsf{b}):\mathsf{S} \in \mathcal{A}$ (again $\mathsf{S} \in \mathsf{R}^{\downarrow}$). In the first case we have $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{R}^{\mathcal{I}_\mathcal{C}}$ by the definition of $\mathcal{I}_\mathcal{C}$ (case 6, $\mathsf{n} = 1, \mathsf{c_i} = \mathsf{d_i}$) and $\mathsf{b}^{\mathcal{I}_\mathcal{C}} \in \mathsf{D}^{\mathcal{I}_\mathcal{C}}$ by induction hypothesis. In the second case there exists the witness c with $\mathsf{c}:\exists\,\mathsf{R}\,.\,\mathsf{D} \in \mathcal{A}$. By definition c cannot be blocked, and by hypothesis $\mathcal{A}$ is complete. So we have an individual b with $(\mathsf{c}, \mathsf{b}):\mathsf{S} \in \mathcal{A}$ $(\mathsf{S} \in \mathsf{R}^{\downarrow})$ and $\mathsf{b}:\mathsf{D} \in \mathcal{A}$. By induction hypothesis we have $\mathsf{b}^{\mathcal{I}_\mathcal{C}} \in \mathsf{D}^{\mathcal{I}_\mathcal{C}}$, and by the definition of $\mathcal{I}_\mathcal{C}$ (case 6, $\mathsf{n} = 1, \mathsf{c_i} \neq \mathsf{d_i}$, $\mathsf{d_i}$ is a witness for $\mathsf{c_i}$, and $\mathsf{a} = \mathsf{c_i}, \mathsf{c} = \mathsf{d_i}$) we have $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{R}^{\mathcal{I}_\mathcal{C}}$.

If $\mathsf{C} = \exists_{\geq n}\,\mathsf{R}$, we prove the hypothesis by contradiction. We assume that $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \notin (\exists_{\geq n}\,\mathsf{R})^{\mathcal{I}_\mathcal{C}}$. Then there exist at most m $(0 \leq m < n)$ distinct S-successors of a with $\mathsf{S} \in \mathsf{R}^{\downarrow}$. Two cases can occur: (1) the individual a is not blocked in $\mathcal{I}_\mathcal{C}$. Then we have less than n S-successors of a in $\mathcal{A}$, and the $\mathsf{R}\exists_{\geq n}$-rule is applicable to a. This contradicts the assumption that $\mathcal{A}$ is complete. (2) a is blocked by an individual c but the same argument as in case (1) holds and leads to the same contradiction.

For $\mathsf{C} = \exists_{\leq n}\,\mathsf{R}$ we show the goal by contradiction. Suppose that $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \notin (\exists_{\leq n}\,\mathsf{R})^{\mathcal{I}_\mathcal{C}}$. Then there exist at least $n+1$ distinct individuals $\mathsf{b}_1{}^{\mathcal{I}_\mathcal{C}}, \ldots, \mathsf{b}_{n+1}{}^{\mathcal{I}_\mathcal{C}}$ such that $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}_i{}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{R}^{\mathcal{I}_\mathcal{C}}$, $i \in 1..n+1$. The following two cases can occur. (1) The individual a is not blocked: We have $n+1$ $(\mathsf{a}, \mathsf{b}_i):\mathsf{S}_i \in \mathcal{A}$ with $\mathsf{S}_i \in \mathsf{R}^{\downarrow}$ and $\mathsf{S}_i \notin T$,

$i \in 1..n+1$. The R$\exists_{\leq n}$ rule cannot be applicable since $\mathcal{A}$ is complete and the b_i are distinct, i.e. $\mathsf{b}_i \not\doteq \mathsf{b}_j \in \mathcal{A}$, $i, j \in 1..n+1$, $i \neq j$. This contradicts the assumption that $\mathcal{A}$ is clash-free. (2) There exists a witness c for a with $(\mathsf{c}, \mathsf{b}_i) : \mathsf{S}_i \in \mathcal{A}$, $\mathsf{S}_i \in \mathsf{R}^{\downarrow}$, and $\mathsf{S}_i \notin T$, $i \in 1..n+1$. This leads to an analogous contradiction. Due to the construction of the canonical interpretation in case of a blocking condition (with c being the witness) and a non-transitive role R (R is required to be a simple role, see the syntactic restrictions for number restrictions and role boxes), there is no $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}_k{}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{R}^{\mathcal{I}_\mathcal{C}}$ if there is no $(\mathsf{c}^{\mathcal{I}_\mathcal{C}}, \mathsf{b}_k{}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{R}^{\mathcal{I}_\mathcal{C}}$ $(\mathsf{k} \in 1..\mathsf{n}+1)$.

