Set Description Languages and Reasoning about Numerical Features of Sets

Hans Jürgen Ohlbach
Dept. of Computer Science, Kings College, Strand, London WC2R 2LS
email: ohlbach@dcs.kcl.ic.ac.uk

Abstract

In this paper a combination methodology is presented, which allows one to use arithmetical algorithms for reasoning about numerical features of sets specified with the description logic $\mathcal{ALC}$ and a few of its variants. The methodology itself can be extended to other set description languages.

1 Introduction

Description logics can be used to specify object models of a given application domain. What they usually cannot do is reason about non-logical features of these sets. In particular cardinality information is difficult to process [3]. For example from (1) *silk-ties* $\Rightarrow$ *ties* and $|\textit{silk-ties}| \geq 100$ (there are more than 100 silk ties) no pure logical inference system can derive $|\textit{ties}| \geq 50$. Or, conversely, from (1) and $\textit{max-cost}(\textit{ties}) \leq 5$, no logical inference system would derive $\textit{max-cost}(\textit{silk-ties}) \leq 7$, although the numerical problems are trivial in this case.

In [1], Ohlbach and Koehler have presented *atomic decomposition* as a general technique for combining formal systems with a Boolean algebra component, i.e. propositional logic, as a *set description language*. The method reduces the problem formulation in a mixed language to a problem in the basic system. For example, in the above case it would reduce '$|\textit{silk-ties}| \geq 100 \Rightarrow |\textit{ties}| \geq 100$' to '$c_{st} \geq 100 \Rightarrow (c_{st} + c_{nst}) \geq 50$' where c_{st} and c_{nst} are ordinary non-negative integer valued variables. '$\textit{max-cost}(\textit{ties}) \leq 5 \Rightarrow \textit{max-cost}(\textit{silk-ties}) \leq 7$' would be reduced to '$max(c_{st}, c_{nst}) \leq 5 \Rightarrow c_{st} \leq 7$'. To do this, one exploits the set theoretic interpretation of (1) in an *atomic* Boolean algebra for decomposing set terms into their atomic components. For the simple case of the two terms *silk-ties* and *ties*, one gets four atomic components: (2) *silk-ties*, *ties*, (3) *silk-ties*, $\neg$*ties*, (4) $\neg$*silk-ties*, *ties*, and (5) $\neg$*silk-ties*, $\neg$*ties*, of which (3) contradicts (1). (2), (4) and (5) are actually the propositional models of (1). In the set theoretic interpretation they represent the atoms of the corresponding Boolean algebra: (2) represents the silk ties, (4) represents the non-silk ties, (5) represents all the rest.

All these sets are disjoint. *ties* consist of *silk-ties* and *non-silk-ties* ((2) and (4) are actually the models satisfying *ties*). Therefore *ties* can be decomposed into $\{\textit{silk-ties}, \textit{non-silk-ties}\}$. *silk-ties* need not be decomposed further.

In the next step one exploits that the *bridging functions* $|.|$ and *max-cost* are *additive*, which allows one to push the computation of the numerical values of the sets down to the atomic level. Additivity means in this case

$$|x \cap y| = \emptyset \Rightarrow |x \cup y| = |x| + |y| \quad \text{and}$$
$$\textit{max-cost}(x \cup y) = max(\textit{max-cost}(x), \textit{max-cost}(y)).$$

Since *silk-ties* and *non-silk-ties* are disjoint, we can decompose $|\textit{ties}|$ into $|\textit{silk-ties}| + |\textit{non-silk-ties}|$ and $\textit{max-cost}(\textit{ties})$ into $max(\textit{max-cost}(\textit{silk-ties}), \textit{max-cost}(\textit{non-silk-ties}))$.

The sets occurring in $|\textit{silk-ties}|$ and $|\textit{non-silk-ties}|$ are completely independent of each other. Therefore these terms can be replaced with ordinary integer valued variables. This finally reduces the problem to a pure arithmetical problem.

In this paper the methodology is generalized from propositional logic as set description language to the description logic $\mathcal{ALC}$. The models of $\mathcal{ALC}$ are Kripke frames, and not Boolean algebras. Nevertheless, one can still decompose the domain of an $\mathcal{ALC}$-interpretation into "atomic" components, use the additivity of the bridging functions to reduce the numerical features of the sets to numerical features of the atoms, and then turn them into numerical variables.

