A Decomposition Rule for Decision Procedures by Resolution-based Calculi

Ullrich Hustadt[1], Boris Motik[2], and Ulrike Sattler[3]

[1] Department of Computer Science, University of Liverpool
Liverpool, UK
U.Hustadt@csc.liv.ac.uk

[2] FZI Research Center for Information Technologies at the University of Karlsruhe
Karlsruhe, Germany
motik@fzi.de

[3] Department of Computer Science, University of Manchester
Manchester, UK
sattler@cs.man.ac.uk

Abstract. Resolution-based calculi are among the most widely used calculi for theorem proving in first-order logic. Numerous refinements of resolution are nowadays available, such as e.g. *basic superposition*, a calculus highly optimized for theorem proving with equality. However, even such an advanced calculus does not restrict inferences enough to obtain decision procedures for complex logics, such as $\mathcal{SHIQ}$. In this paper, we present a new *decomposition* inference rule, which can be combined with any resolution-based calculus compatible with the standard notion of redundancy. We combine decomposition with basic superposition to obtain three new decision procedures: (i) for the description logic $\mathcal{SHIQ}$, (ii) for the description logic $\mathcal{ALCHIQ}b$, and (iii) for answering conjunctive queries over $\mathcal{SHIQ}$ knowledge bases. The first two procedures are worst-case optimal and, based on the vast experience in building efficient theorem provers, we expect them to be suitable for practical usage.

1 Introduction

Resolution-based calculi are nowadays among the most widely used calculi for theorem proving in first-order logic. The reasons for that are twofold. On the theoretical side, the initial resolution calculus was significantly refined to obtain various efficient calculi without losing soundness or completeness (e.g. [2, 15]). On the practical side, implementation techniques for efficient theorem provers have been devised and applied in practice (an overview is given in [21]).

Because of its popularity, resolution is often used as a framework for deciding various fragments of first-order logic. The fundamental principles for deciding a first-order fragment $\mathcal{L}$ by resolution are known from [12]. First, one selects a sound and complete resolution calculus $\mathcal{C}$. Next, one identifies the set of clauses $\mathcal{N}_{\mathcal{L}}$ such that for a finite signature, $\mathcal{N}_{\mathcal{L}}$ is finite and each formula $\varphi \in \mathcal{L}$, when translated into clauses, produces clauses from $\mathcal{N}_{\mathcal{L}}$. Finally, one demonstrates *closure* of $\mathcal{N}_\mathcal{L}$ under $\mathcal{C}$, namely, that applying an inference of $\mathcal{C}$ to clauses from $\mathcal{N}_\mathcal{L}$ produces a clause in $\mathcal{N}_\mathcal{L}$. This is sufficient to obtain a refutation decision procedure for $\mathcal{L}$ since, in the worst case, $\mathcal{C}$ will derive all clauses of $\mathcal{N}_\mathcal{L}$. An overview of decision procedures derived by these principles is given in [8].

The calculus $\mathcal{C}$ should be chosen to restrict inferences as much as possible without losing completeness. Namely, an unoptimized calculus usually performs unnecessary inferences which hinder closure of $\mathcal{N}_\mathcal{L}$ under $\mathcal{C}$. Consider the decision procedure for $\mathcal{SHIQ}^-$ description logic we presented in [11]. This logic provides so-called *number restrictions*, which are translated into first-order logic using counting quantifiers. We translate counting quantifiers into (in)equalities, and decide $\mathcal{SHIQ}^-$ by saturation under *basic superposition* [3, 14]. The prominent feature of basic superposition is the *basicness* restriction, by which superposition into terms introduced by unification can be omitted without compromising completeness. This restriction is crucial to obtain closure under inferences.

Interestingly, this approach does not yield a decision procedure for the slightly more expressive DL $\mathcal{SHIQ}$ [9] ($\mathcal{SHIQ}^-$ allows number restrictions only on roles without subroles). Namely, basic superposition alone is not restrictive enough to limit the term depth in conclusions. Therefore, we present *decomposition*, a new inference rule which can be used to transform certain conclusions. We show that decomposition is sound and complete when combined with basic superposition, which is interesting because of a non-standard approach to lifting used in basic superposition; however, the rule can be combined with any saturation calculus compatible with the standard notion of redundancy [2].

Decomposition indeed solves the motivating problem since it allows us to establish the closure under inferences for $\mathcal{SHIQ}$, and even yields an optimal decision procedure[4]. Furthermore, decomposition proves to be versatile and useful for other decidable fragments of first-order logic: we extend the basic superposition algorithm to handle $\mathcal{ALCHIQ}b$, a description logic providing *safe* Boolean role expressions. As for $\mathcal{SHIQ}$, this algorithm is optimal. Finally, we derive a decision procedure for answering conjunctive queries over $\mathcal{SHIQ}$ knowledge bases. Based on the vast experience in building efficient theorem provers, we believe that these algorithms are suitable for practice.

All results in this paper have been summarized in a technical report [10].

2 Preliminaries

Description Logics. Given a set of role names N_R, a $\mathcal{SHIQ}$ *role* is either some $R \in N_R$ or an *inverse role* R^- for some $R \in N_R$. A $\mathcal{SHIQ}$ *RBox* $KB_\mathcal{R}$ is a finite set of role inclusion axioms $R \sqsubseteq S$ and transitivity axioms $\mathsf{Trans}(R)$, for R and S $\mathcal{SHIQ}$ roles. As usual, for $R \in N_R$, we set $\mathsf{Inv}(R) = R^-$ and $\mathsf{Inv}(R^-) = R$, and we assume that, if $R \sqsubseteq S \in KB_\mathcal{R}$ ($\mathsf{Trans}(R) \in KB_\mathcal{R}$), then $\mathsf{Inv}(R) \sqsubseteq \mathsf{Inv}(S) \in KB_\mathcal{R}$ ($\mathsf{Trans}(\mathsf{Inv}(R)) \in KB_\mathcal{R}$) as well. A role R is *simple* if for each role $S \sqsubseteq^* R$, $\mathsf{Trans}(S) \notin KB_\mathcal{R}$ ($\sqsubseteq^*$ is the reflexive-transitive closure of $\sqsubseteq$).

[4] Optimal under the assumption that numbers in number restrictions are coded in unary.

Given a set of concept names N_C, $\mathcal{SHIQ}$ concepts are inductively defined as follows: each $A \in N_C$ is a $\mathcal{SHIQ}$ concept and, if C is a $\mathcal{SHIQ}$ concept, R a role, S a simple role, and n an integer, then $\neg C$, $C_1 \sqcap C_2$, $\forall R.C$, and $\leq n\, S.C$ are also $\mathcal{SHIQ}$ concepts. As usual, we use $C_1 \sqcup C_2$, $\exists R.C$, $\geq n\, S.C$ as abbreviations for $\neg(\neg C_1 \sqcap \neg C_2)$, $\neg\forall R.\neg C$, and $\neg(\leq (n-1)\, S.C)$. A TBox $KB_{\mathcal{T}}$ is a finite set of concept inclusion axioms $C \sqsubseteq D$. An ABox $KB_{\mathcal{A}}$ is a finite set of axioms $C(a)$, $R(a,b)$, and (in)equalities $a \approx b$ and $a \not\approx b$. A $\mathcal{SHIQ}$ knowledge base KB is a triple $(KB_{\mathcal{R}}, KB_{\mathcal{T}}, KB_{\mathcal{A}})$. The semantics of KB is given by translating it into first-order logic by the operator π from Table 1. The main inference problem is checking KB satisfiability, i.e. determining if a first-order model of $\pi(KB)$ exists.

