Noname manuscript No.
(will be inserted by the editor)

A Superposition Calculus for Abductive Reasoning

M. Echenim and N. Peltier

the date of receipt and acceptance should be inserted later

E-mail: Mnacho.Echenim@imag.fr, Nicolas.Peltier@imag.fr

Keywords Equational First-Order Logic, Abduction, Superposition Calculus, Deductive-Completeness

Mathematics Subject Classification (2000) 03B35 · 68T15

CR Subject Classification F.3.1 · F.4.1 · I.2.3

Abstract We present a modification of the Superposition Calculus that is meant to generate consequences of sets of first-order axioms. This approach is proven to be sound and deductive-complete in the presence of redundancy elimination rules, provided the considered consequences are built on a given finite set of ground terms, represented by constant symbols. In contrast to other approaches, most existing results about the termination of the Superposition calculus can be carried over to our procedure. This ensures in particular that the calculus is terminating for many theories of interest to the SMT community.

1 Introduction

The verification of complex systems is generally based on proving the validity, or, dually, the satisfiability of a logical formula. A standard practice consists in translating the behavior of the system to be verified into a logical formula, and proving that the negation of the formula is unsatisfiable. These formulæ may be domain-specific, so that it is only necessary to test the satisfiability of the formula modulo some background theory, whence the name *Satisfiability Modulo Theories problems*, or *SMT problems*. If the formula is actually satisfiable, this means the system is not error-free, and any model can be viewed as a trace that generates an error. The models of a satisfiable formula can

Univ. Grenoble Alpes, F-38000 Grenoble, France · CNRS, LIG

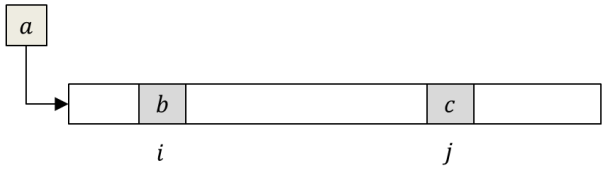

Fig. 1: Insertion into array a of element b at position i and element c at position j.

therefore help the designers of the system guess the origin of the errors and deduce how they can be corrected; this is the main reason for example why state-of-the-art SMT solvers feature automated model building tools (see for instance Caferra, Leitsch, and Peltier, 2004). However, this approach is not always satisfactory. First, there is the risk of an information overkill: indeed, the generated model may be very large and complex, and discovering the origin of the error may require a long and difficult analysis. Second, the model may be too specific, in the sense that it only corresponds to one particular execution of the system and that dismissing this single execution may not be sufficient to fix the system. Also, there are generally many interpretations on different domains that satisfy the formula. In order to understand where the error(s) may come from, it is generally necessary to analyze all of these models and to identify common patterns. This leaves the user with the burden of having to infer the general property that can rule out all the undesired behaviors. A more useful and informative solution would be to directly infer the missing axioms, or hypotheses, that can be added in order to ensure the unsatisfiability of the input formula. These axioms can be viewed as sufficient conditions ensuring that the system is valid. Such conditions must be *plausible* and *economical*: for instance, explanations that contradict the axioms of the considered theories are obviously irrelevant.

In this paper, we present what is, to the best of our knowledge, a novel approach to this debugging problem: we argue that rather than studying one or several models of a formula, more valuable information can be extracted from the properties that hold in *all* the models of the formula. For example, consider the theory of arrays, which is axiomatized as follows (as introduced by McCarthy, 1962):

$$\forall x, z, v.\ \mathrm{select}(\mathrm{store}(x, z, v), z) \simeq v, \tag{1}$$

$$\forall x, z, w, v.\ z \simeq w \vee \mathrm{select}(\mathrm{store}(x, z, v), w) \simeq \mathrm{select}(x, w). \tag{2}$$

These axioms state that if element v is inserted into array x at position z, then the resulting array contains v at position z, and the same elements as in x elsewhere. Assume that to verify that the order in which elements are inserted into a given array does not matter, the satisfiability of the following formula is tested (see also Figure 1):

$$\mathrm{select}(\mathrm{store}(\mathrm{store}(a, i, b), j, c), k) \not\simeq \mathrm{select}(\mathrm{store}(\mathrm{store}(a, j, c), i, b), k).$$

This formula asserts that there is a position k that holds different values in the array obtained from a by first inserting element b at position i and then element c at position j, and in the array obtained from a by first inserting element c at position j and then element b at position i. It turns out that this formula is actually satisfiable, which in this case means that some hypotheses are missing. State-of-the-art SMT solvers such as Yices (Dutertre and de Moura, 2006), veriT (Bouton et al, 2009) or Z3 (de Moura and Bjørner, 2008) can help find out what hypotheses are missing by outputting a model of the formula. In this case, Yices outputs

```
(= b 1) (= c 3) (= i 2) (= k 2) (= j 2)
```

and for this simple example, such a model may be sufficient to quickly understand where the error comes from. However, a simpler and more natural way to determine what hypotheses are missing would be to have a tool that, when fed the formula above, outputs $i \simeq j \wedge b \not\simeq c$, stating that the formula can only be true when elements b and c are distinct, and are inserted at the *same* position in array a. This information permits us to know immediately what additional hypotheses must be made for the formula to be unsatisfiable. In this example, there are two possible hypotheses that can be added: $i \not\simeq j$ or $b \simeq c$.

We investigate what information should be provided to the user and how it can be obtained, by distinguishing a set of ground terms on which additional hypotheses are allowed to be made. These terms may be represented by a particular set of constant symbols, called *abducible constants* or simply *abducibles*, and the problem boils down to determining what ground clauses containing only abducible constants are logically entailed by the formula under consideration, since the negation of any of these clauses can be viewed as a set of additional hypotheses that make the formula unsatisfiable. Indeed, by duality, computing implicants (or explanations) of a formula ϕ is equivalent to computing implicates (i.e., logical consequences) of $\neg\phi$. In order to compute such implicates, we devise a variant of the Superposition calculus (Bachmair and Ganzinger, 1994; Nieuwenhuis and Rubio, 2001) that is deductive-complete for the considered set of abducible constants, i.e., that can generate all the clauses built on abducible constants (using a finite set of predicate symbols including $\simeq$) that are logical consequences of the input clause set up to redundancy. Our procedure is defined by enriching the standard calculus with some new mechanisms allowing the assertion of relevant hypotheses during the proof search. These additional hypotheses are stored as constraints associated with the clauses and are propagated along the derivations. If the empty clause can be generated under a conjunction of hypotheses $\mathcal{X}$, then the conjunction of the original formula and $\mathcal{X}$ is unsatisfiable. An essential feature of this approach is that the conditions are not asserted arbitrarily or eagerly, using a generate-and-test approach (which would be inefficient): instead they are *discovered* on a need basis, either by considering residual equations of unification failures (for positive hypotheses) or by negating some of the literals occurring in the clauses (for negative hypotheses).

Related Work

The generation of implicants (or, by duality, of implicates) of logical formulæ has many applications in system verification and artificial intelligence, and this problem has been thoroughly investigated in the context of propositional logic. The earlier approaches use refinements of the resolution method (Tison, 1967; Kean and Tsiknis, 1990; De Kleer, 1992; Simon and Del Val, 2001), while more recent proposals use decomposition-based procedures (Jackson and Pais, 1990; Henocque, 2002; Matusiewicz et al, 2009, 2011). These methods mainly focus on the efficient representation of information, and develop compact ways of storing and manipulating huge sets of implicates.

In contrast, the approaches handling abductive reasoning in first-order or equational logic are very scarce. It is well-known that the Superposition calculus is not deductive-complete in general, for instance it cannot generate the clause $a \not\simeq b$ from the clause $f(a) \not\simeq f(b)$, although $f(a) \not\simeq f(b) \models a \not\simeq b$. Implicates can be generated automatically from sets of first-order clauses by using the resolution rule (Marquis, 1991). However, when dealing with equational clause sets, the addition of equality axioms leads to inefficiency and divergence in almost all but trivial cases. Knill, Cox, and Pietrzykowski (1992) use a proof technique called *surface resolution* for generating implicates of Horn clauses in equational logic. The proposed approach, based on a systematic flattening of the terms and on the application of the resolution principle with substitutivity axioms, is very general and has some nice theoretical properties, but it is also very inefficient. The search space is huge, because the systematic abstraction of every subterm destroys all ordering or unifiability constraints, and termination is very rare. Mayer and Pirri (1993) describe a tableaux-based (or, dually, a sequent-based) proof procedure for abductive reasoning. The intuitive idea is to apply the usual decomposition rules of propositional logic, and then compute the formulæ that force the closure of all open branches in the tableaux, thus yielding sufficient conditions ensuring unsatisfiability. The approach can be extended to first-order logic, by relying on reverse skolemization techniques in order to eliminate the Skolem symbols introduced inside the branches for handling existential quantifiers. Again, this approach is not well-suited for handling equality, and no termination results are presented. Tran, Ringeissen, Ranise, and Kirchner (2010) show that the Superposition calculus can be used to generate *positive* and *unit* implicates for some specific theories. This approach is closer to ours, since it is based on the Superposition calculus, hence handles equality in an efficient way; however it is more focused.

While the previous approaches rely on usual complete proof procedures for first-order logic, more recent work builds on the recent developments and progresses in the field of Satisfiability Modulo Theories by devising algorithms relying on theory-specific decision procedures. Sofronie-Stokkermans (2010, 2013) devises a technique for generating abductive explanations in local extensions of decidable theories. The approach reduces the considered problem to a formula in the basic theory by instantiating the axioms of the extension. Dillig, Dillig, McMillan, and Aiken (2012) generate an incomplete set of implicants of

formulæ interpreted in decidable theories by combining quantifier-elimination (for discarding useless variables) with model building tools (to construct sufficient conditions for satisfiability). In contrast to these approaches, our method is proof-theoretic, hence it is generic and self-sufficient. The drawback is that it requires the adaptation of usual theorem provers instead of using them as black boxes (see also Example 60 for a comparison of our method with the simplification technique devised by Dillig et al (2012)).

Wernhard (2013) proposes a method to derive abductive explanations from first-order logical programs, under several distinct non-classical semantics, using a reduction to second-order quantifier-elimination. Both the considered framework and the proposed techniques completely depart from our work.

Organization of the Paper

The rest of the paper is structured as follows. In Section 2 we review basic definitions and adapt standard results to our framework. In Section 3 the new Superposition calculus $\mathcal{SA}^{\prec}_{sel}$ is presented, and it is shown in Section 4 that it is deductive-complete for ground clauses built on the set of abducible constants. This first version of the calculus is however very inefficient, because of its poor handling of clauses that are built only on abducible terms and because the set of abducible implicates is huge. In Section 5 some refinements of the calculus are presented, aiming at more efficiency. First it is shown that $\mathcal{SA}^{\prec}_{sel}$ can be easily adapted to search only for some specific classes of implicates (e.g., positive implicates, or implicates of some bounded size), provided the class is closed under subsumption. Second, it is proven that inferences only involving abducible clauses can be discarded, yielding an *implicit* representation of the set of implicates. We then combine the two refinements in order to construct a two-step efficient algorithm to compute sets of prime implicates. The first step consists in using the second refinement of the calculus to generate an implicant ϕ of the entire set of abducible clauses, and the second step consists in using the first refinement to generate a set of prime implicates entailing the formula ϕ, thus yielding a concise representation of the set of implicates.

In Section 6, we show that most termination results holding for the usual Superposition calculus also apply to $\mathcal{SA}^{\prec}_{sel}$. The present paper is a thoroughly expanded and revised version of (Echenim and Peltier, 2012). See Section 5.2 for more details on the relationship of $\mathcal{SA}^{\prec}_{sel}$ with the calculus defined in (Echenim and Peltier, 2012).

2 Preliminaries

2.1 Basic Definitions

The set of *terms* is built as usual on a set of *function symbols* $\mathcal{F}$ including a set of *predicate symbols* $\mathcal{P}$, containing in particular a special constant $\top$,

and a set of *variables* $\mathcal{V}$. Every symbol $f \in \mathcal{F}$ is mapped to a unique *arity* $ar(f) \in \mathcal{N}$. The set $\mathcal{F}_n$ is the set of function symbols of arity n; an element of $\mathcal{F}_0$ is a *constant*. A term whose head is in $\mathcal{P}$ is *boolean*.

An *atom* (or *equation*) is an unordered pair of terms, written $t \simeq s$, where t and s are terms. A *literal* is either an atom or the negation of an atom (i.e., a *disequation*), written $t \not\simeq s$. For every literal l, we denote by l^{c} the complementary literal of l, which is defined as follows: $(t \simeq s)^{\mathrm{c}} \stackrel{\text{def}}{=} t \not\simeq s$ and $(t \not\simeq s)^{\mathrm{c}} \stackrel{\text{def}}{=} t \simeq s$. We use the notation $t \bowtie s$ to denote a literal of the form $t \simeq s$ or $t \not\simeq s$, and $t \not\bowtie s$ then denotes the complementary literal of $t \bowtie s$. As usual, a non-equational atom $p(\mathbf{t})$ where $p \in \mathcal{P}$ and $\mathbf{t}$ is a vector of terms is encoded as an equation $p(\mathbf{t}) \simeq \top$. For readability, such an equation is sometimes written $p(\mathbf{t})$, and $p(\mathbf{t}) \not\simeq \top$ can be written $\neg p(\mathbf{t})$. Predicate symbols are assumed to occur only at root position[1]

A *clause* is a finite multiset of literals, sometimes written as a disjunction. The empty clause is denoted by $\Box$. For technical reasons, we assume that the predicate symbols only occur in atoms of the form $t \simeq \top$, where $t \neq \top$: literals of the form $\top \not\simeq \top$ can be removed from the clauses, and the clauses containing a literal $\top \simeq \top$ can be dismissed; equations of the form $p(\mathbf{t}) \simeq q(\mathbf{s})$ with $p, q \in \mathcal{P} \setminus \{\top\}$ are forbidden[2].

For every clause $C = \{l_1, \ldots, l_n\}$, C^{c} denotes the set of unit clauses $\{\{l_i^{\mathrm{c}}\} \mid i \in [1, n]\}$ and for every set of unit clauses $S = \{\{l_i\} \mid i \in [1, n]\}$, S^{c} denotes the clause $\{l_1^{\mathrm{c}}, \ldots, l_n^{\mathrm{c}}\}$. Throughout the paper, we assume that $\prec$ denotes some fixed reduction ordering on terms (see, e.g., Baader and Nipkow, 1998) such that $\top \prec t$, for all terms $t \neq \top$, extended to literals and clauses as usual[3].

The set of variables occurring in an expression (term, atom, literal, clause) E is denoted by $\mathrm{var}(E)$. If $\mathrm{var}(E) = \emptyset$ then E is *ground*. A *substitution* is a function mapping variables to terms. For every term t and for every substitution σ, we denote by $t\sigma$ the term obtained from t by replacing every variable x by its image w.r.t. σ. The *domain* of a substitution is the set of variables x such that $x\sigma \neq x$. A substitution σ is *ground* if for every x in the domain of σ, $x\sigma$ is ground.

A *position* is a finite sequence of positive integers. A position p *occurs* in a term t if either $p = \varepsilon$ or if $t = f(t_1, \ldots, t_n)$, $p = i.q$ with $i \in [1, n]$ and q is a position in t_i. If p is a position in t, the terms $t|_p$ and $t[s]_p$ are defined as follows: $t|_\varepsilon \stackrel{\text{def}}{=} t$, $t[s]_\varepsilon \stackrel{\text{def}}{=} s$, $f(t_1, \ldots, t_n)|_{i.q} \stackrel{\text{def}}{=} (t_i)|_q$ and $f(t_1, \ldots, t_n)[s]_{i.q} \stackrel{\text{def}}{=} f(t_1, \ldots, t_{i-1}, t_i[s]_q, t_{i+1}, \ldots, t_n)$.

Given a set of constants E, a literal $t \bowtie s$ is *E-flat* if either $t, s \in \mathcal{V} \cup E$ or $t = p(t_1, \ldots, t_n)$, $s = \top$ and $t_1, \ldots, t_n \in \mathcal{V} \cup E$. A clause is *$E$-flat* if all its literals are E-flat. The set of E-flat clauses is denoted by $\mathfrak{C}_{flat}(E)$. A clause is *flat* if it is $\mathcal{F}_0$-flat.

[1] This condition is preserved by the Superposition rules in Section 3, provided literals of the form $p(\mathbf{t}) \bowtie \top$ are never selected in a clause instance in which $p(\mathbf{t})$ occurs at some not root position (note that these literals cannot be maximal).

[2] It is easy to check that the calculus introduced in Section 3 never generates such equations.

[3] The literals $t \simeq s$ and $t \not\simeq s$ are ordered as $\{\{t\}, \{s\}\}$ and $\{\{t, s\}\}$, respectively.

An *interpretation* is a congruence relation on ground terms. An interpretation I *validates* a clause C if for all ground substitutions σ of domain $\mathrm{var}(C)$ there exists $l \in C$ such that either $l = (t \simeq s)$ and $(l, s)\sigma \in I$, or $l = (t \not\simeq s)$ and $(l, s)\sigma \notin I$.

2.2 Abducible Constants and $\mathcal{A}$-Sets

In this section we introduce the notion of an $\mathcal{A}$-set, that provides a convenient way of representing partial interpretations defined on a particular set of constant symbols. Let $\mathcal{A} \subseteq \mathcal{F}_0$ be a set of constants, called the *abducible constants*. The set $\mathcal{A}$ is fixed by the user and contains all constants on which the abducible formulæ can be constructed. We assume that $f(\mathbf{t}) \succ a$, for all $a \in \mathcal{A}$ and $f \notin \mathcal{A}$, and that $q(t_1, \ldots, t_n) \succ p(a_1, \ldots, a_n)$ if $a_1, \ldots, a_n \in \mathcal{A}$, p, q are predicate symbols and $\exists i \in [1, n]\, t_i \succ a_1, \ldots, a_n$. A clause is *elementary* if it is $\mathcal{A}$-flat and contains no symbol in $\mathcal{P}$ (in other words, every literal is of the form $a \bowtie b$ with $a, b \in \mathcal{V} \cup \mathcal{A}$).

Definition 1 An *$\mathcal{A}$-set* is a set of $\mathcal{A}$-flat literals $\mathcal{X}$ satisfying the following properties.

- If $L \in \mathcal{X}$ and L is not ground then L is negative or of the form $p(t_1, \ldots, t_n) \simeq \top$ with $p \in \mathcal{P}$.
- If $\{L[a]_p, a \simeq b\} \subseteq \mathcal{X}$, where $a, b \neq \top$, then $L[b]_p \in \mathcal{X}$.
- $a \simeq a \in \mathcal{X}$, for all $a \in \mathcal{A}$.

An $\mathcal{A}$-set $\mathcal{X}$ is *positive* if it only contains positive literals, and *complete* if for every ground $\mathcal{A}$-flat atom A, $\mathcal{X}$ contains either A or $\neg A$.

Note that all elementary positive literals in $\mathcal{X}$ must be ground whereas negative or non elementary literals possibly contain variables. Informally, a satisfiable $\mathcal{A}$-set can be viewed as a partial interpretation on the constant symbols in $\mathcal{A}$. The positive elementary literals in $\mathcal{X}$ define an equivalence relation between elements on $\mathcal{A}$ and the negative elementary literals specify the equivalence classes that are known to be distinct. Literals of the form $p(t_1, \ldots, t_n) \bowtie \top$ specify the interpretation of predicate symbols on constants of $\mathcal{A}$. Variables correspond to unknown (or unspecified) constant symbols in $\mathcal{A}$. Complete $\mathcal{A}$-sets are total interpretations on $\mathcal{A}$.

