Role Hierarchies and Number Restrictions

Hans Jürgen Ohlbach
Imperial College,
Department of Computing,
180 Queen's Gate,
London SW7 2BZ, England,
Email: h.ohlbach@doc.ic.ac.uk

Jana Koehler
Institute for Computer Science,
Albert Ludwigs University,
Am Flughafen 17,
79110 Freiburg, Germany,
Email: koehler@informatik.uni-freiburg.de

Abstract

We present the atomic decomposition technique as a general method for reducing reasoning about cardinalities of sets, in the DL case the cardinalities of role fillers, to equation solving problems. Then we apply the technique to a particular description logic with complex role terms and role hierarchies and a strong arithmetic component.

1 Introduction

If Henry has two sons and three daughters, he has five children. Therefore the DL-term $|has\text{-}son| = 2 \sqcap |has\text{-}daughter| = 3$ should subsume the term $|has\text{-}child| = 5$.[1] This subsumption relation holds only if we take into account that the role fillers of *has-son* and *has-daughter* form a partitioning of the role fillers of *has-child*. Therefore $|has\text{-}son| + |has\text{-}daughter| = |has\text{-}child|$. Without this information, sons, daughters, and children can be arbitrary sets having arbitrary overlaps with each other as shown in Figure 1. The three sets can be built up from seven mutually disjoint and unseparated areas. We gave these areas names with the following meaning:

c = children, not sons, not daughters;
s = sons, not children, not daughters;
d = daughters, not children, not sons;
cs = children, which are sons, not daughters;
cd = children, which are daughters, not sons;
sd = sons, which are daughters, not children;
csd = children, which are both sons and daughters.

The original sets can now be obtained from their "atomic" components: $children = c \cup cs \cup cd \cup csd$, $sons = s \cup cs \cup sd \cup csd$, $daughters = d \cup cd \cup sd \cup csd$. Moreover, since this decomposition is mutually disjoint

[1] We use the set notation $|r| = n$ to abbreviate the more common notation *atleast* $r\ n \sqcap$ *atmost* $r\ n$. It means that the number of r-role fillers is exactly n.

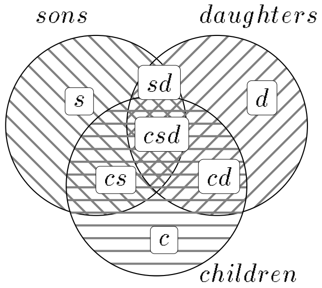

Figure 1: The general structure of the role fillers for *has-son*, *has-daughter* and *has-child*.

and exhaustive, the cardinalities of the sets just add up, and by dropping the cardinality function $|\ldots|$, the above subsumption problem can now be stated as

$$\begin{aligned} &s + cs + sd + csd = 2 \wedge d + cd + sd + csd = 3 \\ &\quad \Rightarrow\ c + cs + cd + csd = 5. \end{aligned} \tag{1}$$

In (1) we interpret the symbols $c, s, \ldots$ directly as the numbers denoting the cardinality of the corresponding role fillers. This makes sense because the sets are finite and mutually disjoint. This way, the subsumption problem has been transformed into a pure linear Diophantine equation problem about non-negative integer variables. If we take into account that the set of sons and daughters partition the set of children ($csd = 0$), all sons are children ($s = 0$ and $sd = 0$), all daughters are children ($d = 0$ and $sd = 0$), and there are no other children besides sons and daughters ($c = 0$), we end up with

$$cs = 2 \wedge cd = 3 \quad \Rightarrow \quad cs + cd = 5 \tag{2}$$

and this is in fact valid.

Let us recall the steps we performed. We started with the most general decomposition of the three sets into their atomic components. This allowed us to turn cardinality terms like $|has\text{-}son|$ into arithmetic terms and we obtained a pure linear Diophantine equation problem. Then we exploited *disjointness*, *subset*, and *exhaustiveness* relations between the different sets. Each such relation made some of the atoms empty and simplified the problem. Finally we ended up with the simple formula (2) we could submit to an arithmetic equation solver.