The difference to the Boolean algebra case are certain interaction axioms which correlate the non-Boolean parts in the $\mathcal{ALC}$-language with the numerical parts. They have the form

$$|\varphi| = 0 \Rightarrow |\exists\, r.\varphi| = 0$$

(If the set φ is empty then the set of objects having some r-successors in φ must be empty too.)

The problems for which we present a solution methodology consists of two parts, a set specification part (T-Box) in the language $\mathcal{ALC}$, and an arithmetical part in a mixed language consisting of $\mathcal{ALC}$-formulae and arithmetical expressions.

This part may describe an inference problem about cardinalities of sets or other numerical features, or a consistency problem involving arithmetics. We show how to use a set specification for eliminating the $\mathcal{ALC}$-formulae and turning the mixed problem into a pure arithmetical problem.

2 The Languages Involved

We need three components in the syntax. The first component is the language $\mathcal{L}_\mathcal{E}$ of some basic arithmetical system $\mathcal{E}$, which comes with its natural semantics. $\Im_\mathcal{E}$ usually represents a solution of an equation system.

The second component is the set description language $\mathcal{L}_\mathcal{S}$. As a concrete example we choose $\mathcal{L}_\mathcal{S} = \mathcal{ALC}$. The syntax of $\mathcal{ALC}$-formulae is:

$$\mathcal{ALC} = \mathbb{C}|\mathcal{ALC} \sqcup \mathcal{ALC}|\mathcal{ALC} \sqcap \mathcal{ALC}|\mathcal{ALC} \equiv \mathcal{ALC}| \\ \neg\mathcal{ALC}|\forall\,\mathbb{R}.\mathcal{ALC}|\exists\,\mathbb{R}.\mathcal{ALC}$$

where $\mathbb{C}$ is a set of *concept names* and $\mathbb{R}$ is a set of *role names*. The semantics is the usual well known Kripke semantics. *Note: $\mathcal{L}_\mathcal{E}$ and $\mathcal{L}_\mathcal{S}$ need not share any symbols!*

As a bridge between these two languages $\mathcal{L}_\mathcal{E}$ and $\mathcal{L}_\mathcal{S}$ we need a distinguished set F of *bridging functions*. One particular bridging function is the cardinality function $|...|$. Bridging functions must be *additive*: if a bridging function is applied to the union of two disjoint sets x and y, then it must be possible to apply the bridging function to x and y separately and then combine the result. For the set cardinality function for example, this is a very natural property: $|x \cup y| = |x| + |y|$, provided x and y are disjoint.

To ensure additivity, we require that the *additivity axioms* are satisfied.

Definition 2.1 (Additivity Axioms) *The additivity axioms for a bridging function $f \in F$ are:*

$$x \sqcap y = \emptyset \Rightarrow f(\ldots, t_{i-1}, x \sqcup y, t_{i+1}, \ldots) \\ = g_i(f(\ldots, t_{i-1}, x, t_{i+1}, \ldots), f(\ldots, t_{i-1}, y, t_{i+1}, \ldots))$$

for each argument position i. $g_i(x, y)$ is some term in $\mathcal{L}_\mathcal{E}$. ■

The combined language is defined as follows:

Definition 2.2 (The Combined Language $\mathcal{L}_{\mathcal{ES}}$) *If $s[x] \in \mathcal{L}_\mathcal{E}$ where x is some term occurring at some position in s, $f \in F$ with arity n, $t_1, \ldots, t_n \in \mathcal{L}_\mathcal{S}$ then $s[x/f(t_1, \ldots, t_n)] \in \mathcal{L}_{\mathcal{ES}}$. No other terms are in $\mathcal{L}_{\mathcal{ES}}$.* ■

The $\mathcal{L}_\mathcal{S}$-part in the combined language is ground with respect to $\mathcal{L}_\mathcal{E}$. That means there are no shared variables in different occurrences or even different arguments of the bridging functions. Since the language $\mathcal{L}_\mathcal{S}$ and $\mathcal{L}_\mathcal{E}$ do not share any symbols, we can define a combined interpretation $\Im_{ES}$ as the union of the interpretations $\Im_\mathcal{E}$ and $\Im_\mathcal{S}$. The interpretation of the bridging function symbols also becomes part of $\Im_{ES}$.