The logic $\mathcal{SHIQ}^{-}$ is obtained from $\mathcal{SHIQ}$ by restricting roles in number restrictions $\leq n\, S.C$ and $\geq n\, S.C$ to *very* simple roles; a role S is *very simple* in $KB_{\mathcal{R}}$ if there is no role S' with $S' \sqsubseteq S \in KB_{\mathcal{R}}$. The restriction $\mathcal{ALCHIQ}$ of $\mathcal{SHIQ}$ is obtained by disallowing transitivity axioms $\mathsf{Trans}(R)$ in RBoxes.

Considering complexity, we must decide how to measure the *size* of concepts and knowledge bases. Here, we simply use their length, and assume *unary coding of numbers*, i.e. $|\leq n\, R.C| = n + 1 + |C|$.

Basic Superposition. We assume the standard notions of first-order clauses with equality: all existential quantifiers have been eliminated using Skolemization; all remaining variables are universally quantified; we only consider the equality predicate, i.e. all non-equational literals A are encoded as $A \approx \top$ in a multi-sorted setting; and we treat $\approx$ as having built-in symmetry. Moreover, we assume the reader to be familiar with standard resolution [2].

Basic superposition [3, 14] is an optimized version of superposition which prohibits superposition into terms introduced by unification in previously performed inferences. Its inferences rules are formalized by distinguishing two parts of a clause: (i) the *skeleton* clause C and (ii) the *substitution* σ representing the cumulative effects of all unifications. Such a representation of a clause $C\sigma$ is called a *closure*, and is written as $C \cdot \sigma$. A closure can conveniently be represented by *marking* the terms in $C\sigma$ occurring at variable positions of C with [ ]. Any position at or beneath a marked position is called a *substitution position*.

The calculus requires two parameters. The first is an *admissible* ordering on terms $\succ$, i.e. a *reduction ordering* total on ground terms. If $\succ$ is total on non-ground terms (as is the case in this paper), it can be extended to literals by associating, with each literal $L = s \circ t$, $\circ \in \{\approx, \not\approx\}$, a complexity measure $c_L = (\max(s,t), p_L, \min(s,t))$, where p_L is 1 if $\circ$ is $\approx$, and 0 otherwise. Now $L_1 \succ L_2$ iff $c_{L_1} \succ c_{L_2}$, where c_{L_i} are compared lexicographically, with $1 \succ 0$. The second parameter of the calculus is a *selection function* which selects an arbitrary set of negative literals in each clause.

The basic superposition calculus is a refutation procedure. If a set of closures N is *saturated up to redundancy* (meaning that all inferences from premises in N are redundant in N), then N is unsatisfiable if and only if it contains the empty closure. A literal $L \cdot \sigma$ is (strictly) maximal w.r.t. a closure $C \cdot \sigma$ if no $L' \in C$ exists, such that $L'\sigma \succ L\sigma$ ($L'\sigma \succeq L\sigma$). A literal $L \cdot \sigma$ is *(strictly) eligible for superposition* in $(C \vee L) \cdot \sigma$ if there are no selected literals in $(C \vee L) \cdot \sigma$ and $L \cdot \sigma$

Table 1. Semantics of $\mathcal{SHIQ}$ by Mapping to FOL

Concepts to FOL:	$\pi_y(A,X) = A(X)$ $\pi_y(C \sqcap D, X) = \pi_y(C,X) \wedge \pi_y(D,X)$
	$\pi_y(\neg C, X) = \neg \pi_y(C,X)$ $\pi_y(\forall R.C, X) = \forall y : R(X,y) \rightarrow \pi_x(C,y)$
	$\pi_y(\leq n\, S.C, X) = \forall y_1, \ldots, y_{n+1} : \bigwedge S(X,y_i) \wedge \bigwedge \pi_x(C,y_i) \rightarrow \bigvee y_i \approx y_j$
Axioms to FOL:	$\pi(C \sqsubseteq D) = \forall x : \pi_y(C,x) \rightarrow \pi_y(D,x)$
	$\pi(R \sqsubseteq S) = \forall x,y : R(x,y) \rightarrow S(x,y)$
	$\pi(\mathsf{Trans}(R)) = \forall x,y,z : R(x,y) \wedge R(y,z) \rightarrow R(x,z)$
KB to FOL:	$\pi(R) = \forall x,y : R(x,y) \leftrightarrow R^-(y,x)$
	$\pi(KB_\mathcal{R}) = \bigwedge_{\alpha \in KB_\mathcal{R}} \pi(\alpha) \wedge \bigwedge_{R \in N_R} \pi(R)$
	$\pi(KB_\mathcal{T}) = \bigwedge_{\alpha \in KB_\mathcal{T}} \pi(\alpha)$
	$\pi(KB_\mathcal{A}) = \bigwedge_{C(a) \in KB_\mathcal{A}} \pi_y(C,a) \wedge \bigwedge_{R(a,b) \in KB_\mathcal{A}} R(a,b) \wedge \bigwedge_{a \approx b \in KB_\mathcal{A}} a \approx b \wedge \bigwedge_{a \not\approx b \in KB_\mathcal{A}} a \not\approx b$
	$\pi(KB) = \pi(KB_\mathcal{R}) \wedge \pi(KB_\mathcal{T}) \wedge \pi(KB_\mathcal{A})$

X is a meta variable and is substituted by the actual variable.
π_x is defined as π_y by substituting $x_{(i)}$ for all $y_{(i)}$, respectively, and π_y for π_x.

Table 2. Inference Rules of the $\mathcal{BS}$ Calculus

Positive superposition: $\dfrac{(C \vee s \approx t) \cdot \rho \quad (D \vee w \approx v) \cdot \rho}{(C \vee D \vee w[t]_p \approx v) \cdot \theta}$	(*i*) $\sigma = \mathsf{MGU}(s\rho, w\rho\|_p)$ and $\theta = \rho\sigma$, (*ii*) $t\theta \not\succeq s\theta$ and $v\theta \not\succeq w\theta$, (*iii*) $(s \approx t) \cdot \theta$ is strictly eligible for superposition, (*iv*) $(w \approx v) \cdot \theta$ is strictly eligible for superposition, (*v*) $s\theta \approx t\theta \not\succeq w\theta \approx v\theta$, (*vi*) $w\|_p$ is not a variable.
Negative superposition: $\dfrac{(C \vee s \approx t) \cdot \rho \quad (D \vee w \not\approx v) \cdot \rho}{(C \vee D \vee w[t]_p \not\approx v) \cdot \theta}$	(*i*) $\sigma = \mathsf{MGU}(s\rho, w\rho\|_p)$ and $\theta = \rho\sigma$, (*ii*) $t\theta \not\succeq s\theta$ and $v\theta \not\succeq w\theta$, (*iii*) $(s \approx t) \cdot \theta$ is strictly eligible for superposition, (*iv*) $(w \not\approx v) \cdot \theta$ is eligible for resolution, (*v*) $w\|_p$ is not a variable.
Reflexivity resolution: $\dfrac{(C \vee s \not\approx t) \cdot \rho}{C \cdot \theta}$	(*i*) $\sigma = \mathsf{MGU}(s\rho, t\rho)$ and $\theta = \rho\sigma$, (*ii*) $(s \not\approx t) \cdot \theta$ is eligible for resolution.
Equality factoring: $\dfrac{(C \vee s \approx t \vee s' \approx t') \cdot \rho}{(C \vee t \not\approx t' \vee s' \approx t') \cdot \theta}$	(*i*) $\sigma = \mathsf{MGU}(s\rho, s'\rho)$ and $\theta = \rho\sigma$, (*ii*) $t\theta \not\succeq s\theta$ and $t'\theta \not\succeq s'\theta$, (*iii*) $(s \approx t) \cdot \theta$ is eligible for superposition.
Ordered Hyperresolution: $\dfrac{E_1 \ldots E_n \quad E}{(C_1 \vee \ldots \vee C_n \vee D) \cdot \theta}$	(*i*) E_i are of the form $(C_i \vee A_i) \cdot \rho$, for $1 \leq i \leq n$, (*ii*) E is of the form $(D \vee \neg B_1 \vee \ldots \vee \neg B_n) \cdot \rho$, (*iii*) σ is the most general substitution such that $A_i\theta = B_i\theta$ for $1 \leq i \leq n$, and $\theta = \rho\sigma$, (*iv*) $A_i \cdot \theta$ is strictly eligible for superposition, (*v*) $\neg B_i \cdot \theta$ are selected, or nothing is selected, $i = 1$ and $\neg B_1 \cdot \theta$ is maximal w.r.t. $D \cdot \theta$.

