Optimizing Reasoning in Description Logics with Qualified Number Restrictions

Volker Haarslev and Ralf Möller
University of Hamburg, Computer Science Department
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany

Abstract

In this extended abstract an optimization technique, the so-called signature calculus, for reasoning with number restrictions in description logics is investigated. The calculus is used to speed-up ABox (and TBox) reasoning in the description logic $\mathcal{ALCQH}_{R^+}$.

1 Introduction

The calculus presented in this paper addresses a major source of inefficiency concerning the treatment of number restrictions in the original tableaux calculi for expressive description logics such as $\mathcal{ALCNH}_{R^+}$ [1] and $\mathcal{SHIQ}$ [4]. The inefficiency is caused by large numbers occurring in at-least or at-most concepts. Due to the interaction of role hierarchies and number restrictions, it is not possible to deal with number restrictions in a simple way by considering representative individuals instead of a number of individuals. This source of inefficiency is illustrated with the following TBox (let $\mathsf{S_1}, \mathsf{S_2}, \mathsf{S_3}, \mathsf{R}$ be role names).

$$\mathbf{S_1} \sqsubseteq \mathsf{R},\ \mathbf{S_2} \sqsubseteq \mathsf{R},\ \mathbf{S_3} \sqsubseteq \mathsf{R}$$
$$\mathbf{C} \sqsubseteq \exists_{\geq 10}\, \mathsf{S_1} \sqcap \exists_{\geq 10}\, \mathsf{S_2} \sqcap \exists_{\geq 10}\, \mathsf{S_3} \sqcap (\forall\, \mathsf{S_1} \,.\, \mathsf{A} \sqcap \mathsf{B}) \sqcap (\forall\, \mathsf{S_2} \,.\, \neg \mathsf{A}) \sqcap (\forall\, \mathsf{S_3} \,.\, \neg \mathsf{A} \sqcap \neg \mathsf{B}) \sqcap \exists_{\leq 20}\, \mathsf{R}$$

In order to test the satisfiability of the concept C, currently implemented calculi for expressive DLs with number restrictions (e.g. [1, 4]) create 10 $\mathsf{S_i}$-successors for each role $\mathsf{S_i}$, $\mathsf{i} \in 1..3$, i.e. we get 30 R-successors but only 20 R-successors are allowed. The value restrictions ensure that only $\mathsf{S_2}$-successors and $\mathsf{S_3}$-successors can be successfully merged. The concept C is still satisfiable if we multiply each number by 10 or 100 etc. However, a merging process has to be applied to an increasing number of individuals possibly causing a combinatorial explosion, which is likely to occur in many concept satisfiability tests, even in the average case.

The so-called signature calculus presented below uses a compact representation, a signature, for large numbers of role successors caused by at-least concepts. Informally speaking, the new calculus is based on the idea to generate only "required" *proxy role successors* for roles (and their "conjunctions") occurring in number restrictions. This is achieved by introducing specific rules for dealing with so-called *signature assertions*. Due to the complexity caused by the combination of role hierarchies and number restrictions there is already a dramatic speed-up for number restrictions with values less than ten. In practical description logic systems these optimizations are of utmost importance in order to ensure that practical problems (e.g. configuration problems) can actually be solved with verified techniques. In the following, we assume the reader is familiar with the syntax and semantics of $\mathcal{ALCNH}_{R^+}$ [1]. In addition to the concept constructors in $\mathcal{ALCNH}_{R^+}$, the language $\mathcal{ALCQH}_{R^+}$ provides qualified number restrictions (see e.g. the logic $\mathcal{SHIQ}$ [4] for the syntax and semantics of qualified number restrictions). In contrast to $\mathcal{SHIQ}$, $\mathcal{ALCQH}_{R^+}$ does not support inverse roles. Furthermore, in contrast to $\mathcal{SHIQ}$, in $\mathcal{ALCQH}_{R^+}$ the unique name assumption is enforced for ABox individuals, i.e. the set of individuals O is divided into the set of old individuals O_O which can be mentioned in an ABox, and into the set of new individuals O_N which are generated only by the set of rules for testing ABox consistency.

