Combining Tableau and Algebraic Methods for Reasoning with Qualified Number Restrictions in Description Logics

Volker Haarslev*, **Martina Timmann***, **Ralf Möller****

*University of Hamburg
Computer Science Department
Vogt-Kölln-Str. 30
22527 Hamburg, Germany
haarslev@informatik.uni-hamburg.de

**Univ. of Appl. Sciences in Wedel
Computer Science Department
Feldstrasse 143
22880 Wedel, Germany
rmoeller@fh-wedel.de

Abstract

The paper investigates an optimization technique for reasoning with *qualified number restrictions* in the description logic $\mathcal{ALCQH}_{R^+}$ (a.k.a. $\mathcal{SHQ}$), which can be seen as one of the cornerstones for reasoning technology in the context of, for instance, the semantic web activity. We present a hybrid architecture where a standard tableaux calculus is combined with a procedure deciding the satisfiability of linear inequations derived from qualified number restrictions. The advances are demonstrated by an empirical evaluation using the description logic system RACER. The evaluation demonstrates a dramatic speed up compared to other known approaches.

1 Introduction

Description logics (DLs) provide an important cornerstone in the development of reasoning technology in the context of the semantic web initiative [1]. In particular, recent developments in optimization techniques for DL systems based on *tableau calculi* demonstrate the practical applicability of formal reasoning systems in the web context. This has been shown not only concerning terminological knowledge (TBoxes) but also w.r.t. assertional knowledge about specific individuals (ABoxes). Furthermore, in the context of the semantic web, it has been argued that expressive description logics are required [2]. In particular, the logics $\mathcal{SHIQ}$ [16] and $\mathcal{SHOQ}$ [14] have been proposed. As a common basis of all these logics for the semantic web, in this paper, we investigate the logic $\mathcal{SHQ}$ (aka. $\mathcal{ALCQH}_{R^+}$ in another naming scheme for DLs [10]).

Over the last few years dedicated optimizations techniques have been proposed for dealing with expressive description logics. These techniques turned out to be very effective for TBoxes and ABoxes. In particular, for the logic $\mathcal{SHIQ}$, with FaCT [17, 13] and with RACER[1] [6, 10], optimization techniques suitable for large-scale practical applications have been investigated [8]. Despite recent advances, there exist only very few proposals for optimization techniques addressing the sources of complexity introduced by so-called *qualified number restrictions*, which are part of the logics discussed above. Number restrictions are often required in practical applications, e.g., in the context of configuration problems [18]. Qualified number restrictions provide for an additional expressivity [12].

[1] RACER download page: http://kogs-www.informatik.uni-hamburg.de/~race/

Syntax	**Semantics**
Concepts	
A	$\mathsf{A}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, A is a concept name
$\neg\mathsf{C}$	$\Delta^{\mathcal{I}} \setminus \mathsf{C}^{\mathcal{I}}$
$\mathsf{C} \sqcap \mathsf{D}$	$\mathsf{C}^{\mathcal{I}} \cap \mathsf{D}^{\mathcal{I}}$
$\mathsf{C} \sqcup \mathsf{D}$	$\mathsf{C}^{\mathcal{I}} \cup \mathsf{D}^{\mathcal{I}}$
$\exists\,\mathsf{R}\,.\,\mathsf{C}$	$\{a \in \Delta^{\mathcal{I}} \mid \exists\, b \in \Delta^{\mathcal{I}} : (a,b) \in \mathsf{R}^{\mathcal{I}},\ b \in \mathsf{C}^{\mathcal{I}}\}$
$\forall\,\mathsf{R}\,.\,\mathsf{C}$	$\{a \in \Delta^{\mathcal{I}} \mid \forall\, b : (a,b) \in \mathsf{R}^{\mathcal{I}} \Rightarrow b \in \mathsf{C}^{\mathcal{I}}\}$
$\exists_{\geq n}\,\mathsf{S}\,.\,\mathsf{C}$	$\{a \in \Delta^{\mathcal{I}} \mid \mathsf{S}^{\sharp}(a,\mathsf{C}) \geq n\}$
$\exists_{\leq m}\,\mathsf{S}\,.\,\mathsf{C}$	$\{a \in \Delta^{\mathcal{I}} \mid \mathsf{S}^{\sharp}(a,\mathsf{C}) \leq m\}$
Roles	
R	$\mathsf{R}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

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

Figure 1: Syntax and Semantics of $\mathcal{ALCQH}_{R^+}$ ($n, m \in \mathbb{N}$, $n > 1, m > 0$, $\|\cdot\|$ denotes set cardinality, $\mathsf{S} \in S$, $\mathsf{S}^{\sharp}(a,\mathsf{C}) = \|\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in \mathsf{S}^{\mathcal{I}},\ b \in \mathsf{C}^{\mathcal{I}}\}\|$).

1.1 The Language $\mathcal{ALCQH}_{R^+}$

We briefly introduce the description logic (DL) $\mathcal{ALCQH}_{R^+}$ (see the tables in Figure 1) using a standard Tarski-style semantics based on an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. $\mathcal{ALCQH}_{R^+}$ extends the description logic $\mathcal{ALCNH}_{R^+}$ [6] by qualified number restrictions. We assume a set of concept names C and a set of role names R. The mutually disjoint subsets P and T of R denote non-transitive and transitive roles, respectively ($R = P \cup T$).

If $\mathsf{R}, \mathsf{S} \in R$ are role names, then the terminological axiom $\mathsf{R} \sqsubseteq \mathsf{S}$ 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 *descendants* of a role R. $\mathsf{R}^{\uparrow} = \{\mathsf{S} \in R \mid \mathsf{R} \sqsubseteq^* \mathsf{S}\}$ is the set of *ancestors* of a role R. 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 descendant. A syntactic restriction holds for the combinability of number restrictions and transitive roles in $\mathcal{ALCQH}_{R^+}$. Number restrictions are only allowed for *simple* roles. This restriction is motivated by an undecidability result in case of an unrestricted combinability [15]. The concept name $\top$ ($\bot$) is used as an abbreviation for $\mathsf{C} \sqcup \neg\mathsf{C}$ ($\mathsf{C} \sqcap \neg\mathsf{C}$).

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. to a given role hierarchy $\mathcal{R}$.[2] We use $\mathsf{C} \doteq \mathsf{D}$ as an abbreviation for $\{\mathsf{C} \sqsubseteq \mathsf{D}, \mathsf{D} \sqsubseteq \mathsf{C}\}$.

The *concept satisfiability problem* is to decide whether a given concept term C is *satisfiable* w.r.t. to $\mathcal{T}$ and $\mathcal{R}$, i.e., whether there exists an interpretation $\mathcal{I}$ such that $\mathcal{I}$ satisfies $\mathcal{T}$ and $\mathcal{R}$ and $\mathsf{C}^{\mathcal{I}} \neq \emptyset$.

1.2 Problems with Qualified Number Restrictions

In [9] a particular tableaux calculus for $\mathcal{ALCQH}_{R^+}$, the so-called signature calculus, is presented. The signature calculus addresses a source of inefficiency which is caused by standard tableau calculi dealing with qualified number restrictions (e.g., see [12] for $\mathcal{ALCQ}$). The

[2]The reference to $\mathcal{R}$ is omitted in the following.

signature calculus offers a compact representation for role successors using so-called proxy individuals. A proxy individual represents a set of role successors. Its corresponding signature can be understood as a representation for the cardinality of a set of qualified role successors (see [9] for details). This compact representation of role successors is largely independent from the values of numbers occurring in qualified number restrictions. New proxy individuals are generated, if required, in order to capture the interaction of number restrictions and role hierarchies. The use of the signature calculus already indicates a dramatic performance gain of several orders of magnitude (see [9]).