is (strictly) maximal w.r.t. $C \cdot \sigma$; $L \cdot \sigma$ is *eligible for resolution* in $(C \vee L) \cdot \sigma$ if it is selected in $(C \vee L) \cdot \sigma$ or there are no selected literals in $(C \vee L) \cdot \sigma$ and $L \cdot \sigma$ is maximal w.r.t. $C \cdot \sigma$. We denote basic superposition with $\mathcal{BS}$ and present its inference rules in Table 2. The ordered hyperresolution rule is a macro inference, combining negative superposition and reflexivity resolution. The closure E is called the *main premise*, and the closures E_i are called the *side premises*. An overview of the completeness proof and compatible redundancy elimination rules are given in [10].

3 Motivation

To motivate the need for decomposition, we give an overview of our procedure for deciding satisfiability of a $\mathcal{SHIQ}^-$ knowledge base KB using $\mathcal{BS}$ from [11] and highlight the problems related to deciding full $\mathcal{SHIQ}$. We assume that KB has an *extensionally reduced* ABox, where all concepts occurring in ABox assertions are atomic. This is without loss of generality, since each axiom $C(a)$, where C is complex, can be replaced with axioms $A_C(a)$ and $A_C \sqsubseteq C$, for A_C a new concept; this transformation is obviously polynomial.

3.1 Deciding $\mathcal{SHIQ}^-$ by $\mathcal{BS}$

Eliminating Transitivity. A minor problem in deciding satisfiability of KB are the transitivity axioms, which, in their clausal form, do not contain so-called *covering* literals (i.e. literals containing all variables of a clause). Such clauses are known to be difficult to handle, so we preprocess KB into an equisatisfiable $\mathcal{ALCHIQ}^-$ knowledge base $\Omega(KB)$. Roughly speaking, we replace each transitivity axiom $\mathsf{Trans}(S)$ with axioms $\forall R.C \sqsubseteq \forall S.(\forall S.C)$, for each R with $S \sqsubseteq^* R$ and C a concept occurring in KB. This transformation is polynomial.

Preprocessing. We next translate $\Omega(KB)$ into a first-order formula $\pi(KB)$ according to Table 1. Assuming unary coding of numbers, $\pi(KB)$ can be computed in polynomial time. To transform $\pi(KB)$ into a set of closures $\Xi(KB)$, we apply the well-known *structural transformation* [16]. Roughly speaking, the structural transformation introduces a new name for each non-atomic subformula of $\pi(KB)$. It is well-known that $\pi(KB)$ and $\Xi(KB)$ are equisatisfiable, and that $\Xi(KB)$ can be computed in polynomial time.

For any KB, all closures from $\Xi(KB)$ are of types from Table 3; we call them $\mathcal{ALCHIQ}^-$-closures. We use the following notation: for a term t, with $\mathbf{P}(t)$ we denote a disjunction of the form $(\neg)P_1(t) \vee \ldots \vee (\neg)P_n(t)$, and with $\mathbf{P}(\mathbf{f}(x))$ we denote a disjunction of the form $\mathbf{P_1}(f_1(x)) \vee \ldots \vee \mathbf{P_m}(f_m(x))$ (notice that this allows each $\mathbf{P_i}(f_i(x))$ to contain positive and negative literals). With $\langle t \rangle$ we denote that the term t may, but need not be marked. In all closure types, some of the disjuncts may be empty. Furthermore, for each function symbol f occurring in $\Xi(KB)$, there is exactly one closure of type 3 containing $f(x)$ unmarked; this closure is called the R^f*-generator*, the disjunction $\mathbf{P^f}(x)$ is called the f*-support*, and R is called the *designated role* for f and is denoted as $\mathsf{role}(f)$.

Parameters for $\mathcal{BS}$. We use $\mathcal{BS}_{DL}$ to denote the $\mathcal{BS}$ calculus parameterized as follows. We use a standard *lexicographic path ordering* [7, 1] (LPO) for comparing terms. LPOs are based on a precedence $>_P$ over function, constant, and predicate symbols. If the precedence is total, LPO is admissible for basic superposition. To decide $\mathcal{ALCHIQ}^-$, we can use any precedence such that $f >_P c >_P p >_P \top$, for any function symbol f, constant c, and predicate symbol p. We select all negative binary literals in a closure. On $\mathcal{ALCHIQ}^-$-closures $\mathcal{BS}_{DL}$ compares only terms with at most one variable, and LPOs are total for such terms. Hence, literals in $\mathcal{ALCHIQ}^-$-closures can be compared as explained in Section 2.

Table 3. Types of $\mathcal{ALCHIQ}^-$-closures

1	$\neg R(x,y) \vee \mathsf{Inv}(R)(y,x)$
2	$\neg R(x,y) \vee S(x,y)$
3	$\mathbf{P^f}(x) \vee R(x, \langle f(x) \rangle)$
4	$\mathbf{P^f}(x) \vee R([f(x)], x)$
5	$\mathbf{P_1}(x) \vee \mathbf{P_2}(\langle \mathbf{f}(x) \rangle) \vee \bigvee \langle f_i(x) \rangle \approx/\not\approx \langle f_j(x) \rangle$
6	$\mathbf{P_1}(x) \vee \mathbf{P_2}([g(x)]) \vee \mathbf{P_3}(\langle \mathbf{f}([g(x)]) \rangle) \vee \bigvee \langle t_i \rangle \approx/\not\approx \langle t_j \rangle$ where t_i and t_j are either of the form $f([g(x)])$ or of the form x
7	$\mathbf{P_1}(x) \vee \bigvee \neg R(x, y_i) \vee \mathbf{P_2}(\mathbf{y}) \vee \bigvee y_i \approx y_j$
8	$\mathbf{R}(\langle \mathbf{a} \rangle, \langle \mathbf{b} \rangle) \vee \mathbf{P}(\langle \mathbf{t} \rangle) \vee \bigvee \langle t_i \rangle \approx/\not\approx \langle t_j \rangle$ where t, t_i and t_j are either some constant b or a functional term $f_i([a])$

Conditions:

(i): In any term $f(t)$, the inner term t occurs marked.
(ii): In all positive equality literals with at least one function symbol, both sides are marked.
(iii): Any closure containing a term $f(t)$ contains $\mathbf{P^f}(t)$ as well.
(iv): In a literal $[f_i(t)] \approx [f_j(t)]$, $\mathsf{role}(f_i) = \mathsf{role}(f_j)$.
(v): In a literal $[f(g(x))] \approx x$, $\mathsf{role}(f) = \mathsf{Inv}(\mathsf{role}(g))$.
(vi): For each $[f_i(a)] \approx [b]$ a *witness* closure of the form $R(\langle a \rangle, \langle b \rangle) \vee D$ exists, with $\mathsf{role}(f_i) = R$, D does not contain functional terms or negative binary literals, and is contained in this closure.

Closure of $\mathcal{ALCHIQ}^-$-closures under Inferences. The following lemma is central to our work, since it implies, together with a bound on the number of $\mathcal{ALCHIQ}^-$-closures, termination of $\mathcal{BS}_{DL}$. The proof is by examining all inferences of $\mathcal{BS}_{DL}$ for all possible types of $\mathcal{ALCHIQ}^-$-closures.