2 Signature Calculus for $\mathcal{ALCQH}_{R^+}$

A calculus for deciding the ABox consistency problem for $\mathcal{ALCQH}_{R^+}$ is introduced in the following. First, a few definitions are required.

Definition 1 (Additional ABox Assertions) Let $\mathsf{a}, \mathsf{b} \in O$, $RS \subseteq R$ (the set of role names), and S be a simple role (see [1, 4]), then the following expressions are also assertional axioms ($n \geq 0, m > 0$):

$\mathsf{a} : \exists'_{\geq m}\, \mathsf{S}$ *(at-least number restriction)*, $(\mathsf{a}, \mathsf{b}) : \langle n, RS \rangle$ *(signature assertion)*.

An interpretation $\mathcal{I}$ satisfies an assertional axiom $\mathsf{a} : \exists'_{\geq m}\, \mathsf{S}$ iff $\mathsf{a}^{\mathcal{I}} \subseteq (\exists_{\geq m}\, \mathsf{S})^{\mathcal{I}}$ and an assertional axiom $(\mathsf{a}, \mathsf{b}) : \langle n, RS \rangle$ iff $\forall x \in \mathsf{a}^{\mathcal{I}},\ y \in \mathsf{b}^{\mathcal{I}} : (x, y) \in \bigcap_{\mathsf{R} \in RS} \mathsf{R}^{\mathcal{I}}$ and $\|\mathsf{b}^{\mathcal{I}}\| = n$.

We assume that every concept term is in negation normal form before the completion rules introduced below are applied (see [1, 4]). If a signature assertion $(\mathsf{a}, \mathsf{b}) : \langle n, RS \rangle$ is element of an ABox $\mathcal{A}$, we say it represents n "identical" role successors. The completion rules for signature assertions require a dedicated operator '$\widetilde{\sqcap}$' for well-formed sets of role names.

Definition 2 (Sub- and Super-Roles, Role Set, Role Conjunction) $\mathsf{R}^{\downarrow}$ ($\mathsf{R}^{\uparrow}$) denotes the sub-roles (super-roles) of R including R. If R is a set of roles, the union of the results of applying the operator to each member is denoted. A *well-formed role set* contains either a single role name or the direct parents of an anonymous (most specific) "role conjunction" whose canonical name is represented by this set. The set resulting from '$\widetilde{\sqcap}$' represents the "role conjunction" of its operands. Let $RS_1, RS_2 \subseteq R$ and $RS = RS_1 \cup RS_2$, then the role conjunction is defined as $RS_1 \,\widetilde{\sqcap}\, RS_2 = \{\mathsf{R} \in RS \mid \neg\exists\, \mathsf{S} \in RS : \mathsf{S} \in (\mathsf{R}^{\downarrow} \setminus \{\mathsf{R}\})\}$, i.e. the "role conjunction" is the union of both role sets without implied super-roles.

For instance, consider the role hierarchy example introduced in the introduction. The "role conjunction" $\{\mathsf{R}\} \,\widetilde{\sqcap}\, \{\mathsf{S}_1, \mathsf{S}_2\}$ yields $\{\mathsf{S}_1, \mathsf{S}_2\}$, i.e. R is not a member of the new set because R is already implied by at least one member of the new set (e.g. $\mathsf{S}_1 \sqsubseteq \mathsf{R}$), while $\{\mathsf{S}_1\} \,\widetilde{\sqcap}\, \{\mathsf{S}_2, \mathsf{S}_3\}$ yields $\{\mathsf{S}_1, \mathsf{S}_2, \mathsf{S}_3\}$.

Given a knowledge base $(\mathcal{T}, \mathcal{A})$, the problem of checking the ABox consistency of $\mathcal{A}$ w.r.t. $\mathcal{T}$ is transformed to the problem of checking the consistency of a so-called augmented ABox w.r.t. an empty TBox.