Definition 2.3 (Problem Specification) *A* problem specification $P = (\mathcal{P}, \mathcal{S}, \mathcal{B})$ *consists of:*

1. *a finite set $\mathcal{P}$ of $\mathcal{L}_{\mathcal{ES}}$ formulae, the* problem formulation*;*
2. *a finite set $\mathcal{S}$ of* set axioms*, (T-Box);*
3. *a set $\mathcal{B}$ of bridging function additivity axioms for each bridging function (Def. 2.1).* ■

In the following we assume that the T-Box is acyclic and a problem specification is in *expanded normal form*, i.e. all defined concepts are replaced by their definitions, and the definitions themselves have been removed from the set axioms.

Definition 2.4 (Set-Models) *An $\mathcal{L}_{\mathcal{ES}}$-interpretation $\Im$ is* a set-model for *a problem specification $P = (\mathcal{P}, \mathcal{S}, \mathcal{B})$ iff $\varphi^\Im = \Im_D$ for all $\varphi \in \mathcal{S}$ ($\Im_D$ is $\Im$'s domain), and $\Im$ satisfies the bridging axioms $\mathcal{B}$.*

A problem specification P is set-consistent *iff it has a set-model. A problem specification P is* consistent *iff it has a set-model which also satisfies the problem formulation $\mathcal{P}$.*

A $\mathcal{L}_\mathcal{S}$-formula ψ is consistent with $\mathcal{S}$ *iff there is a set-model $\Im$ for P with $\psi^\Im \neq \emptyset$.* ■

In the following we only consider set-consistent problem specifications. Set-consistency can be checked with the standard algorithms for $\mathcal{ALC}$.

3 Atomic Decomposition

We eliminate the set part from the problem specification and turn the main problem into a pure problem in the system $\mathcal{E}$. In the first step we decompose the $\mathcal{L}_\mathcal{S}$-formulae occurring as arguments of bridging functions in the problem formulation $\mathcal{P}$ into disjoint joins of meets. Each part corresponds to an atom of a Boolean set algebra.

Definition 3.1 (Atoms) *Given a problem specification $P = (\mathcal{P}, \mathcal{S}, \mathcal{B})$, let F_P be the set of all concept names together with the set of non-Boolean sub-formulae $\forall\,r.\varphi$ and $\exists\,r.\varphi$ occurring in $\mathcal{P}$ and $\mathcal{S}$. The set A_{F_P} of* syntactic atoms *over F_P is defined recursively:*

$$A_\emptyset = \emptyset.$$

$$A_{X \cup \{\psi\}} = \{\varphi \sqcap \psi | \varphi \in X, \varphi \sqcap \psi \text{ is consistent with } \mathcal{S}\} \cup \\ \{\varphi \sqcap \psi' | \varphi \in X, \varphi \sqcap \psi' \text{ is consistent with } \mathcal{S}\}.$$

■

The solution of the main problem in a problem specification may require interpretations where the $\mathcal{L}_\mathcal{S}$-formulae (and the atoms) denote sets with a given finite cardinality. Therefore our main interest is in interpretations with finite domain and, moreover, interpretations where the syntactic atoms denote non-empty sets. To this end we show a variation of the finite model property for $\mathcal{ALC}$ (and modal logic K_m).

Definition 3.2 (NDF-Model) *A* non-degenerated finite set-model *(NDF-model) for a problem specification P is a set-model $\Im$ for P whose domain is finite, and where all the atoms $a \in A_{F_P}$ are non-empty, i.e. $a^\Im \neq \emptyset$.*

Theorem 3.3 (NDF-Models exist) *Each set-consistent problem specification $P = (\mathcal{P}, \mathcal{S}, \mathcal{B})$ has a NDF-model $\Im$.*