This definition of $\mathcal{A}$-sets is given for theoretical purposes only: in practice, they can be more conveniently represented by a set of oriented equations of the form $\{a_i \simeq b_i \mid i \in [1, n]\}$, where $\forall i \in [1, n]\ a_i, b_i \in \mathcal{A}$, $a_i \succ b_i$ and $i \neq j \Rightarrow a_i \neq a_j$, together with a set of irreducible literals of the form $c \not\simeq d$ or $p(c_1, \ldots, c_n) \bowtie \top$, where $\forall i \in [1, n]$, $c, d, c_1, \ldots, c_n \neq a_i$. When convenient, we may represent an $\mathcal{A}$-set by a set $\mathcal{X}$ of equations and disequations, with the intended meaning that we are actually referring to the smallest $\mathcal{A}$-set that contains $\mathcal{X}$.

Example 2 Let $\mathcal{A} = \{a, b, c, d\}$ and $x \in \mathcal{V}$. Then the set

$$\mathcal{X} = \{a \simeq a,\ b \simeq b,\ c \simeq c,\ d \simeq d,\ e \simeq e,\ a \simeq b,\ c \not\simeq a,\ c \not\simeq b,\ d \not\simeq x\}$$

is an $\mathcal{A}$-set. Assuming an ordering such that $a \succ b \succ c \succ d$, it can be more concisely represented by $\{a \simeq b, c \not\simeq b, d \not\simeq x\}$. $\mathcal{X}$ defines a partial interpretation in which a, b are known to be equal and distinct from c, while d is distinct from some unspecified constant x (x can represent a, b, c or e – if x represents d then the set is unsatisfiable). The interpretation is only partial since it can be extended into a total interpretation that satisfies either $a \simeq d$ or $a \not\simeq d$.

Definition 3 For every $\mathcal{A}$-set $\mathcal{X}$ and for every expression (term, atom, literal, clause or clause set) E, we denote by $E_{\downarrow\mathcal{X}}$ the expression obtained from E by replacing every constant $a \in \mathcal{A}$ in E by the smallest (according to $\prec$) constant b in $\mathcal{A}$ such that $a \simeq b \in \mathcal{X}$. We write $t \sim_{\mathcal{A}}^{\mathcal{X}} s$ iff $t_{\downarrow\mathcal{X}} = s_{\downarrow\mathcal{X}}$ and $t \sim_{\mathcal{A}} s$ iff there exists an $\mathcal{A}$-set $\mathcal{X}$ such that $t \sim_{\mathcal{A}}^{\mathcal{X}} s$. This definition is extended to substitutions: we write $\sigma = \theta_{\downarrow\mathcal{X}}$ if $x\sigma = (x\theta)_{\downarrow\mathcal{X}}$ for all variables x, and $\sigma \sim_{\mathcal{A}}^{\mathcal{X}} \theta$ if for all $x \in \mathrm{dom}(\sigma) \cup \mathrm{dom}(\theta)$, $x\sigma \sim_{\mathcal{A}}^{\mathcal{X}} x\theta$.

Proposition 4 *Let C be a clause, σ be a substitution and $\mathcal{X}$ be an $\mathcal{A}$-set. If $(C\sigma)_{\downarrow\mathcal{X}}$ is $\mathcal{A}$-flat (resp. elementary), then so is C.*

Proof The contrapositive is obvious: if C is not $\mathcal{A}$-flat, then it contains a non-boolean term t that is not in $\mathcal{A} \cup \mathcal{V}$. But then, neither $t\sigma$ nor $t\sigma_{\downarrow\mathcal{X}}$ can be in $\mathcal{A} \cup \mathcal{V}$, hence $(C\sigma)_{\downarrow\mathcal{X}}$ cannot be $\mathcal{A}$-flat. The reasoning is similar for elementary clauses.

2.3 $\mathcal{A}$-Unification

$\mathcal{A}$-unification is an extension of unification that, given two terms t and s, aims at computing a substitution σ such that $t\sigma \sim_{\mathcal{A}} s\sigma$, meaning that $t\sigma$ and $s\sigma$ are equal up to a renaming of constants in $\mathcal{A}$. The set of necessary constant renamings is collected and stored in a positive $\mathcal{A}$-set. This set corresponds exactly to residual (non-solvable) equations obtained when applying standard unification algorithms.

Example 5 The terms $f(a, b)$ and $f(x, x)$ (with $a, b \in \mathcal{A}$) are not unifiable in the standard sense, but they are $\mathcal{A}$-unifiable. The substitution $\sigma : \{x \mapsto a\}$ is an $\mathcal{A}$-unifier of these two terms, together with the $\mathcal{A}$-set $\{a \simeq a,\ b \simeq b,\ a \simeq b\}$.

Definition 6 An *$\mathcal{A}$-substitution* is a pair $(\sigma, \mathcal{X})$ where σ is a substitution and $\mathcal{X}$ is a (ground) $\mathcal{A}$-set containing only equations between elements of $\mathcal{A}$. An $\mathcal{A}$-substitution $(\sigma, \mathcal{X})$ is an *$\mathcal{A}$-unifier* of an equation $t \simeq s$ iff $t\sigma \sim_{\mathcal{A}}^{\mathcal{X}} s\sigma$. Two terms admitting an $\mathcal{A}$-unifier are *$\mathcal{A}$-unifiable*.

Intuitively, if $(\sigma, \mathcal{X})$ is an $\mathcal{A}$-unifier of an equation $t \simeq s$, then the equations in $\mathcal{X}$ can be used to reduce t and s to terms that are unifiable in the standard sense.

Definition 7 An $\mathcal{A}$-substitution $(\sigma, \mathcal{X})$ is *more general* than an $\mathcal{A}$-substitution $(\sigma', \mathcal{X}')$, written $(\sigma, \mathcal{X}) \geq_{\mathcal{A}} (\sigma', \mathcal{X}')$, if the two following conditions hold:

- $\mathcal{X} \subseteq \mathcal{X}'$.
- There exists a (standard) substitution θ such that $\forall x \in \mathcal{V}$, $x\sigma' \sim_{\mathcal{A}}^{\mathcal{X}'} x\sigma\theta$.

We write $(\sigma, \mathcal{X}) \sim_{\mathcal{A}} (\sigma', \mathcal{X}')$ if $(\sigma, \mathcal{X}) \geq_{\mathcal{A}} (\sigma', \mathcal{X}')$ and $(\sigma', \mathcal{X}') \geq_{\mathcal{A}} (\sigma, \mathcal{X})$.

Example 8 Let $\mathcal{A} = \{a, b, c\}$, and consider the following substitutions and $\mathcal{A}$-sets:

$$\sigma = \{x \mapsto a,\ y \mapsto c,\ z \mapsto f(a, z')\} \text{ and } \mathcal{X} = \{a \simeq c\},$$
$$\sigma' = \{x \mapsto a,\ y \mapsto b,\ z \mapsto f(b, b),\} \text{ and } \mathcal{X}' = \{a \simeq b,\ b \simeq c\}.$$

By letting $\theta = \{z' \mapsto b\}$, it is simple to verify that $(\sigma, \mathcal{X}) \geq_{\mathcal{A}} (\sigma', \mathcal{X}')$. Note that $\mathcal{X}'$ implicitly contains $a \simeq c$ since as explained above, $\mathcal{X}'$ actually denotes the reflexive and transitive closure of $\{a \simeq b,\ b \simeq c\}$.

It is easy to check that for all terms t, s and for every $\mathcal{A}$-unifier σ of t and s, there exists a most general $\mathcal{A}$-unifier of t, s that is more general than σ. However, most general $\mathcal{A}$-unifiers are not unique, even modulo variable renamings. For example, the equation $f(g(a), g(b)) \simeq f(g(x), g(y))$ admits several most general unifiers, including $(\{x \to a, y \to b\}, \{a \simeq b\})$, $(\{x \to b, y \to a\}, \{a \simeq b\})$, ... which are of course all $\sim_{\mathcal{A}}$-equivalent. $\mathcal{A}$-unifiers are unique up to $\sim_{\mathcal{A}}$-equivalence, and they can be computed by a slight adaptation of the usual unification algorithm (see Appendix A for details).

3 The A-Superposition Calculus

In this section we define an extension of the standard Superposition calculus (Bachmair and Ganzinger, 1994; Nieuwenhuis and Rubio, 2001) with which it is possible to generate all $\mathcal{A}$-flat implicates of a considered clause set. The calculus handles constrained clauses, called $\mathcal{A}$-*clauses*, the constraint part of an $\mathcal{A}$-clause being an $\mathcal{A}$-set containing all the equations and disequations needed to derive the corresponding non-constraint part from the original clause set. Unification is replaced by $\mathcal{A}$-unification, and the $\mathcal{A}$-set of the generated $\mathcal{A}$-unifier is appended to the constraint of the conclusion of the rule. Also, an additional inference rule, called the $\mathcal{A}$-*Assertion* rule, is introduced in order to add disequations to the constraints. Finally, another rule, the $\mathcal{A}$-*Substitutivity* rule, handles constraints built on predicate symbols.

Definition 9 An $\mathcal{A}$-*clause* is a pair $[C \,|\, \mathcal{X}]$ where C is a clause and $\mathcal{X}$ is an $\mathcal{A}$-set. If $\mathcal{X} = \emptyset$, then we may write C instead of $[C \,|\, \emptyset]$.

In what follows, we first define the ordering and selection function the calculus is based upon before presenting the inference rules and redundancy criterion of the $\mathcal{A}$-Superposition calculus. We also prove that the calculus is sound.

3.1 Ordering and Selection Function

We begin by introducing some additional notation and terminology.

Definition 10 For all terms t, s, we write $t \succ_{\mathcal{A}} s$ if for every $\mathcal{A}$-set $\mathcal{X}$ and ground substitution σ, we have $t\sigma_{\downarrow\mathcal{X}} \succ s\sigma_{\downarrow\mathcal{X}}$. This ordering is extended to atoms, literals and clauses as it is done for $\prec$.

Intuitively, $t \succ_{\mathcal{A}} s$ means that t is always greater than s, regardless of the names of the constants in $\mathcal{A}$.

Example 11 If $a, b, c \in \mathcal{A}$ and $f(x) \succ a \succ b \succ c$, then we have $f(b) \succeq_{\mathcal{A}} a$, but $f(a) \not\succeq_{\mathcal{A}} f(b)$, since $f(a)_{\downarrow\{a\simeq c\}} = f(c) \prec f(b) = f(b)_{\downarrow\{a\simeq c\}}$.

Definition 12 A substitution σ is *$\mathcal{X}$-pure* if for all variables $x \in \mathrm{var}(\mathcal{X})$, $x\sigma$ is either a variable or a constant in $\mathcal{A}$.

Definition 13 A function *sel* is a *selection function* for an ordering $>$ iff *sel* maps every clause C to a set of literals in C such that $sel(C)$ either contains a negative literal or contains all literals that are $>$-maximal in C.

We consider a selection function *sel* for the ordering $\succeq_{\mathcal{A}}$, that satisfies the following assumption.

Assumption 14 *The function sel is stable under* $\mathcal{A}$-substitutions*, i.e., for every clause C, for every literal $l \in C$ and for every $\mathcal{A}$-substitution $(\eta, \mathcal{X})$, if $l\eta_{\downarrow\mathcal{X}} \in sel(C\eta_{\downarrow\mathcal{X}})$, then $l \in sel(C)$.*

3.2 Inference Rules

The calculus $\mathcal{SA}^{\prec}_{sel}$ is defined by the rules below (as usual, we assume that the premises share no variables). The standard Superposition calculus (denoted by $\mathcal{SP}^{\prec}_{sel}$) coincides with $\mathcal{SA}^{\prec}_{sel}$ if $\mathcal{A} = \emptyset$.

Remark 15 Following our convention, in all rules, if $\mathcal{X}$, $\mathcal{Y}$ are two $\mathcal{A}$-sets, then $\mathcal{X} \cup \mathcal{Y}$ does not denote the mere union of $\mathcal{X}$ and $\mathcal{Y}$, but rather the smallest $\mathcal{A}$-set containing both $\mathcal{X}$ and $\mathcal{Y}$ (it is obtained by transitive closure from the union of $\mathcal{X}$ and $\mathcal{Y}$). For example, if $\{a, b, c\} \subseteq \mathcal{A}$ with $a \succ b \succ c$, $\mathcal{X} = \{a \simeq a, b \simeq b, c \simeq c, a \simeq b\}$ and $\mathcal{Y} = \{a \simeq a, b \simeq b, c \simeq c, a \simeq c\}$, then $\mathcal{X} \cup \mathcal{Y}$ denotes the $\mathcal{A}$-set $\{a \simeq a, b \simeq b, c \simeq c, a \simeq b, a \simeq c, b \simeq c\}$. Similarly, if $\mathcal{X}$ is an $\mathcal{A}$-set and σ is an $\mathcal{X}$-pure substitution, then $\mathcal{X}\sigma$ denotes the smallest $\mathcal{A}$-set containing $\mathcal{X}\sigma$. For instance, if $\mathcal{X} = \{a \simeq a, b \simeq b, a \simeq b, x \not\simeq y\}$ and $\sigma = \{x \mapsto a\}$, then $\mathcal{X}\sigma = \{a \simeq a, b \simeq b, a \simeq b, a \not\simeq y, b \not\simeq y\}$.

$\mathcal{A}$-Superposition

$$\frac{[C \vee t \bowtie s \,|\, \mathcal{X}], \quad [D \vee u \simeq v \,|\, \mathcal{Y}]}{[C \vee D \vee t[v]_p \bowtie s \,|\, \mathcal{X} \cup \mathcal{Y} \cup \mathcal{E}]\sigma}$$

If $(\sigma, \mathcal{E})$ is an $(\mathcal{X} \cup \mathcal{Y})$-pure most general $\mathcal{A}$-unifier of u and $t|_p$, $v\sigma \not\succeq_{\mathcal{A}} u\sigma$, $s\sigma \not\succeq_{\mathcal{A}} t\sigma$, $(t \bowtie s)\sigma \in sel((C \vee t \bowtie s)\sigma)$, $(u \simeq v)\sigma \in sel((D \vee u \simeq v)\sigma)$ and if $t|_p$ is a variable then $t|_p$ occurs in $\mathcal{X}$.

We shall refer to the left and right premises of the inference rule as the *into* and *from* premises, respectively. The main difference with the usual Superposition rule (besides the replacement of $\succ$ by $\succ_{\mathcal{A}}$ and of unifiers by $\mathcal{A}$-unifiers) is that superposition into a variable is permitted, provided the considered variable occurs in the constraint part of the clause. The reason is that these do not actually represent variables in the usual sense, but rather placeholders for (unknown) constants (see also Example 49).

By definition of the calculus, variables can only occur in the constraints if the $\mathcal{A}$-Assertion rule (see below) is applied on a non-ground literal. This is the case because, by definition of $\mathcal{A}$-unification, the other rules add only ground equations into the constraints. We remark that, in the special case where the set of predicate symbols $\mathcal{P}$ is empty, a non-ground literal can be added to the constraints only if the considered clause is variable-eligible, i.e. contains a selected literal of the form $x \simeq t$, where $x \not\prec t$. This cannot happen if the clause set is variable-inactive[4] (Armando et al, 2009). However, there exist theories of interest that are not variable-inactive, such as the theory of arrays with axioms for constant arrays (e.g., $\forall x\, select(t, x) \simeq c$).

Note that the rule applies if t and u are of the form $p(\mathbf{t}) \bowtie \top$ and $p(\mathbf{s}) \simeq \top$ (with $p = \varepsilon$), in which case $t[v]_p \bowtie s$ is of the form $\top \bowtie \top$. If $\bowtie$ is $\simeq$ then the $\mathcal{A}$-clause is a tautology and can be deleted, and if $\bowtie$ is $\not\simeq$ then the literal $\top \not\simeq \top$ is deleted from the clause as explained previously. The rule is essentially equivalent to Ordered Resolution in this case (see for instance Leitsch, 1997).

$\mathcal{A}$-Reflection

$$\frac{[C \vee t \not\simeq s \,|\, \mathcal{X}]}{[C \,|\, \mathcal{X} \cup \mathcal{E}]\sigma}$$

If $(\sigma, \mathcal{E})$ is an $\mathcal{X}$-pure most general $\mathcal{A}$-unifier of t and s and $(t \not\simeq s)\sigma \in sel((C \vee t \not\simeq s)\sigma)$.

[4] By definition, a clause set S is *variable-inactive* iff the clauses that are deducible from S by the superposition calculus are not variable-eligible.

Equational $\mathcal{A}$-Factorization

$$\frac{[C \vee t \simeq s \vee u \simeq v \,|\, \mathcal{X}]}{[C \vee s \not\simeq v \vee t \simeq s \,|\, \mathcal{X} \cup \mathcal{E}]\sigma}$$

If $(\sigma, \mathcal{E})$ is an $\mathcal{X}$-pure most general $\mathcal{A}$-unifier of t and u, $s\sigma \not\preceq_{\mathcal{A}} t\sigma$, $v\sigma \not\preceq_{\mathcal{A}} u\sigma$ and $(t \simeq s)\sigma \in sel((C \vee t \simeq s \vee u \simeq v)\sigma)$.

For technical convenience, we assume that $s \not\simeq v$ is omitted in the conclusion if $s\sigma = v\sigma$.

$\mathcal{A}$-Assertion

$$\frac{[t \simeq s \vee C \,|\, \mathcal{X}]}{[C \,|\, \mathcal{X} \cup \{t \not\simeq s\}]}$$

If $t, s \in \mathcal{A} \cup \mathcal{V}$, $t \simeq s \in sel(t \simeq s \vee C)$ and $\mathcal{A} \neq \emptyset$.

$$\frac{[p(t_1, \ldots, t_n) \bowtie \top \vee C \,|\, \mathcal{X}]}{[C \,|\, \mathcal{X} \cup \{p(t_1, \ldots, t_n) \not\bowtie \top\}]}$$

If $t_1, \ldots, t_n \in \mathcal{A} \cup \mathcal{V}$, $p(t_1, \ldots, t_n) \bowtie \top \in sel(p(t_1, \ldots, t_n) \bowtie \top \vee C)$ and $\mathcal{A} \neq \emptyset$.

$\mathcal{A}$-Substitutivity Rule

$$\frac{[t_1 \simeq s_1 \vee C_1 \,|\, \mathcal{X}_1] \ldots [t_n \simeq s_n \vee C_n \,|\, \mathcal{X}_n]}{[p(t_1, \ldots, t_n) \bowtie \top \vee C_1 \vee \cdots \vee C_n \,|\, \{p(s_1, \ldots, s_n) \bowtie \top\} \cup \bigcup_{i=1}^{n} \mathcal{X}_i]}$$

If $p \in \mathcal{P}$ and for all $i = 1, \ldots, n$, $t_i, s_i \in \mathcal{A} \cup \mathcal{V}$.

The rule can be applied also by replacing some of the premises $[t_i \simeq s_i \vee C_i \,|\, \mathcal{X}_i]$ by variants of the Reflexivity axiom $x \simeq x$ (note that if all premises are of this form then the conclusion is a tautology).

3.3 Soundness

The interpretation of an $\mathcal{A}$-clause is defined as a logical implication:

Definition 16 An interpretation I *validates* an $\mathcal{A}$-clause $[C \,|\, \mathcal{X}]$ iff for every $\mathcal{X}$-pure ground substitution σ of domain $\mathrm{var}(C) \cup \mathrm{var}(\mathcal{X})$, either $I \not\models \mathcal{X}\sigma$ or $I \models C\sigma$. If $I \models [C \,|\, \mathcal{X}]$ for all interpretations I, then $[C \,|\, \mathcal{X}]$ is a *tautology*.