Definition 3 (Augmented ABox) For an initial ABox $\mathcal{A}$ w.r.t a TBox $\mathcal{T}$ we define its *augmented* ABox $\mathcal{A}'$ by applying the following rules to $\mathcal{A}$. 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. Every assertion of the form $(\mathsf{a}, \mathsf{b}) : \mathsf{R}$ is replaced by $(\mathsf{a}, \mathsf{b}) : \langle 1, \{\mathsf{R}\}\rangle$ and every pair of assertions of the form $(\mathsf{a}, \mathsf{b}) : \langle 1, RS_1\rangle, (\mathsf{a}, \mathsf{b}) : \langle 1, RS_2\rangle$, $RS_1 \neq RS_2$ is replaced by $(\mathsf{a}, \mathsf{b}) : \langle 1, RS_1 \,\widetilde{\sqcap}\, RS_2\rangle$ as long as possible. If $\exists_{\geq n}\, \mathsf{R} \,.\, \mathsf{C}$ occurs in $\mathcal{A}$ then replace this 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 the role hierarchy) as long as possible. From this point on, if we refer to an initial ABox $\mathcal{A}$ we always mean its augmented ABox.

The tableaux calculus also requires the notion of *blocking* the applicability of tableaux rules. This is based on so-called concept sets, an ordering for individuals, and on the notion of blocking individuals.

Definition 4 (Concept Sets, Individual Ordering, blocked) Given an ABox $\mathcal{A}$ and an individual a occurring in $\mathcal{A}$, the *concept set* of a is defined 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 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 $\mathcal{A}$.

Let $\mathcal{A}$ be an ABox and $\mathsf{a}, \mathsf{b} \in O$ be individuals in $\mathcal{A}$. We call a the *blocking individual* of b if the following conditions hold: $\mathsf{a}, \mathsf{b} \in O_N$, $\sigma(\mathcal{A}, \mathsf{a}) \supseteq \sigma(\mathcal{A}, \mathsf{b})$, and $\mathsf{a} \prec \mathsf{b}$. If there exists a blocking individual a for b, then b is said to be *blocked* (by a).

Definition 5 (Potential R-successors) Given an ABox $\mathcal{A}$, $\sharp(\mathsf{a},\mathsf{R},\mathsf{C})_\mathcal{A}$ defines the number of potential R-successors for an individual a mentioned in $\mathcal{A}$ whose R-successors are known as instances of C.

$$\begin{aligned}\sharp(\mathsf{a},\mathsf{R},\mathsf{C})_\mathcal{A} &= \sum_{\alpha\in\mathcal{A}} \mathrm{count}(\mathsf{a},\mathsf{R},\mathsf{C},\alpha)_\mathcal{A}\\ \mathrm{count}(\mathsf{a},\mathsf{R},\mathsf{C},\alpha)_\mathcal{A} &= \begin{cases} n & \text{if } \alpha = (\mathsf{a},\mathsf{b}):\langle n, RS\rangle,\ \mathsf{R}\in RS^{\uparrow},\ \mathsf{C}\in\sigma(\mathcal{A},\mathsf{b}),\\ 0 & \text{otherwise.}\end{cases}\end{aligned}$$

As an abbreviation $\sharp(\mathsf{a},\mathsf{R})_\mathcal{A}$ is used for $\sharp(\mathsf{a},\mathsf{R},\top)_\mathcal{A}$.

Definition 6 (Lower and Upper Bound for R-successors)
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 are known as instances of C).

$$\begin{aligned}\min(\mathsf{a},\mathsf{R})_\mathcal{A} &= \max(\{0\}\cup\{n \mid \mathsf{a}:\exists'_{\geq n}\mathsf{S}\in\mathcal{A},\ \mathsf{S}\in\mathsf{R}^{\downarrow}\}\cup\\ &\qquad\qquad \{\|\{\mathsf{b}\in O_O \mid (\mathsf{a},\mathsf{b}):\langle 1, RS\rangle\in\mathcal{A},\ \mathsf{R}\in RS^{\uparrow}\}\|\})\\ \max(\mathsf{a},\mathsf{R},\mathsf{C})_\mathcal{A} &= \min(\{\infty\}\cup\{n \mid \mathsf{a}:\exists_{\leq n}\mathsf{S}\,.\,\mathsf{C}\in\mathcal{A},\ \mathsf{S}\in\mathsf{R}^{\uparrow}\})\end{aligned}$$

Due to the unique name assumption for old individuals, the number of old R-successors has to be considered in the definition of $\min(\mathsf{a},\mathsf{R})_\mathcal{A}$.