2 Atomic Decomposition of Sets

The introductory example in Figure 1 illustrated the *semantic* side of the decomposition idea. A set of sets is decomposed into basic non-overlapping components. The basis for the *syntactic* representation is a *finite* set $\mathcal{S} = \{s_1, \ldots, s_n\}$ of terms in some term language coming from some application. Each s_i is supposed to denote a subset of some global set $\mathcal{D}$. In the example, we had $\mathcal{S} = \{\textit{has-son}, \textit{has-daughter}, \textit{has-child}\}$. In addition to the symbols in $\mathcal{S}$ we have the fixed symbols $\top$ denoting the whole domain $\mathcal{D}$ and $\perp$ denoting the empty set. A syntactic representation of the atomic decomposition of $\mathcal{S}$ can be obtained by taking the power set of $\mathcal{S}$.

Definition 1 *Let* $\mathcal{S} \stackrel{\text{def}}{=} \{s_1, \ldots, s_n\}$ *be a* finite *set of terms of some term language which we also denote with $\mathcal{S}$ (for the purpose of this paper there is no difference). $\mathcal{S}$ together with $\{\top, \perp\}$ are the* basic set terms.

i) *We define the* atomic decomposition $\alpha_{\mathcal{S}}(\top)$ *as the set of subsets of $\mathcal{S}$. The elements of the decomposition are called the* atoms.

ii) *For a given $s \in \mathcal{S}$ let* $\alpha_{\mathcal{S}}(s) \stackrel{\text{def}}{=} \{w \in \alpha_{\mathcal{S}}(\top) \mid s \in w\}$ *be the $\mathcal{S}$*-decomposition *(or* atomic decomposition *or just simply* decomposition*) of s.* □

If we abbreviate in the 'children' example above *has-child* with c, *has-son* with s etc., we get $\alpha_{\mathcal{S}}(c) = \{c, cs, cd, csd\}$, $\alpha_{\mathcal{S}}(s) = \{s, cs, csd, sd\}$ etc.

Now we can take any interpretation $\mathcal{E}$ for $\mathcal{S}$ which maps the elements of $\mathcal{S}$ to some sets, and extend it to an interpretation $\mathcal{E}' = \alpha_{\mathcal{S}}(\mathcal{E})$ which gives the intended meaning to the atoms in $\alpha_{\mathcal{S}}(\top)$.

Definition 2 *For a set $\mathcal{S}$ of terms and its decomposition $\alpha_{\mathcal{S}}(\top)$ we extend the definition of $\alpha_{\mathcal{S}}$ to interpretations:*
i) Let $\mathcal{E}$ be an interpretation *function which assigns some set $\mathcal{E}(s) \subseteq \mathcal{D}$ of objects in some domain $\mathcal{D}$ to each member of $s \in \mathcal{S}$. $\mathcal{E}(\top) = \mathcal{D}$ and $\mathcal{E}(\perp) = \emptyset$ is required. We define the* extended interpretation function $\alpha_{\mathcal{S}}(\mathcal{E})$ *to be a new function which is like $\mathcal{E}$, but which assigns sets to the atoms as well: for $w \in \alpha_{\mathcal{S}}(\top)$*

$$\alpha_{\mathcal{S}}(\mathcal{E})(w) \stackrel{\text{def}}{=} \bigcap_{r \in w} \mathcal{E}(r) \setminus \bigcup_{s \in \mathcal{S}, s \notin w} \mathcal{E}(s).$$

Since the intersection over an empty index set is $\mathcal{D}$, $\alpha_{\mathcal{S}}(\mathcal{E})(\emptyset) = \mathcal{D} \setminus \bigcup_{s \in \mathcal{S}} \mathcal{E}(s)$ is defined as well.
ii) For a set $\{w_1, \ldots, w_k\}$ of atoms we define
$\alpha_{\mathcal{S}}(\mathcal{E})(\{w_1, \ldots, w_k\}) \stackrel{\text{def}}{=} \alpha_{\mathcal{S}}(\mathcal{E})(w_1) \cup \ldots \cup \alpha_{\mathcal{S}}(\mathcal{E})(w_k)$. □

The definition of $\mathcal{E}' = \alpha_{\mathcal{S}}(\mathcal{E})$ gives the intended meaning to the atoms because each set is decomposed into an exhaustive and mutually disjoint set of atoms using, for example $\mathcal{E}'(u) \cap \mathcal{E}'(v) = \emptyset$ for each $u \neq v \in \alpha_{\mathcal{S}}(\top)$ and $\mathcal{E}(r) = \bigcup_{w \in \alpha_{\mathcal{S}}(r)} \mathcal{E}'(w)$ for each $r \in \mathcal{S}$ hold. Details of this and further proofs can be found in [7].