Note that in particular, an $\mathcal{A}$-clause $[C \,|\, \mathcal{X}]$ is a tautology if $\mathcal{X}$ is unsatisfiable, if $C_{\downarrow \mathcal{X}}$ contains two complementary literals or a literal of the form $t \simeq t$, or if a literal in C occurs in $\mathcal{X}$.

Theorem 17 *Let S be a set of $\mathcal{A}$-clauses. If C is generated from S by one of the rules of $\mathcal{SA}^{\prec}_{sel}$ then $S \models C$.*

Proof It suffices to prove that all the rules are sound, i.e., that the conclusion of the rule is a logical consequence of the premises. This is due to the fact that if $(\sigma, \mathcal{E})$ is an $\mathcal{A}$-unifier of $t \simeq s$, then the $\mathcal{A}$-clause $[t\sigma \simeq s\sigma \,|\, \mathcal{E}]$ is valid in all interpretations. Then the proof follows by a straightforward inspection of the rules, as in the usual case.

3.4 Redundancy

We adapt the standard redundancy criterion to $\mathcal{A}$-clauses. We begin by defining quasi-positive clauses, for which restricted redundancy criteria are necessary.

Definition 18 An $\mathcal{A}$-clause is *quasi-positive* if the only negative literals occurring in it are of the form $p(\mathbf{t}) \not\simeq \top$, where $p \in \mathcal{P}$.

Definition 19 An $\mathcal{A}$-clause $[C \,|\, \mathcal{X}]$ is *$\mathcal{A}$-redundant* in a set of $\mathcal{A}$-clauses S if either $[C \,|\, \mathcal{X}]$ is a tautology, or for every ground substitution θ of the variables in $[C \,|\, \mathcal{X}]$ such that $\mathcal{X}\theta$ is a satisfiable $\mathcal{A}$-set, one of the following conditions hold.

- There exists an $\mathcal{A}$-clause $[D \,|\, \mathcal{Y}]$ and a substitution σ such that $D\sigma \subseteq C\theta$ and $\mathcal{Y}\sigma \subseteq \mathcal{X}\theta$.
- Either $\mathcal{A} = \emptyset$ or $C\theta$ is not both $\mathcal{A}$-flat and quasi-positive, and there exist $\mathcal{A}$-clauses $[D_i \,|\, \mathcal{Y}_i]$ and substitutions σ_i $(1 \leq i \leq n)$, such that:
  - $\mathcal{Y}_i\sigma_i \subseteq \mathcal{X}\theta$ for all $i = 1, \ldots, n$,
  - $\mathcal{X}\theta, D_1\sigma_1, \ldots, D_n\sigma_n \models C\theta$,
  - $C\theta \succeq_{\mathcal{A}} D_1\sigma_1, \ldots, D_n\sigma_n$.

When applied to standard clauses (with $\mathcal{A} = \emptyset$), this notion of redundancy coincides with the usual criterion (see for instance Bachmair and Ganzinger, 1994; Nieuwenhuis and Rubio, 2001). It is easy to check that the standard redundancy detection rules such as subsumption, tautology deletion or equational simplification, are particular cases of this redundancy criterion. Note that the second item in Definition 19 is similar to the usual redundancy criterion of the Superposition calculus (see, e.g, Bachmair and Ganzinger, 1994), with the following differences: (i) the entire constraint part of the considered $\mathcal{A}$-clause may be used to infer the clausal part, disregarding any ordering condition, (ii) the condition only applies to clauses that are not both $\mathcal{A}$-flat and quasi-positive. For the clauses that are $\mathcal{A}$-flat and quasi-positive, redundancy testing is limited to tautology deletion and subsumption (this is necessary to ensure completeness, see Remark 26).

Example 20 Let $\mathcal{A} = \{a, b, c\}$. The $\mathcal{A}$-clause $[a \not\simeq c \vee b \not\simeq c \vee f(x) \simeq d \,|\, a \not\simeq b]$ is $\mathcal{A}$-redundant in any set S, since for all ground substitutions θ, $a \not\simeq b \models (a \not\simeq c \vee b \not\simeq c \vee f(x) \simeq d)\theta$.

The $\mathcal{A}$-clause $[f(a, b) \simeq c \vee g(a) \simeq d \,|\, a \not\simeq b]$ is $\mathcal{A}$-redundant in $\{f(a, x) \simeq c \vee a \simeq b\}$. Indeed, let $\sigma = \{x \mapsto b\}$, then $a \not\simeq b, f(a, x)\sigma \simeq c \vee a \simeq b \models f(a, b) \simeq c \vee g(a) \simeq d$ and $f(a, b) \simeq c \vee g(a) \simeq d \succeq_{\mathcal{A}} f(a, x)\sigma \simeq c \vee a \simeq b$.

Definition 21 A set S is $\mathcal{SA}^{\prec}_{sel}$-*saturated* if every $\mathcal{A}$-clause that can be derived from $\mathcal{A}$-clauses in S by a rule in $\mathcal{SA}^{\prec}_{sel}$ is redundant in S.

3.5 Examples

We provide simple application examples to illustrate the way the calculus can be used to generate implicates.

Example 22 Let $S = \{g(f(x)) \simeq d, f(a) \simeq a, g(b) \simeq b, d \simeq c\}$. Assume that $\mathcal{A} = \{a, b, c\}$. By applying the $\mathcal{A}$-Superposition rule on the terms $f(x)$ and $f(a)$, we derive the clause $g(a) \simeq d$ (note that this application of the rule is equivalent to the usual one). Then the $\mathcal{A}$-Superposition rule can be applied again on the terms $g(a)$ and $g(b)$. The unification yields the constraints $a \simeq b$, hence the following $\mathcal{A}$-clause is derived: $[b \simeq d \,|\, a \simeq b]$. The Assertion rule cannot be applied on $b \simeq d$, since this literal is not $\mathcal{A}$-flat. Instead, the application of the $\mathcal{A}$-Superposition rule on the term d (note that we must have $d \succ b, c$ since $d \notin \mathcal{A}$ and $b, c \in \mathcal{A}$) yields: $[b \simeq c \mid a \simeq b]$. Finally, the Assertion rule can be applied on $b \simeq c$ since this literal is $\mathcal{A}$-flat, thus generating $[\Box \mid b \not\simeq c \wedge a \simeq b]$. This $\mathcal{A}$-clause is equivalent to the clause $b \simeq c \vee a \not\simeq b$, and we have $S \models b \simeq c \vee a \not\simeq b$.

The second example involves predicate symbols.

Example 23 We consider two functions f and g such that f and $y \mapsto g(x, y)$ are increasing, together with abducible constants a, b, i and j. The aim is to determine under which conditions the property $f(g(a, i)) \leq f(g(b, j))$ holds. The problem is formalized as follows (where $t \leq s$ stands for $(t \leq s) \simeq \top$ and $\leq \in \mathcal{P}$, $x, y, u \in \mathcal{V}$): $S = \{x \not\leq y \vee f(x) \leq f(y), x \not\leq y \vee g(u, x) \leq g(u, y), f(g(a, i)) \not\leq f(g(b, j))\}$. For conciseness, the axioms corresponding to $\leq$ (e.g., transitivity) are omitted since they play no role in our context.

The Superposition rule applies on the first and last clauses, yielding $g(a, i) \not\leq g(b, j)$. Then the rule applies again from the latter clause into the second one, and it generates: $[i \not\leq j \,|\, \{a \simeq b\}]$. Finally the $\mathcal{A}$-Assertion rule yields the $\mathcal{A}$-clause: $[\Box \,|\, \{i \leq j, a \simeq b\}]$, meaning that the desired property is fulfilled if $i \leq j$ and $a \simeq b$ hold.

The $\mathcal{A}$-Assertion rule is necessary to guarantee deductive completeness, as shown in the example below.

Example 24 Consider the (satisfiable) clause set: $S \stackrel{\text{def}}{=} \{y \simeq x \vee f(x, x, y) \simeq a, f(a, b, c) \not\simeq a\}$, where $\mathcal{A} \stackrel{\text{def}}{=} \{a, b, c\}$ and x, y are variables. It is simple to verify that $S \models a \not\simeq b \vee c \simeq a$, and the calculus is designed to generate from S a clause of the form $[\Box \,|\, \mathcal{X}]$, where $\neg\mathcal{X} \equiv a \not\simeq b \vee c \simeq a$. In order to generate such a clause, it is clear that one has to unify $f(x, x, y)$ and $f(a, b, c)$, since the unification of $f(a, b, c)$ and a leads to an immediate failure, so that the Reflection rule is not applicable. This is feasible only if the condition $a \simeq b$ is

added to the constraints of the obtained clause, yielding a constrained clause of the form: $[c \simeq a \,|\, a \simeq b]$. The literal $c \simeq a$ can only be deleted using the $\mathcal{A}$-Assertion rule, appending the disequation $c \not\simeq a$ to the constraints, thus obtaining the required $\mathcal{A}$-clause: $\{[\Box \,|\, \{a \simeq b, c \not\simeq a\}]$.

The last example shows that the $\mathcal{A}$-Substitutivity rule is also needed for completeness.

Example 25 Consider the clause set: $S \stackrel{\text{def}}{=} \{a \simeq b\}$. It is clear that $S \models p(a) \simeq \top \vee p(b) \not\simeq \top$ for any predicate symbol p of arity 1, but $[\Box \,|\, \{p(a) \not\simeq \top, p(b) \simeq \top\}]$ cannot be generated without the help of the $\mathcal{A}$-Substitutivity rule. The above implicate is indeed obtained as follows: The $\mathcal{A}$-Substitutivity rule applies from $a \simeq b$, yielding $[p(a) \simeq \top \,|\, p(b) \simeq \top]$, and the desired result is obtained by applying the $\mathcal{A}$-Assertion rule. Note that the equation $p(a) \simeq p(b)$ does not need to be inferred in our context since predicate symbols are allowed only in atoms of the form $t \simeq \top$. Considering implicates built on arbitrary function symbols (with nested applications) would lead to divergence since, e.g., an infinite number of clauses of the form $f^n(a) \simeq f^n(b)$ (with $n \in \mathcal{N}$) could be derived from the above clause.

Remark 26 The previous example also shows the importance of the restriction on the redundancy criterion. Indeed, if the criterion is relaxed by removing the condition "$C\theta$ is not $\mathcal{A}$-flat and quasi-positive" in the second item of Definition 19, then the $\mathcal{A}$-clause $[p(a) \simeq \top \,|\, p(b) \simeq \top]$ is redundant in S, because $a \simeq b \prec p(a) \simeq \top$ and $a \simeq b, p(b) \simeq \top \models p(a) \simeq \top$. Thus, no non-redundant inferences apply on S and the implicate $p(a) \simeq \top \vee p(b) \not\simeq \top$ cannot be generated.

4 Deductive Completeness

We show in this section that $\mathcal{SA}^{\prec}_{sel}$ is deductive-complete for the clauses in $\mathfrak{C}_{flat}(\mathcal{A})$. More precisely, we prove that for any $\mathcal{SA}^{\prec}_{sel}$-saturated set S and clause $C \in \mathfrak{C}_{flat}(\mathcal{A})$, if $S \models C$ then S contains an $\mathcal{A}$-clause of the form $[\Box \,|\, \mathcal{Y}]$ where $\mathcal{Y}^{\mathrm{c}} \models C$. The result is obtained in the following way. Given such a set S and clause C, we consider the smallest $\mathcal{A}$-set $\mathcal{X}$ that contains C^{c}, and construct a set of standard ground clauses $\Phi(S, \mathcal{X})$ such that:

- $\Phi(S, \mathcal{X})$ contains all ground instances of clauses in S, as well as a set of unit clauses equivalent to $\mathcal{X} \equiv C^{\mathrm{c}}$.
- $\Phi(S, \mathcal{X})$ is saturated under a slightly adapted version of the Superposition calculus which is refutationally complete.

Since $S \cup C^{\mathrm{c}}$ is unsatisfiable and the considered calculus is refutationally complete, these two properties together will entail that $\Phi(S, \mathcal{X})$ contains the empty clause. Finally, we show that this is possible only if S contains an $\mathcal{A}$-clause of the required form.

First, we formally define the notions of $\mathcal{A}$-implicates and prime $\mathcal{A}$-implicates.

Definition 27 Let S be a set of $\mathcal{A}$-clauses. A clause C is an *$\mathcal{A}$-implicate* of S if it satisfies the following conditions.

- C is $\mathcal{A}$-flat and ground.
- C is not a tautology.
- $S \models C$.

C is a *prime $\mathcal{A}$-implicate* of S if, moreover, $C \models D$ holds for every $\mathcal{A}$-implicate D of S such that $D \models C$. We denote by $I_{\mathcal{A}}(S)$ the set of $\mathcal{A}$-implicates of S.

Definition 28 We denote by $\mathcal{C}_{\mathcal{A}}(S)$ the set of clauses of the form $(\mathcal{X}\sigma)^{\mathrm{c}}$, where $[\Box \,|\, \mathcal{X}] \in S$ and σ maps each variable x in $\mathcal{X}$ to some constant symbol $a \in \mathcal{A}$ in such a way that $\mathcal{X}\sigma$ is satisfiable[5]. We write $S \sqsubseteq S'$ if for every clause $C' \in S'$, there exists $C \in S$ such that $C \models C'$.

Our goal is to prove that $\mathcal{C}_{\mathcal{A}}(S) \sqsubseteq I_{\mathcal{A}}(S)$ when S is $\mathcal{SA}^{\prec}_{sel}$-saturated, i.e., that every prime implicate of S occurs in $\mathcal{C}_{\mathcal{A}}(S)$ (up to equivalence).

4.1 Definition of $\Phi(S, \mathcal{X})$

Let α and β be two arbitrarily chosen function symbols not occurring in S, where $ar(\alpha) = 1$, $\beta \in \mathcal{P}$ and $ar(\beta) = 0$. We assume that $\forall a \in \mathcal{A}, \beta \succ \alpha(a)$ and that $\forall g \notin \mathcal{A} \cup \{\alpha\}, g(\mathbf{t}) \succ \beta$.

For every clause C and clause set S, $sup(C, S)$ denotes the set of clauses inductively defined as follows.

- $C \in sup(C, S)$.
- If $D \in sup(C, S)$ and D' is obtained by applying the standard Superposition rule into D from a positive and elementary clause in S, then $D' \in sup(C, S)$.

A clause set S is *non-redundant* iff for every clause $C \in S$, C is not redundant in $S \backslash \{C\}$. For every clause set S, it is easy to obtain a non-redundant subset of S that is equivalent to S by first removing equivalent clauses and then deleting every clause C that is redundant in $S \setminus \{C\}$.

We define the set of standard ground clauses $\Phi(S, \mathcal{X})$ as well as a selection function sel_{Φ} as follows.

Definition 29 Let S be a set of $\mathcal{A}$-clauses and let $\mathcal{X}$ be an $\mathcal{A}$-set. We denote by $\Phi(S, \mathcal{X})$ the set

$$\Phi(S, \mathcal{X}) \stackrel{\text{def}}{=} \Phi_1(S, \mathcal{X}) \uplus \Phi_2(S, \mathcal{X}) \uplus \Phi_3(S, \mathcal{X}) \uplus \Phi_4(S, \mathcal{X}) \uplus \Phi_5(S, \mathcal{X}),$$

where for $i = 1, \ldots, 5$, $\Phi_i(S, \mathcal{X})$ is defined as follows:

1. $\Phi_1(S, \mathcal{X})$ is the set of clauses of the form $D\sigma_{\downarrow \mathcal{X}} \vee C'$, where $[D \,|\, \mathcal{Y}] \in S$, σ is a ground substitution of domain $\mathrm{var}(D)$ such that $\mathcal{Y}\sigma \subseteq \mathcal{X}$ and $x\sigma_{\downarrow \mathcal{X}} = x\sigma$ for all $x \in \mathrm{var}(D)$, and C' is defined as follows:

[5] In other words, σ is such that for every $u \not\simeq v \in \mathcal{X}$, $u\sigma \neq v\sigma$, and for all literals $l, m \in \mathcal{X}$, we have $l\sigma \neq (m\sigma)^{\mathrm{c}}$.

- $C' \stackrel{\text{def}}{=} \Box$ if $D\sigma$ is $\mathcal{A}$-flat and quasi-positive;
- $C' \stackrel{\text{def}}{=} (\beta \not\simeq \top)$ otherwise.

The selection function sel_Φ is defined on $\Phi_1(S, \mathcal{X})$ as follows: $sel_\Phi(D\sigma_{\downarrow\mathcal{X}} \vee C')$ contains all literals $l_{\downarrow\mathcal{X}}$ such that $l \in sel(D\sigma)$ and one of the following holds:

- l is negative,
- $sel(D\sigma)$ is positive and $l_{\downarrow\mathcal{X}}$ is $\succ$-maximal in $D\sigma_{\downarrow\mathcal{X}} \vee C'$.

2. $\Phi_2(S, \mathcal{X})$ is the set of unit clauses of the form $c \simeq c_{\downarrow\mathcal{X}}$, where $c \in \mathcal{A}$ and $c \neq c_{\downarrow\mathcal{X}}$. The selection function is defined on $\Phi_2(S, \mathcal{X})$ by: $sel_\Phi(c \simeq c_{\downarrow\mathcal{X}}) \stackrel{\text{def}}{=} \{c \simeq c_{\downarrow\mathcal{X}}\}$.
3. $\Phi_3(S, \mathcal{X})$ is the set of non-redundant clauses in

$$\bigcup_{a \not\simeq b \in \mathcal{X}} sup(\alpha(a_{\downarrow\mathcal{X}}) \not\simeq \alpha(b_{\downarrow\mathcal{X}}), \Phi_1(S, \mathcal{X})),$$

and for all $C \in \Phi_3(S, \mathcal{X})$, $sel_\Phi(C)$ contains all negative literals in C.
4. $\Phi_4(S, \mathcal{X})$ is the set of non-redundant clauses in

$$\bigcup_{l \in \mathcal{X},\, l = p(a_1, \ldots, a_n) \bowtie \top} sup(l_{\downarrow\mathcal{X}}, \Phi_1(S, \mathcal{X})),$$

and for all $C \in \Phi_3(S, \mathcal{X})$, $sel_\Phi(C)$ contains all literals of the form $t \bowtie \top$ in C.
5. $\Phi_5(S, \mathcal{X}) = \{\beta \simeq \top\} \cup \{\alpha(u) \not\simeq \alpha(v) \vee u \simeq v \mid u, v \in \mathcal{A}, u = u_{\downarrow\mathcal{X}}, v = v_{\downarrow\mathcal{X}}, u \neq v\}$. We let $sel_\Phi(\beta \simeq \top) \stackrel{\text{def}}{=} \{\beta \simeq \top\}$, and $sel_\Phi(\alpha(u) \not\simeq \alpha(v) \vee u \simeq v) \stackrel{\text{def}}{=} \{\alpha(u) \not\simeq \alpha(v)\}$.

It is easy to verify that the sets $\Phi_i(S, \mathcal{X})$ with $i = 1, \ldots, 5$ are disjoint. The *type* of a clause $C \in \Phi(S, \mathcal{X})$ is the number i such that $C \in \Phi_i(S, \mathcal{X})$.