2.1 Completion Rules

We are now ready to define the *completion rules* that are intended to generate a so-called completion (see below) of an initial ABox $\mathcal{A}$. In the following, the term $\sim\mathsf{C}$ is used for the negation normal form of $\neg\mathsf{C}$.

Definition 7 (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** $\mathcal{A}' = \mathcal{A}\cup\{\mathsf{a}:\mathsf{D}\}$

SigEmpty The signature cleanup rule (applied only after SigSplit and SigMerge).
if $(\mathsf{a},\mathsf{b}):\langle 0, RS\rangle\in\mathcal{A}$
then $\mathcal{A}' = \mathcal{A}\setminus(\{(\mathsf{a},\mathsf{b}):\langle 0, RS\rangle\}\cup\{\mathsf{b}:\mathsf{C} \mid \mathsf{b}:\mathsf{C}\in\mathcal{A}\})$

R∀C The role value restriction rule.

if 1. $\mathsf{a}:\forall\,\mathsf{R}\,.\,\mathsf{C} \in \mathcal{A}$, and

2. $\exists\,\mathsf{b} \in O : (\mathsf{a},\mathsf{b}):\langle n, RS\rangle \in \mathcal{A},\ \mathsf{R} \in RS^{\uparrow}$, and

3. $\mathsf{b}:\mathsf{C} \notin \mathcal{A}$

then $\mathcal{A}' = \mathcal{A} \cup \{\mathsf{b}:\mathsf{C}\}$

R∀$_+$C The transitive role value restriction rule.

if 1. $\mathsf{a}:\forall\,\mathsf{R}\,.\,\mathsf{C} \in \mathcal{A}$, and

2. $\exists\,\mathsf{b} \in O,\ \mathsf{T} \in \mathsf{R}^{\downarrow} \cap RS^{\uparrow} \cap T : (\mathsf{a},\mathsf{b}):\langle n, RS\rangle \in \mathcal{A}$, 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∀$_x$ 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∃$_{\geq n}$ The number restriction exists rule (for signatures).

if 1. $\mathsf{a}:\exists_{\geq n}\,\mathsf{S} \in \mathcal{A}$, and

2. a is not blocked, and

3. $\sharp(\mathsf{a},\mathsf{S})_{\mathcal{A}} < n$, and

then $\mathcal{A}' = (\mathcal{A} \setminus \{\mathsf{a}:\exists_{\geq n}\,\mathsf{S}\}) \cup \{\mathsf{a}:\exists'_{\geq n}\,\mathsf{S}, (\mathsf{a},\mathsf{b}):\langle n, \{\mathsf{S}\}\rangle\}$, $\mathsf{b} \in O_N$ new in $\mathcal{A}$.

SigSplit The signature split rule (nondeterministic).

if 1. $\mathsf{a}:\exists_{\leq m}\,\mathsf{R}\,.\,\mathsf{C} \in \mathcal{A}$, and

2. $\sharp(\mathsf{a},\mathsf{R})_{\mathcal{A}} > m$, and

3. $\mathcal{M}_{\mathsf{R}} = \{\alpha \in \mathcal{A} \mid \alpha = (\mathsf{a},\mathsf{b}):\langle n, RS\rangle,\ \mathsf{R} \in RS^{\uparrow},\ \{\sim\mathsf{C},\mathsf{C}\} \cap \sigma(\mathcal{A},\mathsf{b}) = \emptyset\}$, and

4. $\mathcal{M}_{\mathsf{R}} \neq \emptyset$

then select $(\mathsf{a},\mathsf{b}):\langle n, RS\rangle \in \mathcal{M}_{\mathsf{R}}$ and $\mathsf{E} \in \{\sim\mathsf{C},\mathsf{C}\}$ such that

if $\mathsf{b} \in O_O$

then (case a: *choose* $\sim\mathsf{C}$ *or* C *for old individual*)

$\mathcal{A}' = \mathcal{A} \cup \{\mathsf{b}:\mathsf{E}\}$

elsif $\exists\,\mathsf{c} \in O_N : \mathsf{c} \neq \mathsf{b},\ (\mathsf{a},\mathsf{c}):\langle k, RS\rangle \in \mathcal{A},\ \sigma(\mathcal{A},\mathsf{c}) = \sigma(\mathcal{A},\mathsf{b}) \cup \{\mathsf{E}\}$

then (case b: *change splitting*)