Lemma 1. *Let $\Xi(KB) = N_0, \ldots, N_i \cup \{C\}$ be a $\mathcal{BS}_{DL}$-derivation, where C is the conclusion derived from premises in N_i. Then C is either an $\mathcal{ALCHIQ}^-$-closure or it is redundant in N_i.*

Termination and Complexity Analysis. Let $|KB|$ denote the size of KB with numbers coded in unary. It is straightforward to see that, given a knowledge base KB, the size of a set of non-redundant $\mathcal{ALCHIQ}^-$-closures over the vocabulary from $\Xi(KB)$ is exponentially bounded in $|KB|$: let r be the number of role names, a the number of atomic concept names, c the number of constants, f the number of Skolem function symbols occurring in $\Xi(KB)$, and v the maximal number of variables in a closure. Obviously, r, a, and c are linear in $|KB|$ and, for unary coding of numbers, f and v are also linear in $|KB|$. Thus we have at most $(f+1)^2(v+c)$ terms of depth at most 2, which, together with the possible marking, yields at most $t = 2(f+1)^2(v+c)$ terms in a closure. This yields at most $at + rt^2$ atoms, which, together with the equality literals, and allowing each atom to occur negatively, gives at most $\ell = 2(at + (r+1)t^2)$ literals in a closure. Each closure can contain an arbitrary subset of these literals, so the total number of closures is bounded by 2^ℓ. Thus we obtain an exponential bound on the size of the set of closures that $\mathcal{BS}_{DL}$ can derive. Each inference step can be carried out in exponential time, so, since $\mathcal{BS}_{DL}$ is a sound and complete refutation procedure [3], we have the following result:

Theorem 1 ([11]). *For an $\mathcal{ALCHIQ}^-$ knowledge base KB, saturating $\Xi(KB)$ by $\mathcal{BS}_{DL}$ with eager application of redundancy elimination rules decides satisfiability of KB and runs in time exponential in $|KB|$, for unary coding of numbers.*

3.2 Removing the Restriction to Very Simple Roles

For a $\mathcal{SHIQ}$ knowledge base KB containing number restrictions on roles which are not very simple, the saturation of $\Xi(KB)$ may contain closures whose structure corresponds to Table 3, but for which conditions (iii) – (vi) do not hold; we call such closures $\mathcal{ALCHIQ}$-closures. Let KB be the knowledge base containing axioms (1) – (9):

$$R \sqsubseteq T \Rightarrow \neg R(x,y) \vee T(x,y) \quad (1)$$
$$S \sqsubseteq T \Rightarrow \neg S(x,y) \vee T(x,y) \quad (2)$$
$$C \sqsubseteq \exists R.\top \Rightarrow \neg C(x) \vee R(x, f(x)) \quad (3)$$
$$\top \sqsubseteq \exists S^-.\top \Rightarrow S^-(x, g(x)) \quad (4)$$
$$\top \sqsubseteq \leq 1\, T \Rightarrow \neg T(x,y_1) \vee \neg T(x,y_2) \vee y_1 \approx y_2 \quad (5)$$
$$\exists S.\top \sqsubseteq D \Rightarrow \neg S(x,y) \vee D(x) \quad (6)$$
$$\exists R.\top \sqsubseteq \neg D \Rightarrow \neg R(x,y) \vee \neg D(x) \quad (7)$$
$$\top \sqsubseteq C \Rightarrow C(x) \quad (8)$$
$$\neg S^-(x,y) \vee S(y,x) \quad (9)$$

$$S([g(x)], x) \quad (10)$$
$$\neg C(x) \vee T(x, [f(x)]) \quad (11)$$
$$T([g(x)], x) \quad (12)$$
$$\neg C([g(x)]) \vee [f(g(x))] \approx x \quad (13)$$
$$\neg C([g(x)]) \vee R([g(x)], x) \quad (14)$$
$$D([g(x)]) \quad (15)$$
$$\neg D([g(x)]) \vee \neg C([g(x)]) \quad (16)$$
$$\neg C([g(x)]) \quad (17)$$
$$\square \quad (18)$$

Consider a saturation of $\Xi(KB)$ by $\mathcal{BS}_{DL}$ producing closures (10) – (13). For (13), Condition (v) is not satisfied: $\mathsf{role}(f) = R \neq \mathsf{Inv}(\mathsf{role}(g)) = \mathsf{Inv}(S^-) = S$. This is because in (5), a number restriction was stated on a role that is not very simple. Now (13) can be superposed into (3), resulting in (14), which is obviously not an $\mathcal{ALCHIQ}$-closure.

If KB were an $\mathcal{ALCHIQ}^-$ knowledge base, Condition (v) would hold, so we would be able to assume that a closure $R([g(x)], x)$ exists. This closure would subsume (14), so we would simply throw (14) away and continue saturation.

Since Condition (v) does not hold, a subsuming closure does not exist, so in order not to lose completeness, we must keep (14) and perform further inferences with it. This might cause termination problems: in general, (14) might be resolved with some closure of type 6 of the form $C([g(h(x))])$, producing a closure of the form $R([g(h(x))], [h(x)])$. The term depth in the binary literal is now two, and it may be used to derive closures with ever deeper terms. Thus, the set of derivable closures becomes infinite, and we cannot conclude that the saturation necessarily terminates.

A careful analysis reveals that various refinements of the ordering and the selection function will not help. Furthermore, the inference deriving (14) is necessary. Namely, KB is unsatisfiable, and the empty closure is derived through steps (15) – (18), which require (14).

4 Transformation by Decomposition

To solve the problems outlined in Subsection 3.2, we introduce *decomposition*, a transformation that can be applied to the conclusions of some $\mathcal{BS}$ inferences. It is a general technique not limited to description logics. In the following, for $\mathbf{x}$ a vector of distinct variables $x_1, \ldots, x_n$, and $\mathbf{t}$ a vector of (not necessarily distinct) terms $t_1, \ldots, t_n$, let $\{\mathbf{x} \mapsto \mathbf{t}\}$ denote the substitution $\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$, and let $Q([\mathbf{t}])$ denote $Q([t_1], \ldots, [t_n])$.

Definition 1. *Let $C \cdot \rho$ be a closure and N a set of closures. A* decomposition *of $C \cdot \rho$ w.r.t. N is a pair of closures $C_1 \cdot \rho \vee Q([\mathbf{t}])$ and $C_2 \cdot \theta \vee \neg Q(\mathbf{x})$ where $\mathbf{t}$ is a vector of n terms, $\mathbf{x}$ is a vector of n distinct variables, $n \geq 0$, satisfying these conditions: (*i*) $C = C_1 \cup C_2$, (*ii*) $\rho = \theta\{\mathbf{x} \mapsto \mathbf{t}\}$, (*iii*) $\mathbf{x}$ is exactly the set of free variables of $C_2\theta$, and (*iv*) if $C_2 \cdot \theta \vee \neg Q'(\mathbf{x}) \in N$, then $Q = Q'$, otherwise Q is a new predicate not occurring in N. The closure $C_2 \cdot \theta$ is called the* fixed *part, the closure $C_1 \cdot \rho$ is called the* variable *part and the predicate Q is called the* definition predicate*. An application of decomposition is often written as $C \cdot \rho \rightsquigarrow C_1 \cdot \rho \vee Q([\mathbf{t}]),\ C_2 \cdot \theta \vee \neg Q(\mathbf{x})$.*

Let ξ be a $\mathcal{BS}$ inference with a most general unifier σ on a literal $L_m \cdot \eta$ from a main premise $D_m \cdot \eta$ and with a side premise $D_s \cdot \eta$. The conclusion of ξ is eligible for decomposition *if, for each ground substitution τ such that $\xi\tau$ satisfies the constraints of $\mathcal{BS}$, we have $\neg Q(\mathbf{t})\tau \prec L_m\eta\sigma\tau$. With $\mathcal{BS}^+$ we denote the $\mathcal{BS}$ calculus where decomposition can be applied to conclusions of eligible inferences.*

The definition of eligibility is defined to cover the most general case. In the following, we use a simpler test: ξ is eligible for decomposition if $\neg Q(\mathbf{t}) \prec L_m\eta\sigma$, or a literal $L \in D_s$ exists such that $\neg Q(\mathbf{t}) \prec L\eta\sigma$. The latter is a sufficient approximation, since $L\eta\sigma\tau \prec L_m\eta\sigma\tau$ for each τ as in Definition 1.