Example 30 Let $\mathcal{A} = \{a, b, c, d, e\}$, and $\mathcal{X}$ be the reflexive-transitive closure of $\{a \simeq b, c \simeq d, b \not\simeq e\}$, where $a \succ b \succ c \succ d \succ e$. Consider the set of clauses

$$S = \{f(a) \simeq c \vee a \not\simeq b,\ b \simeq c,\ c \simeq d,\ [g(x, y) \simeq f(d) \,|\, y \not\simeq e],\ [f(x) \simeq x \,|\, a \simeq c]\}.$$

Then $\Phi(S, \mathcal{X})$ is decomposed as follows:

$\Phi_1(S, \mathcal{X})$: This set consists of $f(b) \simeq d \vee a \not\simeq b \vee \beta \not\simeq \top$, $b \simeq d$, $d \simeq d$ and $g(t, b) \simeq f(d) \vee \beta \not\simeq \top$, where t ranges over the set of all ground terms. The constants a and c occurring in S are respectively replaced by $b = a_{\downarrow\mathcal{X}}$ and $d = c_{\downarrow\mathcal{X}}$ in $\Phi_1(S, \mathcal{X})$. The $\mathcal{A}$-clause $[f(x) \simeq x \,|\, a \simeq c]$ generates no clauses in $\Phi_1(S, \mathcal{X})$, since $(a \simeq c) \notin \mathcal{X}$.
$\Phi_2(S, \mathcal{X})$: $\{a \simeq b, c \simeq d\}$.
$\Phi_3(S, \mathcal{X})$: $\{\alpha(b) \not\simeq \alpha(e), \alpha(d) \not\simeq \alpha(e)\}$. The first clause is constructed from $(b \not\simeq e) \in \mathcal{X}$, the second one is generated by Superposition into $\alpha(b) \not\simeq \alpha(e)$ from the clause $b \simeq d$ above.
$\Phi_4(S, \mathcal{X})$: $\emptyset$. There is no predicate symbols other than $\simeq$.

$\Phi_5(S, \mathcal{X})$: This set consists of the following clauses:

$$\{\beta \simeq \top,\ \alpha(b) \not\simeq \alpha(d) \vee b \simeq d,\ \alpha(b) \not\simeq \alpha(e) \vee b \simeq e,\ \alpha(d) \not\simeq \alpha(e) \vee d \simeq e\}\,.$$

Remark 31 The addition of α is irrelevant from a semantic point of view, since by construction, $\alpha(a) \not\simeq \alpha(b)$ if and only if $a \not\simeq b$ for all $a, b \in \mathcal{A}$; it is thus possible to replace all atoms of the form $\alpha(x) \not\simeq \alpha(y)$ by $x \not\simeq y$. However, this technical trick ensures that all the clauses of type 3 are strictly greater than all elementary $\mathcal{A}$-flat clauses in $\Phi(S, \mathcal{X})$, which plays a crucial rôle in the proof of Lemma 36. Similarly, the addition of the literal $\beta \not\simeq \top$ does not affect the semantics of the clause set (since by definition $\beta \simeq \top$ occurs in this set), but ensures that all clauses of type 1 that are not quasi-positive are strictly greater than all clauses of types 2 or 3.

Proposition 32 *For all sets of clauses S and $\mathcal{A}$-sets $\mathcal{X}$, sel_Φ is a selection function for the ordering $\succ$.*

Proof We must check that for every clause $C \in \Phi(S, \mathcal{X})$, $sel_\Phi(C)$ contains either a negative literal in C or all $\succ$-maximal literals in C (see Definition 29 for the notations). This is immediate for clauses of type 2 and 5, since $sel_\Phi(C) = C$. For clauses of type 3, we observe that C necessarily contains a negative literal, obtained from the literal $\alpha(a_{\downarrow\mathcal{X}}) \not\simeq \alpha(b_{\downarrow\mathcal{X}})$ by Superposition. Similarly, all $\mathcal{A}$-clauses of type 4 contains a (unique) literal of the form $p(a_1, \ldots, a_n) \bowtie \top$, that is necessarily maximal. Now assume that C is a clause of type 1, i.e., that $C = D\sigma_{\downarrow\mathcal{X}} \vee D''$ for some $[D \,|\, \mathcal{Y}]$ in S and $D'' \subseteq \{\beta \not\simeq \top\}$. If we suppose that $sel_\Phi(C)$ contains no negative literal, then the same must hold for $sel(D\sigma)$, thus $sel(D\sigma)$ necessarily contains all $\succeq_{\mathcal{A}}$-maximal literals in $D\sigma$. Furthermore, by definition of $\succeq_{\mathcal{A}}$, for all $m \in D\sigma$, if $m_{\downarrow\mathcal{X}}$ is $\succ$-maximal in $D\sigma_{\downarrow\mathcal{X}}$, then m is $\succeq_{\mathcal{A}}$-maximal in $D\sigma$, which entails that $sel_\Phi(C)$ contains all $\succ$-maximal literals in C (note that if $D'' \neq \Box$ then $D\sigma$ is not elementary, since otherwise $D\sigma$ would contain a negative $\succeq_{\mathcal{A}}$-maximal literal and thus $sel(D\sigma)$ would contain a negative literal, hence $D\sigma \succ D''$).

Proposition 33 *Let S_{init} be a set of standard clauses and let S be a set of clauses generated from S_{init} by $\mathcal{SA}^{\prec}_{sel}$. Then $\Phi(S, \mathcal{X}) \models S_{init} \equiv S$.*

Proof Let $S' \stackrel{\text{def}}{=} \Phi(S, \mathcal{X})$ and consider the set of standard clauses S_{cl} occurring in S, i.e., $S_{cl} \stackrel{\text{def}}{=} \{C \mid [C \,|\, \emptyset] \in S\}$. Since $\mathcal{SA}^{\prec}_{sel}$ is sound, $S_{init} \models S$. Furthermore, if a standard clause is $\mathcal{A}$-redundant in a set of $\mathcal{A}$-clauses, then it is also redundant w.r.t. the standard clauses in this set, by definition of the redundancy criterion. Thus $S_{cl} \equiv S \equiv S_{init}$.

By construction, S' contains all the clauses that can be obtained from ground instances of clauses in S_{cl}, by replacing every constant a by $a_{\downarrow\mathcal{X}}$ and possibly adding literals of the form $\beta \not\simeq \top$. Since S' contains all atoms of the form $a \simeq a_{\downarrow\mathcal{X}}$ where $a \neq a_{\downarrow\mathcal{X}}$ as well as the atom $\beta \simeq \top$, we deduce that $S' \models S_{cl}$, and that $S' \models S_{init} \equiv S$.

4.2 Saturatedness of $\Phi(S, \mathcal{X})$

The next lemma states that $\Phi(S, \mathcal{X})$ is saturated w.r.t. a slight restriction of the usual Superposition calculus. We shall also use a refined version of the redundancy criterion.

Definition 34 A set of ground clauses S is *weakly saturated* w.r.t. an inference rule in $\mathcal{SP}^{\prec}_{sel_\Phi}$ if every application of the rule on a set of premises $\{C_1, \ldots, C_n\} \subseteq S$ (with $n = 1, 2$) yields a clause C such that there exists a set $\{D_1, \ldots, D_m\} \subseteq S$ with $\forall i \in [1, m]$, $D_i \prec \max_\prec(\{C_1, \ldots, C_n\})$ and $\{D_1, \ldots, D_m\} \models C$.

In contrast to the notion of saturatedness (see Definition 21), the clauses D_i are compared with the maximal parent of C and not with C itself. Since a clause is always smaller than its parents, this entails that every saturated set is weakly saturated, but the converse does not hold.

Lemma 35 *Let S be a set of ground clauses that is weakly saturated w.r.t. all rules in $\mathcal{SP}^{\prec}_{sel_\Phi}$. The set S is satisfiable iff it does not contain $\Box$.*

Proof See (Bachmair and Ganzinger, 1994) or (Nieuwenhuis and Rubio, 2001, Theorem 4.8).

Lemma 36 below is the main technical result that is used to prove the completeness of $\mathcal{SA}^{\prec}_{sel}$.

Lemma 36 *Let S be an $\mathcal{SA}^{\prec}_{sel}$-saturated set of $\mathcal{A}$-clauses and let $\mathcal{X}$ be a ground and satisfiable $\mathcal{A}$-set. The set $\Phi(S, \mathcal{X})$ is weakly saturated under all inference rules in $\mathcal{SP}^{\prec}_{sel_\Phi}$, except for Equational Factorization on positive $\mathcal{A}$-flat clauses.*

Before giving the proof of Lemma 36 (on pages 24–31), we need to introduce some additional terminology and establish some intermediate results. This part is quite technical and can be skipped at a first reading

Definition 37 Let S be a set of $\mathcal{A}$-clauses and $\mathcal{X}$ be an $\mathcal{A}$-set. Let $u \bowtie v \vee C \vee D$ be a clause of type 1 in $\Phi(S, \mathcal{X})$, where $\bowtie \in \{\simeq, \not\simeq\}$ and $D \subseteq \{\beta \not\simeq \top\}$, and consider an $\mathcal{A}$-clause $[u' \bowtie v' \vee C' \,|\, \mathcal{Y}] \in S$ and a substitution σ such that[6] $(u'\sigma)_{\downarrow\mathcal{X}} = u$, $(v'\sigma)_{\downarrow\mathcal{X}} = v$, $(C'\sigma)_{\downarrow\mathcal{X}} = C$ and $\mathcal{Y}\sigma \subseteq \mathcal{X}$. The term occurrence u is *superposable* in $u \bowtie v \vee C$ if u' occurs in $\mathcal{Y}$ whenever it is a variable.

Proposition 38 *Let S be a set of $\mathcal{A}$-clauses and $\mathcal{X}$ be an $\mathcal{A}$-set. Let C be a clause of type 1 in $\Phi(S, \mathcal{X})$ and a, b be constants in $\mathcal{A}$ such that $a_{\downarrow\mathcal{X}} = a$ and $b_{\downarrow\mathcal{X}} = b$. Let P be a set of non-superposable occurrences of a in C. Then there exists a set P' of occurrences of a in C that contains P, and a clause D in $\Phi(S, \mathcal{X})$ such that D is obtained from C by replacing all occurrences of a in P' by b.*

[6] If several terms u' satisfying the above conditions exist then one of them is chosen arbitrarily.

Proof By definition, there exists an $\mathcal{A}$-clause $[C' \,|\, \mathcal{Y}] \in S$ and a substitution σ such that $C = C'\sigma_{\downarrow\mathcal{X}} \vee C''$, $C'' \subseteq \{\beta \not\simeq \top\}$ and $\mathcal{Y}\sigma \subseteq \mathcal{X}$. Since P is a set of non-superposable occurrences in C, the subterms of C' at the positions in P are variables $x_1, \ldots, x_n$ not occurring in $\mathcal{Y}$.

Consider the substitution θ coinciding with σ, except that $\forall i \in [1, n]$, $x_i\theta \stackrel{\text{def}}{=} b$. Since the variables x_i $(1 \leq i \leq n)$ do not occur in $\mathcal{Y}$, θ and σ coincide on $\mathcal{Y}$, hence $\mathcal{Y}\theta \subseteq \mathcal{X}$. This means that $\Phi(S, \mathcal{X})$ must contain the clause of type 1 $C'\theta_{\downarrow\mathcal{X}} \vee C''$ (note that C'' is not affected because $C'\sigma$ is $\mathcal{A}$-flat and positive exactly when $C'\theta$ satisfies the same property). By definition, since $a_{\downarrow\mathcal{X}} = a$ and $b_{\downarrow\mathcal{X}} = b$, $C'\theta_{\downarrow\mathcal{X}} \vee C''$ is therefore obtained from $C = C'\sigma_{\downarrow\mathcal{X}} \vee C''$ by replacing some occurrences of a by b, and in particular, all the occurrences in P are replaced.

Note that P' can be a strict superset of P: for example, if $S = \{x \simeq c \vee x \simeq d\}$, then $a \simeq c \vee a \simeq d \in \Phi(S, \emptyset)$, position 1.1 is not superposable in $a \simeq c \vee a \simeq d$, and it is clear that $b \simeq c \vee b \simeq d \in \Phi(S, \emptyset)$ but $b \simeq c \vee a \simeq c \notin \Phi(S, \emptyset)$.

Proposition 39 *Let $[C \,|\, \mathcal{X}]$ be a ground $\mathcal{A}$-clause; assume that $\mathcal{X}$ is satisfiable and that C is $\mathcal{A}$-flat and quasi-positive. Then $[C \,|\, \mathcal{X}]$ is a tautology if and only if $C_{\downarrow\mathcal{X}}$ is either a tautology or contains a literal that also occurs in $\mathcal{X}$. In particular, if C is elementary and positive then $[C \,|\, \mathcal{X}]$ is a tautology exactly when $C_{\downarrow\mathcal{X}}$ is a tautology.*

Proof Assume that $C_{\downarrow\mathcal{X}}$ is not a tautology and contains no literal in $\mathcal{X}$. Let I be the interpretation such that $\forall a, b \in \mathcal{A}$, $I \models a \simeq b$ iff $a_{\downarrow\mathcal{X}} = b_{\downarrow\mathcal{X}}$, and for all $a_1, \ldots, a_n$ where $a_{i\downarrow\mathcal{X}} = a_i$, $I \models p(a_1, \ldots, a_n) \simeq \top$ iff either $p(a_1, \ldots, a_n) \simeq \top \in \mathcal{X}$ or $p(a_1, \ldots, a_n) \not\simeq \top \in C_{\downarrow\mathcal{X}}$. Note that I is well-defined, since $\mathcal{X}$ and $C_{\downarrow\mathcal{X}}$ share no literals and neither of them contains complementary literals. By definition, I validates all positive literals in $\mathcal{X}$. If $a \not\simeq b \in \mathcal{X}$ and $I \not\models a \not\simeq b$, then $a_{\downarrow\mathcal{X}} = b_{\downarrow\mathcal{X}}$, hence $\mathcal{X} \models a \simeq b$, which means that $\mathcal{X}$ is unsatisfiable, and this contradicts the hypothesis of the lemma. Similarly, if $p(a_1, \ldots, a_n) \not\simeq \top \in \mathcal{X}$ and $I \models p(a_1, \ldots, a_n) \simeq \top$ then since $\mathcal{X}$ is satisfiable, $p(a_1, \ldots, a_n) \not\simeq \top$ must occur in $C_{\downarrow\mathcal{X}}$, which contradicts the hypothesis that $\mathcal{X}$ and $C_{\downarrow\mathcal{X}}$ share no literals. Therefore, $I \models \mathcal{X}$. Now consider a literal $l \in C$. Since C is $\mathcal{A}$-flat and quasi-positive, l is of the form $a \simeq b$ or $p(a_1, \ldots, a_n) \bowtie \top$. If l is of the form $a \simeq b$ and $a_{\downarrow\mathcal{X}} = b_{\downarrow\mathcal{X}}$ then $C_{\downarrow\mathcal{X}}$ is a tautology, and this is impossible by hypothesis. Thus $a_{\downarrow\mathcal{X}} \neq b_{\downarrow\mathcal{X}}$ and $I \not\models a \simeq b$. Now assume that l is of the form $p(a_1, \ldots, a_n) \bowtie \top$ where $\bowtie \in \{\simeq, \not\simeq\}$ and that $I \models l$. Let $m \stackrel{\text{def}}{=} p(a_{1\downarrow\mathcal{X}}, \ldots, a_{n\downarrow\mathcal{X}}) \bowtie \top$. Since $I \models l, \mathcal{X}$, it is clear that $I \models m$. If $\bowtie$ is $\not\simeq$ this implies by definition of I that $m \notin C_{\downarrow\mathcal{X}}$, a contradiction. If $\bowtie$ is $\simeq$, then either $m \in \mathcal{X}$ or $m^{\mathrm{c}} \in C_{\downarrow\mathcal{X}}$. In the first case m occurs in both $\mathcal{X}$ and $C_{\downarrow\mathcal{X}}$, and in the second case, both m and m^{c} occur in $C_{\downarrow\mathcal{X}}$ which is a tautology; thus we get a contradiction in both cases. Therefore, I is a counter-model of $[C \,|\, \mathcal{X}]$.

The converse is straightforward.

Note that the previous property does not hold if C is not quasi-positive; for example, $[a \not\simeq b \,|\, a \not\simeq b]$ is a tautology but the unit clause $(a \not\simeq b)_{\downarrow\{a \not\simeq b\}} = a \not\simeq b$ is not.

Lemma 40 *Let S be an $\mathcal{SA}^{\prec}_{sel}$-saturated set of $\mathcal{A}$-clauses and $\mathcal{X}$ be an $\mathcal{A}$-set. For $i = 1, 2$, let $u_i \simeq v_i \vee C_i$ be an $\mathcal{A}$-flat clause of type 1 in $\Phi(S, \mathcal{X})$, and assume that $u_i \neq v_i$. If the following conditions hold:*

- *$u_1 = u_2 \neq \top$,*
- *u_1 is superposable in $u_1 \simeq v_1 \vee C_1$,*
- *for $i = 1, 2$, $u_i \simeq v_i \in sel_\Phi(u_i \simeq v_i \vee C_i)$,*
- *$v_1 \simeq v_2 \vee C_1 \vee C_2$ is not a tautology,*

then there is a clause of type 1 in $\Phi(S, \mathcal{X})$ that is contained in $v_1 \simeq v_2 \vee C_1 \vee C_2$.

Proof For $i = 1, 2$, since $u_i \simeq v_i \vee C_i$ is of type 1, there exists an $\mathcal{A}$-clause $[t_i \simeq s_i \vee D_i \,|\, \mathcal{Y}_i] \in S$ and a substitution σ_i such that $\mathcal{Y}_i\sigma_i \subseteq \mathcal{X}$, $D_i\sigma_{i\downarrow\mathcal{X}} = C_i$, $t_i\sigma_{i\downarrow\mathcal{X}} = u_i$ and $s_i\sigma_{i\downarrow\mathcal{X}} = v_i$.

Let $\sigma = \sigma_1\sigma_2$. Since $u_1 = u_2$, we have $t_1\sigma = t_1\sigma_1 \sim^{\mathcal{X}}_{\mathcal{A}} t_2\sigma_2 = t_2\sigma$, hence $(\sigma, \mathcal{X})$ is an $\mathcal{A}$-unifier of $t_1 \simeq t_2$. Let $(\eta, \mathcal{Z})$ be a most general $\mathcal{A}$-unifier of $t_1 \simeq t_2$, then $\mathcal{Z} \subseteq \mathcal{X}$, and there exists a ground substitution σ' such that $\forall x$, $x\eta\sigma' \sim^{\mathcal{X}}_{\mathcal{A}} x\sigma$. Now, $[(t_i \simeq s_i)\eta\sigma']_{\downarrow\mathcal{X}} = u_i \simeq v_i$, which is selected in $u_i \simeq v_i \vee C_i$, and since sel is stable under $\mathcal{A}$-substitutions, $(t_i \simeq s_i)\eta \in sel((t_i \simeq s_i \vee D_i)\eta)$.

By hypothesis $v_1, v_2 \in \mathcal{A} \cup \mathcal{V}$, hence $s_1, s_2 \in \mathcal{A} \cup \mathcal{V}$. By definition of $\succ_{\mathcal{A}}$, this implies that $s_i\eta \not\succeq_{\mathcal{A}} t_i\eta$: indeed, s_1, s_2 can be replaced by the minimal constant $\top$, either by instantiation or by rewriting of constants in $\mathcal{A}$. Note also that $s_i\eta \neq t_i\eta$ since otherwise we would have $u_i = v_i$, which contradicts the hypotheses of the lemma.