However, there still exists a large class of problems (using qualified number restrictions) which cannot be efficiently dealt with by the signature calculus. One of these problems is illustrated as follows. Let us assume a role name R, the concept names $\mathsf{C}_1, \ldots, \mathsf{C}_7, \mathsf{P}_1, \ldots, \mathsf{P}_4$, the axioms $\mathsf{C}_3 \sqsubseteq \mathsf{C}_1$, $\mathsf{C}_4 \sqsubseteq \mathsf{C}_1$, $\mathsf{C}_5 \sqsubseteq \mathsf{C}_1 \sqcap \mathsf{C}_2$, $\mathsf{C}_6 \sqsubseteq \mathsf{C}_2$, $\mathsf{C}_7 \sqsubseteq \mathsf{C}_2$, and the concept D defined as follows.

$$
\begin{aligned}
\mathbf{D} \doteq\ & \exists_{\geq 2}\, \mathsf{R} \,.\, (\mathsf{C}_3 \sqcap \mathsf{P}_1) \sqcap \exists_{\geq 3}\, \mathsf{R} \,.\, (\mathsf{C}_4 \sqcap \mathsf{P}_1) \sqcap \exists_{\geq 1}\, \mathsf{R} \,.\, (\mathsf{C}_5 \sqcap \neg \mathsf{P}_1 \sqcap \mathsf{P}_2 \sqcap \mathsf{P}_3 \sqcap \neg \mathsf{P}_4) \sqcap \\
& \exists_{\geq 1}\, \mathsf{R} \,.\, (\mathsf{C}_5 \sqcap \neg \mathsf{P}_1 \sqcap \mathsf{P}_2 \sqcap \neg \mathsf{P}_3 \sqcap \neg \mathsf{P}_4) \sqcap \exists_{\geq 2}\, \mathsf{R} \,.\, (\mathsf{C}_5 \sqcap \neg \mathsf{P}_1 \sqcap \neg \mathsf{P}_2 \sqcap \neg \mathsf{P}_4) \sqcap \\
& \exists_{\geq 3}\, \mathsf{R} \,.\, (\mathsf{C}_6 \sqcap \mathsf{P}_1) \sqcap \exists_{\geq 2}\, \mathsf{R} \,.\, (\mathsf{C}_7 \sqcap \neg \mathsf{P}_1)
\end{aligned}
$$

Then, the concept term $\mathsf{D} \sqcap \exists_{\leq 7}\, \mathsf{R} \,.\, \mathsf{C}_1 \sqcap \exists_{\leq 7}\, \mathsf{R} \,.\, \mathsf{C}_2 \sqcap \exists_{\leq 7}\, \mathsf{R} \,.\, (\mathsf{C}_1 \sqcup \mathsf{C}_2)$ is *satisfiable* while the concept term $\mathsf{D} \sqcap (\exists_{\leq 6}\, \mathsf{R} \,.\, \mathsf{C}_1 \sqcup \exists_{\leq 6}\, \mathsf{R} \,.\, \mathsf{C}_2 \sqcup \exists_{\leq 6}\, \mathsf{R} \,.\, (\mathsf{C}_1 \sqcup \mathsf{C}_2))$ is *not satisfiable*. Even with advances such as the signature calculus, RACER (and other comparable DL systems) cannot compute the (un)satisfiability of either concept e.g.,term within a reasonable amount of time (e.g., ≤ 100 seconds). Note that the numbers used in the example are all rather small.

In [20] mathematical programming and atomic decomposition is presented as the basic TBox inference technique for a large class of modal and description logics. The proposed techniques seem to be well suited for dealing with qualified number restrictions. However, the approach in [20] is not based on a tableaux calculus and does not investigate blocking strategies as required by the description logic $\mathcal{ALCQH}_{R^+}$. In this paper we report on the integration of an algebraic reasoner into the tableaux-based DL reasoner RACER. Due to space restrictions, we focus on TBox reasoning.

2 Dealing with Qualified Number Restrictions

In the following we present a tableaux algorithm which decides the satisfiability of $\mathcal{ALCQH}_{R^+}$ concepts. It is based on a hybrid architecture combining a tableaux calculus with a reasoner about sets of linear inequations. The calculus is inspired by [20] and [9]. The calculus uses internal data structures, which are sometimes called "constraints" (e.g., [3]). In other contexts, in particular if the DL supports so-called ABoxes, the data structures are also called "ABox assertions" (see e.g., [6]).

2.1 A Tableaux Calculus for $\mathcal{ALCQH}_{R^+}$

First, we introduce ABox assertions used as input for the tableaux algorithm. Let C be a concept term, R be a role name, O be the set of individual names (disjoint from the set of concepts names and the set of role names), $\mathsf{a}, \mathsf{b} \in O$ be individual names, and $x \notin O$, then the following expression are ABox assertions: (1) $\mathsf{a} : \mathsf{C}$ (*instance assertion*), (2) $(\mathsf{a}, \mathsf{b}) : \mathsf{R}$ (*role assertion*), and (3) $\forall\, x \,.\, (x : \mathsf{C})$ (*universal concept assertion*). An interpretation $\mathcal{I}$ satisfies an assertional axiom $\mathsf{a} : \mathsf{C}$ iff $\mathsf{a}^{\mathcal{I}} \in \mathsf{C}^{\mathcal{I}}$, $(\mathsf{a}, \mathsf{b}) : \mathsf{R}$ iff $(\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}}) \in \mathsf{R}^{\mathcal{I}}$, and $\forall\, x \,.\, (x : \mathsf{C})$ iff $\mathsf{C}^{\mathcal{I}} = \Delta^{\mathcal{I}}$. An

ABox $\mathcal{A}$ is *satisfiable* iff there exists an interpretation $\mathcal{I}$ which satisfies all assertions in $\mathcal{A}$ and all axioms in $\mathcal{T}$ w.r.t. $\mathcal{R}$.

Given a TBox $\mathcal{T}$, and a role hierarchy $\mathcal{R}$, a concept D is satisfiable iff the ABox $\mathcal{A} = \{\mathsf{a}:\mathsf{D}\}$ extended in accordance with the following rules is satisfiable. 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 usual negation normal form. Every concept of the form $\exists\,\mathsf{R}\,.\,\mathsf{C}$ occurring in $\mathcal{A}$ is replaced by $(\exists_{\geq 1}\,\mathsf{R}' \sqcap \forall\,\mathsf{R}'\,.\,\mathsf{C})$ and every $\exists_{\geq n}\,\mathsf{R}\,.\,\mathsf{C}$ by $(\exists_{\geq n}\,\mathsf{R}' \sqcap \forall\,\mathsf{R}'\,.\,\mathsf{C})$, with $\mathsf{R}' \in R$ fresh in $\mathcal{A}$ and $\mathsf{R}' \sqsubseteq \mathsf{R}$ added to $\mathcal{R}$.

$\mathcal{ALCQH}_{R^+}$ supports transitive roles and GCIs. In order to guarantee the termination of the tableaux calculus, the notion of *blocking* an individual for the applicability of tableau rules is introduced as follows. Given an ABox $\mathcal{A}$ and an individual a occurring in $\mathcal{A}$, we define the *concept set* of a as $\sigma(\mathcal{A},\mathsf{a}) := \{\top\} \cup \{\mathsf{C} \mid \mathsf{a}:\mathsf{C} \in \mathcal{A}\}$. We define an *individual ordering* '$\prec$' for individuals (elements of O) occurring in an ABox $\mathcal{A}$. If $\mathsf{b} \in O$ is introduced into $\mathcal{A}$, then $\mathsf{a} \prec \mathsf{b}$ for all individuals a already present in $\mathcal{A}$. Let $\mathcal{A}$ be an ABox and $\mathsf{a}, \mathsf{b} \in O$ be individuals in $\mathcal{A}$. We call a a blocking individual of b if all of the following conditions hold: (1) $\sigma(\mathcal{A},\mathsf{a}) \supseteq \sigma(\mathcal{A},\mathsf{b})$, (2) $\mathsf{a} \prec \mathsf{b}$. If there exists a blocking individual a for b, then b is said to be *blocked* (by a).