$\mathcal{A}' = (\mathcal{A} \setminus \{(\mathsf{a},\mathsf{c}):\langle k, RS\rangle,\ (\mathsf{a},\mathsf{b}):\langle n, RS\rangle\}) \cup$
$\{(\mathsf{a},\mathsf{c}):\langle k{+}1, RS\rangle,\ (\mathsf{a},\mathsf{b}):\langle n{-}1, RS\rangle\}$

else (case c: *split individual*)

$\mathcal{A}' = (\mathcal{A} \setminus \{(\mathsf{a},\mathsf{b}):\langle n, RS\rangle\}) \cup \{\mathsf{c}:\mathsf{D} \mid \mathsf{D} \in \sigma(\mathcal{A},\mathsf{b})\} \cup$
$\{(\mathsf{a},\mathsf{c}):\langle \mathit{1}, RS\rangle,\ \mathsf{c}:\mathsf{E},\ (\mathsf{a},\mathsf{b}):\langle n{-}1, RS\rangle\}$

with $\mathsf{c} \in O_N$ new in $\mathcal{A}$,

eliminate 0-signatures in $\mathcal{A}'$.

SigMerge The signature merge rule (nondeterministic).

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

2. $\mathcal{M}_{\mathsf{R}} = \{\alpha \in \mathcal{A} \mid \alpha = (\mathsf{a}, \mathsf{b}) : \langle n, RS \rangle,\ \mathsf{R} \in RS^{\uparrow},\ \mathsf{C} \in \sigma(\mathcal{A}, \mathsf{b})\}$

then select $\{(\mathsf{a}, \mathsf{b}_1) : \langle n_1, RS_1 \rangle,\ (\mathsf{a}, \mathsf{b}_2) : \langle n_2, RS_2 \rangle\} \subseteq \mathcal{M}_{\mathsf{R}}$ such that

1. $\mathsf{b}_1 \neq \mathsf{b}_2$, and

2. either $\mathsf{b}_1, \mathsf{b}_2 \in O_N$ or $\mathsf{b}_1 \in O_N$, $\mathsf{b}_2 \in O_O$

if $\mathsf{b}_1, \mathsf{b}_2 \in O_N$

then

if $\exists\, i, j \in 1..2 : RS_i{}^{\uparrow} \subseteq RS_j{}^{\uparrow},\ i \neq j$

then (case a: *decrement super-role signature*)

$\mathcal{A}' = (\mathcal{A} \setminus \{(\mathsf{a}, \mathsf{b}_i) : \langle n_i, RS_i \rangle\}) \cup \{(\mathsf{a}, \mathsf{b}_i) : \langle n_i - 1, RS_i \rangle\} \cup \{\mathsf{b}_j : \mathsf{D} \mid \mathsf{D} \in \sigma(\mathcal{A}, \mathsf{b}_i)\}$

elsif $\exists\, \mathsf{c} \in O_N : (\mathsf{a}, \mathsf{c}) : \langle n, RS_1 \,\tilde{\sqcap}\, RS_2 \rangle \in \mathcal{M}_{\mathsf{R}}$

then (case b: *increment common sub-role signature*)

$\mathcal{A}' = (\mathcal{A} \setminus \{(\mathsf{a}, \mathsf{b}_1) : \langle n_1, RS_1 \rangle, (\mathsf{a}, \mathsf{b}_2) : \langle n_2, RS_2 \rangle, (\mathsf{a}, \mathsf{c}) : \langle n, RS_1 \,\tilde{\sqcap}\, RS_2 \rangle\}) \cup$
$\{(\mathsf{a}, \mathsf{b}_1) : \langle n_1 - 1, RS_1 \rangle, (\mathsf{a}, \mathsf{b}_2) : \langle n_2 - 1, RS_2 \rangle, (\mathsf{a}, \mathsf{c}) : \langle n+1, RS_1 \,\tilde{\sqcap}\, RS_2 \rangle\} \cup$
$\{\mathsf{c} : \mathsf{D} \mid \mathsf{D} \in \sigma(\mathcal{A}, \mathsf{b}_1) \cup \sigma(\mathcal{A}, \mathsf{b}_2)\}$

else (case c: *create common sub-role signature*)