Proof: Let $a \in A_{F_P}$. Since a is consistent with $\mathcal{S}$, there is a set-model $\Im_a$ for $\mathcal{S}$ with $a^{\Im_a} \neq \emptyset$. Selective filtration with respect to all (but finitely many) $\mathcal{L}_\mathcal{S}$-subformulae occurring in $\mathcal{P}$ and $\mathcal{S}$ yields a model $\Im'_a$ for $\mathcal{S}$ with finite domain. The domain elements are a representative system of the (finitely many) equivalence classes of $\Im_a$'s domain which yield the same truth value for all the $\mathcal{L}_\mathcal{S}$-subformulae in $\mathcal{P}$ and $\mathcal{S}$. An element in $a^{\Im_a}$ has to be chosen as a representative of the correpsonding equivalence class. The relations $r^{\Im'_a}$ are defined: $(x, y) \in r^{\Im'_a}$ iff $(x', y') \in r^{\Im_a}$ for some x', y' in the equivalence classes of x and y respectively. Structural induction now proves that $\Im'_a$ is a set-model for P.

$\Im$ itself is defined as the disjoint union of all the $\Im'_a$. Since there are only finitely many atoms in A_{F_P}, the domain of $\Im$ is finite. Since $\varphi^{\Im_a} = \mathcal{D}_a$ for all $\varphi \in \mathcal{S}$ (where $\mathcal{D}_a$ is $\Im_a$'s domain), and since $\Im$'s domain $\mathcal{D}$ is the union of all $\mathcal{D}_a$, $\varphi^\Im = \mathcal{D}$ for all $\varphi \in \mathcal{S}$. Moreover, since each $\mathcal{D}_a$ contains at least one element $x \in a^{\Im_a}$, and $\mathcal{D}$ is the *disjoint* union of the $\mathcal{D}_a$, $x \in a^\Im$ as well, i.e. $a^\Im \neq \emptyset$. ■

Lemma 3.4 (A_{F_P} is a Partitioning) *For each NDF-model $\Im$ of a problem specification P, $\{a^\Im \mid a \in A_{F_P}\}$ is a partitioning of $\Im$'s domain $\mathcal{D}$.*

Proof: By induction on the construction of A_{F_P} one can easily prove that all $a^\Im$ are disjoint. The fact that the union of all the $a^\Im$ is $\Im$'s domain can be proved by induction on the number of elements in F_P. The base case is that F_P contains just one element ψ. If ψ is inconsistent with $\mathcal{S}$, i.e. $\psi^\Im = \emptyset$ then $\psi'^\Im = \mathcal{D}$. If ψ' is inconsistent with $\mathcal{S}$, i.e. $\psi'^\Im = \emptyset$ then $\psi^\Im = \mathcal{D}$. If both are consistent then $A_{F_P} = \{\psi, \psi'\}$ and therefore $\psi^\Im \cup \psi'^\Im = \mathcal{D}$.

For the induction step we argue
$(X \sqcap \psi)^\Im \cup (X \sqcap \psi')^\Im = ((X \sqcap \psi) \sqcup (X \sqcap \psi'))^\Im = X^\Im = \mathcal{D}$. ■

The next lemma states that each $\mathcal{L}_\mathcal{S}$-formula occurring in a problem specification is either true for *all* elements of a partition in the partitioning specified by the syntactic atoms, or it is true in *no* element of the partition. This means that all formulae can be represented as joins of the corresponding atoms.

Lemma 3.5 *For all $\mathcal{L}_\mathcal{S}$-formulae φ occurring in $\mathcal{S}$ and $\mathcal{P}$ of a problem specification $P = (\mathcal{P}, \mathcal{S}, \mathcal{B})$, for all NDF-models $\Im$ for P and for all atoms $a \in A_{F_P}$: $a^\Im \cap \varphi^\Im = \emptyset$ or $a^\Im \cap \varphi^\Im = a^\Im$.*

Proof: by induction on the structure of φ. If φ is a concept name then either φ is a conjunct in a or φ' is a conjunct in a. In the first case $a^\Im \subseteq \varphi^\Im$ and in the second case $a^\Im \cap \varphi^\Im = \emptyset$. The induction steps for the Boolean connectives are straightforward applications of the induction hypotheses. The arguments for the induction steps for the existential and universal quantifiers are as in the base case (non-Boolean formulae in atoms are treated like concept names). ■