Given an ABox $\mathcal{A}$, $\sharp(\mathsf{a},\mathsf{R})_{\mathcal{A}}$ defines the number of potential R-successors for an individual a mentioned in $\mathcal{A}$.

$$\sharp(\mathsf{a},\mathsf{R})_{\mathcal{A}} = \sum_{\alpha \in \mathcal{A}} \mathrm{count}(\mathsf{a},\mathsf{R},\alpha)_{\mathcal{A}},\ \ \mathrm{count}(\mathsf{a},\mathsf{R},\alpha)_{\mathcal{A}} = \begin{cases} n & \text{if } \alpha = \mathsf{a}:\exists_{\geq n}\,\mathsf{R}',\ \mathsf{R}' \in \mathsf{R}^{\downarrow} \\ 0 & \text{otherwise} \end{cases}$$

Given an ABox $\mathcal{A}$, $\min(\mathsf{a},\mathsf{R})_{\mathcal{A}}$ defines the minimal number of required and $\max(\mathsf{a},\mathsf{R},\mathsf{C})_{\mathcal{A}}$ the maximal number of allowed R-successors for an individual a mentioned in $\mathcal{A}$ (whose R-successors satisfy C).

$$\begin{aligned} \min(\mathsf{a},\mathsf{R})_{\mathcal{A}} &= \max(\{0\} \cup \{n \mid \mathsf{a}:\exists_{\geq n}\,\mathsf{S} \in \mathcal{A},\ \mathsf{S} \in \mathsf{R}^{\downarrow}\} \\ \max(\mathsf{a},\mathsf{R},\mathsf{C})_{\mathcal{A}} &= \min(\{\infty\} \cup \{n \mid \mathsf{a}:\exists_{\leq n}\,\mathsf{S}\,.\,\mathsf{C} \in \mathcal{A},\ \mathsf{S} \in \mathsf{R}^{\uparrow}\}) \end{aligned}$$

We are now ready to define the *completion rules* that are intended to generate a so-called completion (see also below) of an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$.

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

R$\sqcup$ The disjunction rule.
  if $\mathsf{a}:\mathsf{C} \sqcup \mathsf{D} \in \mathcal{A}$, and $\{\mathsf{a}:\mathsf{C},\ \mathsf{a}:\mathsf{D}\} \cap \mathcal{A} = \emptyset$
  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 $\mathsf{a}:\forall\,\mathsf{R}\,.\,\mathsf{C} \in \mathcal{A}$, and $\exists\,\mathsf{b} \in O, \mathsf{S} \in \mathsf{R}^{\downarrow} : (\mathsf{a},\mathsf{b}):\mathsf{S} \in \mathcal{A}$, and $\mathsf{b}:\mathsf{C} \notin \mathcal{A}$
  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 $\exists\,\mathsf{b} \in O$, $\mathsf{T} \in \mathsf{R}^{\downarrow}$, $\mathsf{T} \in T$, $\mathsf{S} \in \mathsf{T}^{\downarrow} : (\mathsf{a},\mathsf{b}):\mathsf{S} \in \mathcal{A}$, and
  2. $\mathsf{b}:\forall\,\mathsf{T}\,.\,\mathsf{C} \notin \mathcal{A}$
  then $\mathcal{A}' = \mathcal{A} \cup \{\mathsf{b}:\forall\,\mathsf{T}\,.\,\mathsf{C}\}$

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

R$\exists_{\geq n}$ The number restriction exists rule.

if 1. $\mathsf{a}:\exists_{\geq n}\,\mathsf{R} \in \mathcal{A}$, and a is not blocked, and

2. $\neg\exists\,\mathsf{b_1},\ldots,\mathsf{b_n} \in O,\ \mathsf{S_1},\ldots,\mathsf{S_n} \in \mathsf{R}^{\downarrow} : \{(\mathsf{a},\mathsf{b_k}):\mathsf{S_k} \mid \mathsf{k} \in 1..\mathsf{n}\} \subseteq \mathcal{A}$

then $\mathcal{A}' = \mathcal{A} \cup \{(\mathsf{a},\mathsf{b_k}):\mathsf{R} \mid \mathsf{k} \in 1..\mathsf{n}\}$ where $\mathsf{b_1},\ldots,\mathsf{b_n} \in O$ are not used in $\mathcal{A}$

Merge The qualified number restriction merge rule.

if 1. $\exists\,\mathsf{a}, \mathsf{C}$ mentioned in $\mathcal{A} : \sharp(\mathsf{a},\mathsf{R})_{\mathcal{A}} > \max(\mathsf{a},\mathsf{R},\mathsf{C})_{\mathcal{A}}$, and

2. $\hat{R} = \{\mathsf{R}' \in P \mid \mathsf{a}:\exists_{\leq m}\,\mathsf{R}'\,.\,\mathsf{D} \in \mathcal{A}\}$, and

3. $\mathcal{M}^{\mathsf{R}}_{\geq} = \{\mathsf{a}:\exists_{\geq n}\,\mathsf{S} \in \mathcal{A} \mid \mathsf{S} \in \mathsf{R}'^{\downarrow},\ \mathsf{R}' \in \hat{R}\}$, $\mathcal{M}^{\mathsf{R}}_{\leq} = \{\mathsf{a}:\exists_{\leq m}\,\mathsf{S}\,.\,\mathsf{D} \in \mathcal{A} \mid \mathsf{S} \in \hat{R}\}$

then $\langle \mathrm{SAT}, M, \mathcal{R}'\rangle \leftarrow \mathrm{ISAT}(\mathcal{A},\mathcal{R},\mathcal{M}^{\mathsf{R}}_{\geq},\mathcal{M}^{\mathsf{R}}_{\leq})$

if SAT

then $\mathcal{A}' = (\mathcal{A} \setminus \mathcal{M}^{\mathsf{R}}_{\geq}) \cup M$, $\mathcal{R} = \mathcal{R}'$

else $\mathcal{A}' = \mathcal{A} \cup \{\mathsf{a}:\bot\}$

The qualified number restriction merge rule needs some explanation, it is fairly non-standard. Intuitively speaking, the idea of the new merge rule can be summarized as follows. It is invoked whenever there exists an individual with potential successors for a role R such that the number of these successors violates an at-most restriction for an ancestor role of R. If this is the case, the rule calls the procedure ISAT (see Section 2.2) with $\mathcal{A}$, $\mathcal{R}$, and the set $\mathcal{M}^{\mathsf{R}}_{\geq}$ of at-least and the set $\mathcal{M}^{\mathsf{R}}_{\leq}$ of at-most assertions. If the inequations derived from both sets are satisfiable, ISAT returns an extended role hierarchy $\mathcal{R}'$ and a set M of new assertions such that these assertions satisfy all restrictions from $\mathcal{M}^{\mathsf{R}}_{\geq}$ and $\mathcal{M}^{\mathsf{R}}_{\leq}$ and, thus, the assertions from M replace the ones from $\mathcal{M}^{\mathsf{R}}_{\geq}$. If the inequations are unsatisfiable, the ABox $\mathcal{A}'$ is marked as contradictory (see below).

Given an ABox $\mathcal{A}$, more than one rule might be applicable to $\mathcal{A}$. The order is determined by the *completion strategy* which is defined as follows. A *meta rule* controls the priority between individuals: Apply a tableaux rule to an individual $\mathsf{b} \in O$ only if no rule is applicable to another individual $\mathsf{c} \in O$ such that $\mathsf{c} \prec \mathsf{b}$. The completion rules are always applied in the following order: (1) All non-generating rules (R$\sqcap$, R$\sqcup$, R$\forall$C, R$\forall_+$C, R$\forall_x$); (2) Qualified number restriction merge rule; (3) Number restriction exists rule (R$\exists_{\geq n}$). In the following we always assume that the completion strategy is observed. It ensures that rules are applied to individuals w.r.t. the ordering '$\prec$' and the number restriction exists rule is only applied to individuals if the qualified number restriction merge rule is not applicable.