$\mathcal{A}' = (\mathcal{A} \setminus \{(\mathsf{a}, \mathsf{b}_1) : \langle n_1, RS_1 \rangle, (\mathsf{a}, \mathsf{b}_2) : \langle n_2, RS_2 \rangle\}) \cup$
$\{(\mathsf{a}, \mathsf{b}_1) : \langle n_1 - 1, RS_1 \rangle, (\mathsf{a}, \mathsf{b}_2) : \langle n_2 - 1, RS_2 \rangle, (\mathsf{a}, \mathsf{c}) : \langle 1, RS_1 \,\tilde{\sqcap}\, RS_2 \rangle\} \cup$
$\{\mathsf{c} : \mathsf{D} \mid \mathsf{D} \in \sigma(\mathcal{A}, \mathsf{b}_1) \cup \sigma(\mathcal{A}, \mathsf{b}_2)\}$, $\mathsf{c} \in O_N$ new in $\mathcal{A}$

else (case d: *merge with old individual* b_2)

$\mathcal{A}' = (\mathcal{A} \setminus \{(\mathsf{a}, \mathsf{b}_1) : \langle n, RS_1 \rangle, (\mathsf{a}, \mathsf{b}_2) : \langle 1, RS_2 \rangle\}) \cup$
$\{(\mathsf{a}, \mathsf{b}_1) : \langle n - 1, RS_1 \rangle, (\mathsf{a}, \mathsf{b}_2) : \langle 1, RS_1 \,\tilde{\sqcap}\, RS_2 \rangle\} \cup \{\mathsf{b}_2 : \mathsf{D} \mid \mathsf{D} \in \sigma(\mathcal{A}, \mathsf{b}_1)\}$,

eliminate 0-signatures in $\mathcal{A}'$.

After applying the SigSplit or SigMerge rule there might be "empty" signatures with a number restriction equal to 0. Empty signatures and the corresponding concept assertions for their role successors are immediately removed by the clean-up rule SigEmpty.

We call the rules SigSplit, and SigMerge (as well as the disjunction rule, see [1]) *nondeterministic* rules since they can be applied in different ways to the same set of assertions. The remaining rules are called *deterministic* rules. Moreover, we call the rules R∃C and $\text{R}\exists_{\geq n}$ *generating* rules since they are the only rules that increase the total number of role successors of already existing individuals. If the signature merge rule has been applied to an individual a and a non-transitive role R in an ABox $\mathcal{A}$, then it holds that $\sharp(\mathsf{a}, \mathsf{R})_{\mathcal{A}'} = \sharp(\mathsf{a}, \mathsf{R})_{\mathcal{A}} - 1$.

Given an initial ABox $\mathcal{A}$, more than one rule might be applicable to $\mathcal{A}$. This is controlled by a completion strategy in accordance with the ordering for new individuals (Definition 4). Basically, rules to younger individuals are only applied after all rules are applied to older individuals. In addition, non-generating rules are to be applied before generating rules. Due to space constraints we refer to [1] for details.

The calculus also has to check whether so-called clash triggers are applicable.

Definition 8 (Clash Triggers) We assume the same naming conventions as used above. Let $\mathcal{A}$ be an ABox and $\mathcal{A}'$ be its augmented ABox. The ABoxes $\mathcal{A}, \mathcal{A}'$ are called *contradictory* if one of the following *clash triggers* is applicable to $\mathcal{A}'$. If none of the clash triggers is applicable to $\mathcal{A}'$, then $\mathcal{A}$ and $\mathcal{A}'$ are called *clash-free*.

- *Primitive clash*: $\exists \{\mathsf{a}:\mathsf{A}, \mathsf{a}:\neg\mathsf{A}\} \subseteq \mathcal{A}'$, where A is a concept name.
- *Number restriction clash*:
  - $\exists\,\mathsf{a}$ mentioned in $\mathcal{A}'$, $\mathsf{R} \in P : \sharp(\mathsf{a},\mathsf{R})_{\mathcal{A}'} < \min(\mathsf{a},\mathsf{R})_{\mathcal{A}'}$ **or**
  - $\exists \{(\mathsf{a},\mathsf{b}):\langle n, RS\rangle, \mathsf{b}:\mathsf{C}\} \subseteq \mathcal{A}' : \mathsf{R} \in RS^{\uparrow}, n > \max(\mathsf{a},\mathsf{R},\mathsf{C})_{\mathcal{A}'}$.