Definition 3.6 (Decomposition) *For a problem specification $P = (\mathcal{P}, \mathcal{S}, \mathcal{B})$ we define a decomposition function $\alpha_P(\varphi) = \bigsqcup_{a \in A_{F_P}, a \sqcap \varphi \text{ consistent with } \mathcal{S}} a$.*
The decomposition $\alpha_P(\mathcal{P})$ of the problem formulation is defined as the application of the decomposition function α_P to all $\mathcal{L}_\mathcal{S}$-type arguments of the bridging functions occurring in $\mathcal{P}$. ■

Lemma 3.7 (Equivalence Preserving)
For each $\mathcal{L}_\mathcal{S}$-formula φ occurring in a problem specification $P = (\mathcal{P}, \mathcal{S}, \mathcal{B})$ and for each NDF-model for P: $\varphi^\Im = \alpha_P(\varphi)^\Im$.

Proof: Since A_{F_P} defines a partitioning (Lemma 3.4)

$$\begin{aligned}
\varphi^\Im &= \textstyle\bigcup_{a \in A_{F_P}} (a^\Im \cap \varphi^\Im) \\
&= \textstyle\bigcup_{a \in A_{F_P}, a^\Im \cap \varphi^\Im \neq \emptyset} (a^\Im \cap \varphi^\Im) \\
&= \textstyle\bigcup_{a \in A_{F_P}, a \sqcap \varphi \text{ consistent with } \mathcal{S}} a^\Im \quad \text{(Lemma 3.5)} \\
&= \alpha_P(\varphi)^\Im
\end{aligned}$$

■

Since the extensions of the atoms are disjoint, the disjointness conditions of the additivity axioms are fulfilled. In a second step we can therefore apply the additivity axioms as equivalence preserving rewrite rules to push the bridging functions down to the atomic level. We end up with a modified problem formulation where the arguments of the bridging functions are only atoms $\varphi_1 \sqcap \ldots \sqcap \varphi_n$. Let us call this the *normalized problem formulation.*

The atoms are completely independent of each other, and we can replace the embedded atoms $f(\varphi_1 \sqcap \ldots \sqcap \varphi_n)$ with *names* $a_{f(\varphi_1 \sqcap \ldots \sqcap \varphi_n)}$. More precisely:

Definition 3.8 (Elimination of Bridging Fcts.) *The replacement operation β replaces in a decomposed problem formulation $P' = (\alpha_P(\mathcal{P}), \mathcal{S}, \mathcal{B})$ all bridging function symbols f with corresponding $\mathcal{L}_\mathcal{E}$-terms. β introduces for each term $f(a_1, \ldots, a_n)$ where the a_i are the atoms, a new $\mathcal{L}_\mathcal{E}$-constant symbol $f'_{a_1,\ldots,a_n}$ and replaces terms $f(a_1, \ldots, a_n$ with $f'_{a_1,\ldots,a_n}$.*

In addition, for each atom a occurring in $\alpha_P(\mathcal{P})$ and for each $\exists r.\psi \sqcap \forall r.\psi_1 \sqcap \ldots \sqcap \forall r.\psi_k$ which is part of a, β adds interaction axioms

$\beta(\alpha_P(|\psi \sqcap \psi_1 \sqcap \ldots \sqcap \psi_k|) = 0 \Rightarrow |a| = 0)$
to $\beta(\alpha_P(\mathcal{P}))$. ∎

The interaction axioms inserted by β are characteristic for the language $\mathcal{ALC}$. The soundness and completeness proof for β ensures that these axioms completely characterize the interaction between the quantifiers and the arithmetic part of $\mathcal{L}_{\mathcal{ES}}$.

Lemma 3.9 (Soundness and Completeness of β) *$\mathcal{P}' = \alpha_P(\mathcal{P})$ is satisfiable in the theory given by the $\mathcal{S}$ and $\mathcal{B}$-part of a decomposed problem formulation $P' = (\alpha_P(\mathcal{P}), \mathcal{S}, \mathcal{B})$ if and only if $\beta(\mathcal{P}')$ is satisfiable.*

Proof: If $\mathcal{P}'$ is satisfied by some interpretation $\Im'$ we just choose $\Im(f'_{a_1,\ldots,a_n})(\vec{x}) \stackrel{\text{def}}{=} \Im'(f(a_1,\ldots,a_n,\vec{x}))$. This guarantees that $\beta(\mathcal{P}')$ is satisfied as well.