We assume the same naming conventions as used above. An ABox $\mathcal{A}$ is called *contradictory* if the following *clash trigger* is applicable: $\mathsf{a}:\bot \in \mathcal{A}$ or $\{\mathsf{a}:\mathsf{A}, \mathsf{a}:\neg\mathsf{A}\} \subseteq \mathcal{A}$, where A is a concept name. If the clash trigger is not applicable to $\mathcal{A}$, then $\mathcal{A}$ is called *clash-free*. Any ABox containing a clash is obviously unsatisfiable. 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 called a *completion* of $\mathcal{A}$. The purpose of the calculus is to generate a completion for an ABox $\mathcal{A}$ in order to prove the satisfiability of $\mathcal{A}$. For a given ABox $\mathcal{A}$ and a role hierarchy $\mathcal{R}$, the calculus can be invoked by $\mathsf{ASAT}(\mathcal{A},\mathcal{R})$. The calculus applies the completion rules as given above. It stops the application of rules, if a clash occurs. The calculus answers "*true*" if a completion can be derived, and "*false*" otherwise.

2.2 The Algebraic Reasoner

This section introduces an algebraic reasoner which is integrated into our tableaux calculus for $\mathcal{ALCQH}_{R^+}$. The reasoner decides the satisfiability of a set $\mathcal{M}$ of assertions (of the form $\mathsf{a}:\exists_{\geq n}\,\mathsf{R}$ or $\mathsf{a}:\exists_{\leq n}\,\mathsf{R}\,.\,\mathsf{D}$) w.r.t. to a given ABox $\mathcal{A}$ and a role hierarchy $\mathcal{R}$ (and a TBox $\mathcal{T}$).

The reasoner has been inspired by [20]. In contrast to [20] our approach supports cyclic terminologies, general concept inclusions, and transitive roles. The task of the algebraic reasoner can be divided into two parts. The first step derives a set $\mathcal{S}$ of linear inequations from $\mathcal{M}$, $\mathcal{A}$, and $\mathcal{R}$ using the ASAT test. The second step decides the satisfiability of $\mathcal{S}$ with the help of a Simplex procedure which allows only solutions in $\mathbb{N}$ [4]. The algebraic reasoner can be called via $\mathsf{ISAT}(\mathcal{A},\mathcal{R},\mathcal{M}^{\mathsf{R}}_{\leq},\mathcal{M}^{\mathsf{R}}_{\geq})$ where $\mathcal{A}$ is an ABox, $\mathcal{R}$ a role hierarchy, and $\mathcal{M}^{\mathsf{R}}_{\leq}$ ($\mathcal{M}^{\mathsf{R}}_{\geq}$) contains assertions of the form $\mathsf{a} : \exists_{\leq n} \mathsf{R} . \mathsf{D}$ ($\mathsf{a} : \exists_{\geq n} \mathsf{R}$) with $\mathcal{M} = \mathcal{M}^{\mathsf{R}}_{\leq} \cup \mathcal{M}^{\mathsf{R}}_{\geq} \subseteq \mathcal{A}$.

2.2.1 Derivation of Inequations

The derivation of inequations is based on a partitioning of sets of potential role successors whose cardinalities play the role of variables in the inequations. The partitioning into disjoint subsets is necessary since the variables (i.e., the cardinalities) in the inequations have to be independent from one another (see [20] for a discussion). This can be achieved by (1) determining the union of all sets of potential role successors which are referred to in $\mathcal{M}$ and (2) partitioning this union into pairwise disjoint subsets such that every potential role successor set can be represented as the union of these partitions.

For the partitioning of potential role successors we need a few definitions. Let us assume that $\mathsf{R}^{\mathsf{a}} = \{b \in \Delta^{\mathcal{I}} \mid (\mathsf{a}^{\mathcal{I}}, b) \in \mathsf{R}^{\mathcal{I}}\}$ denotes the *potential* R-*successors* of an individual a and $\overline{\mathsf{R}^{\mathsf{a}}} = \Delta^{\mathcal{I}} \setminus \mathsf{R}^{\mathsf{a}}$ the complement of R^{a}. Let $RS = \{\mathsf{R}_1, \ldots, \mathsf{R}_\mathsf{n}\} \subseteq R$, then $RS^{\mathsf{a}} = \{\mathsf{x}_1 \cap \cdots \cap \mathsf{x}_\mathsf{n} \mid \mathsf{x}_\mathsf{i} \in \{\mathsf{R}^{\mathsf{a}}_\mathsf{i}, \overline{\mathsf{R}^{\mathsf{a}}_\mathsf{i}}\},\ \mathsf{R}_\mathsf{i} \in RS,\ \mathsf{i} \in 1..\mathsf{n}\}$ defines the set of all possible *partitions* of potential role successors w.r.t a and RS. For every role in RS either R^{a} or $\overline{\mathsf{R}^{\mathsf{a}}}$ is part of an intersection in RS^{a}. Without loss of generality we can make the following assumptions: (1) The partition $\overline{\mathsf{R}^{\mathsf{a}}_1} \cap \cdots \cap \overline{\mathsf{R}^{\mathsf{a}}_\mathsf{n}}$ can be ignored since it is not needed in the following; (2) The emptiness of a partition due to assertions for a can be determined by only considering the non-negated parts of the partition since restrictions for a set $\overline{\mathsf{R}^{\mathsf{a}}}$ of potential role fillers cannot be expressed; (3) Role hierarchies are observed in intersections because $\overline{\mathsf{R}^{\mathsf{a}}} \subseteq \overline{\mathsf{S}^{\mathsf{a}}} \Leftrightarrow \mathsf{S}^{\mathsf{a}} \subseteq \mathsf{R}^{\mathsf{a}}$. A canonical name for partitions is needed. Let us assume that an ordering '$\lessdot$' is defined on all elements of R. The canonical name[3] for a partition $\mathsf{x}_1 \cap \cdots \cap \mathsf{x}_\mathsf{n} \in RS^{\mathsf{a}}$ is defined as $\mathsf{R}^{\mathsf{a}}_1 \ldots \mathsf{R}^{\mathsf{a}}_\mathsf{m}$ if $\{\mathsf{R}_1, \ldots, \mathsf{R}_\mathsf{m}\}$ (with $\mathsf{m} \leq \mathsf{n}$) is the set of role names mentioned in $\mathsf{x}_1 \cap \cdots \cap \mathsf{x}_\mathsf{n}$ and $\mathsf{R}_1 \lessdot \cdots \lessdot \mathsf{R}_\mathsf{m}$. By analogy, $\mathsf{r}_1 \ldots \mathsf{r}_\mathsf{m}$ is the canonical name for a variable denoting the cardinality of $\mathsf{x}_1 \cap \cdots \cap \mathsf{x}_\mathsf{n}$. For instance, the canonical name of $\mathsf{R}^{\mathsf{a}}_1 \cap \mathsf{R}^{\mathsf{a}}_2 \cap \overline{\mathsf{R}^{\mathsf{a}}_3}$ is $\mathsf{R}_1\mathsf{R}_2$ and the cardinality of $\mathsf{R}^{\mathsf{a}}_1 \cap \mathsf{R}^{\mathsf{a}}_2 \cap \overline{\mathsf{R}^{\mathsf{a}}_3}$ is represented as $\mathsf{r}_1\mathsf{r}_2$, if $RS = \{\mathsf{R}_1, \mathsf{R}_2, \mathsf{R}_3\}$.