Based on the symbols in $\mathcal{S}$ we can now define *set-terms* constructed with the usual set connectives $\cap$, $\cup$, and $\setminus$.

Definition 3 *Let again $\mathcal{S}$ be a finite set of terms, the basic set terms. We define the set $S_{\mathcal{S}}$ of (composed)* set-terms based on $\mathcal{S}$

$$S_{\mathcal{S}} ::= \mathcal{S} \mid S_{\mathcal{S}} \cup S_{\mathcal{S}} \mid S_{\mathcal{S}} \cap S_{\mathcal{S}} \mid S_{\mathcal{S}} \setminus S_{\mathcal{S}} \qquad \square$$

The atomic decomposition $\alpha_{\mathcal{S}}(\varphi)$ of a set-term φ is defined in the obvious way by taking the standard definition of the set connectives.

For all set-terms $S_{\mathcal{S}}$, one can compute the cardinality of the set associated to the set-term by adding up the cardinalities of the atoms in its decomposition.

Corollary 1 *For a finite set $\mathcal{S}$ of terms, an interpretation function $\mathcal{E}$ and a set-term φ with decomposition $\alpha_{\mathcal{S}}(\varphi) = \{w_1, \ldots, w_k\}$:*
i) $\mathcal{E}(\varphi) = \alpha_{\mathcal{S}}(\mathcal{E})(\alpha_{\mathcal{S}}(\varphi)) = \alpha_{\mathcal{S}}(\mathcal{E}(w_1)) \cup \ldots \cup \alpha_{\mathcal{S}}(\mathcal{E}(w_k))$.
ii) $|\mathcal{E}(\varphi)| = |\alpha_{\mathcal{S}}(\mathcal{E})(w_1)| + \ldots + |\alpha_{\mathcal{S}}(\mathcal{E})(w_k)|$. □

The decomposition of a set $\mathcal{S}$ with n elements yields 2^n atoms. They correspond to the set of all models of $\mathcal{S}$ taken as propositional variables. Relationships like disjointness and subset, however, may cause many atoms to denote empty sets. This information can be made available as an extra part of the specification, maybe as a set $\mathcal{H}$ of propositional logic formulae. For example *has-daughter* $\Rightarrow$ *has-child* expresses that the *has-daughter*-role is a sub-role of *has-child*. Computing the decomposition $\alpha_{\mathcal{H}}(\top)$ for the role relationship specification $\mathcal{H}$ now amounts to computing the set of all models of $\mathcal{H}$. For a particular class of role hierarchy specifications, which include subset, disjointness and partitioning declarations, we gave in [7] algorithms which perform better than the general exponential model generation algorithm.

The next step is to embed *cardinality terms* $|\varphi|$ where φ is a set term, into a general logic $\mathcal{L}$ with equality, negation and numbers, typically systems of *non-negative linear Diophantine* equations and inequations. Let $\mathcal{S}$ be a finite set of terms which are *disjoint* to the terms in $\mathcal{L}$. We extend $\mathcal{L}$ to $\mathcal{L}_{\mathcal{S}}$ by allowing cardinality terms $|\varphi|$ over set-terms $S_{\mathcal{S}}$ to be well formed terms as well. The integration is quite independent of the particular structure of the logic.

The semantics of $\mathcal{L}$ is extended in a straightforward way to a semantics of $\mathcal{L}_{\mathcal{S}}$: if $\mathcal{E}_{\mathcal{L}}$ is an interpretation for $\mathcal{L}$ and $\mathcal{E}$ is an interpretation for the set terms which maps them to *finite*[2] sets and which, in addition, maps

[2] The restriction to finite sets means that one can reason with cardinal numbers in the same way as with ordinal

$|\ldots|$ to the cardinality function, then $\mathcal{E}_\mathcal{S} \stackrel{\text{def}}{=} \mathcal{E}_\mathcal{L} \cup \mathcal{E}$ is an interpretation for $\mathcal{L}_\mathcal{S}$.