For the other direction, suppose $\Im$ satisfies $\beta(\mathcal{P}')$. ($\beta(\mathcal{P}')$ may for example be a system of arithmetical equations. $\Im$ is in this case a solution of this system, which assigns a number to each variable $f'_{a_1,\ldots,a_n}$.)

We have to construct a new interpretation $\Im'$ which maps the atoms a to sets and the bridging functions f to corresponding functions operating on these sets, and which satisfies the $\mathcal{S}$ and $\mathcal{B}$-parts. In particular if f is the cardinality function, and, say $\Im(f'_a) = 3$, we must make sure that $\Im'$ maps a to a set with three elements.

By construction of the atoms a as meets of $\mathcal{L}_{\mathcal{S}}$-formulae which are consistent with $\mathcal{S}$ (and with $\mathcal{B}$ because a does not contain bridging functions), we know there is a NDF-model $\Im''$ for P (Lemmma 3.2) with $a^{\Im''} \neq \emptyset$. First of all we need to modify $\Im''$ such that $|a^{\Im''}| = \Im(\beta(|a|)$. Since $\Im''$ is a NDF-model for P, $a^{\Im''}$ has finitely many elements and is not empty. But $a^{\Im''}$ may have too many elements, in which case we must delete some, or it may have not enough elements, in which case we have to copy an element sufficiently often.

If one or more elements of $a^{\Im''}$ have to be deleted, we delete them, together with the corresponding parts of the relations (interpretation of the roles). However, if an element w is deleted, and there was another element v with $(v, w) \in r^{\Im''}$ and $v \in (\exists r.\psi \sqcap \forall r.\psi_1 \sqcap \ldots \sqcap \forall r.\psi_k)^{\Im''}$ we need to modify $r^{\Im''}$ and find a w' such that $(v, w') \in r^{\Im''}$ and $w' \in (\psi \sqcap \psi_1 \sqcap \ldots \sqcap \psi_k)^{\Im''}$. The interaction axioms introduced by β guarantee that $\Im$ does not force $\psi \sqcap \psi_1 \sqcap \ldots \sqcap \psi_k$ to be empty, because otherwise the atom containing v would be empty as well. Therefore there is a suitable w'.

With this modification one can delete elements from $a^{\Im''}$ and the resulting interpretation is still a set-model for P.

If there are not enough elements in $a^{\Im''}$ an element of $a^{\Im''}$ needs to be copied (together with outgoing r-relations) sufficiently often such that $a^{\Im''}$ gets enough elements. With structural induction one can show that the new interpretation is still a set-model for P.

By deleting and copying elements from the $a^{\Im''}$ we can therefore construct an interpretation $\Im'$ such that for the cardinality terms $\Im'(|a|) = \Im(\beta(|a|))$.

For the interpretation of the terms $f(a_1,\ldots,a_n,\vec{s})$ with other bridging functions, define the interpretation of the bridging functions f to be

$$\Im'(f)(a_1,\ldots,a_k,\vec{x}) \stackrel{\text{def}}{=} \Im(f'_{a_1,\ldots,a_n})(\vec{x}).$$

This is possible because neither the atoms a_i nor the bridging functions f occur in $\beta(\mathcal{P}')$. The terms $f(a_1,\ldots,a_n,\vec{s})$ in $\mathcal{P}'$ are now interpreted exactly like the terms $f'_{a_1,\ldots,a_n}(\vec{s})$ in $\beta(\mathcal{P}')$. Nothing else is changed and therefore $\mathcal{P}'$ is satisfied as well. ∎

As a consequence of the assumption that the reduction of the interaction axioms is satisfiability preserving, the fact that the atomic decomposition and the application of the additivity axioms is an equivalence transformation, and finally Lemma 3.9 we get a general soundness and completeness result.