If $\mathsf{C} = \exists\, \mathsf{f}_1, \ldots, \mathsf{f}_\mathsf{n} \,.\, \mathsf{P}$ we show that there exist concrete objects $y_1, \ldots, y_n \in \Delta_\mathcal{D}$ such that $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, y_1) \in \mathsf{f}_1{}^{\mathcal{I}_\mathcal{C}}, \ldots, (\mathsf{a}^{\mathcal{I}_\mathcal{C}}, y_n) \in \mathsf{f}_\mathsf{n}{}^{\mathcal{I}_\mathcal{C}}$ and $(y_1, \ldots, y_n) \in \mathsf{P}^{\mathcal{I}_\mathcal{C}}$. The R$\exists$P rule generates assertions $(\mathsf{a}, \mathsf{x}_1) : \mathsf{f}_1, \ldots, (\mathsf{a}, \mathsf{x}_\mathsf{n}) : \mathsf{f}_\mathsf{n}, (\mathsf{x}_1, \ldots, \mathsf{x}_\mathsf{n}) : \mathsf{P}$. Since $\mathcal{A}$ is clash-free there is no concrete domain clash. Hence there exists a variable assignment α that maps $\mathsf{x}_1, \ldots, \mathsf{x}_\mathsf{n}$ to elements of $\Delta_\mathcal{D}$. The conjunction of concrete domain predicates is satisfiable and $(\mathsf{x}_1{}^{\mathcal{I}_\mathcal{C}}, \ldots, \mathsf{x}_\mathsf{n}{}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{P}^{\mathcal{I}_\mathcal{C}}$. By definition of $\mathcal{I}_\mathcal{C}$ it holds that $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{x}_1{}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{f}_1{}^{\mathcal{I}_\mathcal{C}}, \ldots, (\mathsf{a}^{\mathcal{I}_\mathcal{C}}, \mathsf{x}_\mathsf{n}{}^{\mathcal{I}_\mathcal{C}}) \in \mathsf{f}_\mathsf{n}{}^{\mathcal{I}_\mathcal{C}}$. Thus, there exist $y_1, \ldots, y_n$ such that the above-mentioned requirements are fulfilled and therefore $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in (\exists\, \mathsf{f}_1, \ldots, \mathsf{f}_\mathsf{n} \,.\, \mathsf{P})^{\mathcal{I}_\mathcal{C}}$

If $\mathsf{C} = \forall\, \mathsf{f} \,.\, \bot_\mathcal{D}$ then we show that $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in (\forall\, \mathsf{f} \,.\, \bot_\mathcal{D})^{\mathcal{I}_\mathcal{C}}$. Because $\mathcal{A}$ is clash-free, there cannot be an assertion $(\mathsf{a}, \mathsf{x}) : \mathsf{f} \in \mathcal{A}$ for some x in O_c and an $\mathsf{f} \in F$. Thus, it does not hold that there exists $(\mathsf{a}^{\mathcal{I}_\mathcal{C}}, y) \in \mathsf{f}^{\mathcal{I}_\mathcal{C}}$ and hence $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in (\forall\, \mathsf{f} \,.\, \bot_\mathcal{D})^{\mathcal{I}_\mathcal{C}}$.

If $\forall\, x \,.\, x : \mathsf{D} \in \mathcal{A}$, then –due to the completeness of $\mathcal{A}$– for each individual a in $\mathcal{A}$ we have $\mathsf{a} : \mathsf{D} \in \mathcal{A}$ and, by the previous cases, $\mathsf{a}^{\mathcal{I}_\mathcal{C}} \in \mathsf{D}^{\mathcal{I}_\mathcal{C}}$. Thus, $\mathcal{I}_\mathcal{C}$ satisfies $\forall\, x \,.\, x : \mathsf{D}$. Finally, since $\mathcal{I}_\mathcal{C}$ satisfies all assertions in $\mathcal{A}$, $\mathcal{I}_\mathcal{C}$ satisfies $\mathcal{A}$.