If $\mathcal{M}^{\mathsf{R}}_{\leq} \subseteq \mathcal{M} \subseteq \mathcal{A}$ contains an assertion of the form $\mathsf{a} : \exists_{\leq m} \mathsf{R} . \mathsf{D}$, the algebraic reasoner internally considers it as the conjunction $\{\mathsf{a} : \exists_{\leq m} \mathsf{R}', \mathsf{a} : \forall \mathsf{R}' . \mathsf{D}, \mathsf{a} : \forall \widetilde{\mathsf{R}'} . \neg\mathsf{D}\}$ where $\mathsf{R}', \widetilde{\mathsf{R}'} \in R$ are fresh in $\mathcal{A}$, $\mathsf{R}' \sqsubseteq \mathsf{R}$ and $\widetilde{\mathsf{R}'} \sqsubseteq \mathsf{R}$ are added to $\mathcal{R}$, and $\widetilde{\mathsf{R}'}$ represents the complement of R' relative to R. The reasoner has to ensure that R^{a} is split into two disjoint subsets such that $\mathsf{R}^{\mathsf{a}} = \mathsf{R}'^{\mathsf{a}} \cup \widetilde{\mathsf{R}'^{\mathsf{a}}}$ with $\mathsf{R}'^{\mathsf{a}} \subseteq \mathsf{D}^{\mathcal{I}}$, $\widetilde{\mathsf{R}'^{\mathsf{a}}} \subseteq (\neg\mathsf{D})^{\mathcal{I}}$, and $\widetilde{\mathsf{R}'^{\mathsf{a}}} = \mathsf{R}^{\mathsf{a}} \setminus \mathsf{R}'^{\mathsf{a}}$. The split is performed for each element of $\mathcal{M}^{\mathsf{R}}_{\leq}$. The scheme for canonical names of partitions and cardinality variables is properly extended, e.g., the canonical name of $\mathsf{R}^{\mathsf{a}}_1 \cap \mathsf{R}^{\mathsf{a}}_2 \cap \widetilde{\mathsf{R}^{\mathsf{a}}_3}$ is $\mathsf{R}_1\mathsf{R}_2\widetilde{\mathsf{R}_3}$ and its cardinality is represented by $\mathsf{r}_1\mathsf{r}_2\widetilde{\mathsf{r}_3}$.

As mentioned above, a partition can be the empty set due to assertions in the ABox $\mathcal{A}$. For instance, if we assume $\mathsf{R}_1 \sqsubseteq \mathsf{R}$, $\mathsf{R}_2 \sqsubseteq \mathsf{R}$, and $\mathsf{a} : \forall \mathsf{R}_1 . \mathsf{A} \sqcap \forall \mathsf{R}_2 . \neg\mathsf{A} \sqcap \exists_{\geq 2} \mathsf{R}_1 \sqcap \exists_{\geq 2} \mathsf{R}_2 \sqcap \exists_{\leq 3} \mathsf{R}$, the partition $\mathsf{R}^{\mathsf{a}}_1 \cap \mathsf{R}^{\mathsf{a}}_2$ must be empty due to the clash between A and $\neg\mathsf{A}$. In our hybrid approach this is decided by the procedure $\mathsf{RS\_SAT}(\mathcal{A}, \mathcal{R}, \mathsf{a}, RC)$ where $\mathcal{A}$ is an ABox, $\mathcal{R}$ a

[3]The reference to a is omitted if the context is obvious, i.e., $\mathsf{R}_1 \ldots \mathsf{R}_\mathsf{m}$ is also used.

role hierarchy, a an individual, and RC a set of role names. Let $\mathsf{b} \in O$ be new in $\mathcal{A}$. The procedure $\mathsf{RS\_SAT}$ returns *true* if $\mathsf{ASAT}(\mathcal{A} \cup \{(\mathsf{a},\mathsf{b}) : \mathsf{S} \mid \mathsf{S} \in RC\}, \mathcal{R})$ returns *true*. Otherwise $\mathsf{RS\_SAT}$ returns *false*. The call of ASAT is necessary in order to correctly deal with ABoxes where the blocking technique (see Section 2.1) must be applied.

The set $\mathcal{S}$ of inequations is computed as follows. We assume that the algebraic reasoner has been called via $\mathsf{ISAT}(\mathcal{A},\mathcal{R},\mathcal{M}^{\mathsf{R}}_{\leq},\mathcal{M}^{\mathsf{R}}_{\geq})$. Let RS be the set of role names mentioned in $\mathcal{M}^{\mathsf{R}}_{\geq}$, $\mathcal{M}^{\mathsf{R}}_{\leq}$, $\mathcal{AA}$ contain the assertions resulting from the above-mentioned transformation, $\mathcal{R}'$ be the role hierarchy extended by the transformation, and AR contain the role names mentioned in $\mathcal{AA}$. We also assume the operator $roles(\mathsf{X})$ which computes for a partition X the role names involved in this intersection, e.g., $roles(\mathsf{R}^{\mathsf{a}}_1\mathsf{R}^{\mathsf{a}}_2\widetilde{\mathsf{R}^{\mathsf{a}}_3})$ returns $\{\mathsf{R}_1, \mathsf{R}_2, \widetilde{\mathsf{R}_3}\}$. Let $RS'^{\mathsf{a}} = (RS \cup AR)^{\mathsf{a}}$ be the partitioning w.r.t. a and $RS \cup AR$. For each $\mathsf{X} \in RS'^{\mathsf{a}}$ the test $\mathsf{RS\_SAT}(\mathcal{A}', \mathcal{R}', \mathsf{a}, \mathsf{C}, roles(\mathsf{X}))$ is performed with $\mathcal{A}' = (\mathcal{A} \setminus \mathcal{M}^{\mathsf{R}}_{\geq}) \cup \{\alpha \in \mathcal{AA} \mid \alpha = \mathsf{a} : \forall \mathsf{R} \,.\, \mathsf{C}\}$. If the test returns *false*, the inequation $x \leq 0$ is added to $\mathcal{S}$ with x the canonical name for X. Let $P_{\mathsf{R}} = \{\mathsf{X} \in RS'^{\mathsf{a}} \mid \mathsf{R} \in roles(\mathsf{X})\}$. For each $\mathsf{a} : \exists_{\geq n} \mathsf{R} \in \mathcal{M}^{\mathsf{R}}_{\geq}$, the inequation $\mathsf{x}_1 + \cdots + \mathsf{x}_k \geq \mathsf{n}$ is added to $\mathcal{S}$ with $\mathsf{x}_1, \ldots, \mathsf{x}_k$ the canonical names of all partitions from P_{R}. For each $\mathsf{a} : \exists_{\leq m} \mathsf{R} \in \mathcal{M}^{\mathsf{R}}_{\leq} \cup \mathcal{AA}$, the inequation $\mathsf{x}_1 + \cdots + \mathsf{x}_k \leq \mathsf{m}$ is added to $\mathcal{S}$ with $\mathsf{x}_1, \ldots, \mathsf{x}_k$ the canonical names of all partitions from P_{R}. Finally, for every variable x mentioned in $\mathcal{S}$, the inequation $\mathsf{x} \geq 0$ is added to $\mathcal{S}$.