As defined in [1], 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 whose augmented ABox contains a clash is obviously inconsistent. The purpose of the calculus is to generate a completion for an initial ABox $\mathcal{A}$ in order to prove the consistency of $\mathcal{A}$ or the inconsistency of $\mathcal{A}$ if no completion can be derived. For a given initial ABox $\mathcal{A}$, the calculus applies the completion rules from Definition 7. It stops if a clash occurs, it answers "*yes*" if a completion can be derived, and "*no*" otherwise.

The rules SigSplit and especially SigMerge are quite complex and need an explanation. The rule SigSplit splits from a signature a new one by adding an assertion for its proxy individual or shifts between two "matching" signatures the number or role successors by 1. SigSplit is applicable if the following conditions are met. An assertion of the form $\mathsf{a}:\exists_{\leq m}\,\mathsf{R}\,.\,\mathsf{C}$ is a member of $\mathcal{A}$, the number of potential role successors for R is greater than m, and there exists a non-empty set $\mathcal{M}_{\mathsf{R}}$ which contains all signature assertion of the form $(\mathsf{a},\mathsf{b}):\langle n, RS\rangle$ such that $\mathsf{R} \in RS^{\uparrow}$ and neither $\sim\mathsf{C}$ nor C is contained in the concept set $\sigma(\mathcal{A},\mathsf{b})$. Then, the rule SigSplit non-deterministically selects a signature assertion $(\mathsf{a},\mathsf{b}):\langle n, RS\rangle$ from the set $\mathcal{M}_{\mathsf{R}}$ and a concept E from the set $\{\sim\mathsf{C},\mathsf{C}\}$. The following cases are distinguished.

Case a: If b is an old individual, the rule simply adds the assertion $\mathsf{b}:\mathsf{E}$ to $\mathcal{A}$.

Case b: If there already exists a signature assertion $(\mathsf{a},\mathsf{c}):\langle k, RS\rangle$ in $\mathcal{A}$ where c is a new individual different from b and $\sigma(\mathcal{A},\mathsf{c}) = \sigma(\mathcal{A},\mathsf{b}) \cup \{\mathsf{E}\}$, the number of role successors of the assertion with c is incremented and the assertion with b is decremented.

Case c: Otherwise a new assertion with cardinality 1 is added to $\mathcal{A}$ and the assertion with b is decremented. The SigSplit rule could create a single signature assertion representing too many potential successors for a role R. However, this leads to a contradictory ABox due to the clash triggers defined in Definition 8.

The rule SigMerge is applicable to an individual a if a has more potential R-successors than allowed by applicable at most restrictions and there exists a set $\mathcal{M}_{\mathsf{R}}$ containing the signature assertions for the descendants of the role R. Then, a pair of assertions is non-deterministically selected such that the following conditions hold:

- The signature assertions use proxy names that are not equal to each other, i.e. we have two distinct signature assertions.
- The proxy individuals are either both new individuals or one is a new individual and the second one is an old individual. Due to the unique name assumption, two signature assertions for old proxy individuals may never be merged.

If these conditions are met for a selected pair of signature assertions, then the SigMerge rule distinguishes four mutually exclusive cases (a-d). If both proxy individuals are new ones, then the cases a-c are considered, otherwise the case d is applicable.

Case a: If one role set is a subset or equal to the second role set, then the counter of the (super)role signature (with the smaller set) is decremented by 1 and the concept assertions for the proxy individual of the (super)role signature are added to the proxy individual of the (sub)role signature.

Case b: If there already exists a role conjunction signature, then decrement the counters of the signature pair, increment the counter of the role conjunction signature, and add the concept assertions of the proxy individuals of the signature pair to the proxy individual of the role conjunction signature.

Case c: This case corresponds to case b but a new role conjunction signature with counter value 1 is added.

Case d: The proxy individual b_2 is an old individual and due to the unique name assumption the counter of the signature assertion with the proxy individual b_1 is decremented and the concept assertions for b_1 are added to b_2.