E.g., consider superposition from $[f(g(x))] \approx [h(g(x))]$ into $C(x) \vee R(x, f(x))$ resulting in $D = C([g(x)]) \vee R([g(x)], [h(g(x))])$. The conclusion is not an $\mathcal{ALCHIQ}$-closure, so keeping it might lead to non-termination. D can be decomposed into $C([g(x)]) \vee Q_{R,f}([g(x)])$ and $\neg Q_{R,f}(x) \vee R(x, [h(x)])$, which are both $\mathcal{ALCHIQ}$-closures. The inference is eligible for decomposition if we ensure that $\neg Q_{R,f}(g(x)) \prec R(g(x), h(g(x)))$ (e.g. by using $R >_P Q_{R,f}$ in LPO).

The soundness and completeness proofs for $\mathcal{BS}^+$ are given in [10]; here we present the intuition behind these results. As shown by Lemma 2, decomposition is sound: it merely introduces a new name for $C_2 \cdot \theta$. Any model of $C \cdot \rho$ can be extended to a model of $C_1 \cdot \rho \vee Q([\mathbf{t}])$ and $C_2 \cdot \theta \vee \neg Q(\mathbf{x})$ by adjusting the interpretation of Q.

Lemma 2. *Let $N_0, \ldots, N_i$ be a $\mathcal{BS}^+$-derivation, and let I_0 be a model of N_0. Then for $i > 1$, N_i has a model I_i such that, if the inference deriving N_i from N_{i-1} involves a decomposition step as specified in Definition 1 introducing a new predicate Q, then $I_i = I_{i-1} \cup \{Q(\mathbf{s}) \mid \mathbf{s}$ is a vector of ground terms such that $C_2\theta\{\mathbf{x} \mapsto \mathbf{s}\}$ is true in $I_{i-1}\}$; otherwise $I_i = I_{i-1}$.*

The notion of *variable irreducibility* is a central concept in the completeness proof of basic superposition. Roughly speaking, a closure $C \cdot \rho\tau$ is a *variable* *irreducible ground instance* of $C \cdot \rho$ w.r.t. a ground and convergent rewrite system R if substitution positions in $C \cdot \rho\tau$ are not reducible by rewrite rules in R. We use this to prove completeness, by showing that decomposition is compatible with the usual notion of redundancy for $\mathcal{BS}$ [3, 14], as shown by Lemma 3. We do so in two steps. First, the eligibility criterion ensures that (*) ground instances of $C_1 \cdot \rho \vee Q([\mathbf{t}])$ and $C_2 \cdot \theta \vee \neg Q(\mathbf{x})$ are smaller than the corresponding ground instances of $D_m \cdot \eta$. Second, (**) for each variable irreducible ground instance $C \cdot \rho\tau$ of $C \cdot \rho$, there are variable irreducible ground instances E_1 and E_2 of $C_1 \cdot \rho \vee Q([\mathbf{t}])$ and $C_2 \cdot \theta \vee \neg Q(\mathbf{x})$, respectively, such that $\{E_1, E_2\} \models C \cdot \rho\tau$. Property (**) holds since the terms $\mathbf{t}$ are extracted from the substitution part of $C \cdot \rho$. Effectively, (**) means that decomposition does not lose "relevant" variable irreducible ground instances of $C \cdot \rho$ which are used in the proof. Actually, closures $C_1 \cdot \rho \vee Q([\mathbf{t}])$ and $C_2 \cdot \theta \vee \neg Q(\mathbf{x})$ can have "excessive" variable irreducible ground instances without a counterpart ground instance of $C \cdot \rho$. However, this is not a problem, since decomposition is sound.

Lemma 3. *Let ξ be a $\mathcal{BS}$ inference applied to premises from a closure set N resulting in a closure $C \cdot \rho$. If $C \cdot \rho$ can be decomposed into closures $C_1 \cdot \rho \vee Q([\mathbf{t}])$ and $C_2 \cdot \theta \vee \neg Q(\mathbf{x})$ which are both redundant in N, then the inference ξ is redundant in N.*

Soundness and compatibility with the notion of redundancy imply that $\mathcal{BS}^+$ is a sound and complete calculus, as shown by Theorem 2. Note that, to obtain the saturated set N, we can use any fair saturation strategy [2]. Furthermore, the decomposition rule can be applied an infinite number of times in a saturation, and it is even allowed to introduce an infinite number of definition predicates. In the latter case, we just need to ensure that the term ordering is well-founded.

Theorem 2. *For N_0 a set of closures of the form $C \cdot \{\}$, let N be a set of closures obtained by saturating N_0 under $\mathcal{BS}^+$. Then N_0 is satisfiable if and only if N does not contain the empty closure.*

For a resolution calculus $\mathcal{C}$ other than $\mathcal{BS}$, Lemma 2 applies as well. Furthermore, if $\mathcal{C}$ is compatible with the standard notion of redundancy [2], Lemma 3 holds as well: (*) holds for $\mathcal{C}$ identically, and (**) is needed only for $\mathcal{BS}$, due to a non-standard lifting strategy. Hence, decomposition can be combined with any such calculus.

Related Work. In [17] and [6] a similar rule for splitting without backtracking was considered, and in [18] a similar separation rule was introduced to decide fluted logic. Decomposition allows replacing complex terms with simpler ones, so it is different from splitting (which does not allow component clauses to contain common variables) or separation (which links component clauses only by literals without functional terms). Furthermore, by the eligibility criterion we make decomposition compatible with the standard notion of redundancy. Thus, decomposition becomes a full-fledged inference rule and can be applied an infinite number of times in a saturation. Finally, combining decomposition with basic superposition is not trivial, due to a non-standard approach to lifting.

5 Applications of Decomposition

To show the usefulness of decomposition, in this section, we use it to extend the algorithm from Section 3 to obtain three new decision procedures.

5.1 Deciding $\mathcal{ALCHIQ}$

Definition 2. *$\mathcal{BS}^{+}_{DL}$ is the modification of the $\mathcal{BS}_{DL}$ calculus where conclusions are decomposed, whenever possible, as follows, for an arbitrary term t:*

$$\begin{array}{rc} D \cdot \rho \vee R([t]\,,[f(t)]) \rightsquigarrow & D \cdot \rho \vee Q_{R,f}([t]) \\ & \neg Q_{R,f}(x) \vee R(x,[f(x)]) \\ D \cdot \rho \vee R([f(t)]\,,[t]) \rightsquigarrow & D \cdot \rho \vee Q_{\mathsf{Inv}(R),f}([t]) \\ & \neg Q_{\mathsf{Inv}(R),f}(x) \vee R([f(x)]\,,x) \end{array}$$

and where the precedence of the LPO is $f >_P c >_P p >_P Q_{S,f} >_P \top$, for any function symbol f, constant symbol c, non-definition predicate p and definition predicate $Q_{S,f}$.