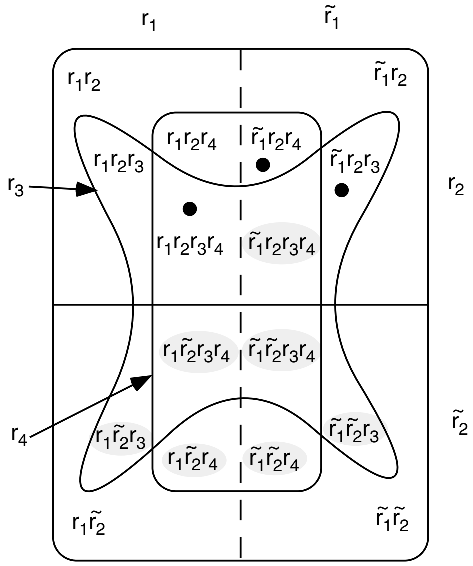
2.2.2 Satisfiability of Inequations

This step decides the satisfiability of $\mathcal{S}$ with the help of a Simplex procedure which allows only solutions in $\mathbb{N}$ [4]. It is implemented with sparse arrays as basic data structures for matrix representation. ISAT returns a set M of transformed assertions and a role hierarchy $\mathcal{R}'$ if $\mathcal{S}$ is satisfiable. The assertions from M are used by the tableaux calculus for replacing the assertions from $\mathcal{M}^{\mathsf{R}}_{\geq}$ in $\mathcal{A}$ such that no assertion from $\mathcal{M}^{\mathsf{R}}_{\geq}$ violates an assertion from $\mathcal{M}^{\mathsf{R}}_{\leq}$ anymore and $(\mathcal{A} \setminus \mathcal{M}^{\mathsf{R}}_{\geq}) \cup M$ is satisfiable iff $\mathcal{A}$ is satisfiable. The assertions in M are computed as follows. Let $Sol = \{\mathsf{x}_1 = \mathsf{l}_1, \ldots, \mathsf{x}_k = \mathsf{l}_k\}$ be the solution for $\mathcal{S}$ with $\mathsf{x}_1, \ldots, \mathsf{x}_k$ variable names mentioned in $\mathcal{S}$ and $\mathsf{l}_1, \ldots, \mathsf{l}_k \in \mathbb{N}$, $\mathsf{l}_i > 0$, $\mathsf{i} \in 1..\mathsf{k}$. For each equation $\mathsf{x} = \mathsf{l} \in Sol$ the assertion $\mathsf{a} : \exists_{\geq l} \mathsf{RC}$ is added to M, $\mathsf{RC} \sqsubseteq \mathsf{R}$ is added to $\mathcal{R}'$ for all $\mathsf{R} \in roles(\mathsf{x})$, and $\mathsf{S} \sqsubseteq \mathsf{RC}$ is added to $\mathcal{R}'$ for all $\mathsf{S} \neq \mathsf{RC}$ with $\mathsf{S} \sqsubseteq \mathsf{R} \in \mathcal{R}'$. Furthermore, all assertion in $\{\alpha \in \mathcal{AA} \mid \alpha = \mathsf{a} : \forall \mathsf{T} \,.\, \mathsf{C}\}$ are added to M.

2.2.3 Example

In order to give the reader a better intuition we illustrate the reasoning procedure with the assertions shown at the top of Figure 2b. Let us assume that $\mathsf{ASAT}(\mathcal{A}, \emptyset)$ with the ABox $\mathcal{A} = \{\mathsf{a} : \exists_{\leq 1} \mathsf{R} \,.\, \mathsf{B} \sqcap \mathsf{C},\ \mathsf{a} : \exists_{\leq 3} \mathsf{R} \,.\, \mathsf{A},\ \mathsf{a} : \exists_{\geq 2} \mathsf{R} \,.\, \mathsf{A} \sqcap \mathsf{B},\ \mathsf{a} : \exists_{\geq 2} \mathsf{R} \,.\, \mathsf{A} \sqcap \mathsf{C}\}$ is called. The only tableaux rule applicable to assertions in $\mathcal{A}$ is the qualified number restriction merge rule. This rule calls $\mathsf{ISAT}(\mathcal{A}, \mathcal{R}, \mathcal{M}^{\mathsf{R}}_{\leq}, \mathcal{M}^{\mathsf{R}}_{\geq})$ with the ABox $\mathcal{A}' = \{\mathsf{a} : \exists_{\leq 1} \mathsf{R} \,.\, \mathsf{B} \sqcap \mathsf{C},\ \mathsf{a} : \exists_{\leq 3} \mathsf{R} \,.\, \mathsf{A},\ \mathsf{a} : \exists_{\geq 2} \mathsf{R}_3,\ \mathsf{a} : \forall \mathsf{R}_3 \,.\, \mathsf{A} \sqcap \mathsf{B},\ \mathsf{a} : \exists_{\geq 2} \mathsf{R}_4,\ \mathsf{a} : \forall \mathsf{R}_4 \,.\, \mathsf{A} \sqcap \mathsf{C}\}$, $\mathcal{R} = \{\mathsf{R}_3 \sqsubseteq \mathsf{R},\ \mathsf{R}_4 \sqsubseteq \mathsf{R}\}$, $\mathcal{M}^{\mathsf{R}}_{\leq} = \{\mathsf{a} : \exists_{\leq 1} \mathsf{R} \,.\, \mathsf{B} \sqcap \mathsf{C},\ \mathsf{a} : \exists_{\leq 3} \mathsf{R} \,.\, \mathsf{A}\}$, and $\mathcal{M}^{\mathsf{R}}_{\geq} = \{\mathsf{a} : \exists_{\geq 2} \mathsf{R}_3,\ \mathsf{a} : \exists_{\geq 2} \mathsf{R}_4\}$. After applying the transformation rules from Section 2.1 and 2.2.1 we get the transformed assertions in Figure 2b. The partitioning RS^{a} with the set $RS = \{\mathsf{R}, \mathsf{R}_1, \widetilde{\mathsf{R}_1}, \mathsf{R}_2, \widetilde{\mathsf{R}_2}, \mathsf{R}_3, \mathsf{R}_4\}$ is shown as Venn diagram in Figure 2a.[4] Each region in the diagram is labeled by the canonical variable name representing the cardinality of the associated partition. The enclosing rounded rectangle denotes the set R^{a} which is split horizontally *and* vertically in two halves. The vertical (labeled r_1, $\widetilde{\mathsf{r}_1}$) and the horizontal

[4]Without loss of generality R may be omitted in the partitioning of RS^{a}.



(a) Venn diagram of the partitioning of R^a (regions are labeled by variables).

Original assertions (satisfiable):
$a : \exists_{\leq 1} R . B \sqcap C$, $a : \exists_{\leq 3} R . A$,
$a : \exists_{\geq 2} R . A \sqcap B$, $a : \exists_{\geq 2} R . A \sqcap C$

Transformed assertions ($\forall_{i \in 1..4} R_i \sqsubseteq R \in \mathcal{R}$):
$a : \exists_{\leq 1} R_1$, $a : \forall R_1 . B \sqcap C$, $a : \forall \widetilde{R_1} . \neg(B \sqcap C)$,
$a : \exists_{\leq 3} R_2$, $a : \forall R_2 . A$, $a : \forall \widetilde{R_2} . \neg A$,
$a : \exists_{\geq 2} R_3$, $a : \forall R_3 . A \sqcap B$, $a : \exists_{\geq 2} R_4$, $a : \forall R_4 . A \sqcap C$

Contents of $\mathcal{S}$ ($\forall x$ mentioned in $\mathcal{S}$: $x \geq 0$):
$r_1r_2 + r_1\tilde{r}_2 + r_1r_2r_3 + r_1r_2r_4 + r_1r_2r_3r_4 \leq 1$,
$r_1r_2 + \tilde{r}_1r_2 + r_1r_2r_3 + r_1r_2r_4 + \tilde{r}_1r_2r_3 + \tilde{r}_1r_2r_4 + r_1r_2r_3r_4 \leq 3$,
$r_1r_2r_3 + \tilde{r}_1r_2r_3 + r_1r_2r_3r_4 \geq 2$,
$r_1r_2r_4 + \tilde{r}_1r_2r_4 + r_1r_2r_3r_4 \geq 2$,
$r_1\tilde{r}_2r_3 \leq 0$, $r_1\tilde{r}_2r_4 \leq 0$, $\tilde{r}_1\tilde{r}_2r_3 \leq 0$, $\tilde{r}_1\tilde{r}_2r_4 \leq 0$,
$r_1\tilde{r}_2r_3r_4 \leq 0$, $\tilde{r}_1r_2r_3r_4 \leq 0$, $\tilde{r}_1\tilde{r}_2r_3r_4 \leq 0$

(b) Original/Transformed concept and inequations derived from partitioning of left subfigure, number restrictions, and RS_SAT tests.

Figure 2: Partitioning of R^a and derived inequations.

halves (labeled r_2, $\tilde{r}_2$) together form four mutually disjoint quadrants. The star-like shape (labeled r_3) denotes R^a_3, while the smaller rounded rectangle (labeled r_4) denotes R^a_4.