Since u_1 is superposable, either t_1 is not a variable or t_1 occurs in $\mathcal{Y}_1$, hence by definition of $\mathcal{SA}^{\prec}_{sel}$, the $\mathcal{A}$-Superposition from $[t_2 \simeq s_2 \vee D_2 \,|\, \mathcal{Y}_2]$ into $[t_1 \simeq s_1 \vee D_1 \,|\, \mathcal{Y}_1]$ upon the terms t_1 and t_2 generates $[(s_1 \simeq s_2 \vee D_1 \vee D_2)\eta \,|\, \mathcal{Y}_1\eta \cup \mathcal{Y}_2\eta \cup \mathcal{Z}]$. Now, the $\mathcal{A}$-clause $[(s_1 \simeq s_2 \vee D_1 \vee D_2)\eta\sigma' \,|\, \mathcal{Y}_1\eta\sigma' \cup \mathcal{Y}_2\eta\sigma' \cup \mathcal{Z}]$ must be $\mathcal{A}$-redundant in S, because S is $\mathcal{SA}^{\prec}_{sel}$-saturated. This clause cannot be a tautology; indeed, for $i = 1, 2$, since $u_i \simeq v_i \in sel_\Phi(u_i \simeq v_i \vee C_i)$ and $u_i \simeq v_i \vee C_i \in \mathfrak{C}_{flat}(\mathcal{A})$, C_i must be positive by definition of the selection function sel_Φ, and cannot contain a symbol in $\mathcal{P}$ (otherwise the literal containing this symbol would be strictly greater than $u_i \simeq v_i$). By hypothesis, $v_1 \simeq v_2 \vee C_1 \vee C_2$ is not a tautology and since C_1, C_2 are positive and elementary, we deduce by Proposition 39 that $[(s_1 \simeq s_2 \vee D_1 \vee D_2)\eta\sigma' \mid \mathcal{Y}_1\eta\sigma' \cup \mathcal{Y}_2\eta\sigma' \cup \mathcal{Z}]$ is not a tautology either. Thus, by Definition 19, there exists an $\mathcal{A}$-clause $[E \,|\, \mathcal{Z}'] \in S$ and a substitution θ such that $E\theta \subseteq (s_1 \simeq s_2 \vee D_1 \vee D_2)\eta\sigma'$ and $\mathcal{Z}'\theta \subseteq \mathcal{Y}_1\eta\sigma' \cup \mathcal{Y}_2\eta\sigma' \cup \mathcal{Z} \subseteq \mathcal{X}$. Therefore, $\Phi(S, \mathcal{X})$ contains the clause $E\theta_{\downarrow\mathcal{X}}$ that is contained in $[(s_1 \simeq s_2 \vee D_1 \vee D_2)\eta\sigma']_{\downarrow\mathcal{X}} = v_1 \simeq v_2 \vee C_1 \vee C_2$.

The following result is a straightforward consequence of Definition 19.

Proposition 41 *If $[C \,|\, \mathcal{X}]$ is redundant in a set S, then for any $\mathcal{A}$-substitution $(\sigma, \mathcal{Y})$, $[C\sigma \,|\, \mathcal{X} \cup \mathcal{Y}]$ is also redundant in S.*

Proposition 42 *Let S be an $\mathcal{SA}^{\prec}_{sel}$-saturated set of $\mathcal{A}$-clauses and $\mathcal{X}$ be a ground $\mathcal{A}$-set. If $\Phi(S, \mathcal{X})$ contains a non-tautological clause $D \subseteq C \vee a \simeq b \vee a \simeq b$, where $C \vee a \simeq b$ is positive and elementary then $C \vee a \simeq b$ is redundant in $\Phi(S, \mathcal{X})$.*

Proof If D contains at most one occurrence of $a \simeq b$, then necessarily $D \subseteq C \vee a \simeq b$ and the proof is immediate. Otherwise, since $a \simeq b \vee a \simeq b \subseteq D$, the latter cannot be of type 2; it is therefore of type 1, thus there exists an $\mathcal{A}$-clause $[D' \,|\, \mathcal{Y}] \in S$ and a substitution σ such that $D'\sigma_{\downarrow\mathcal{X}} = D$ and $\mathcal{Y}\sigma \subseteq \mathcal{X}$. D' is of the form $C' \vee u \simeq v \vee u' \simeq v'$, where $C'\sigma_{\downarrow\mathcal{X}} \subseteq C$, $u\sigma_{\downarrow\mathcal{X}} = u'\sigma_{\downarrow\mathcal{X}} = a$ and $v\sigma_{\downarrow\mathcal{X}} = v'\sigma_{\downarrow\mathcal{X}} = b$. By Proposition 4, D' is $\mathcal{A}$-flat; thus the literal $u \simeq v$ is necessarily $\succeq_{\mathcal{A}}$-maximal in D', and the $\mathcal{A}$-Factorization rule applied to $[D' \,|\, \mathcal{Y}]$ generates $[(C' \vee u \simeq v \vee v \not\simeq v')\theta \,|\, \mathcal{Y}\theta \cup \mathcal{Z}]$, where $(\theta, \mathcal{Z})$ is the m.g.u. of u and u', or simply $[(C' \vee u \simeq v)\theta \,|\, \mathcal{Y}\theta \cup \mathcal{Z}]$, if $v\theta = v'\theta$. We assume that $v\theta \neq v'\theta$, the proof when they are equal is simpler. Since $(\sigma, \mathcal{X})$ is an instance of $(\theta, \mathcal{Z})$, by Proposition 41 the clause $[(C' \vee u \simeq v \vee v \not\simeq v')\sigma \,|\, \mathcal{X}]$ must be redundant in S, and since $v\sigma_{\downarrow\mathcal{X}} = v'\sigma_{\downarrow\mathcal{X}} = b$, it is equivalent to $[(C' \vee u \simeq v)\sigma \,|\, \mathcal{X}]$. This $\mathcal{A}$-clause cannot be a tautology; otherwise, by Proposition 39, $(C' \vee u \simeq v)\sigma_{\downarrow\mathcal{X}} \equiv D$ would also be a tautology. Therefore, by Definition 19, there exists an $\mathcal{A}$-clause $[E \,|\, \mathcal{U}] \in S$ and a substitution η such that $E\eta \subseteq (C' \vee u \simeq v \vee v \not\simeq v')\sigma$ and $\mathcal{U}\eta \subseteq \mathcal{X}$. By definition of $\Phi(S, \mathcal{X})$, the clause $E\eta_{\downarrow\mathcal{X}}$ occurs in $\Phi(S, \mathcal{X})$. If $E\eta \subseteq (C' \vee u \simeq v)\sigma$ then the proof is completed. Otherwise, E is of the form $E' \vee w \not\simeq w'$, where $E'\eta \subseteq (C' \vee u \simeq v)\sigma$, $w\eta = v\sigma$ and $w'\eta = v'\sigma$. Note that w and w' cannot both be equal to $\top$, since otherwise $w \not\simeq w'$ would have been removed from the $\mathcal{A}$-clause, thus the literal $w \not\simeq w'$ is necessarily $\succeq_{\mathcal{A}}$-maximal in $E' \vee w \not\simeq w'$, and it must be selected; therefore, the $\mathcal{A}$-Reflection rule can be applied on this clause. Since $(\eta, \mathcal{X})$ is a unifier of w and w', necessarily, the $\mathcal{A}$-clause $[E'\eta \,|\, \mathcal{X}]$ is redundant in S. By Definition 19, S contains a clause $[E'' \,|\, \mathcal{U}']$ and there exists a substitution μ such that $E''\mu \subseteq E'\eta$ and $\mathcal{U}'\mu \subseteq \mathcal{X}$. We conclude that $E'\eta_{\downarrow\mathcal{X}} \subseteq C \vee a \simeq b$ must be redundant in $\Phi(S, \mathcal{X})$.

Proposition 43 *There exists a set of clauses $U \subseteq \Phi_2(S, \mathcal{X}) \cup \Phi_3(S, \mathcal{X}) \cup \Phi_4(S, \mathcal{X}) \cup \Phi_5(S, \mathcal{X})$ such that U contains no occurrence of β, and $U \models \mathcal{X}$*

Proof Consider the following sets:

$$X_1 \stackrel{\text{def}}{=} \{a \simeq a_{\downarrow\mathcal{X}} \,|\, a \in \mathcal{X},\, a \neq a_{\downarrow\mathcal{X}}\},$$
$$X_2 \stackrel{\text{def}}{=} \left\{(a \not\simeq b)_{\downarrow\mathcal{X}} \,|\, a \not\simeq b \in \mathcal{X}\right\},$$
$$X_3 \stackrel{\text{def}}{=} \left\{(f(a_1, \ldots, a_n) \bowtie \top)_{\downarrow\mathcal{X}} \,|\, f(a_1, \ldots, a_n) \bowtie \top \in \mathcal{X}\right\}.$$

It is clear that $\mathcal{X} \equiv X_1 \cup X_2 \cup X_3$ and that $X_1 \subseteq \Phi_2(S, \mathcal{X})$. By letting $X_2' \stackrel{\text{def}}{=} \bigcup_{a \not\simeq b \in X_2} \{\alpha(a) \not\simeq \alpha(b)\}$, we have $X_2' \models X_2$; the set $U \stackrel{\text{def}}{=} X_1 \cup X_2' \cup X_3$ entails $\mathcal{X}$, it is a subset of $\Phi_2(S, \mathcal{X}) \cup \Phi_3(S, \mathcal{X}) \cup \Phi_4(S, \mathcal{X})$ and contains no occurrence of β.

Proposition 44 *Let S be an $\mathcal{SA}^{\prec}_{sel}$-saturated set of $\mathcal{A}$-clauses and let $\mathcal{X}$ be a ground and satisfiable $\mathcal{A}$-set. Consider a positive and elementary clause $F = b_1 \simeq b_2 \vee C \in \Phi_1(S, \mathcal{X})$, where $b_1 \not\simeq b_2 \in \mathcal{X}$. If C is not a tautology, then there is a clause in $\Phi(S, \mathcal{X})$ that is contained in C.*

Proof By definition of $\Phi_1(S, \mathcal{X})$, S contains an $\mathcal{A}$-clause $[u \simeq v \vee E \,|\, \mathcal{Y}]$ and there is a substitution θ such that $u\theta_{\downarrow\mathcal{X}} = b_1$, $v\theta_{\downarrow\mathcal{X}} = b_2$, $E\theta_{\downarrow\mathcal{X}} = C$ and $\mathcal{Y}\theta \subseteq \mathcal{X}$. Note that $E\theta_{\downarrow\mathcal{X}}$ must be positive and elementary, and so must $E\theta$. The $\mathcal{A}$-Assertion rule generates the $\mathcal{A}$-clause $[E \,|\, \mathcal{Y} \cup \{u \not\simeq v\}]$, and since $\mathcal{X}$ is satisfiable and $b_1 \not\simeq b_2 \in \mathcal{X}$, necessarily $b_1 \neq b_2$, so that $\mathcal{Y}\theta \cup \{u \not\simeq v\}\,\theta$ is also satisfiable. If $[E \,|\, \mathcal{Y} \cup \{u \not\simeq v\}]\theta$ is a tautology, then so is $E\theta_{\downarrow\mathcal{X}}$ by Proposition 39, which is impossible by hypothesis. Thus, since S is $\mathcal{SA}^{\prec}_{sel}$-saturated and $E\theta$ is $\mathcal{A}$-flat and positive, by Definition 19, S contains an $\mathcal{A}$-clause $[E' \,|\, \mathcal{Y}']$ and there exists a substitution θ' such that $E'\theta' \subseteq E\theta$ and $\mathcal{Y}'\theta' \subseteq \mathcal{Y}\theta \cup \{u \not\simeq v\}\,\theta$. But since $(u \not\simeq v)_{\downarrow\theta} = (b_1 \not\simeq b_2) \in \mathcal{X}$, this means that $\Phi(S, \mathcal{X})$ contains the clause $E'\theta'_{\downarrow\mathcal{X}} \subseteq E\theta_{\downarrow\mathcal{X}} = C$, hence the result.

We establish a result on the form of clauses of types 3 or 4 in $\Phi(S, \mathcal{X})$.

Lemma 45 *Any clause C of type 3 (resp. 4) in $\Phi(S, \mathcal{X})$ is of the form $\alpha(a_1) \not\simeq \alpha(a_2) \vee C'$ (resp. $p(a_1, \ldots, a_n) \bowtie \top \vee C'$) where:*

1. *$a_1, a_2 \in \mathcal{A}$ (resp. $a_1, \ldots, a_n \in \mathcal{A}$)*
2. *C' is positive and elementary.*
3. *$\mathcal{X}$ contains a clause of the form $b_1 \not\simeq b_2$ (resp. $p(b_1, \ldots, b_n)$) such that for all $i \in [1, 2]$ (resp. $i \in [1, n]$), one of the following conditions holds:*
   - *$a_i = b_i$,*
   - *$a_i \prec b_i$ and $\Phi(S, \mathcal{X})$ contains a clause of the form $a_i \simeq b_i \vee C_i$ with $C_i \subseteq C'$ and $C_i \prec (a_i \simeq b_i)$.*

Proof By definition, C is a non-redundant clause that is obtained from a clause of the form $\alpha(b_1) \not\simeq \alpha(b_2)$ with $b_1 \not\simeq b_2 \in \mathcal{X}$ (resp. from a clause $p(b_1, \ldots, b_n) \in \mathcal{X}$) by applying Superposition inferences from positive elementary clauses in $\Phi(S, \mathcal{X})$. We prove the result by induction on the number of Superposition inferences leading to the generation of C. The base case is immediate: it suffices to consider $a_i = b_i$ and $C' = \Box$. Assume that C is generated by a Superposition inference from a positive elementary clause into a clause D. Without loss of generality we assume that the considered derivation is minimal (w.r.t. the number of inference steps). By the induction hypothesis, D is of the form $\alpha(a_1) \not\simeq \alpha(a_2) \vee D'$ (resp. $p(a_1, \ldots, a_n) \bowtie \top \vee D'$), where a_1, a_2 (resp. $a_1, \ldots, a_n$) and D' satisfy the required properties. By definition of the selection function sel_Φ, only the literal $\alpha(a_1) \not\simeq \alpha(a_2)$ (resp. $p(a_1, \ldots, a_n)$) is selected, hence the replacement necessarily occurs in this literal. Without loss of generality, we may assume that the Superposition inference occurs upon the constant a_1, from a clause of the form $a_1 \simeq a'_1 \vee E'$, with $a_1 \succ a'_1$. Then necessarily, C is of the form $\alpha(a'_1) \not\simeq \alpha(a_2) \vee D' \vee E'$ (resp. $p(a'_1, a_2, \ldots, a_n) \bowtie \top \vee D' \vee E'$). If $b_1 = a_1$, then the proof is completed, since the clause $a'_1 \simeq a_1 \vee E'$ fulfills the requirement for Item 3 to hold. Otherwise, by the induction hypothesis, $\Phi(S, \mathcal{X})$ contains a clause of the form $a_1 \simeq b_1 \vee D_1$, where $D_1 \subseteq D'$. Assume that a_1 is not superposable in $a_1 \simeq b_1 \vee D_1$. Then by Proposition 38, this entails that $\Phi(S, \mathcal{X})$ contains a clause of type 1 of the form $a'_1 \simeq b_1 \vee D'_1$, where D'_1 is obtained from D_1 by replacing occurrences of a_1 by a'_1 (note that b_1 is

not replaced, since $b_1 \neq a_1$). By replacing the Superposition inference upon b_1 in the derivation leading to C by a Superposition from $a'_1 \simeq b_1 \vee D'_1$, we get a clause D'' of the form $\alpha(a'_1) \not\simeq \alpha(a_2) \vee C''$ (resp. $p(a'_1, a_2, \ldots, a_n) \bowtie \top \vee C''$), where $C'' \subseteq D' \vee D'_1$. Clause D'' satisfies the following properties.

- D'' is a clause of type 3 (resp. 4) in $\Phi(S, \mathcal{X})$.
- $D'' \preceq C$, since $a'_1 \prec a_1$.
- $a'_1 \simeq a_1 \vee E' \prec C$, since by definition of the ordering, $\alpha(x) \succ c$ and $p(\mathbf{x}) \succ c$ for every $c \in \mathcal{A}$.
- $D'', a'_1 \simeq a_1 \vee E' \models C$.

The number of inferences in the derivation is strictly less than that of C (since the sequence of Superposition inferences replacing b_1 by a_1 and then a_1 by a'_1 has been replaced by a single replacement of b_1 by a'_1), which by minimality of the derivation entails that $D'' \neq C$. Thus $D'' \prec C$, and C is redundant, which contradicts the definition of clauses of type 4.

This means that a_1 is necessarily superposable in $a_1 \simeq b_1 \vee D_1$. We distinguish two cases.

- The clause $a'_1 \simeq b_1 \vee D_1 \vee E'$ is a tautology. Since this clause is positive, this entails that it contains a literal of the form $t \simeq t$ (otherwise the interpretation mapping all constants to distinct elements would falsify the clause). Since $b_1 \succeq a_1$ and $a_1 \succ a'_1$, we have $b_1 \neq a'_1$, hence the literal $t \simeq t$ occurs in $D_1 \vee E'$. But then, C, which contains $D_1 \vee E'$ is also redundant, and this is impossible by definition of clauses of type 4.
- The clause $b_1 \simeq a'_1 \vee D_1 \vee E'$ is not a tautology. Since a_1 is superposable in $a_1 \simeq b_1 \vee D_1$, by Lemma 40, we deduce that there is a clause of type 1 in $\Phi(S, \mathcal{X})$ that is contained in $(a'_1 \simeq b_1) \vee D_1 \vee E'$. If this clause is contained in $D_1 \vee E'$ then it is also contained in C which is redundant and the proof is completed; otherwise it is of the form $(a'_1 \simeq b_1) \vee D'_1$, where $D'_1 \subseteq D_1 \vee E' \subseteq D' \vee E'$, which proves that the above property holds for C.

We are now in a position to provide the proof of Lemma 36.

Proof (of Lemma 36) We have to prove that every clause generated from $\Phi(S, \mathcal{X})$ by an inference in $\mathcal{SP}^{\prec}_{sel_\Phi}$ except for Equational Factorization on positive $\mathcal{A}$-flat clauses is a logical consequence of some clauses in $\Phi(S, \mathcal{X})$ that are strictly smaller than the maximal premise of the inference. Note that this condition necessarily holds if the conclusion is redundant in $\Phi(S, \mathcal{X})$, since a clause cannot be greater than its maximal premise. We distinguish several cases, depending on the types of the clauses involved in the inference.

Clauses of type 2.

By definition, every such clause is of the form $c \simeq c_{\downarrow\mathcal{X}}$, where $c \neq c_{\downarrow\mathcal{X}}$ and by construction, $c \succ c_{\downarrow\mathcal{X}}$. Constant c cannot occur in other clauses in $\Phi(S, \mathcal{X})$, since all its occurrences are replaced by $c_{\downarrow\mathcal{X}}$. Thus the clause $c \simeq c_{\downarrow\mathcal{X}}$ cannot interact with any other clause, because of the ordering restrictions of the Superposition calculus.

Clauses of type 5.

By construction, constant β only occurs in literals of the form $\beta \not\simeq \top$ and $\beta \simeq \top$. By definition of sel_Φ, the literal $\beta \not\simeq \top$ is never selected, thus the clause $\beta \simeq \top$ cannot interact with other clauses in $\Phi(S, \mathcal{X})$. Now, consider a clause of the form $\alpha(u) \not\simeq \alpha(v) \vee u \simeq v$. By definition, $u = u_{\downarrow\mathcal{X}}$, and u cannot be the maximal term of a selected literal in $\Phi(S, \mathcal{X})$. Since α occurs only in negative literals, no literal can interact with $\alpha(u) \not\simeq \alpha(v)$, and since $u \neq v$, the Reflection rule does not apply either.

Clauses of type 3.

Let C be a clause of type 3. By definition, only negative literals are selected in C, thus the only inference rules that can be applied on C are the Reflection rule and the Superposition rule into C, where the "from" premise is necessarily a clause of type 1 in $\Phi(S, \mathcal{X})$. By Case 3 of Definition 29, all the non-redundant clauses that can be generated by the Superposition inference rule are already in $\Phi_3(S, \mathcal{X})$. Thus, we only consider the case where the Reflection inference rule applied on C generates a clause D.