For the logic $\mathcal{L}$ we assume that the usual logical notions apply, i.e. we can speak of unsatisfiable and satisfiable formulae, tautologies, a satisfiability relation $\mathcal{E} \models \varphi$ between interpretations and formulae, and an entailment relation $\varphi \models \psi$. In the case of Diophantine equations, $\mathcal{E} \models \varphi$ means, $\mathcal{E}$ is a solution for φ, and $\varphi \models \psi$ means that all solutions for φ are also solutions for ψ.

The language $\mathcal{L}_\mathcal{S}$ is the language for formulating problems involving reasoning about cardinalities of sets. The actual process of inference, however, must take place in $\mathcal{L}$. Therefore, we replace a cardinality term $|\varphi|$ with its atomic decomposition $\alpha_\mathcal{H}(\varphi) = \{w_1, \ldots, w_k\}$, and further by the arithmetic terms denoting the sum of their cardinalities $\Sigma_\mathcal{H}(|\varphi|) \stackrel{\text{def}}{=} w_1 + \ldots + w_k$. For a $\mathcal{L}_\mathcal{S}$-formula ψ and a decomposition $\alpha_\mathcal{H}$ of set terms let $\alpha_\mathcal{H}(\psi)$ be the result of replacing all cardinality terms $|\varphi|$ occurring in ψ with $\Sigma_\mathcal{H}(|\varphi|)$.

The translation is sound because the interpretation of the sum terms is the same as the interpretation of the cardinality terms. It is complete because we can show that from an interpretation (solution) for the translated formula which assigns numbers to the atoms, sets with suitable cardinalities can be constructed such that the original formula is satisfiable.

Theorem 1 *For any $\mathcal{L}_\mathcal{S}$-formula φ and $\mathcal{L}_\mathcal{S}$-interpretation $\mathcal{E}$: If $\mathcal{E} \models \varphi$ then $\alpha^A_\mathcal{S}(\mathcal{E}) \models \alpha_\mathcal{H}(\varphi)$, where $\alpha^A_\mathcal{S}(\mathcal{E})$ interprets the atoms as the cardinalities of the sets.*
If $\alpha_\mathcal{H}(\varphi)$ is satisfiable then φ is satisfiable. □

3 The Description Logic $\mathcal{TF}^{++}$

So far we have a language for talking about cardinalities of sets and, by means of atomic decompositions, we can translate the set cardinality terms into ordinary sum-terms. We show how this idea can be put to work for reasoning with role hierarchies and number restriction in the Description Logic $\mathcal{TF}$ [6]. The general method, however, is not restricted to this language.

$\mathcal{TF}$ has only conjunction, the universal quantifier and the number restrictions *atleast* and *atmost*. The simple structure of $\mathcal{TF}$ allows for a straightforward application of the atomic decomposition technique. Moreover, we get some quite powerful extensions of the language for free. The extended language $\mathcal{TF}^{++}$ is limited to conjunction and universal quantification as well, but it is much more expressive on the arithmetic side because we allow for general graph-like role hierarchies and for roles denoted by composed set-terms (Def. 3. Number restrictions *atleast* and *atmost* have been replaced by general arithmetic constraints. Well-formed $\mathcal{TF}^{++}$declaration are:

---

numbers.

$$\begin{aligned}
\textit{cat-lovers} &\stackrel{\text{def}}{=} \textit{person} \sqcap \\
&\quad |\textit{has-cat}| > |\textit{has-son}| + |\textit{has-daughter}| \\
500er &\stackrel{\text{def}}{=} \textit{car} \sqcap \\
&\quad \textit{cubic-capacity} = 500 \cdot |\textit{has-cylinder}| \\
VIP\textit{-friend} &\stackrel{\text{def}}{=} \textit{person} \sqcap \\
&\quad \forall(\textit{has-friend} \setminus \textit{has-relative}).VIP
\end{aligned}$$

Given a set of symbols (role names) $\mathcal{R}$ and another set of symbols (concept names) $\mathcal{C}$, disjoint to $\mathcal{R}$, let $\mathcal{L}_\mathcal{R}$ be a suitable arithmetic language. Arithmetic constraints are formed over $\mathcal{L}_\mathcal{R}$ and since the role names are to be decomposed in our setting, $\mathcal{R}$ plays the role of $\mathcal{S}$ from the previous section. Free variables in $\mathcal{L}_\mathcal{R}$-formulae, as for example *cubic-capacity*, are allowed. They are interpreted as (number valued) functional roles.