The first step of ISAT derives the set $\mathcal{S}$ of inequations (see Figure 2b) imposed by the sets $\mathcal{M}^R_{\leq}$, $\mathcal{M}^R_{\geq}$, and the results of the RS_SAT tests. The RS_SAT tests for seven subsets (marked in Figure 2a by a gray oval background) return *false*, i.e., these sets must be empty due to assertions in $\mathcal{A}$. A side condition for the Simplex procedure, which is not stated in $\mathcal{S}$ for sake of brevity, requires for each variable x mentioned in $\mathcal{S}$ that $x \geq 0$. The first four inequations are a direct translation from the transformed concept (in the same order) on the basis of the partitioning displayed in Figure 2a. The remaining seven inequations reflect the results of the RS_SAT tests. For better readability, the empty partitions have been omitted in the first four inequations. The set $\mathcal{S}$ of inequations is satisfiable by setting $r_1r_2r_3r_4 = 1$, $\tilde{r}_1r_2r_3 = 1$, $\tilde{r}_1r_2r_4 = 1$. All other variables mentioned in $\mathcal{S}$ are set to zero. The non-empty partitions are marked by black dots in Figure 2a. ISAT returns the set $M = \{a : \exists_{\geq 1} RC_1$, $a : \exists_{\geq 1} RC_2$, $a : \exists_{\geq 1} RC_3$, $a : \forall R_1 . (B \sqcap C)$, $a : \forall \widetilde{R_1} . \neg(B \sqcap C)$, $a : \forall R_2 . A$, $a : \forall \widetilde{R_2} . \neg A\}$ and $\mathcal{R}' = \{RC_1 \sqsubseteq R_i \mid i \in 1..4\} \cup \{RC_2 \sqsubseteq \widetilde{R_1}, RC_2 \sqsubseteq R_2, RC_2 \sqsubseteq R_4\} \cup \{RC_3 \sqsubseteq \widetilde{R_1}, RC_3 \sqsubseteq R_2, RC_3 \sqsubseteq R_3\}$.

The example concept becomes unsatisfiable if it is slightly changed to $\exists_{\geq 2} R . (A \sqcap B) \sqcap \exists_{\geq 2} R . (A \sqcap C) \sqcap \exists_{\leq 1} R . (B \sqcap C) \sqcap \exists_{\leq 2} R . A$. The partitioning and the results of the RS_SAT are still valid for this concept but the inequations are now unsatisfiable due to $r_1r_2 + \tilde{r}_1r_2 + r_1r_2r_3 + r_1r_2r_4 + \tilde{r}_1r_2r_3 + \tilde{r}_1r_2r_4 + r_1r_2r_3r_4 \leq 2$. This inequation can be rewritten as $r_1r_2 + \tilde{r}_1r_2 + r_1r_2r_4 + \tilde{r}_1r_2r_4 \leq 0$ due to $r_1r_2r_3 + \tilde{r}_1r_2r_3 + r_1r_2r_3r_4 \geq 2$ and it follows that $r_1r_2r_4 = 0$ and $\tilde{r}_1r_2r_4 = 0$. Together with $r_1r_2r_4 + \tilde{r}_1r_2r_4 + r_1r_2r_3r_4 \geq 2$ this implies $r_1r_2r_3r_4 \geq 2$ which contradicts the inequation $r_1r_2 + r_1\tilde{r}_2 + r_1r_2r_3 + r_1r_2r_4 + r_1r_2r_3r_4 \leq 1$.

2.3 Proof Sketch for the Hybrid Approach

Due to lack of space we give only a short sketch for the proof of our hybrid approach. In general, we already know that the concept satisfiability problem for the logic $\mathcal{ALCQH}_{R^+}$ is decidable since it is a subset of $\mathcal{SHIQ}$ [16]. Thus, it remains to show that our particular set of tableau rules together with the algebraic reasoner still decide the concept satisfiability problem. In [20] a proof is given for a particular class of description logics. On the one hand, we already mentioned that the approach from [20] is not applicable to the logic $\mathcal{ALCQH}_{R^+}$ due to the expressivity of $\mathcal{ALCQH}_{R^+}$. On the other hand, we argue that the proof in [20] is applicable to our approach to some extent provided the possibly recursive call of ASAT (via ISAT) for testing the emptiness of potential role successor sets is sound, complete, and terminating.

The termination problem depends on the structure of the tableau rules and the completion strategy. All rules except the SigMerge rule add new assertions to $\mathcal{A}$. Possible terminological cycles and transitive roles are properly dealt with by standard rules and the blocking technique employed. The SigMerge rule calls the ISAT procedure, if an at-most restriction for a role R is possibly violated. It is important to note that all "related" at-least and at-most restrictions are handled together by the ISAT procedure. The critical point is the recursive call of ASAT with a modified ABox and an possibly extended role hierarchy in order to test the emptiness of a potential role successor set. We argue that the at-least restrictions from $\mathcal{M}^{\mathsf{R}}_{\geq}$ may be removed from $\mathcal{A}$ for such a test since they cannot influence the result of an ASAT test. This is caused by the property of the $\mathcal{ALCQH}_{R^+}$ calculus that a role successor for an individual a cannot add assertions for a to $\mathcal{A}$. The omission also prevents possible cycles since the ISAT procedure cannot be called again with the same sets of at-least and at-most restrictions. If the ISAT procedure returns true, the SigMerge rule may replace the original at-least restrictions by the new set M since the new assertions do not violate any at-most restriction and their role successors are proven to be satisfiable.

The soundness problem can be addressed in a similar manner. If the tableaux calculus terminates and returns *true*, it is rather straightforward to construct a canonical interpretation from the completion of $\mathcal{A}$ that satisfies $\mathcal{A}$, $\mathcal{R}$, and $\mathcal{T}$. We conjecture that proof for $\mathcal{ALCNH}_{R^+}$ in [6] can be easily adapted to $\mathcal{ALCQH}_{R^+}$. The completeness problem can be handled in a similar way.

3 Evaluation

In order to indicate the advancement of this new architecture, we compared the performance of the hybrid architecture against settings where a standard tableaux calculus were used. A set of four benchmark problems were generated. The increased difficulty of the problems is caused by exponentially increasing the size of numbers used in at-least and at-most concepts. Each of the four problems exists in two variants (a 'test concept' is satisfiable vs. unsatisfiable). A problem basically employs concept terms of the form $\exists_{\leq n}\,\mathsf{R} \sqcap \exists_{\geq m_1}\,\mathsf{R_1} \sqcap \exists_{\geq m_2}\,\mathsf{R_2} \sqcap \exists_{\geq m_3}\,\mathsf{R_3} \sqcap \forall\,\mathsf{R_2}\,.\,\mathsf{C} \sqcap \forall\,\mathsf{R_3}\,.\,\neg\mathsf{C}$ with $\mathsf{R_i} \sqsubseteq \mathsf{R}, \mathsf{i} \in 1..3$. The (un)satisfiability of these terms has to be proven. A term is made satisfiable by choosing values for n, m_i such that $\max(m_1, m_2 + m_3) \leq n$ or unsatisfiable if $\max(m_1, m_2 + m_3) > n$.

Figure 3 demonstrates the result of this benchmark w/out algebraic reasoning. The performance gain in Figure 3 is dramatic since these problems can now be solved in constant time (usually below 0.02 seconds). The speed enhancement also scales up for problems with *qualified* number restrictions. The type of "intractable" problems from the introduction can