Theorem 3.10 (Soundness and Completeness) *A problem specification $(\mathcal{P}, \mathcal{S}, \mathcal{B})$ is satisfiable iff the transformed specification $\beta(\alpha_p(\mathcal{P}))$ is satisfiable.* ∎

The inference procedure derived from this theorem comprises the following steps: For the problem specification $P = (\mathcal{P}, \mathcal{S}, \mathcal{B})$ in expanded normal form first compute the syntactic atoms by checking for all the combinations $\varphi_1 \sqcap \ldots \sqcap \varphi_n$ of positive and negative concept names and non-Boolean $\mathcal{L}_{\mathcal{S}}$-formula occurring in $\mathcal{P}$ and $\mathcal{S}$ whether they are consistent with $\mathcal{S}$. The consistent ones are kept, the other ones ignored. Now decompose the $\mathcal{L}_{\mathcal{S}}$-formula occurring in $\mathcal{P}$ into sets of syntactic atoms. Use the additivity axioms in $\mathcal{B}$ to push the bridging functions down to the level of single atoms. Then replace the resulting 'bridging terms' with variables or composed $\mathcal{L}_{\mathcal{E}}$-terms, add the interaction axioms for the existential quantifications and solve the resulting problem with an $\mathcal{E}$-problem solver.

4 Other Set Description Languages

The parts of the methodology presented above, wich actually depend on $\mathcal{ALC}$ as SDL are the consistency test (for each SDL there needs to be a suitable algorithm), the finite model property for showing the existence of NDF-models (Theorem 3.2), the interaction axioms introduced by β (Def. 3.8), and the corresponding parts in the proof of Lemma 3.9 dealing with deleting and copying objects in the domain.

The definition of syntactic atoms (Def. 3.1) can be easily adjusted: *all* non-Boolean $\mathcal{L}_{\mathcal{S}}$-formulae in the problem specification become part of F_P.

There are no problems with variants of $\mathcal{ALC}$ where the roles have special properties (reflexivity, symmetry, transitivity). If the $\mathcal{L}_{\mathcal{S}}$-consistency test is sound and complete for these properties then the methodology works.

The extension of $\mathcal{ALC}$ with number-valued functions also only needs an adjusted consistency test. Number restrictions in the $\mathcal{L}_S$-language, however, require more interaction axioms. *atleast* $3\ r \sqcap \forall r.\psi$, for example, requires the existence of at least 3 objects in the extension of ψ. Therefore corresponing interaction axioms need to be inserted by β.

5 A Final Example

We illustrate the whole procedure with an optimization problem from marketing research.

For political polls a number of at least 2300 U.S. people have to be surveyed, at least 1000 of them have to be 30 years or younger and 600 between 31 and 50. At least 15 % of the surveyed people have to live in a state bordering Mexico and not more than 20 % of those who are 51 or more live in a state bordering Mexico. The prices of interviewing people in each age and region category are 7.50 $ for those younger than 30 and living in a state bordering Mexico and 6.90 $ for those who do not live in a state bordering Mexico, and so on.

The sets of people to be interviewed are axiomatized in $\mathcal{ALC}$ with number-valued functions (*age* in this case). One possibility is:

$$
\begin{array}{lcl}
30^- & = & int \sqcap age \leq 30 \\
30^+ & = & int \sqcap (age \leq 30)' \wedge age \leq 50 \\
50^+ & = & int \sqcap (age \leq 50)' \\
bs & = & int \sqcap (\exists\, resident\text{-}in.bordering\text{-}state) \\
nbs & = & int \sqcap (\exists\, resident\text{-}in.bordering\text{-}state)' \\
int & \neq & bordering\text{-}state
\end{array}
$$

where int denotes the set of people to be interviewed, nb denotes the set of people to be intereviewed and living in the bordering states, nbs denotes the set of people to be interviewed not living in a bordering state, 30^- denotes the set of people younger than 30, 30^+ denotes the set of people between 30 and 50, and finally 50^+ denotes the set of people older than 50.

The arithmetic part of the specification can now be formulated using the notions introduced above:

$$
\begin{array}{l}
|int| \geq 2300, |30^-| \geq 1000, |30^+| \geq 600, \\
|bs| \geq 0.15 * |int|, \\
|50^+ \sqcap bs| \leq 0.20 * |50^+|, \\
costs(bs \sqcap 30^-) = 7.50 * |bs \sqcap 30^-|, \\
costs(bs \sqcap 30^+) = 6.80 * |bs \sqcap 30^+|, \quad \ldots
\end{array}
$$

The problem is to find numbers for the cardinalities of the sets such that $costs(int)$ is minimal.