By Lemma 45, C is of the form $\alpha(a_1) \not\simeq \alpha(a_2) \vee C'$, where $\mathcal{X}$ contains a clause of the form $b_1 \not\simeq b_2$ and, for $i = 1, 2$, either $b_i = a_i$ or $\Phi(S, \mathcal{X})$ contains a positive and elementary clause (of type 1) of the form $(b_i \simeq a_i) \vee C_i$, where $C_i \subseteq C'$. Furthermore, for the Reflection rule to apply, we must have $a_1 = a_2$.

If $b_1 = a_1$ and $b_2 = a_2$, then this entails that $b_1 = b_2$, which is impossible since $b_1 \not\simeq b_2 \in \mathcal{X}$, and this set is satisfiable.

Assume that $b_1 = a_1$ and $b_2 \neq a_2$, the other case is symmetrical. Then, as mentioned above, by Lemma 45, $\Phi(S, \mathcal{X})$ must contain a clause $F = (b_2 \simeq a_2) \vee C_2$, where $C_2 \subseteq C'$, and since $a_2 = a_1 = b_1$, we have $F = (b_2 \simeq b_1) \vee C_2$. By Proposition 44, either C_2 is a tautology, in which case C' is also a tautology, or there is a clause in $\Phi(S, \mathcal{X})$ that is contained in C_2, and therefore also in C'. In both cases, we deduce that C' is redundant in $\Phi(S, \mathcal{X})$.

If $b_1 \neq a_1$, $b_2 \neq a_2$ and one of a_1, a_2, say, a_1, is superposable in $(b_1 \simeq a_1) \vee C_1$, then by Lemma 40, there is a clause F in $\Phi(S, \mathcal{X})$ that is contained in $(b_1 \simeq b_2) \vee C_1 \vee C_2$. Since $C_1 \vee C_2 \subseteq C'$, if $F \subseteq C_1 \vee C_2$ then $F \subseteq C'$ and the latter is redundant in $\Phi(S, \mathcal{X})$. Otherwise, F is of the form $(b_1 \simeq b_2) \vee C''$, where $C'' \subseteq C_1 \vee C_2$. By Proposition 44, either C'' is a tautology, in which case C' is also a tautology, or $\Phi(S, \mathcal{X})$ contains a clause $D' \subseteq C''$. In both cases, we deduce that C' is redundant in $\Phi(S, \mathcal{X})$.

Now assume that $b_1 \neq a_1$, $b_2 \neq a_2$ and that neither a_1 nor a_2 is superposable. By Proposition 38, $\Phi(S, \mathcal{X})$ contains a clause of the form $b_1 \simeq b_2 \vee G_1$, where G_1 is obtained from C_1 by replacing occurrences of a_1 by b_2, and by Proposition 44, we deduce that $\Phi(S, \mathcal{X})$ contains a clause $G'_1 \subseteq G_1$. Thus, since $\alpha(a_1) \not\simeq \alpha(a_2) \vee C_1 \vee C_2 \subseteq C$ and $a_1 = a_2$, we have:

$$G'_1, b_2 \simeq a_2 \vee C_2 \models G_1, b_2 \simeq a_2 \vee C_2 \models C_1 \vee C_2 \models C.$$

Since C contains an occurrence of α, it is strictly greater than G'_1 and $b_2 \simeq a_2 \vee F_2$; thus C is redundant, and cannot be a clause of type 3, a contradiction.

Clauses of type 4.

By Lemma 45, $\mathcal{X}$ contains a unit clause L of the form $p(b_1, \ldots, b_n) \bowtie \top$, and C is of the form $p(a_1, \ldots, a_n) \bowtie \top \vee C'$, where for every $i \in [1, n]$, one of the two following conditions holds:

1. $a_i = b_i$.
2. $\Phi(S, \mathcal{X})$ contains a positive elementary clause of the form $a_i \simeq b_i \vee C_i$, where $a_i \prec b_i$, and $(a_i \simeq b_i) \succ C_i$, $C_i \subseteq C'$.

Without loss of generality, we assume that L is of the form $p(b_1, \ldots, b_n) \simeq \top$, so that C is of the form $p(a_1, \ldots, a_n) \simeq \top \vee C'$. The only rule that can be applied on C (besides Superposition from elementary positive clauses for which the proof follows immediately from Case 4 of Definition 29) is the Superposition rule on the term $p(a_1, \ldots, a_n)$, and in this case the other premise must be of the form $p(a_1, \ldots, a_n) \not\simeq \top \vee F$. The generated clause is $C' \vee F$, since literals of the form $\top \not\simeq \top$ are deleted.

Let I denote the set of indices $i \in [1, n]$ such that $a_i \neq b_i$ and let $I' \stackrel{\text{def}}{=} [1, n] \setminus I$. By definition of $\Phi(S, \mathcal{X})$, for each index $i \in I$, there exists an $\mathcal{A}$-clause $[a'_i \simeq b'_i \vee C'_i \,|\, \mathcal{Y}_i] \in S$ and a substitution σ_i such that $a'_i\sigma_{i\downarrow\mathcal{X}} = a_i$, $b'_i\sigma_{i\downarrow\mathcal{X}} = b_i$, $C'_i\sigma_{i\downarrow\mathcal{X}} = C_i$, and $\mathcal{Y}_i\sigma_i \subseteq \mathcal{X}$. We define $E \stackrel{\text{def}}{=} \bigvee_{i\in I} C_i$ and $E' \stackrel{\text{def}}{=} \bigvee_{i\in I} C'_i$. Recall that $C_i \subseteq C'$ for all $i \in I$, hence $E \subseteq C'$; it is therefore sufficient to prove that $E \vee F$ is redundant in $\Phi(S, \mathcal{X})$.

Consider a set of pairwise distinct fresh variables $\{x_i \,|\, i \in I'\}$, and for all $i \in [1, n]$, consider the $\mathcal{A}$-clause $[c_i \simeq d_i \vee E'_i \,|\, \mathcal{Y}'_i]$ defined as follows:

- if $i \in I$, then $c_i \stackrel{\text{def}}{=} a'_i$, $d_i \stackrel{\text{def}}{=} b'_i$, $E'_i \stackrel{\text{def}}{=} C'_i$ and $\mathcal{Y}'_i \stackrel{\text{def}}{=} \mathcal{Y}_i$;
- otherwise $c_i \stackrel{\text{def}}{=} d_i \stackrel{\text{def}}{=} x_i$, $E'_i \stackrel{\text{def}}{=} \Box$ and $\mathcal{Y}'_i \stackrel{\text{def}}{=} \emptyset$.

The $\mathcal{A}$-Substitutivity rule applied to these clauses[7] generates the $\mathcal{A}$-clause $[F \,|\, \mathcal{U}]$, where $F = p(c_1, \ldots, c_n) \simeq \top \vee E'$ and $\mathcal{U} = \{p(d_1, \ldots, d_n) \simeq \top\} \cup \bigcup_{i\in[1,n]} \mathcal{Y}'_i$, and this $\mathcal{A}$-clause, which is $\mathcal{A}$-flat and quasi-positive, must be redundant in S. In particular, if μ is the substitution such that $a'_i\mu = a'_i\sigma_i$ for all $i \in I$, and $x_i\mu = a_i$ for all $i \in I'$, then by the redundancy criterion, either $[F \,|\, \mathcal{U}]\mu$ is a tautology (Case (i)), or there exists an $\mathcal{A}$-clause $[D \,|\, \mathcal{Y}]$ and a substitution θ such that $D\theta \subseteq F\mu$ and $\mathcal{Y}\theta \subseteq \mathcal{U}\mu$ (Case (ii)). We let $\mathcal{U}' \stackrel{\text{def}}{=} \bigcup_{i\in[1,n]} \mathcal{Y}'_i$.

If $[F \,|\, \mathcal{U}]\mu$ is a tautology (Case (i)), then by Proposition 39, $F\mu_{\downarrow\mathcal{U}\mu}$ is either a tautology, or contains a literal occurring in $\mathcal{U}\mu$. Assume that $F\mu_{\downarrow\mathcal{U}\mu}$ is a tautology. Then so is $F\mu_{\downarrow\mathcal{X}}$, because $F\mu_{\downarrow\mathcal{U}\mu} = F\mu_{\downarrow\mathcal{U}'\mu}$ and $\mathcal{U}'\mu \subseteq \mathcal{X}$. Now, since $(F\mu)_{\downarrow\mathcal{X}} = p(a_1, \ldots, a_n) \simeq \top \vee E$ and E is elementary, this means that E must be a tautology, and $E \vee F$ is thus redundant in $\Phi(S, \mathcal{X})$. We now

[7] Note that the clauses corresponding to indices $i \in I'$ are variants of the reflexivity axiom. The other clauses occur in S.

assume that $F\mu_{\downarrow\mathcal{U}\mu}$ contains a literal l occurring in $\mathcal{U}\mu$. If $l \in \mathcal{U}'$, then it is necessarily positive, hence $l_{\downarrow\mathcal{X}}$ is a tautology and $E \vee F$ is redundant in $\Phi(S, \mathcal{X})$. We may therefore assume in what follows that $l = (p(c_1, \ldots, c_n)\mu \simeq \top)_{\downarrow\mathcal{U}\mu} = p(d_1, \ldots, d_n)\mu \simeq \top$, so that $l_{\downarrow\mathcal{X}} = p(a_1, \ldots, a_n) \simeq \top = p(b_1 \ldots, b_n) \simeq \top$, from which we deduce that $(a_1, \ldots, a_n) = (b_1, \ldots, b_n)$.

If $[F \,|\, \mathcal{U}]\mu$ is not a tautology (Case (ii)), then $[D \,|\, \mathcal{Y}]$ is necessarily $\mathcal{A}$-flat and quasi-positive; also, $(D\theta)_{\downarrow\mathcal{X}} \subseteq (F\mu)_{\downarrow\mathcal{X}} = p(a_1, \ldots, a_n) \simeq \top \vee E$, and $\mathcal{Y}\theta \subseteq \mathcal{U}\mu \subseteq \mathcal{X}$. If $(D\theta)_{\downarrow\mathcal{X}} \subseteq E$, then it is clear that $E \vee F$ is redundant in $\Phi(S, \mathcal{X})$; we may therefore assume that D is of the form $p(\mathbf{s}) \simeq \top \vee D'$, where $p(\mathbf{s})\theta_{\downarrow\mathcal{X}} = p(a_1, \ldots, a_n)$ and $D'\theta_{\downarrow\mathcal{X}} \subseteq E$. By definition of the ordering, $p(\mathbf{s}\theta_{\downarrow\mathcal{X}}) \simeq \top$ is strictly greater than any literal in $D'\theta_{\downarrow\mathcal{X}}$.

We then distinguish two cases, depending on the type of the other premise in the Superposition inference, namely $p(a_1, \ldots, a_n) \not\simeq \top \vee F$.

1. If $p(a_1, \ldots, a_n) \not\simeq \top \vee F$ is of type 1, then there exists an $\mathcal{A}$-clause $[p(\mathbf{t}) \not\simeq \top \vee F' \,|\, \mathcal{Z}] \in S$ and a substitution θ' such that ${p(\mathbf{t})\theta'}_{\downarrow\mathcal{X}} = p(a_1, \ldots, a_n)$, $F = F'{\theta'}_{\downarrow\mathcal{X}} \vee F''$ and $\mathcal{Z}\theta' \subseteq \mathcal{X}$, where $F'' \subseteq \{\beta \not\simeq \top\}$. Then:
   - In Case (i), we only need to consider the case where $(a_1, \ldots, a_n) = (b_1, \ldots, b_n)$, by the preceding remark. Furthermore, the $\mathcal{A}$-Assertion rule applies on $[p(\mathbf{t}) \not\simeq \top \vee F' \,|\, \mathcal{Z}]$, yielding $[F' \,|\, \mathcal{Z} \cup \{p(\mathbf{t}) \simeq \top\}]$. Since S is saturated under the $\mathcal{A}$-Assertion rule, this $\mathcal{A}$-clause is redundant in S. Since $(p(\mathbf{t}\theta') \simeq \top)_{\downarrow\mathcal{X}} = (p(a_1, \ldots, a_n) \simeq \top) = (p(b_1, \ldots, b_n) \simeq \top) \in \mathcal{X}$, this entails that $F = F'{\theta'}_{\downarrow\mathcal{X}} \vee F''$ is redundant in $\Phi(S, \mathcal{X})$; hence so is $E \vee F$.
   - In Case (ii), since ${p(\mathbf{t})\theta'}_{\downarrow\mathcal{X}} = p(a_1, \ldots, a_n) = p(\mathbf{s})\theta_{\downarrow\mathcal{X}}$, $\mathbf{t}$ and $\mathbf{s}$ have an $\mathcal{A}$-unifier $(\mu', \mathcal{U}')$, that is more general than $(\theta\theta', \mathcal{X})$. Furthermore, $(p(\mathbf{s}) \simeq \top)\mu'$ and $(p(\mathbf{t}) \not\simeq \top)\mu'$ must be selected in $D\mu'$ and $(p(\mathbf{t}) \not\simeq \top \vee F')\mu'$ respectively, because the selection function is stable under $\mathcal{A}$-substitution, and $p(\mathbf{s}\theta_{\downarrow\mathcal{X}}) \simeq \top$ and ${p(\mathbf{t}\theta'}_{\downarrow\mathcal{X}}) \not\simeq \top$ must be selected in $D\theta_{\downarrow\mathcal{X}}$ and $(p(\mathbf{s}) \not\simeq \top \vee F'){\theta'}_{\downarrow\mathcal{X}}$ respectively. Thus the Superposition rule applies on $[p(\mathbf{s}) \simeq \top \vee D' \,|\, \mathcal{Y}]$ and $[p(\mathbf{t}) \not\simeq \top \vee F' \,|\, \mathcal{Z}]$, generating $[(D' \vee F') \,|\, \mathcal{Y} \cup \mathcal{Z} \cup \mathcal{U}']\mu'$. The $\mathcal{A}$-clause $[D'\theta \vee F'\theta' \,|\, \mathcal{X}]$ is therefore redundant in S, hence $E \vee F$ is redundant in $\Phi(S, \mathcal{X})$.
2. Otherwise, $p(a_1, \ldots, a_n) \not\simeq \top \vee F$ must be of type 4, F must be positive and elementary, and by the same reasoning as previously, we prove that $\mathcal{X}$ contains a clause $p(b'_1, \ldots, b'_n) \not\simeq \top$, and there exists an $\mathcal{A}$-set $\mathcal{Z}' \subseteq \mathcal{X}$ such that either $[p(a_1, \ldots, a_n) \not\simeq \top \vee F \mid \{p(b'_1, \ldots, b'_n) \not\simeq \top\} \cup \mathcal{Z}']$ is a tautology (Case (iii)), or there exist an $\mathcal{A}$-clause $[p(\mathbf{s}') \not\simeq \top \vee D'' \mid \mathcal{Y}']$ and a substitution θ' such that ${\mathbf{s}'\theta'}_{\downarrow\mathcal{X}} = (a_1, \ldots, a_n)$, ${D''}_{\downarrow\mathcal{X}}\theta \subseteq F$ and $\mathcal{Y}'\theta \subseteq \{p(b'_1, \ldots, b'_n) \bowtie \top\} \cup \mathcal{Z}'$ (Case (iv)).
   Using Proposition 39 and the same reasoning as above, we only consider the subcase of Case (iii) where $(a_1, \ldots, a_n) = (b'_1, \ldots, b'_n)$. Also, we note that Cases (i) and (iii) cannot hold simultaneously (otherwise we would have $(b_1, \ldots, b_n) = (a_1, \ldots, a_n) = (b'_1, \ldots, b'_n)$, hence $\mathcal{X}$ would contain both $p(a_1, \ldots, a_n) \simeq \top$ and $p(a_1, \ldots, a_n) \not\simeq \top$ and would be unsatisfiable). By symmetry, we may assume that (i) does not hold. Then:

- In Case (iii), we can apply the $\mathcal{A}$-Assertion rule on $[p(\mathbf{s}) \simeq \top \vee D' \,|\, \mathcal{Y}]$ to generate $[D' \,|\, \mathcal{Y} \cup \{p(\mathbf{s}) \not\simeq \top\}]$. Since $(a_1, \ldots, a_n) = (b'_1, \ldots, b'_n)$, we have $\mathcal{Y}\theta \cup \{(p(\mathbf{s}_{\downarrow\mathcal{X}}) \not\simeq \top)\theta\} \subseteq \{p(b_1, \ldots, b_n) \simeq \top, p(b'_1, \ldots, b'_n) \not\simeq \top\} \subseteq \mathcal{X}$, and $D'\theta \subseteq E$ is therefore redundant in $\Phi(S, \mathcal{X})$.
- In Case (iv), it is easy to check that we can apply the $\mathcal{A}$-Superposition rule on $[p(\mathbf{s}) \bowtie \top \vee D' \,|\, \mathcal{Y}]$ and $[p(\mathbf{s}') \not\bowtie \top \vee D'' \,|\, \mathcal{Y}']$ to generate an $\mathcal{A}$-clause of the form $[D' \vee D'' \,|\, \mathcal{Y} \cup \mathcal{Y}' \cup \mathcal{U}']\mu'$, where $(\mu', \mathcal{U}')$ is more general than $(\theta\theta', \mathcal{X})$. Thus $E \vee F$ is redundant in $\Phi(S, \mathcal{X})$.

Clauses of type 1.

All inferences involving a clause of type 2, 3 or 4 have already been considered, we now focus on inferences involving only clauses of type 1. We assume the Superposition rule is applied; the proof for the unary inference rules is similar. Let $C = u \simeq v \vee D$ and $E = t \bowtie s \vee F$ be two clauses of type 1 in $\Phi(S, \mathcal{X})$. Assume that the Superposition rule applies from C into E, upon the terms u and $t|_p$, yielding $t[v]_p \bowtie s \vee F \vee D$, where $t|_p = u$, $u \succ v$, $t \succ s$, $u \simeq v \in sel_\Phi(C)$ and $t \bowtie s \in sel_\Phi(E)$. Note that this implies that $u \simeq v$ is strictly maximal in C. We prove that the clause $t[v]_p \bowtie s \vee F \vee D$ is redundant in $\Phi(S, \mathcal{X})$. Note that by definition of sel_Φ, $t \bowtie s$ cannot be $\beta \not\simeq \top$. By definition, S contains two $\mathcal{A}$-clauses $C' = [u' \simeq v' \vee D' \,|\, \mathcal{Y}]$ and $E' = [t' \bowtie s' \vee F' \,|\, \mathcal{Z}]$ and there exist substitutions σ and θ such that:

- $u'\sigma_{\downarrow\mathcal{X}} = u$, $v'\sigma_{\downarrow\mathcal{X}} = v$, $D'\sigma_{\downarrow\mathcal{X}} \vee D'' = D$, $\mathcal{Y}\sigma \subseteq \mathcal{X}$ and $D'' \subseteq \{\beta \not\simeq \top\}$,
- $t'\theta_{\downarrow\mathcal{X}} = t$, $s'\theta_{\downarrow\mathcal{X}} = s$, $F'\theta_{\downarrow\mathcal{X}} \vee F'' = F$, $\mathcal{Z}\theta \subseteq \mathcal{X}$ and $F'' \subseteq \{\beta \not\simeq \top\}$.