Definition 4 *The set C of* concept terms *over $\mathcal{C}$, $\mathcal{R}$ and some set $\mathcal{F}$ of functional roles is defined as:*

$$C ::= \mathcal{C} \mid \mathcal{L}_{\mathcal{R}\cup\mathcal{F}} \mid C \sqcap C \mid \forall S'_\mathcal{R}.\mathcal{L}$$

where $S'_\mathcal{R} \stackrel{\text{def}}{=} S_\mathcal{R} \cup (\top, \bot)$. □

Definition 5 *An interpretation $\mathcal{E} = (\mathcal{D}, \Im)$ for concept terms and a given role hierarchy specification $\mathcal{H}$ over some set $\mathcal{R}$ of role names consists of a set $\mathcal{D}$ and an $\mathcal{L}$-interpretation $\Im$. $\Im$ consists of two parts, $\Im_\mathcal{L}$ which interprets the pure $\mathcal{L}$-symbols, and $\Im_C$ which interprets the main $\mathcal{TF}^{++}$parts. In particular, $\Im_C$ assigns*
i) a subset of the domain $\mathcal{D}$ to each concept name in $\mathcal{C}$,
ii) disjoint binary relations on $\mathcal{D}$ to each atom in $\alpha_\mathcal{H}(\top)$ together with the corresponding binary relations for the role names themselves and for the role terms,
iii) functions from domain elements to numbers for each functional role in $\mathcal{F}$;
iv) the usual interpretation to $\cap$, $\cup$, $\setminus$ and $|...|$.

For some $a \in \mathcal{D}$ we define $\Im_a$ to be like $\Im$, but for role names $r \in \mathcal{R}$ and atoms $w \in \alpha_\mathcal{H}(\top)$ we require $\Im_a(w) \stackrel{\text{def}}{=} \{b \mid (a, b) \in \Im_C(w)\}$ (role fillers). For $f \in \mathcal{F}$ let $\Im_a(f) \stackrel{\text{def}}{=} \Im_C(f)(a)$. $\Im_a$ is turned inductively into an $\mathcal{L}_\mathcal{R}$-interpretation for interpreting $\mathcal{L}_\mathcal{R}$ formulae. Let $\mathcal{E}_a \stackrel{\text{def}}{=} (\mathcal{D}, \Im_a)$.

An interpretation is turned into an interpretation for concept terms in the usual way, i.e. $\mathcal{E}(c) = \Im(c)$ if c is a concept name, $\mathcal{E}(\varphi \sqcap \psi) = \mathcal{E}(\varphi) \cap \mathcal{E}(\psi)$, etc. For arithmetic terms $\varphi \in \mathcal{L}_\mathcal{R}$ we define $\mathcal{E}(\varphi)$ as the set of all $a \in \mathcal{D}$ where $|\Im_a(r)|$ is finite for all r occurring in φ and $\Im_a \models \varphi$. For two concept terms φ and ψ we define: $\varphi \models \psi$ iff $\mathcal{E}(\varphi) \subseteq \mathcal{E}(\psi)$ for all *interpretations $\mathcal{E}$.* □

We also use $\Im_a$ for interpreting complex arithmetic formulae and assume that it is accompanied with an interpretation for $\mathcal{L}_\mathcal{R}$-formulae. For example, if $\varphi \stackrel{\text{def}}{=} |r| + |s| = 3$ and $\Im_a$ maps r to the set $\{b\}$ and s to the set $\{c, d\}$ then $\Im_a(|r|) = 1$ and $\Im_a(|s|) = 2$. Assuming that $\Im_a$ understands the $+$ sign properly, we find $\Im_a(|r| + |s|) = 3$, and thus $\Im_a \models |r| + |s| = 3$.

We call an interpretation $\mathcal{E}$ a *model for* a concept term φ if $\mathcal{E}(\varphi) \neq \emptyset$. If $\mathcal{E}(\varphi) = \emptyset$ for all interpretations $\mathcal{E}$ then

φ is *inconsistent* or *unsatisfiable*, otherwise it is called *consistent* or *satisfiable*.

A considerable reduction of the exponentially many atoms is possible in $\mathcal{TF}^{++}$ by exploiting the following theorem.