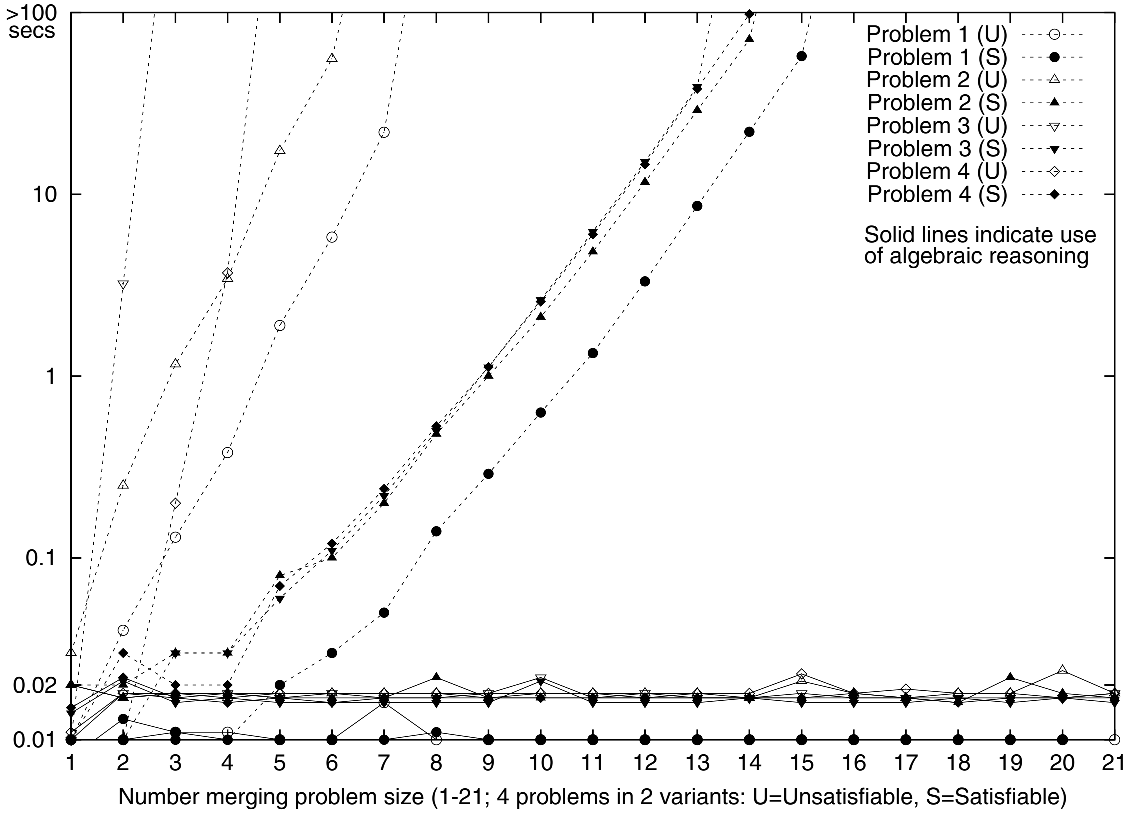


Figure 3: RACER: benchmark problems w/out algebraic reasoning.

now be handled as well. Using the algebraic reasoner they can be solved below 0.1 seconds. On the basis of this concept we also generated a set of benchmarks where the numbers occurring in the number restrictions are increased in the same way as for the benchmarks from Figure 3. There is no need to display the results in detail since the algebraic reasoner can solve these problems in constant time (below 0.4 seconds) but standard tableau algorithms cannot even solve the problem with the lowest difficulty within 100 seconds.

However, there exist problems such that the number of required role successor satisfiability tests, and, in turn, the number of variables required for the Simplex procedure, might increase exponentially in the worst case. This can be illustrated with concepts of the form $\exists\, \mathsf{R} \,.\, \mathsf{C_1} \sqcap \ldots \sqcap \exists\, \mathsf{R} \,.\, \mathsf{C_n} \sqcap \exists_{\leq m}\, \mathsf{R}$, $m < n$. The algebraic reasoner has to consider $\mathcal{O}(2^n)$ variables for the Simplex procedure and $\mathcal{O}(2^n)$ role successor satisfiability tests in the worst case. A similar situation might arise if more and more qualified at-most restrictions are added to a concept satisfiability test since each restriction increases the number of required role successor satisfiability tests by a factor of two (see also the example from Section 2.2.3).

One might argue that this type of optimization is not relevant for actual applications since example knowledge bases rarely contain number restrictions with numbers greater than one. However, this does not hold for technical application domains such as the configuration of technical devices, e.g., see the work described in [18]. In this context number restrictions with values greater than one are quite natural ("an engine with at least 4 valves", a "hub with at least 16 connections", etc).

Another argument in favor of this optimization technique is the support of simple reasoning about integers. Many knowledge bases developed for the CLASSIC system make use of min/max reasoning about integer intervals (e.g., "the price is between 200 and 300 dollars"). This can easily be mapped to a concept description $\exists\, \mathsf{price} \,.\, (\exists_{\geq 200}\, \mathsf{has\_value} \sqcap \exists_{\leq 300}\, \mathsf{has\_value})$ where has_value is a feature used to represent the restrictions on the price interval. Of course, restrictions on integers or reals can be more adequately modeled with so-called concrete domains which are supported by RACER (version 1.6, see [11, 7] for more details).

4 Conclusion and Outlook

In this paper we have presented a hybrid architecture for efficiently dealing with number restrictions in combination with role hierarchies and/or qualified number restrictions in the DL $\mathcal{ALCQH}_{R^+}$. The architecture has been implemented for concept satisfiability tests and evaluated in the ABox description logic system RACER. In contrast to [20] our approach is integrated into a tableaux calculus and can deal with GCIs, transitive roles, and cyclic terminologies. We are currently investigating how to extend this approach to ABox reasoning for $\mathcal{ALCQH}_{R^+}$.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[2] Jeen Broekstra, Michel Klein, Dieter Fensel, and Ian Horrocks. Adding formal semantics to the Web: building on top of RDF Schema. In *Proc. of the ECDL 2000 Workshop on the Semantic Web*, 2000.

[3] M. Buchheit, F.M. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. *Journal of Artificial Intelligence Research*, 1:109–138, 1993.

[4] R.E. Gomory. An algorithm for integer solutions to linear programs. In R.L. Graves and P. Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 269–302. McGraw-Hill, New York, 1963.

[5] R. Goré, A. Leitsch, and T. Nipkow, editors. *Proceedings of the International Joint Conference on Automated Reasoning, IJCAR'2001, June 18-23, 2001, Siena, Italy*, LNCS. Springer-Verlag, Berlin, June 2001.

[6] V. Haarslev and R. Möller. Expressive ABox reasoning with number restrictions, role hierarchies, and transitively closed roles. In A.G. Cohn, F. Giunchiglia, and B. Selman, editors, *Proceedings of Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2000), Breckenridge, Colorado, USA, April 11-15, 2000*, pages 273–284, April 2000.

[7] V. Haarslev and R. Möller. Description of the RACER system. In *Proceedings of the Workshop on Methods for Modalities 2 (M4M-2), University of Amsterdam, November 29-30, 2001*, November 2001.

[8] 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, August 4-10, 2001, Seattle, Washington, USA, pages 161–166, August 2001.

[9] V. Haarslev and R. Möller. Optimizing reasoning in description logics with qualified number restriction. In McGuinness and Patel-Schneider [19], pages 142–151.

[10] V. Haarslev and R. Möller. RACER system description. In Goré et al. [5], pages 701–705.

[11] V. Haarslev, R. Möller, and M. Wessel. The description logic $\mathcal{ALCNH}_{R^+}$ extended with concrete domains: A practically motivated approach. In Goré et al. [5], pages 29–44.

[12] B. Hollunder and F. Baader. Qualifying number restrictions in concept languages. In J. Allen, R. Fikes, and E. Sandewall, editors, *Second International Conference on Principles of Knowledge Representation, Cambridge, Mass., April 22-25, 1991*, pages 335–346, April 1991.

[13] I. Horrocks. FaCT and iFaCT. In *Proc. of the 1999 Description Logic Workshop (DL'99)*, pages 133–135, 1999.

[14] I. Horrocks and U. Sattler. Ontology reasoning in the $\mathcal{SHOQ}(\mathcal{D})$ description logic. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)*, pages 199–204, August 2001.

[15] 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-Verlag, September 1999.

[16] I. Horrocks, U. Sattler, and S. Tobies. Reasoning with individuals for the description logic SHIQ. In David McAllester, editor, *Proc. of the 17th Int. Conf. on Automated Deduction (CADE 2000)*, number 1831 in LNCS. Springer-Verlag, Berlin, 2000.

[17] Ian Horrocks. Using an expressive description logic: FaCT or fiction? In *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 636–647, 1998.

[18] Deborah L. McGuinness. An industrial-strength description logic-based configurator platform. *IEEE Intelligent Systems*, July/August:69–77, 1998.

[19] D.L. McGuinness and P. Patel-Schneider, editors. *Proceedings of the International Workshop on Description Logics (DL'2001), Aug. 1-3, 2001, Stanford, CA, USA*, August 2001.

[20] H.J. Ohlbach and J. Köhler. Modal logics, description logics and arithmetic reasoning. *Artificial Intelligence*, 109(1-2):1–31, 1999.