For a (possibly inverse) role S and a function symbol f, the predicate $Q_{S,f}$ is unique. Since $R([f(x)]\,,x)$ and $\mathsf{Inv}(R)(x,[f(x)])$ are logically equivalent by the operator π, it is safe to use $Q_{\mathsf{Inv}(R),f}$ as the definition predicate for $R([f(x)]\,,x)$.

Inferences of $\mathcal{BS}_{DL}$, when applied to $\mathcal{ALCHIQ}$-closures, derive an $\mathcal{ALCHIQ}$-closure even if conditions (iii) – (vi) are not enforced. The only exception is the superposition from a closure of type 5 or 6 into a closure of type 3, but such closures are decomposed by $\mathcal{BS}^{+}_{DL}$ into two $\mathcal{ALCHIQ}$-closures; the inference is eligible for decomposition since $\neg Q_{R,f}(t) \prec R(t,g(t))$ (which is the maximal literal of the closure of type 3 after unification). Furthermore, $Q_{S,f}$ is unique for a pair of S and f, so the number of definition predicates is polynomially bounded. This allows us to derive an exponential bound on the number of $\mathcal{ALCHIQ}$-closures as in Theorem 1 and thus to obtain a decision procedure.

Theorem 3. *For an $\mathcal{ALCHIQ}$ knowledge base KB, saturation of $\Xi(KB)$ by $\mathcal{BS}^{+}_{DL}$ decides satisfiability of KB, and runs in time exponential in $|KB|$.*

5.2 Safe Role Expressions

A prominent limitation of $\mathcal{ALCHIQ}$ is the rather restricted form of role expressions that may occur in a knowledge base. This can be overcome by allowing for *safe* Boolean role expressions in TBox and ABox axioms. The resulting logic is called $\mathcal{ALCHIQ}b$, and can be viewed as the "union" of $\mathcal{ALCHIQ}$ and $\mathcal{ALCIQ}b$ [20]. Using safe expressions, it is possible to state negative or disjunctive knowledge regarding roles. Roughly speaking, safe role expressions are built using union, disjunction, and *relativized* negation of roles. This allows for statements such as $\forall x,y : \mathit{isParentOf}(x,y) \rightarrow \mathit{isMotherOf}(x,y) \vee \mathit{isFatherOf}(x,y)$, but does not allow for "fully negated" statements such as: $\forall x,y : \neg \mathit{isMotherOf}(x,y) \rightarrow \mathit{isFatherOf}(x,y)$. The safety restriction is needed for the algorithm to remain in ExpTime; namely, it is known that reasoning with non-safe role expressions is NExpTime-complete [13].

Definition 3. *A* role expression *is a finite expression built over the set of roles using the connectives* $\sqcup$, $\sqcap$ *and* $\neg$ *in the usual way. A role expression* E *is* safe *if each conjunction of the disjunctive normal form of* E *contains at least one non-negated atom. The description logic* $\mathcal{ALCHIQ}b$ *is obtained from* $\mathcal{ALCHIQ}$ *by allowing concepts* $\exists E.C$, $\forall E.C$, $\geq n\, E.C$ *and* $\leq n\, E.C$, *inclusion axioms* $E \sqsubseteq F$ *and ABox axioms* $E(a,b)$, *where* E *is a safe role expression, and* F *is any role expression. The semantics of* $\mathcal{ALCHIQ}b$ *is obtained by extending the operator* π *from Table 1 in the obvious way.*

We assume w.l.o.g. that all concepts in KB contain only atomic roles, since one can always replace a role expression with a new atomic role and add a corresponding role inclusion axiom. Hence, the only difference to the case of $\mathcal{ALCHIQ}$ logic is that KB contains axioms of the form $E \sqsubseteq F$, where E is a safe role expression. Such an axiom is equivalent to the first-order formula $\varphi = \forall x, y : \pi(\neg E \sqcup F)$. Assume that E is in disjunctive normal form; since it is safe, $\neg E$ is equivalent to a conjunction of disjuncts, where each disjunct contains at least one negated atom. Hence, translation of φ into first-order logic produces closures of the form $\Gamma = \neg R_1(x,y) \vee \ldots \vee \neg R_n(x,y) \vee S_1(x,y) \vee \ldots \vee S_m(x,y)$, where $n \geq 1, m \geq 0$. Computing the disjunctive normal form might introduce an exponential blow-up, so to compute $\Xi(KB)$ we use structural transformation, which runs in polynomial time, but also produces only closures of type Γ.

Next, we consider saturation of $\Xi(KB)$ using $\mathcal{BS}^+_{DL}$, and define $\mathcal{ALCHIQ}b$-closures to be of the form as specified in Table 3 where closures of type 2 are replaced with closures of the form Γ above. Since in $\mathcal{BS}^+_{DL}$ all negative binary literals are selected and a closure of type 3 always contains at least one negative binary literal, it can participate only in a hyperresolution inference with closures of type 3 or 4. Due to the occurs-check in unification, side premises are either all of type 3 or all of type 4. Hyperresolvents can have two forms, which are decomposed, whenever possible, as follows, for $\mathbf{S}(s,t) = S_1(s,t) \vee \ldots \vee S_m(s,t)$:

$$\mathbf{P}(x) \vee \mathbf{S}(x, [f(x)]) \rightsquigarrow \begin{array}{c} \neg Q_{S_i,f}(x) \vee S_i(x, [f(x)]) \text{ for } 1 \leq i \leq m \\ \mathbf{P}(x) \vee Q_{S_1,f}(x) \vee \ldots \vee Q_{S_m,f}(x) \end{array}$$
$$\mathbf{P}(x) \vee \mathbf{S}([f(x)], x) \rightsquigarrow \begin{array}{c} \neg Q_{\mathsf{Inv}(S_i),f}(x) \vee S_i([f(x)], x) \text{ for } 1 \leq i \leq m \\ \mathbf{P}(x) \vee Q_{\mathsf{Inv}(S_1),f}(x) \vee \ldots \vee Q_{\mathsf{Inv}(S_m),f}(x) \end{array}$$

Again, we decompose a non-$\mathcal{ALCHIQ}b$-closure into several $\mathcal{ALCHIQ}b$-closures. Hence, we may establish the bound on the size of the closure set as in Subsection 5.1, to obtain the following result:

Theorem 4. *For an* $\mathcal{ALCHIQ}b$ *knowledge base* KB, *saturation of* $\Xi(KB)$ *by* $\mathcal{BS}^+_{DL}$ *decides satisfiability of* KB *in time exponential in* $|KB|$.

5.3 Conjunctive Queries over $\mathcal{SHIQ}$ Knowledge Bases

Conjunctive queries [5] are a standard formalism for relational queries. Here, we present an algorithm for answering conjunctive queries over a $\mathcal{SHIQ}$ knowledge base KB. To eliminate transitivity axioms, we encode KB into an equisatisfiable

$\mathcal{ALCHIQ}$ knowledge base $\Omega(KB)$ [11]. Unfortunately, this transformation does not preserve entailment of ground non-simple roles. Hence, in the following we prohibit the use of non-simple roles in conjunctive queries (such roles can still be used in KB), and focus on $\mathcal{ALCHIQ}$.