Theorem 2 *Let $\mathcal{H}$ be a role hierarchy specification for some roles $\mathcal{R}$ and φ be a satisfiable concept term. If for two role names r and s, there is neither a common sub-role nor a common super-role in the role hierarchy, and neither $r \cup s$ nor $r \cap s$ occur in φ explicitly or implicitly, then there is always a model where r and s are disjoint relations.* $\square$

For the proof, we first show the finite model property using selected filtration and then the existence of disjoint models using a copying technique, cf. [7]. Technically we can exploit this result by decomposing connected components of the role hierarchy separately. This way, the generation of exponentially many intersection atoms is avoided. In the extreme case, where the role hierarchy is flat, this optimized decomposition yields just one single atom per role. The number of atoms now depends primarily on the structure of the connected components in a role hierarchy, the narrower the better.

Linear and non-linear programming tools as, available through http://www.mcs.anl.gov/home/otc/Guide/faq/ can serve as the primary inference engines to identify tautological or unsatisfiable $\mathcal{TF}^{++}$ concepts and to decide subsumption.

Theorem 3 *A concept formula $\varphi \stackrel{\text{def}}{=} C \sqcap N \sqcap U$ where C are the concept names, N are the arithmetic formulae and U are the universal quantifications, is a tautology (written $\models \varphi$), i.e. $\mathcal{E}(\varphi)$ is the whole domain for all interpretations $\mathcal{E}$, iff i) C is empty and ii) $\alpha_{\mathcal{H}}(N)$ is a tautology and iii) for all '$\forall\, r.\psi$' $\in U$: ψ is a tautology.*

As a prerequisite to decide consistency and subsumption a normal form is developed which decomposes universal quantifications into atomic components as well.

Definition 6 *If $\varphi = \bigwedge_{r \in R} \forall\, r.\varphi_r$ is a* universal concept term *where R is a set of role terms we define*

$$\alpha_{\mathcal{H}}(\varphi) \stackrel{\text{def}}{=} \bigwedge_{w \in \bigcup_{r \in R} \alpha_{\mathcal{H}}(r)} \forall\, w.\Big(\bigwedge_{w \in \alpha_{\mathcal{H}}(r)} \varphi_r\Big).$$

For an atom w let $\alpha_{\mathcal{H}w}(\varphi) \stackrel{\text{def}}{=} \bigwedge_{w \in \alpha_{\mathcal{H}}(r)} \varphi_r$. $\square$

Given the proper interpretation of the atoms, the decomposition of quantifications is, in fact, an equivalence transformation.

Lemma 1 *For all role terms r, universal concept terms $\varphi = \bigwedge_{r \in R} \forall\, r.\varphi_r$ and interpretations $\mathcal{E} = (\mathcal{D}, \Im)$:*

$$\mathcal{E}(\varphi) = \alpha_{\mathcal{H}}(\mathcal{E})(\alpha_{\mathcal{H}}(\varphi)).$$ $\square$

The normal form for concept terms, we are going to define now, makes implicit information explicit and it even makes the universal quantifications redundant for the purpose of the consistency check.

Definition 7 *Let $\varphi = C \sqcap N \sqcap U$ be a concept term where C is a conjunction of concept names, N is a conjunction of $\mathcal{L}_{\mathcal{R}}$-formulae and U is a conjunction of universal quantifications (all conjunctions may be empty).*

The normal form $\eta(\varphi)$ of φ is

$$\eta(\varphi) \stackrel{\text{def}}{=} \begin{cases} C \sqcap \mathcal{A}_{\mathcal{H}}(\varphi) \sqcap \alpha'_{\mathcal{R}}(U) & \text{if } \mathcal{A}_{\mathcal{H}}(\varphi) \text{ is consistent} \\ \text{`inconsistent'} & \text{otherwise} \end{cases}$$

where $$\mathcal{A}_{\mathcal{H}}(\varphi) \stackrel{\text{def}}{=} \alpha_{\mathcal{H}}(N) \ \sqcap \bigwedge_{w \in \alpha_{\mathcal{H}}(\top),\, \mathcal{A}_{\mathcal{H}}(\alpha_{\mathcal{H}w}(\varphi)) \text{ is inconsistent}} w = 0$$

and $$\alpha'_{\mathcal{R}}(U) \stackrel{\text{def}}{=} \bigwedge_{\text{`}\forall\, w.\psi\text{'} \in \alpha_{\mathcal{H}}(U),\, \mathcal{A}_{\mathcal{H}}(\varphi) \not\models w=0 \text{ and } \not\models \psi} \forall\, w.\eta(\psi).$$ $\square$