In the fist step one has to replace the defined concepts by their definitions, which is straightforward in this case.

The set F_P is $\{int, bordering\text{-}state, age \leq 30, age \leq 50, \exists\, resident\text{-}in.bordering\text{-}state\}$.

The set formulae to be decomposed are int, $age \leq 30$, $age \leq 50$ and $bs \stackrel{\text{def}}{=} \exists\, resident\text{-}in.bordering\text{-}state$. There are six atoms which are consistent with the above definitions of these notions:

$$
\begin{array}{lcl}
b30^- & = & int \sqcap bs \sqcap age \leq 30 \\
b30^+ & = & int \sqcap bs \sqcap (age \leq 30)' \sqcap (age \leq 50) \\
b50^+ & = & int \sqcap bs \sqcap (age \leq 50)'
\end{array}
$$

together with $nb30^-$, $nb30^+$ and $nb50^+$, where bs' occurrs instead of bs. These six atoms correspond to the six different sets into which the set of people to be interviewed can be partitioned.

Using these six atoms one can decompose the set terms into sets of atoms, and the functions operating on Boolean terms, $|\ldots|$ and *cost* into new number valued variables. For example the term int can be replaced by $b30^- \sqcap nb30^- \sqcap b30^+ \sqcap nb30^+ \sqcap b50^+ \sqcap nb50^+$. The 'cardinality term' $|int|$ itself can therefore be replaced by a term denoting the sum of the cardinality of the atoms: $n_{b30^-} + n_{nb30^-} + n_{b30^+} + n_{nb30^+} + n_{b50^+} + n_{nb50^+}$. n_{b30^-} etc. are new non-negative integer valued variables. The *costs* terms are treated in a similar way. For example the term $costs(bs \wedge 30^-)$ is replaced with c_{b30^-} where c_{b30^-} is a positive real-valued variable. The result of all these transformations is

$$
\begin{array}{l}
n_{b30^-} + n_{nb30^-} + n_{b30^+} + n_{nb30^+} + n_{b50^+} + n_{nb50^+} \\
\geq 2300 \\
n_{b30^-} + n_{nb30^-} \geq 1000 \\
n_{b30^+} + n_{nb30^+} \geq 600 \\
n_{b30^-} + n_{b30^+} + n_{b50^+} = 0.15 * (n_{b30^-} + n_{nb30^-} \\
+ n_{b30^+} + n_{nb30^+} + n_{b50^+} + n_{nb50^+}) \\
n_{b50^+} \leq 0.2 * (n_{b50^+} + n_{nb50^+}) \\
c_{b30^-} = 7.50 * n_{b30^-} \qquad etc.
\end{array}
$$

This is now a standard representation of an optimization problem with objective function $c_{b30^-} + c_{nb30^-} + c_{b30^+} + c_{nb30^+} + c_{b50^+} + c_{nb50^+}$ to be minimized. The solution is $c_{nb30^-} = 1000$, $c_{b30^+} = 600$, $c_{b50^+} = 140$, $c_{nb50^+} = 560$.

6 Summary

Sophisticated inference procedures as for example mathematical programming techniques are usually restricted to the language they have been developed for. Any extension of the language is extremely difficult to incorporate into the procedure.

In this contribution I have developed a technique for incorporating the SDL $\mathcal{ALC}$ into such a language without affecting the calculus. By decomposing the set terms into their atomic components we are able to eliminate the set part completely from a mixed problem specification and reduce the given problem to an arithmetical problem.

References

[1] Hans Jürgen Ohlbach and Jana Koehler. How to extend a formal system with a Boolean Algebra component. In P.H. Schmidt W. Bibel, editor, *Automated Deduction. A Basis for Applications*, volume III, pages 57–75. Kluwer Academic Publishers, 1998.

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

[3] Baader, F., Buchheit, M., Hollunder, B., Cardinality restricitions on concepts, Artificial Intelligence, Vol 88, pp 195-213, 1996.