Definition 4. *Let KB be an $\mathcal{ALCHIQ}$ knowledge base, and let $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ be sets of* distinguished *and* non-distinguished *variables, written as $\mathbf{x}$ and $\mathbf{y}$, respectively. A* conjunctive query *over KB, denoted as $Q(\mathbf{x}, \mathbf{y})$, is a conjunction of DL-atoms of the form $(\neg)A(s)$ or $R(s,t)$, where s and t are individuals from KB or distinguished or non-distinguished variables. The basic inferences for conjunctive queries are:*

- Query answering. *An* answer *of a query $Q(\mathbf{x}, \mathbf{y})$ w.r.t. KB is an assignment θ of individuals to distinguished variables, such that $\pi(KB) \models \exists \mathbf{y} : Q(\mathbf{x}\theta, \mathbf{y})$.*
- Query containment. *A query $Q_2(\mathbf{x}, \mathbf{y_2})$ is* contained *in a query $Q_1(\mathbf{x}, \mathbf{y_1})$ w.r.t. KB if $\pi(KB) \models \forall \mathbf{x} : [\exists \mathbf{y_2} : Q_2(\mathbf{x}, \mathbf{y_2}) \rightarrow \exists \mathbf{y_1} : Q_1(\mathbf{x}, \mathbf{y_1})]$.*

Query containment is reducible to query answering by well-known transformations of first-order formulae: $Q_2(\mathbf{x}, \mathbf{y_1})$ is contained in $Q_1(\mathbf{x}, \mathbf{y_2})$ w.r.t. KB if and only if $\mathbf{a}$ is an answer to $Q_1(\mathbf{x}, \mathbf{y_1})$ over $KB \cup \{Q_2(\mathbf{a}, \mathbf{b})\}$, where $\mathbf{a}$ and $\mathbf{b}$ are vectors of new distinct individuals, not occurring in $Q_1(\mathbf{x}, \mathbf{y_1})$, $Q_2(\mathbf{x}, \mathbf{y_2})$ and KB. Therefore, in the rest we only consider query answering.

Let KB be an $\mathcal{ALCHIQ}$ knowledge base. Obviously, for a conjunctive query $Q(\mathbf{x}, \mathbf{y})$, the assignment θ such that $\theta\mathbf{x} = \mathbf{a}$, is an answer of the query w.r.t. KB if and only if the set of closures $\Gamma' = \Xi(KB) \cup \{\neg Q(\mathbf{a}, \mathbf{y})\}$ is unsatisfiable, where $\neg Q(\mathbf{a}, \mathbf{y})$ is the closure obtained by negating each conjunct of $Q(\mathbf{a}, \mathbf{y})$.

A conjunctive query $Q(\mathbf{a}, \mathbf{y})$ is *weakly connected* if its literals cannot be decomposed into two subsets not sharing common variables. W.l.o.g. we assume that $Q(\mathbf{a}, \mathbf{y})$ is weakly connected: if $Q(\mathbf{a}, \mathbf{y})$ can be split into n weakly connected mutually variable-disjoint subqueries $Q_1(\mathbf{a_1}, \mathbf{y_1}), \ldots, Q_n(\mathbf{a_n}, \mathbf{y_n})$, then $\pi(KB) \models \bigwedge_{1 \leq i \leq n} \exists \mathbf{y_i} : Q_i(\mathbf{a_i}, \mathbf{y_i})$ if and only if $\pi(KB) \models \exists \mathbf{y_i} : Q_i(\mathbf{a_i}, \mathbf{y_i})$ for all $1 \leq i \leq n$. The subqueries $Q_i(\mathbf{a_i}, \mathbf{y_i})$ can be computed in polynomial time, so this assumption does not increase the complexity of reasoning.

A slight problem arises if $\neg Q(\mathbf{a}, \mathbf{y})$ contains unmarked constants: assuming that $a_i \in \mathbf{a_i}$ and $a'_i \in \mathbf{a'_i}$ for $i \in \{1, 2\}$, a superposition of $a_1 \approx a'_1 \vee a_2 \approx a'_2$ into $\neg Q_1(\mathbf{a_1}, \mathbf{y_1})$ and $\neg Q_2(\mathbf{a_2}, \mathbf{y_2})$ may produce a closure $\neg Q_1(\mathbf{a'_1}, \mathbf{y_1}) \vee \neg Q_2(\mathbf{a'_2}, \mathbf{y_2})$. Such an inference produces a conclusion with more variables than each of its premises, thus leading to non-termination. To prevent this, we apply the structural transformation to $\neg Q(\mathbf{a}, \mathbf{y})$ and replace Γ' with Γ, where for each $a \in \mathbf{a}$, $\mathcal{O}_a$ is a new predicate unique for a, x_a is a new variable unique for a, and $\mathbf{x_a}$ is the vector of variables obtained from $\mathbf{a}$ by replacing each $a \in \mathbf{a}$ with x_a:

$$\Gamma = \Xi(KB) \cup \{\neg Q(\mathbf{x_a}, \mathbf{y}) \vee \bigvee_{a \in \mathbf{a}} \neg \mathcal{O}_a(x_a)\} \cup \bigcup_{a \in \mathbf{a}} \{\mathcal{O}_a(a)\}$$

The sets Γ' and Γ are obviously equisatisfiable. In the rest we write $\neg \mathcal{O}_{\mathbf{a}}(\mathbf{x_a})$ for $\bigvee_{a \in \mathbf{a}} \neg \mathcal{O}_a(x_a)$. We now define the calculus for deciding satisfiability of Γ:

Definition 5. *$\mathcal{BS}^{+}_{CQ}$ is the extension of the $\mathcal{BS}^{+}_{DL}$ calculus, where the selection function is as follows: if a closure C contains a literal $\neg\mathcal{O}_a(x_a)$, then all such literals are selected; otherwise, all negative binary literals are selected. The precedence for LPO is $f >_P c >_P p >_P \mathcal{O}_a >_P Q_{R,f} >_P p_{a,b} >_P \top$. In addition to decomposition inferences from Definition 2, the following decompositions are performed whenever possible, where the t_i are of the form $f_{i,1}(\ldots f_{i,m}(x)\ldots)$:*

$$(\neg)A_1([t_1]) \vee \ldots \vee (\neg)A_n([t_n]) \rightsquigarrow \begin{array}{c} Q_{(\neg)A_1,t_1}(x) \vee \ldots \vee Q_{(\neg)A_n,t_n}(x) \\ \neg Q_{(\neg)A_i,t_i}(x) \vee (\neg)A_i([t_i]),\ 1 \leq i \leq n \end{array}$$

$$C \cdot \rho \vee \mathcal{O}_a(\langle b \rangle) \rightsquigarrow \begin{array}{c} C \cdot \rho \vee p_{a,b} \\ \neg p_{a,b} \vee \mathcal{O}_a(b) \end{array}$$

Definition 6. *The class of $\mathcal{CQ}$-closures w.r.t. a conjunctive query $Q(\mathbf{a},\mathbf{y})$ over an $\mathcal{ALCHIQ}$ knowledge base KB is the generalization of closures from Table 3 obtained as follows:*

- *Conditions (*iii*) – (*vi*) are dropped.*
- *Closure types 5 and 6 are replaced with a new type $5'$, which contains all closures C satisfying each of the following conditions:*
  1. *C contains only equality, unary or propositional literals.*
  2. *C contains only one variable x.*
  3. *The depth of a term in C is bounded by the number of literals of $Q(\mathbf{a},\mathbf{y})$.*
  4. *If C contains a term of the form $f(t)$, then all terms of the same depth in C are of the form $g(t)$, and all terms of smaller depth are (not necessarily proper) subterms of t.*
  5. *Only the outmost position of a term in C can be unmarked, i.e. each functional term is either of the form $[f(t)]$ or of the form $f([t])$.*
  6. *Equality and inequality literals in C can have the form $[f(t)] \circ [g(t)]$ or $[f(g(t))] \circ [t]$ for $\circ \in \{\approx, \not\approx\}$.*
- *Closure type 8 is modified to allow unary and (in)equality literals to contain unary terms whose depth is bounded by the number of literals in $Q(\mathbf{a},\mathbf{y})$; only outermost positions in a term can be unmarked; all (in)equality literals are of the form $[f(a)] \circ [b]$, $[f(t)] \circ [g(t)]$, $[f(g(t))] \circ [t]$ or $\langle a \rangle \circ \langle b \rangle$, for $\circ \in \{\approx, \not\approx\}$ and t a ground term; and a closure can contain propositional literals $(\neg)p_{a,b}$.*
- *A new* query closure *type contains closures of the form $\neg Q([\mathbf{a}],\mathbf{y}) \vee \mathbf{p}$, where $Q([\mathbf{a}],\mathbf{y})$ is weakly connected, it contains at least one binary literal and $\mathbf{p}$ is a possibly empty disjunction of propositional literals $\mathbf{p} = \bigvee (\neg)p_{a,b}$.*
- *A new* initial closure *type contains closures of the form $\neg\mathcal{O}_{\mathbf{a}}(\mathbf{x_a}) \vee \neg Q(\mathbf{x_a},\mathbf{y})$.*