$\mathcal{A}_{\mathcal{H}}(\varphi)$ in the normal form of $\eta(\varphi)$ is the arithmetic constraint part. It consists of the original constraint part $\alpha_{\mathcal{H}}(N)$ where the cardinality terms have been replaced by the corresponding sum terms, plus some equations $w = 0$ which come from quantifications $\forall\, w.\psi$ where ψ is inconsistent, and therefore there cannot be any w-successors. $\alpha'_{\mathcal{R}}(U)$ is the decomposed and reduced quantification part where all quantifications over empty atomic role components and all tautologies have been eliminated.

Theorem 4 *A concept term φ is consistent if and only if the arithmetic constraint part $\mathcal{A}_{\mathcal{H}}(\varphi)$ in $\eta(\varphi)$ is consistent.* $\square$

The theorem follows from the facts that (1) whenever φ is satisfiable, then the arithmetic part of $\eta(\varphi)$ is satisfiable and (2) the consistency of the arithmetic constraint part of the normal form for concept terms is sufficient to guarantee consistency of the original term.

Testing subsumption means figuring out whether $\mathcal{E}(\varphi) \subseteq \mathcal{E}(\psi)$ holds for two concept terms φ and ψ for *all* interpretations $\mathcal{E}$. In our case we make use of our normal form for concept terms where the arithmetic information is comprised in the arithmetic constraint part and the quantifications are decomposed into their atomic components. The structure of the normalized φ is $\varphi = C_\varphi \sqcap N_\varphi \sqcap U_\varphi$ where C_φ is just a conjunction of concept names, N_φ is the constraint part and U_φ are the decomposed quantifications. The same holds for $\psi = C_\psi \sqcap N_\psi \sqcap U_\psi$.

In order to verify $\varphi \models \psi$ we have to prove each conjunctive part in ψ from φ. The normal form allows us to separate the problem. The C_ψ-part can only follow from the C_φ-part, which is a pure propositional problem. The constraint part N_ψ can only follow from the N_φ-part. The atomic components $\forall\, w.c'$ of ψ can follow from corresponding $\forall\, w.c$ components of φ, if $c \models c'$ or N_φ forces $w = 0$.

Theorem 5 *Let $\varphi = C_\varphi \sqcap N \sqcap U$ be a* consistent *concept term with normal form $\eta(\varphi) = C_\varphi \sqcap N_\varphi \sqcap U_\varphi$. Let $\psi = C_\psi \sqcap N' \sqcap U'$ be another concept term with normal form $\eta(\psi) = C_\psi \sqcap N_\psi \sqcap U_\psi$. C_φ and C_ψ are conjunctions of concept names, N and N' are the arithmetic constraint parts, and U and U' are the universal quantifications.*
We have $\varphi \models \psi$ if and only if
i) $C_\psi \subseteq C_\varphi$,
ii) $N_\varphi \models N_\psi$ and
iii) for all '$\forall\, w.c'$' $\in U_\psi$:
a) $N_\varphi \models w = 0$ or
b) there is some '$\forall\, w.c$' $\in U_\varphi$ with $c \models c'$. □

Let us illustrate the subsumption checking procedure with an example taken from [6]. The task is to show that a concept $D \stackrel{\text{def}}{=} \textit{atleast}\ 3\ \ r$ subsumes

$$C \stackrel{\text{def}}{=} \forall (r \cap p).a \sqcap \forall (r \cap q).\textit{not-a} \sqcap \textit{atleast}\ 2\ \ (r \cap p) \sqcap \textit{atleast}\ 2\ \ (r \cap q)$$

where a and *not-a* have been axiomatized such that their conjunction is inconsistent, see [7]. To show $C \models D$ following Theorem 5 we need the normal forms $\eta(C)$ and $\eta(D)$ and begin with the decomposition of the role set $S = \{r, p, q\}$ according to Definition 1. We obtain $\alpha_{\mathcal{H}}(r) = \{r, rp, rq, rpq\}$, $\alpha_{\mathcal{H}}(p) = \{p, rp, pq, rpq\}$, $\alpha_{\mathcal{H}}(q) = \{q, rq, pq, rpq\}$. Now we are able to decompose the universal quantifiers in C following Definition 6 into

$$\forall\, rp.a \sqcap \forall\, rpq.(a \sqcap \textit{not-a}) \sqcap \forall\, rq.\textit{not-a}.$$