First assume that there is a strict prefix q of p such that $t'|_q$ is a variable x. Then x cannot occur in $\mathcal{Z}$, since otherwise $x\theta$ would be a constant in $\mathcal{A}$ (because $\mathcal{Z}\theta \subseteq \mathcal{X}$), and q would not be a strict prefix of p. Let θ' be the substitution coinciding with θ, except for the value of x, and such that $x\theta'$ is obtained from $x\theta$ by replacing all occurrences of u by v. Since θ and θ' coincide on all the variables in $\mathcal{Z}$, necessarily $\mathcal{Z}\theta' \subseteq \mathcal{X}$. Furthermore, since $(t' \bowtie s' \vee F')\theta'$ is $\mathcal{A}$-flat and positive exactly when $(t' \bowtie s' \vee F')\theta$ is $\mathcal{A}$-flat and positive, we deduce that ${(t' \bowtie s' \vee F')\theta'}_{\downarrow\mathcal{X}} \vee F'' \in \Phi(S, \mathcal{X})$, and this clause is such that

$$\begin{aligned} {(t' \bowtie s' \vee F')\theta'}_{\downarrow\mathcal{X}} \vee F'', u \simeq v \vee D &\models {(t' \bowtie s' \vee F')\theta}_{\downarrow\mathcal{X}} \vee F'' \vee D,\ u \simeq v \vee D \\ &= t \bowtie s \vee F \vee D, u \simeq v \vee D \\ &\models t[v]_p \bowtie s \vee F \vee D. \end{aligned}$$

If ${(t' \bowtie s' \vee F')\theta'}_{\downarrow\mathcal{X}} \vee F'' = t[v]_p \bowtie s \vee F \vee D$ then $t[v]_p \bowtie s \vee F \vee D$ already occurs in $\Phi(S, \mathcal{X})$ hence the proof is completed. Otherwise ${(t' \bowtie s' \vee F')\theta'}_{\downarrow\mathcal{X}} \vee F'' \prec t[u]_p \bowtie s \vee F$. If $p \neq \varepsilon$ or $\bowtie = \not\simeq$, then necessarily $u \simeq v \prec t[u]_p \bowtie s$, since $u \succ v$. Furthermore, $D \prec u \simeq v$, hence ${(t' \bowtie s' \vee F')\theta'}_{\downarrow\mathcal{X}} \vee F'', u \simeq v \vee D \prec t[u]_p \bowtie s \vee F$, and the clause $t[v]_p \bowtie s \vee F \vee D$ is therefore a logical consequence of clauses of $\Phi(S, \mathcal{X})$ that are strictly smaller than one of its premises, the proof is thus completed.

If $p = \varepsilon$ and $\bowtie = \simeq$, then we have $E = u \simeq s \vee F$, and the generated clause is $v \simeq s \vee D \vee F$. If $v = s$ then this clause is a tautology, and is trivially redundant in $\Phi(S, \mathcal{X})$. Otherwise, assume w.l.o.g. that $v \prec s$ (since the same inference can be performed by considering E as the "from" premise, the two parent clauses play symmetric roles), then $u \simeq v \prec u \simeq s$, and as in the previous case, the clause $v \simeq s \vee F \vee D$ is therefore a logical consequence of clauses that are strictly smaller than one of its premises.

Now assume that there is no strict prefix q of p such that $t'|_q$ is a variable x. Necessarily, p must be a position in t'. Since $u = t|_p$, we have $u'\sigma \sim^{\mathcal{X}}_{\mathcal{A}} t'|_p\theta$, hence u' and $t'|_p$ are $\mathcal{A}$-unifiable. Let $(\eta, \mathcal{X}')$ be a most general $\mathcal{A}$-unifier of u' and $t'|_p$. Since $(\sigma\theta, \mathcal{X})$ is an $\mathcal{A}$-unifier of u' and $t'|_p$ we have $\mathcal{X}' \subseteq \mathcal{X}$ and there exists a substitution η' such that $\eta\eta' \sim^{\mathcal{X}}_{\mathcal{A}} \sigma\theta$. Since $u'\sigma_{\downarrow\mathcal{X}} = u \succ v = v'\sigma_{\downarrow\mathcal{X}}$, we have $v'\eta \not\succeq_{\mathcal{A}} u'\eta$, and similarly, $t'\eta \not\succeq_{\mathcal{A}} s'\eta$. Furthermore, since the selection function sel is stable by $\mathcal{A}$-substitution, $(t' \bowtie s')\eta$ and $(u' \simeq v')\eta$ must be selected in $C'\eta$ and $E'\eta$ respectively. Thus the $\mathcal{A}$-Superposition rule applied to C' and E', generates $[(t'[v']_p \bowtie s' \vee D' \vee F')\eta \mid (\mathcal{Y} \cup \mathcal{Z})\eta \cup \mathcal{X}']$. Since S is $\mathcal{SA}^{\prec}_{sel}$-saturated, this clause is $\mathcal{A}$-redundant in S, and so is $[(t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta' \mid (\mathcal{Y} \cup \mathcal{Z})\eta\eta' \cup \mathcal{X}']$.

Suppose that $(t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta'$ is $\mathcal{A}$-flat and quasi-positive. If $[(t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta' \mid (\mathcal{Y} \cup \mathcal{Z})\eta\eta' \cup \mathcal{X}']$ is a tautology, then by Proposition 39, $(t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta'_{\downarrow\mathcal{X}}$ either is a tautology or contains a literal $q(\mathbf{t}) \bowtie \top$ occurring in $\mathcal{X}$. In both cases, $(t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta'_{\downarrow\mathcal{X}}$ is redundant in $\Phi(S, \mathcal{X})$. Otherwise, by Definition 19, S contains an $\mathcal{A}$-clause $[G \mid \mathcal{U}]$ and there exists a substitution μ such that $G\mu \subseteq (t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta'$ and $U\mu \subseteq (\mathcal{Y} \cup \mathcal{Z})\eta\eta' \cup \mathcal{X}' \subseteq \mathcal{X}$. The clause $G\mu$ must be positive and $\mathcal{A}$-flat, hence by Case 1 of Definition 29, $\Phi(S, \mathcal{X})$ contains $G\mu_{\downarrow\mathcal{X}} \vee \Box = G\mu$, and $G\mu \subseteq (t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta'_{\downarrow\mathcal{X}} = t[v]_p \bowtie s \vee D \vee F$.

If $(t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta'$ is not $\mathcal{A}$-flat or not quasi-positive, then there exist n $\mathcal{A}$-clauses $[C_1 \mid \mathcal{X}_1], \ldots, [C_n \mid \mathcal{X}_n]$ and substitutions $\gamma_1, \ldots, \gamma_n$ such that:

- $\forall i \in [1, n]\, \mathcal{X}_i\gamma_i \subseteq (\mathcal{Y} \cup \mathcal{Z})\eta\eta' \cup \mathcal{X}'$,
- $\mathcal{X}'', C_1\gamma_1, \ldots, C_n\gamma_n \models (t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta'$,
- $(t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta'_{\downarrow\mathcal{X}} \succeq_{\mathcal{A}} C_1\gamma_1, \ldots, C_n\gamma_n$.

Since $\mathcal{X}'' \subseteq \mathcal{X}$, we deduce that $\mathcal{X}, C_1\gamma_1, \ldots, C_n\gamma_n \models (t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta'$. Also, by definition of $\succeq_{\mathcal{A}}$, we have $C_1\gamma_{1\downarrow\mathcal{X}}, \ldots, C_n\gamma_{n\downarrow\mathcal{X}} \preceq (t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta'_{\downarrow\mathcal{X}}$. But since $\mathcal{X}_1\gamma_1, \ldots, \mathcal{X}_n\gamma_n \subseteq (\mathcal{Y} \cup \mathcal{Z})\eta\eta' \cup \mathcal{X}' \subseteq \mathcal{X}$, $\Phi(S, \mathcal{X})$ contains clauses of the form $C_i\gamma_{i\downarrow\mathcal{X}} \vee G_i$ $(1 \leq i \leq n)$, where $G_i \subseteq \{\beta \not\simeq \top\}$. By Proposition 43, $\mathcal{X}$ is a logical consequence of a subset of $\Phi_2(S, \mathcal{X}) \cup \Phi_3(S, \mathcal{X}) \cup \Phi_4(S, \mathcal{X}) \cup \Phi_5(S, \mathcal{X})$ that contains no occurrence of β. Since $(t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta'$ is either not $\mathcal{A}$-flat or not positive, $t[v]_p \bowtie s \vee F \vee D$ contains $\beta \not\simeq \top$, and must be strictly greater than all clauses of type 2, 3, 4 or 5 that do not contain any occurrence of β. Thus $(t'[v']_p \bowtie s' \vee D' \vee F')\eta\eta'_{\downarrow\mathcal{X}} = t[v]_p \bowtie s \vee F \vee D$ is redundant in $\Phi(S, \mathcal{X})$.

Remark 46 The set $\Phi(S, \mathcal{X})$ is not saturated under Equational Factorization, because the literal $\beta \not\simeq \top$ is not added to the clauses that are positive and

$\mathcal{A}$-flat, and such clauses can have non-positive descendants. For example, $\{a \simeq b \vee a \simeq c, b \not\simeq c\}$ is $\mathcal{SA}^{\prec}_{sel}$-saturated, but $\Phi(S, \emptyset) = \{a \simeq b \vee a \simeq c, b \not\simeq c \vee \beta \not\simeq \top, \beta \simeq \top\}$ is not.

Corollary 47 *Let S be an $\mathcal{SA}^{\prec}_{sel}$-saturated set of $\mathcal{A}$-clauses and let $\mathcal{X}$ be a ground and satisfiable $\mathcal{A}$-set. If $\Phi(S, \mathcal{X})$ is unsatisfiable then it contains $\Box$.*

Proof Let S' be the smallest set of (standard ground) clauses such that S' contains all clauses C satisfying the following properties:

- C is generated by one of the rules in $\mathcal{SP}^{\prec}_{sel_\Phi}$ from $\Phi(S, \mathcal{X}) \cup S'$.
- C is not a logical consequence of the set of clauses in $\Phi(S, \mathcal{X}) \cup S'$ that are strictly smaller than the maximal premise of C.

Let $S'' = \Phi(S, \mathcal{X}) \cup S'$. Intuitively, S'' is the $\mathcal{SA}^{\prec}_{sel}$-closure of $\Phi(S, \mathcal{X})$ modulo redundancy. By definition S'' must be unsatisfiable and weakly $\mathcal{SP}^{\prec}_{sel}$-saturated, hence S'' contains the empty clause. For any term t, we denote by $\mathrm{P}^+(t)$ the set of positive clauses in $\Phi(S, \mathcal{X})$ that contain no term $s \succeq t$. We prove that the clauses in S' are $\mathcal{A}$-flat and of the form $c \simeq a' \vee a \not\simeq b \vee C'$, where:

1. C' is positive,
2. $c \succ a'$, $a' \succeq a$ and $a' \succeq b$,
3. $\mathrm{P}^+(c) \models C' \vee a \simeq a' \vee b \simeq a'$,
4. $\Phi(S, \mathcal{X})$ contains a positive $\mathcal{A}$-flat clause $C'' \subseteq c \simeq a \vee c \simeq b \vee D$ of type 1 such that $\{D\} \cup \mathrm{P}^+(c) \models C'$ and $D \preceq C'$.

This immediately implies that $\Box \notin S'$, hence that $\Box \in \Phi(S, \mathcal{X})$. The proof is by structural induction on S'. Let $C \in S'$. Note that C cannot be redundant in $\Phi(S, \mathcal{X})$, by definition of S' since the conclusion of an inference rule cannot be greater than its maximal premise.

- Assume that C is derived by the Reflection inference rule. Then, since $\Phi(S, \mathcal{X})$ is weakly saturated under Reflection, the parent of C must occur in S', hence by the induction hypothesis, it must be of the form $c \simeq a' \vee a \not\simeq b \vee C'$, where $\Phi(S, \mathcal{X})$ contains a clause $C'' \subseteq c \simeq a \vee c \simeq b \vee D$ such that $\{D\} \cup \mathrm{P}^+(c) \models C'$, $\mathrm{P}^+(c) \models C' \vee a \simeq a'$ and $D \preceq C'$.
  By definition of the Reflection rule we have $a = b$ and by Proposition 42 the clause $c \simeq a \vee D$ is necessarily redundant in $\Phi(S, \mathcal{X})$. But $C = c \simeq a' \vee C'$ is redundant in $\{c \simeq a \vee D\} \cup \mathrm{P}^+(c)$ by Condition 3 above, since $a = b$. Therefore, C is redundant in $\Phi(S, \mathcal{X})$, which is impossible.
- Assume that C is derived by Factorization. Then C is of the form $c \simeq a \vee a \not\simeq b \vee C'$, and its parent is $c \simeq a \vee c \simeq b \vee C'$. Note that this parent clause must be positive, otherwise $c \simeq a$ would not be selected, and that it is of type 1. Thus, it cannot occur in S', and $c \simeq a \vee c \simeq b \vee C' \in \Phi(S, \mathcal{X})$. It is simple to verify that the induction hypothesis holds on C.
- Assume that C is generated by a Superposition from C_1 into C_2. Then one of the premises is necessarily in S', and by the induction hypothesis, it contains a negative literal. Since a positive literal is selected in the first

premise of the inference rule, we deduce that $C_1 = a \simeq d \vee C_1'$, where $a \succ d$, $C_2 = c \simeq a' \vee a \not\simeq b \vee C_2'$, and $C = c \simeq a' \vee b \not\simeq d \vee C_1' \vee C_2'$. Note that C_1 must be of type 1; furthermore, $a \neq b$, since otherwise the Reflection rule would apply upon C_2, $c \simeq a' \vee C_2'$ would be redundant in $\Phi(S, \mathcal{X})$ and so would C. We prove that C verifies the induction hypothesis.

1. Since C_1 is a positive clause and C_2' is positive by the induction hypothesis, it is clear that $C_1' \vee C_2'$ is positive.
2. Since $a' \succeq a \succ d$, we have $c \succ a'$, $a' \succeq b$ and $a' \succeq d$.
3. By the induction hypothesis, $\mathrm{P}^+(c) \models C_2' \vee a \simeq a' \vee b \simeq a'$. Since $c \succ a' \succeq a \succ d$, we deduce that $C_1 \in \mathrm{P}^+(c)$, and therefore $\mathrm{P}^+(c) \models C_2' \vee C_1' \vee d \simeq a' \vee b \simeq a'$.
4. By the induction hypothesis, there is a positive clause $C_2'' \in \Phi(S, \mathcal{X})$ of type 1 such that $C_2'' \subseteq c \simeq a \vee c \simeq b \vee D_2$, where $\{D_2\} \cup \mathrm{P}^+(c) \models C_2'$ and $D_2 \preceq C_2'$. If C_2'' does not contain literal $c \simeq a$, then the proof is immediate, by letting $C'' \stackrel{\text{def}}{=} C_2''$ and $D \stackrel{\text{def}}{=} D_2$. Otherwise, C_2'' is of the form $c \simeq a \vee E$, where $E \subseteq c \simeq b \vee D_2$. If a is superposable in $c \simeq a \vee E$, then by Lemma 40, there is a clause in $\Phi(S, \mathcal{X})$ that is contained in $c \simeq d \vee C_1' \vee E \subseteq c \simeq b \vee c \simeq d \vee C_1' \vee D_2$, and the proof is completed. Otherwise, by Proposition 38, since C_2'' is of type 1, $\Phi(S, \mathcal{X})$ contains a clause $c \simeq d \vee E'$, where E' is obtained from E by replacing some occurrences of a by d. Since $E \subseteq c \simeq b \vee D_2$ and $a \neq b, c$, we deduce that $E' \subseteq c \simeq b \vee D_2'$, for a clause D_2' obtained from D_2 by replacing some occurrences of a by d. Since $a \simeq d \vee C_1' \in \mathrm{P}^+(c)$, we deduce that $\{D_2'\} \cup \mathrm{P}^+(c) \models C_1' \vee D_2$, hence $\{D_2'\} \cup \mathrm{P}^+(c) \models C_1' \vee C_2'$. Now $a \succ d$, so that $D_2' \preceq D_2 \preceq C_2'$, and the clause $d \simeq c \vee E' \subseteq d \simeq c \vee b \simeq c \vee D_2'$ fulfills the required property.

4.3 Deductive Completeness Theorem

The previous results lead to the following theorem, which states that the calculus $\mathcal{SA}^{\prec}_{sel}$ can be used to generate all ground implicates built on $\mathcal{A}$.

Theorem 48 *Let S_{init} be a set of standard clauses and let S be a set of $\mathcal{A}$-clauses obtained from S_{init} by $\mathcal{SA}^{\prec}_{sel}$-saturation. Then $\mathcal{C}_{\mathcal{A}}(S) \sqsubseteq I_{\mathcal{A}}(S_{init})$.*

Proof Let $C \in I_{\mathcal{A}}(S_{init})$, let $\mathcal{X}$ be the smallest $\mathcal{A}$-set containing C^{c} and let $S' \stackrel{\text{def}}{=} \Phi(S, \mathcal{X})$. Note that $\mathcal{X}$ is ground since C is ground. Since $\mathcal{X}$ is equivalent to C^{c} and C is not a tautology, this $\mathcal{A}$-set is satisfiable. We first prove that S' is equivalent to $S'' \stackrel{\text{def}}{=} S \cup C^{\mathrm{c}} \cup \{\beta \simeq \top, \alpha(u) \not\simeq \alpha(v) \vee u \simeq v \mid u, v \in \mathcal{A}\}$, and therefore unsatisfiable. By Proposition 33 $S' \models S_{init} \equiv S$; since $\Phi_2(S, \mathcal{X}) \cup \Phi_3(S, \mathcal{X}) \subseteq S'$, we have $S' \models C^{\mathrm{c}}$, and since $\Phi_5(S, \mathcal{X}) \subseteq S'$, we conclude that $S' \models S''$. We now show that S'' entails all clauses in S'.

Clauses in $\Phi_5(S, \mathcal{X})$. All the clauses in $\Phi_5(S, \mathcal{X})$ are in S'', and the result is obvious.

Clauses in $\Phi_2(S, \mathcal{X})$. For all $c \in \mathcal{A}$, $C^{\mathrm{c}} \models c \simeq c_{\downarrow\mathcal{X}}$. Since $C^{\mathrm{c}} \subseteq S''$, we have the result.

Clauses in $\Phi_1(S, \mathcal{X})$. Let $[D \,|\, \mathcal{Y}] \in S$, and consider a ground substitution σ such that $\mathcal{Y}\sigma \subseteq \mathcal{X}$ and $x\sigma_{\downarrow\mathcal{X}} = x\sigma$ for all $x \in \mathrm{var}(D)$. Then $S'' \models D\sigma$, and since $\mathcal{Y}\sigma \subseteq \mathcal{X} \equiv C^{\mathrm{c}}$, $S'' \models (D\sigma)_{\downarrow\mathcal{X}}$. But $\beta \simeq \top \in S''$, thus $S'' \models (D\sigma)_{\downarrow\mathcal{X}} \vee C'$, regardless of whether $C' = \Box$ or $C' = (\beta \not\simeq \top)$.

Clauses in $\Phi_4(S, \mathcal{X})$. These clauses are all in C^{c}, hence the result is obvious.

Clauses in $\Phi_3(S, \mathcal{X})$. Consider a literal $a \not\simeq b \in \mathcal{X}$. Since the clause $a \simeq b \vee \alpha(a) \not\simeq \alpha(b)$ occurs in S'', we have $S'' \models \alpha(a) \not\simeq \alpha(b)$; therefore, $S'' \models \mathit{sup}(\alpha(a_{\downarrow\mathcal{X}}) \not\simeq \alpha(b_{\downarrow\mathcal{X}}), \Phi_1(S, \mathcal{X}))$.

Since S'' is unsatisfiable by construction, so is S' and by Corollary 47, S' contains the empty clause. This means that S must contain an $\mathcal{A}$-clause of the form $[\Box \,|\, \mathcal{Y}]$ where $\mathcal{Y}\theta \subseteq C^{\mathrm{c}}$. By definition $\mathcal{C}_{\mathcal{A}}(S)$ contains the clause $(\mathcal{Y}\theta)^{\mathrm{c}}$ and since $\mathcal{Y}\theta \subseteq C^{\mathrm{c}}$ we have $(\mathcal{Y}\theta)^{\mathrm{c}} \models C$.