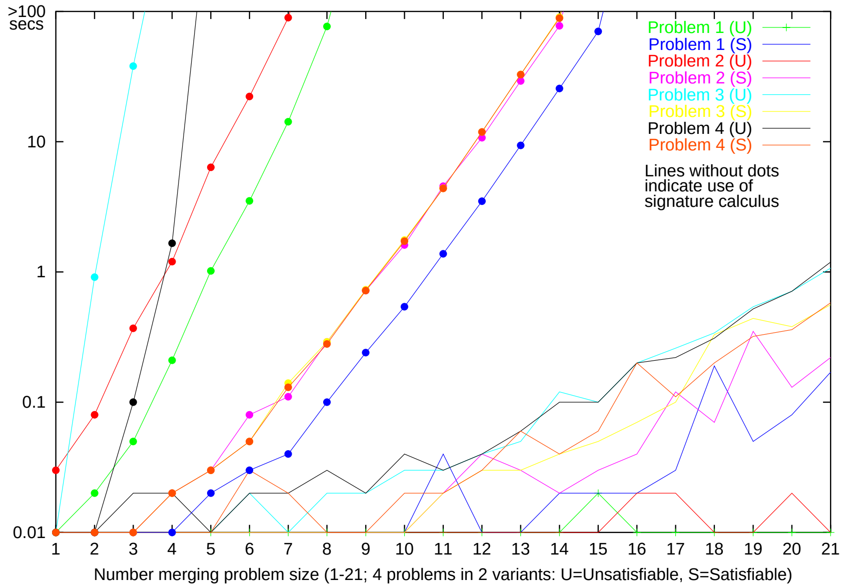

Figure 1: Runtime results for benchmark problems. The lines with dots indicate the runtimes with the standard calculus; the lines without dots demonstrate the performance gain of the signature calculus.

3 Evaluation

In order to evaluate the effectiveness of an implemented version of this calculus, a set of four dedicated benchmark problems has been generated. The increased difficulty of the problems is caused by exponentially increasing the size of numbers used in at-least and at-most concepts which, in turn, cause an exponential number of new concept and role assertions. Each of the four problems exists in two variants (a 'test concept' is consistent vs. inconsistent). The problems use role hierarchies and number restrictions. 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 $\mathsf{R}_\mathsf{i} \sqsubseteq \mathsf{R}, \mathsf{i} \in 1..3$. The (in)consistency of these terms has to be proven. A term is made consistent by choosing values for n, m_i such that $\max(m_1, m_2+m_3) \leq n$ or inconsistent if $\max(m_1, m_2+m_3) > n$. For details on the benchmark generation, we refer to [2]. Figure 1 demonstrates the result of this benchmark with the DL system RACER (note the use of a logarithmic scale). Qualitatively speaking, the use of the signature calculus indicates a dramatic performance gain of several orders of magnitude for this class of inference problems.

4 Conclusion

Our investigations indicate that the signature calculus is advantageous for satisfiable concept terms with number restrictions such as those discussed above. However, the signature calculus is no panacea for all kinds of inference problems involving number restrictions. The techniques described in this extended abstract complement the algebraic optimization techniques investigated in [3]. Both techniques can be applied in different circumstances which can be automatically detected.

5 Acknowledgments

The work presented in this extended abstract and in particular the work on the proofs (see [2]) has been done in collaboration with Stephan Tobies, RWTH Aachen. All deficiencies are due to our own faults, of course.

References

[1] 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), Breckenridge, Colorado, USA, 2000*, pages 273–284, April 2000.

[2] V. Haarslev and R. Möller. Optimizing reasoning with number restrictions. Technical report, University of Hamburg, Computer Science Department, August 2001.

[3] V. Haarslev, M. Timmann, and R. Möller. Combining tableaux and algebraic methods for reasoning with qualified number restrictions. In this volume.

[4] I. Horrocks, U. Sattler, and S. Tobies. Reasoning with individuals for the description logic SHIQ. In David MacAllester, editor, *Proceedings of the 17th International Conference on Automated Deduction (CADE-17)*, Lecture Notes in Computer Science, Germany, 2000. Springer Verlag.