Since $a \sqcap \textit{not-a}$ is inconsistent we obtain

$$\alpha'_{\mathcal{R}}(U_C) = \forall\, rp.a \sqcap \forall\, rq.\textit{not-a}$$

and the first equation for $\mathcal{A}_{\mathcal{H}}(C)$ with $rpq = 0$. Using this equation to simplify the normal form $rp + rpq \geq 2 \wedge rq + rpq \geq 2$ of $\textit{atleast}\ 2\ \ (r \cap p) \sqcap \textit{atleast}\ 2\ \ (r \cap q)$ we obtain $\mathcal{A}_{\mathcal{H}}(C) = rp \geq 2 \wedge rq \geq 2$. Normalizing D leads to a single inequation

$$\eta(D) = \mathcal{A}_{\mathcal{H}}(D) = r + rp + rq + rpq \geq 3$$

Since C_C and C_D are empty there is nothing to show for concept names. Similarly for the universal quantification part because $\alpha'_{\mathcal{R}}(U_D) = \emptyset$. Thus it remains to prove that $\mathcal{A}_{\mathcal{H}}(C) \models \mathcal{A}_{\mathcal{H}}(D)$ by showing that

$$rp \geq 2 \wedge rq \geq 2 \Rightarrow r + rp + rq + rpq \geq 3$$

is valid, which is obvious.

4 Summary and Outlook

The atomic decomposition method integrates arithmetic reasoning about cardinalities of sets into symbolic knowledge representation languages. It allows for applications of arithmetic equation solvers for dealing efficiently with numbers. For the case of the DL $\mathcal{TF}$ we showed how to deal with role hierarchies with a number of set theoretic relationships between different roles, together with complex arithmetic constraints on the number of role fillers and the value of functional roles. Restricted to constraints about number-valued functional roles, the treatment is very similar to the treatment of concrete domains [1]. The arithmetic parts in $\mathcal{TF}^{++}$ are slightly different to the arithmetic parts described in [2], because in $\mathcal{TF}^{++}$ all free variables in the arithmetic formulae are interpreted as functional roles. $\mathcal{TF}^{++}$ has no explicit quantifiers over integer variables, which avoids all the problems discussed in [2]. We also don't have composition of roles as in [4]. The arithmetic part in $\mathcal{TF}^{++}$ is stronger than the number restrictions in [3], where no role hierarchies are considered, but the treatment of the arithmetic information by equation solvers is quite similar.

$\mathcal{TF}^{++}$ is still a relatively simple language. In [8] we therefore extended the ideas to a much more complex language including negation, disjunction, existential quantification, qualified number restrictions, the collection of objects operator, and A-Box reasoning. Qualified number restrictions are treated very similarly to the way described in [5].

References

[1] F. Baader and P. Hanschke. A Scheme for Integrating Concrete Domains into Concept Languages. In Proc. of the IJCAI-91, pages 452–457, Morgan Kaufmann, 1991.

[2] F. Baader and U. Sattler. Description Logics with Symbolic Number Restrictions In Proceedings of ECAI-96, pages 283–287, John Wiley & Sons Ltd, 1996.

[3] D. Calvanese and M. Lenzerini. Making Object-Oriented Schemas more Expressive Proceedings of PODS-94, pages 243-254, ACM Press and Addison Wesley.

[4] F. Baader and U. Sattler. Number Restrictions on Complex Roles in Description Logics. In Proceedings of KR-96

[5] Q. Elhaik, M.-C. Rousset and M.-C. Gaudel. A proposal for a Glass-Box Approach for Subsumption Checking. In proceedings of DL-96. See also the technical report available at http://www.lri.fr/˜quentin.

[6] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems.* LNAI 422. Springer, 1990.

[7] H. J. Ohlbach and J. Koehler. Reasoning about sets via atomic decomposition. Technical report TR-96-031, ICSI Berkeley, 1996. http://www.informatik.uni-freiburg/˜koehler/kr.html.

[8] H. J. Ohlbach. T-Box and A-Box Reasoning using Atomic Decompositions. Imperial College, 1997. Manuscript, available at http://theory.doc.ic.ac.uk/˜ho1/papers/ALCpp.ps.gz.