Lemma 3 (Completeness). *Let $\mathcal{A}_\mathcal{T}$ be an augmented ABox be a role box. If $\mathcal{A}_\mathcal{T}$ is consistent, then there exists at least one completion $\mathcal{A}'$ being computed by applying the completion rules.*

Proof. By contraposition: Obviously, an ABox containing a clash is inconsistent. If there does not exists a completion of $\mathcal{A}_\mathcal{T}$, then it follows from Proposition 1 that the ABox $\mathcal{A}_\mathcal{T}$ is inconsistent.

Lemma 4 (Termination). *The calculus described above terminates on every (augmented) input ABox.*

Proof. The termination of the calculus is shown by specifying an upper limit on the number of assertions that can result from an (augmented) input ABox of a certain length n. Compared to $\mathcal{ALCNH}_{R^+}$ in the termination proof for $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ the additional constructs for concrete domains have to be considered. Basically, since features do not "interact" with value and number restrictions (see the completion rules), the same upper limit $O(2^{4n})$ for a completion can be derived. For details see [6].

Theorem 1 (Decidability). *Let $\mathcal{D}$ be an admissible concrete domain. Checking whether an $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ knowledge base $(\mathcal{T}, \mathcal{A})$ is consistent is a decidable problem.*

Proof. Given a knowledge base $(\mathcal{T}, \mathcal{A})$, an augmented ABox $\mathcal{A}_\mathcal{T}$ can be constructed in linear time. The claim follows from Lemmas 1, 2, 3, and 4.

5 Conclusion

We presented a tableaux calculus deciding the knowledge base consistency problem for the description logic $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$. Applications of the logic in the context of configuration problems have been sketched. The Cylinder example demonstrates that some requirements of a model-based configuration system are fulfilled by $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$. The calculus presented in this paper can be used to solve "simple" configuration problems in which the configuration space can be described by an $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ knowledge base (see [6] for an analysis of the models resulting from the canonical interpretation). We conjecture that concrete domains without features chains can also be included in description logics with inverse roles and qualified number restrictions.

A highly optimized variant of the calculus for the sublogic $\mathcal{ALCNH}_{R^+}$ is already implemented in the ABox description logic system RACE. RACE is available at http://kogs-www.informatik.uni-hamburg.de/˜race/. RACE will be extended with support for reasoning with concrete domains in the near future. With this paper we provide a sound basis for practical extensions of expressive DL systems such that, for instance, construction problems can be effectively solved with description logic reasoning techniques.

References

1. F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Twelfth International Conference on Artificial Intelligence, 1991*, pages 452–457, 1991. A longer version appeared as Tech. Report DFKI-RR-91-10.
2. F. Baader and P. Hanschke. Extensions of concept languages for a mechanical engineering application. In H.J. Ohlbach, editor, *Proceedings, GWAI-92: 16th German Conference on Artificial Intelligence*, pages 132–143. Springer, 1992.
3. M. Buchheit, R. Klein, and W. Nutt. Configuration as model construction: The constructive problem solving approach. In F. Sudweeks and J. Gero, editors, *Proc. 4th Int. Conf. on Artificial Intelligence in Design*. Kluwer, Dordrecht, 1994.
4. V. Haarslev and R. Möller. Consistency testing: The RACE experience. In *Proceedings International Conference Tableaux'2000*. Springer-Verlag, 2000.
5. V. Haarslev and R. Möller. Expressive ABox reasoning with number restrictions, role hierachies, and transitively closed roles. In A.G. Cohn, F. Giunchiglia, and B. Selman, editors, *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2000)*, 2000.
6. V. Haarslev, R. Möller, and M. Wessel. The description logic $\mathcal{ALCNH}_{R+}$ extended with concrete domains. Technical Report FBI-HH-M-290/00, University of Hamburg, Computer Science Department, August 2000.
7. I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99)*, number 1705 in LNAI, pages 161–180. Springer, 1999.
8. C. Lutz. The complexity of reasoning with concrete domains (revised version). LTCS-Report 99-01, LuFG Theoretical Computer Science, RWTH Aachen, 1999.
9. J.R. Wright, E.S. Weixelbaum, G.T. Vesonder, K. Brown, S.R. Palmer, J.I. Berman, and H.H. Moore. A knowledge-based configurator that supports sales, engineering, and manufacturing at AT&T network systems. *AI Magazine*, 14(3):69–80, 1993.