We show the closure of $\mathcal{CQ}$-closures under $\mathcal{BS}^{+}_{CQ}$ in [10]. Roughly speaking, since all literals $\neg\mathcal{O}_a(x_a)$ are selected, the only possible inference for an initial closure is hyperresolution with $\neg p_{a,b} \vee \mathcal{O}_a(b)$ or $\mathcal{O}_a(a)$, generating a query closure with marked terms. Propositional symbols $p_{a,b}$ are used to decompose closures resulting from superposition into $\mathcal{O}_a(b)$; since such literals are smallest in any closure, they cannot participate in inferences with a closure of type $5'$.

Consider an inference with a closure $\neg Q([\mathbf{a}],\mathbf{y}) \vee \mathbf{p}$ such that $Q([\mathbf{a}],\mathbf{y})$ is weakly connected. Since all constants are marked, superposition into such a closure is not possible. The only possible inference is hyperresolution with side premises of type 3, 4 and 8 with a unifier σ. If $Q([\mathbf{a}], \mathbf{y})$ contains a constant or if some side premise is ground, then $Q([\mathbf{a}], \mathbf{y})\sigma$ is ground because $Q([\mathbf{a}], \mathbf{y})$ is weakly connected. Otherwise, since the query closure is weakly connected, the hyperresolution produces a closure of the form $\bigvee(\neg)A_i([t_i])$ with t_i of the form $f_{i,1}(\ldots f_{i,m}(x)\ldots)$. This closure does not satisfy condition 4 of $\mathcal{CQ}$-closures, so it is decomposed into several closures of type 5′; eligibility is ensured since $\neg Q_{(\neg)A_i,t_i}(x) \prec (\neg)A_i(t_i)$, and $(\neg)A_i(t_i)$ originates from some side premise $E_j\sigma$. All side premises contain at most one functional term of depth one, so the depth of functional terms in the conclusion is bounded by the length of the maximal path in $Q([\mathbf{a}], \mathbf{y})$, which is bounded by $|Q(\mathbf{a}, \mathbf{y})|$.

To build a term of the form $f_1(\ldots f_m(x)\ldots)$, one selects a subset of at most $|Q(\mathbf{a}, \mathbf{y})|$ function symbols; the number of such subsets is exponential in $|Q(\mathbf{a}, \mathbf{y})|$. This gives an exponential bound on the closure length, and a doubly exponential bound on the number of $\mathcal{CQ}$-closures, leading to the following result:

Theorem 5. *For a conjunctive query $Q(\mathbf{a}, \mathbf{y})$ over an $\mathcal{ALCHIQ}$ knowledge base KB, saturation of Γ by $\mathcal{BS}^+_{CQ}$ decides satisfiability of Γ in time doubly exponential in $|KB| + |Q(\mathbf{a}, \mathbf{y})|$.*

6 Conclusion

We have proposed *decomposition*, a general inference rule applicable to any resolution calculus compatible with the standard notion of redundancy. This rule transforms certain conclusions of the calculus at hand, and thus can be used to turn a resolution calculus into a decision procedure.

For three decidable fragments of first-order logic, we present three decision procedures obtained by combining basic superposition with decomposition, and by choosing an appropriate term ordering and selection function. More precisely, we obtain two new decision procedures for checking satisfiability of $\mathcal{SHIQ}$ and $\mathcal{ALCHIQ}b$ knowledge bases, and a procedure for answering conjunctive queries over $\mathcal{SHIQ}$ knowledge bases. The first two procedures are worst-case optimal, and we expect them to be suitable for implementation due to the vast experience in building saturation theorem provers. An implementation of these algorithms is under way, and we hope to soon be able to confirm our expectations.

In addition, we plan to extend the algorithm for $\mathcal{ALCHIQ}b$ to support arbitrary role expressions, and to find a way to handle transitivity directly within our calculus, to avoid the reduction and to allow transitive roles in queries.

Acknowledgements

This work was funded by the EU IST project DIP 507483. The presented results have been derived from Deliverable D1.4 - Mechanisms and Algorithms for Scalable Hybrid Reasoning.

References

1. F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
2. L. Bachmair and H. Ganzinger. Resolution Theorem Proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 19–99. Elsevier Science, 2001.
3. L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic Paramodulation. *Information and Computation*, 121(2):172–192, 1995.
4. D. Calvanese, G. De Giacomo, and M. Lenzerini. On the Decidability of Query Containment under Constraints. In *Proc. PODS 1998*, pages 149–158. ACM Press, 1998.
5. A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proc. STOC 1977*, pages 77–90. ACM Press, 1977.
6. H. de Nivelle. Splitting through new proposition symbols. In *Proc. LPAR 2001*, volume 2250 of *LNAI*, pages 172–185. Springer, 2001.
7. N. Dershowitz and D.A. Plaisted. Rewriting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 535–610. Elsevier Science, 2001.
8. C. Fermüller, A. Leitsch, U. Hustadt, and T. Tammet. Resolution Decision Procedures. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 1791–1849. Elsevier Science, 2001.
9. I. Horrocks, U. Sattler, and S. Tobies. Practical Reasoning for Very Expressive Description Logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.
10. U. Hustadt, B. Motik, and U. Sattler. Reasoning for Description Logics around $\mathcal{SHIQ}$ in a Resolution Framework. Technical Report 3-8-04/04, FZI, Germany, 2004. http://www.fzi.de/wim/publikationen.php?id=1172.
11. U. Hustadt, B. Motik, and U. Sattler. Reducing $\mathcal{SHIQ}^-$ Description Logic to Disjunctive Datalog Programs. In *Proc. KR 2004*, pages 152–162. AAAI Press, 2004.
12. W. H. Joyner Jr. Resolution Strategies as Decision Procedures. *J. ACM*, 23(3):398–417, 1976.
13. C. Lutz and U. Sattler. The Complexity of Reasoning with Boolean Modal Logics. In *Advances in Modal Logics*, volume 3. CSLI Publications, Stanford, 2001.
14. R. Nieuwenhuis and A. Rubio. Theorem Proving with Ordering and Equality Constrained Clauses. *J. Logic and Computation*, 19(4):312–351, 1995.
15. R. Nieuwenhuis and A. Rubio. Paramodulation-Based Theorem Proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 371–443. Elsevier Science, 2001.
16. D. A. Plaisted and S. Greenbaum. A Structure-preserving Clause Form Transformation. *J. Symbolic Logic and Computation*, 2(3):293–304, 1986.
17. A. Riazanov and A. Voronkov. Splitting Without Backtracking. In *Proc. IJCAI 2001*, pages 611–617. Morgan Kaufmann, 2001.
18. R. A. Schmidt and U. Hustadt. A Resolution Decision Procedure for Fluted Logic. In D. McAllester, editor, *Proc. CADE 2000*, pages 433–448. Springer, 2000.
19. S. Tessaris. *Questions and answers: reasoning and querying in Description Logic*. PhD thesis, University of Manchester, UK, 2001.
20. S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, Germany, 2001.
21. C. Weidenbach. Combining Superposition, Sorts and Splitting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 1965–2013. Elsevier Science, 2001.