Theorem 48 entails that the $\mathcal{A}$-implicates of a clause set S can be computed by applying the rules of $\mathcal{SA}^{\prec}_{sel}$ on S and collecting the negation of the constraints of the $\mathcal{A}$-clauses whose clausal part is empty. Note that the theorem does not hold if S is not obtained by $\mathcal{SA}^{\prec}_{sel}$-saturation from a set of standard clauses; this is due to the fact that no inference is performed on the literals occurring in the constraints. For example, the set $S = \{[\Box \mid a \simeq b], [\Box \mid a \not\simeq b]\}$ is clearly unsatisfiable and $\mathcal{SA}^{\prec}_{sel}$-saturated, however we have $\mathcal{C}_{\mathcal{A}}(S) = \{a \simeq b, a \not\simeq b\} \not\subseteq I_{\mathcal{A}}(S)$, since $\Box \in I_{\mathcal{A}}(S)$. We also provide an example showing that the theorem does not hold if $\mathcal{A}$-Superposition into the variables occurring in the constraints is not allowed.

Example 49 Let $S \stackrel{\text{def}}{=} \{x \simeq a \vee x \simeq c, x \simeq b \vee x \simeq d\}$ and $C \stackrel{\text{def}}{=} e \simeq a \vee e \simeq b \vee c \simeq d$. It is straightforward to verify that $S \models C$. The only way of generating an $\mathcal{A}$-clause $[\Box \,|\, \mathcal{X}]$ such that $\mathcal{X}\sigma \models C^{\mathrm{c}}$ is to apply the Superposition rule on the literals $x \simeq c$ and $x \simeq d$ upon the term x, which is usually forbidden. This can be done by first applying the $\mathcal{A}$-Assertion rule on the literals $x \simeq a$ and $x \simeq b$, yielding $[x \simeq c \,|\, \{x \not\simeq a\}]$ and $[x \simeq d \,|\, \{x \not\simeq b\}]$. Then it is possible to apply the Superposition on the term x since it occurs in the constraints. This yields $[c \simeq d \,|\, \{x \not\simeq a, x \not\simeq b\}]$, and by applying the $\mathcal{A}$-Assertion rule again, we obtain the $\mathcal{A}$-clause $[\Box \,|\, \{x \not\simeq a, x \not\simeq b, c \not\simeq d\}]$, which satisfies the required property.

5 Refinements

Theorem 48 proves that $\mathcal{SA}^{\prec}_{sel}$-saturation permits to obtain the prime $\mathcal{A}$-implicates of any set of clauses. This set may still be very large, it could thus require a lot of time to be generated and be difficult to handle. In this section we introduce some refinements of the calculus $\mathcal{SA}^{\prec}_{sel}$, showing that at almost no cost, it is possible to generate only those prime $\mathcal{A}$-implicates of a clause set S that satisfy properties that are closed under subsumption (see Definition 50), or to obtain a more concise representation of all the $\mathcal{A}$-implicates of S.

5.1 Imposing Additional Restriction on the Implicates

The first refinement is rather straightforward: it consists in investigating how the calculus can be adapted to generate implicates satisfying additional arbitrary restrictions (e.g., for generating implicates of some bounded cardinality, or positive implicates). We show that some restrictions can be imposed on the constraint part of all the $\mathcal{A}$-clauses occurring in the search space without losing deductive completeness; in other words, inferences yielding to $\mathcal{A}$-clauses whose constraints do not fulfill the considered restriction can be blocked. This is possible if these implicates belong to some class that is closed under subsumption. More formally:

Definition 50 A set of clauses $\mathfrak{P}$ is *closed under subsumption* if for every $C \in \mathfrak{P}$ and for every clause D such that $D\sigma \subseteq C$ for some substitution σ, we have $D \in \mathfrak{P}$. An $\mathcal{A}$-clause $[C \,|\, \mathcal{X}]$ is $\mathfrak{P}$-*compatible* if $\mathcal{X}^{\mathrm{c}} \in \mathfrak{P}$.

Proposition 51 *Let $\mathfrak{P}$ be a set of clauses that is closed under subsumption, and let $[E \,|\, \mathcal{Z}]$ be an $\mathcal{A}$-clause generated by an $\mathcal{SA}^{\prec}_{sel}$-rule, with $[C \,|\, \mathcal{X}]$ as a premise. If $[E \,|\, \mathcal{Z}]$ is $\mathfrak{P}$-compatible, then so is $[C \,|\, \mathcal{X}]$.*

Proof We only consider the case where $[E \mid \mathcal{Z}]$ is generated by the $\mathcal{A}$-Superposition rule applied to $[C \mid \mathcal{X}]$ and $[D \mid \mathcal{Y}]$, the case for the unary inference rules is similar. Then by definition, $\mathcal{Z} = (\mathcal{X} \cup \mathcal{Y} \cup \mathcal{E})\sigma$, where $(\sigma, \mathcal{E})$ is an $(\mathcal{X} \cup \mathcal{Y})$-pure $\mathcal{A}$-substitution, and we have

$$\mathcal{X}^{\mathrm{c}}\sigma \subseteq [(\mathcal{X} \cup \mathcal{Y} \cup \mathcal{E})\sigma]^{\mathrm{c}} \;=\; \mathcal{Z}^{\mathrm{c}}.$$

Since $\mathfrak{P}$ is closed under subsumption, we deduce that $[C \,|\, \mathcal{X}]$ is $\mathfrak{P}$-compatible.

$\mathcal{SA}^{\prec}_{sel}(\mathfrak{P})$ denotes the calculus $\mathcal{SA}^{\prec}_{sel}$ in which all inferences that generate non-$\mathfrak{P}$-compatible $\mathcal{A}$-clause are blocked. The following theorem shows that the calculus $\mathcal{SA}^{\prec}_{sel}(\mathfrak{P})$ is deductive complete for the clauses in $\mathfrak{C}_{\mathit{flat}}(\mathcal{A}) \cap \mathfrak{P}$.

Theorem 52 *Let S_{init} be a set of standard clauses and let S be a set of $\mathcal{A}$-clauses obtained from S_{init} by $\mathcal{SA}^{\prec}_{sel}(\mathfrak{P})$-saturation. If $\mathfrak{P}$ is closed under subsumption, then $\mathcal{C}_{\mathcal{A}}(S) \sqsubseteq I_{\mathcal{A}}(S_{init}) \cap \mathfrak{P}$.*

Proof A simple induction together with Proposition 51 proves that all the ancestors of $\mathfrak{P}$-compatible clauses generated by $\mathcal{SA}^{\prec}_{sel}$ are necessarily $\mathfrak{P}$-compatible themselves. Since $\square \in \mathfrak{P}$ for all sets $\mathfrak{P}$ that are closed under subsumption, all the clauses in S_{init} must be $\mathfrak{P}$-compatible, hence the result.

Examples of classes of clauses that are closed under subsumption include the following sets that are of some practical interest:

- The set of clauses C such that there exists a substitution σ such that $C\sigma$ is equivalent to a clause of length at most k.
- The set of positive (resp. negative) clauses.
- The set of implicants of some formula ϕ.

Note also that the class of clause sets that are closed under subsumption is closed under union and intersection, which entails that these criteria can be combined easily.

5.2 Discarding the Inferences on $\mathcal{A}$-flat Clauses

In this section we impose a restriction on the calculus that consists in preventing inferences on $\mathcal{A}$-literals. The obtained calculus is not complete since it does not generate all $\mathcal{A}$-implicates in general, but it is complete in a restricted sense: every $\mathcal{A}$-implicate is a logical consequence of the set of $\mathcal{A}$-flat clauses generated by the calculus.

Definition 53 We denote by $\mathcal{SAR}^{\prec}_{sel}$ the calculus $\mathcal{SA}^{\prec}_{sel}$ in which no inference upon $\mathcal{A}$-literals is allowed, except for the $\mathcal{A}$-Assertion and $\mathcal{A}$-Reflection rules. We denote by $\Psi(S, \mathcal{X})$ the set obtained from $\Phi(S, \mathcal{X})$ by deleting, in every clause $C \in \Phi(S, \mathcal{X})$, each literal l such that the unit clause l^{c} belongs to $\mathcal{X} \cup \{\beta \simeq \top\}$.

Example 54 Consider the set of clauses and $\mathcal{A}$-set from Example 30. The set $\Psi(S, \mathcal{X})$ contains the following clauses:

- $f(b) \simeq d$, $b \not\simeq d$, $d \simeq d$ and $g(t, b) \simeq f(d)$, where t ranges over the set of all ground terms;
- $a \simeq b$ and $c \simeq d$;
- $\alpha(b) \not\simeq \alpha(e)$ and $\alpha(d) \not\simeq \alpha(e)$;
- $\beta \simeq \top$, $\alpha(b) \not\simeq \alpha(d) \vee b \simeq d$, $\alpha(b) \not\simeq \alpha(e)$ and $\alpha(d) \not\simeq \alpha(e) \vee d \simeq e$.

Proposition 55 *For all sets of $\mathcal{A}$-clauses S and $\mathcal{A}$-sets $\mathcal{X}$, $\Phi(S, \mathcal{X}) \equiv \Psi(S, \mathcal{X})$.*

$\mathcal{SAR}^{\prec}_{sel}$ essentially simulates the calculus in (Echenim and Peltier, 2012), but there are some important differences: in particular our previous approach does not handle variable-active axioms and is complete only for implicates containing no predicate symbol other than $\simeq$. This entails that for example, an implicate of the form $p(c_1, \ldots, c_n) \simeq d$ can only be generated if a new constant c is added to $\mathcal{A}$, along with the axiom $c \Leftrightarrow p(c_1, \ldots, c_n)$. It is clear that applying this operation on all ground atoms is costly from a practical point of view. This is avoided with the new calculus $\mathcal{SAR}^{\prec}_{sel}$, thanks to the addition of new inference rules.

Lemma 56 *Let S be an $\mathcal{SAR}^{\prec}_{sel}$-saturated set of $\mathcal{A}$-clauses and let $\mathcal{X}$ be a complete and satisfiable $\mathcal{A}$-set. The set $\Psi(S, \mathcal{X})$ is $\mathcal{SP}^{\prec}_{sel_\Phi}$-saturated.*

Proof We prove that every $\mathcal{A}$-flat clause of type 1 in $\Psi(S, \mathcal{X})$ is redundant in $\Psi(S, \mathcal{X})$. Let $C = a \bowtie b \vee C'$ be such a clause, by definition of $\Phi(S, \mathcal{X})$, we have $a = a_{\downarrow \mathcal{X}}$ and $b = b_{\downarrow \mathcal{X}}$. If $a = b$ then either $\bowtie = \simeq$, in which case $a \bowtie b \vee C$ is a tautology, or $\bowtie = \not\simeq$, in which case $(a \not\simeq a)^{\mathrm{c}} \in \mathcal{X}$, and C cannot occur in $\Psi(S, \mathcal{X})$. Thus $a \neq b$, and since $\mathcal{X}$ is complete, we deduce that $a \not\simeq b \in \mathcal{X}$ and that $\alpha(a) \not\simeq \alpha(b)$ occurs in $\Phi(S, \mathcal{X})$. This implies that $\bowtie$ is $\not\simeq$, since otherwise the literal $a \simeq b$ would have been deleted from the clause. Thus, C is of the form $a \not\simeq b \vee C'$; it is not positive, and by construction, it contains the literal $\beta \not\simeq \top$. We deduce that $\alpha(a) \not\simeq \alpha(b) \models C$ and that $\alpha(a) \not\simeq \alpha(b) \prec C$; C is

therefore redundant. This implies that the only non-redundant inferences that can be applied on clauses in $\Psi(S, \mathcal{X})$ are upon literals that are not $\mathcal{A}$-flat. The restriction on the calculus $\mathcal{SAR}^{\prec}_{sel}$ does not affect such inferences, thus, as shown in the proof of Lemma 36, they can be simulated by inferences on the corresponding $\mathcal{A}$-clauses in S.

The next theorem states a form of completeness for the restricted calculus $\mathcal{SAR}^{\prec}_{sel}$, which is weaker than that of the calculus $\mathcal{SA}^{\prec}_{sel}$ (compare with Theorem 48) and similar to that of (Echenim and Peltier, 2012). The proof is based on the following result.

Proposition 57 *Let S be a set of $\mathcal{A}$-clauses and let $\mathcal{X}$ be a complete $\mathcal{A}$-set. Then $\Psi(S, \mathcal{X}) \models S$.*

Proof By Proposition 55 we have $\Psi(S, \mathcal{X}) \equiv \Phi(S, \mathcal{X})$. Let $[C \,|\, \mathcal{Y}]$ be a clause in S and let σ be a ground $\mathcal{Y}$-pure substitution. If $\mathcal{Y}\sigma \not\subseteq \mathcal{X}$, then there exists $l \in \mathcal{Y}\sigma$ such that $l \notin \mathcal{X}$ and since $\mathcal{X}$ is complete we deduce that $l^{\mathrm{c}} \in \mathcal{X}$, which entails that $\mathcal{X} \models (\mathcal{Y}\sigma)^{\mathrm{c}}$, and thus $\Phi(S, \mathcal{X}) \models (\mathcal{Y}\sigma)^{\mathrm{c}}$ (since by Proposition 43 $\Phi(S, \mathcal{X}) \models \mathcal{X}$). Otherwise, we must have $C\sigma_{\downarrow\mathcal{X}} \vee C' \in \Phi(S, \mathcal{X})$, for some $C' \subseteq \{\beta \not\simeq \top\}$, and since $\beta \simeq \top \in \Phi(S, \mathcal{X})$ we deduce that $\Phi(S, \mathcal{X}) \models C\sigma$.

Theorem 58 *Let S be an $\mathcal{SAR}^{\prec}_{sel}$-saturated set of $\mathcal{A}$-clauses. Then $\mathcal{C}_{\mathcal{A}}(S) \models I_{\mathcal{A}}(S)$.*

Proof We prove the contrapositive, i.e., that every counter-model of $I_{\mathcal{A}}(S)$ is a counter-model of $\mathcal{C}_{\mathcal{A}}(S)$. Let $\mathcal{M}$ be a counter-model of $I_{\mathcal{A}}(S)$ and let $\mathcal{X}$ be the corresponding $\mathcal{A}$-set, i.e. the set containing all $\mathcal{A}$-flat literals that are true in $\mathcal{M}$. By definition, $\mathcal{X}$ is complete and satisfiable. By Proposition 57, $\Psi(S, \mathcal{X}) \models S \models I_{\mathcal{A}}(S)$. Since $\Psi(S, \mathcal{X}) \models \mathcal{X}$ and $\mathcal{X} \cup I_{\mathcal{A}}(S)$ is unsatisfiable, we deduce that $\Psi(S, \mathcal{X})$ is unsatisfiable; but $\Psi(S, \mathcal{X})$ is $\mathcal{SP}^{\prec}_{sel_\Phi}$-saturated by Lemma 56, hence $\square \in \Psi(S, \mathcal{X})$. We deduce that S contains an $\mathcal{A}$-clause of the form $[C \,|\, \mathcal{Y}]$ and there exists a substitution σ such that $C^{\mathrm{c}}\sigma_{\downarrow\mathcal{X}} \cup \mathcal{Y}\sigma \subseteq \mathcal{X}$. Without loss of generality, we assume that C is the clause with the least number of literals satisfying this property. Suppose that C is nonempty. Then $sel_\Phi(C\sigma_{\downarrow\mathcal{X}})$ contains at least one literal $(u \bowtie v)\sigma_{\downarrow\mathcal{X}}$ and C is of the form $u \bowtie v \vee D$. If $\bowtie$ is $\simeq$, then the $\mathcal{A}$-Assertion rule can be applied to this literal, yielding the $\mathcal{A}$-clause $[D \,|\, \mathcal{Y} \cup \{u \not\simeq v\}]$. Since S is $\mathcal{SAR}^{\prec}_{sel}$-saturated, this $\mathcal{A}$-clause must be $\mathcal{A}$-redundant and by Definition 19, S contains an $\mathcal{A}$-clause $[D' \,|\, \mathcal{Y}']$, such that, for some substitution θ, $D'\theta \subseteq D\sigma$ and $(\mathcal{Y}' \cup \{u \not\simeq v\})\theta \subseteq \mathcal{Y}\sigma$ (note that $D\sigma$ cannot be a tautology because $D^{\mathrm{c}}\sigma \subseteq \mathcal{X}$ and $\mathcal{X}$ is satisfiable). This is impossible because then $[D' \,|\, \mathcal{Y}']$ would then satisfy the above restriction, thus contradicting the minimality of C. If $\bowtie$ is $\not\simeq$ then $(u \simeq v)\sigma_{\downarrow\mathcal{X}}$ must occur in $\mathcal{X}$ since $C^{\mathrm{c}}\sigma_{\downarrow\mathcal{X}} \subseteq \mathcal{X}$; this implies that $u\sigma_{\downarrow\mathcal{X}} = v\sigma_{\downarrow\mathcal{X}}$, hence that $u\sigma \sim^{\mathcal{X}}_{\mathcal{A}} v\sigma$. Thus the $\mathcal{A}$-Reflection rule applies, yielding $[D \,|\, \mathcal{Y} \cup \mathcal{E}]\eta$, where $(\eta, \mathcal{E})$ is the most general unifier of u and v. There exists a substitution σ' such that $\sigma \sim^{\mathcal{X}}_{\mathcal{A}} \eta\sigma'$, and by the same reasoning as previously, since S is $\mathcal{SAR}^{\prec}_{sel}$-saturated, it contains an $\mathcal{A}$-clause $[D' \,|\, \mathcal{Y}']$ and there exists a substitution θ' such that $D'\theta' \subseteq$

$D\eta\sigma'$ and $\mathcal{Y}'\theta' \subseteq \mathcal{Y}\eta\sigma' \cup \mathcal{E}$. But then $(D'\theta'_{\downarrow\mathcal{X}})^{\mathrm{c}} \subseteq (D\eta\sigma'_{\downarrow\mathcal{X}})^{\mathrm{c}} = (D\sigma_{\downarrow\mathcal{X}})^{\mathrm{c}} \subseteq \mathcal{X}$, and $\mathcal{Y}'\theta'_{\downarrow\mathcal{X}} \subseteq (\mathcal{Y}\eta\sigma' \cup \mathcal{E})_{\downarrow\mathcal{X}} = (\mathcal{Y}\sigma \cup \mathcal{E})_{\downarrow\mathcal{X}} \subseteq \mathcal{X}$. Again, this contradicts the minimality of C. Therefore, C is empty, and $(\mathcal{Y}\sigma)^{\mathrm{c}} \in \mathcal{C}_{\mathcal{A}}(S)$. Now $\mathcal{Y}\sigma \subseteq \mathcal{X}$, thus $\mathcal{M} \not\models (\mathcal{Y}\sigma)^{\mathrm{c}}$, which proves that $\mathcal{M}$ is indeed a counter-model of $\mathcal{C}_{\mathcal{A}}(S)$, and the proof is completed.

The main differences between the calculi $\mathcal{SA}^{\prec}_{sel}$ and $\mathcal{SAR}^{\prec}_{sel}$ can be summarized as follows.

- The calculus $\mathcal{SA}^{\prec}_{sel}$ explicitly generates all prime implicates in $I_{\mathcal{A}}(S)$, whereas $\mathcal{SAR}^{\prec}_{sel}$ only generates a finite representation of them, in the form of an $\mathcal{A}$-flat implicant S' of $I_{\mathcal{A}}(S)$. The formula S' can still contain redundancies and some additional post-processing step is required to generate explicitly the prime implicates of S' if needed. Any algorithm for generating prime implicates of propositional clause sets can be used for this purpose, since flat ground equational clause sets can be reduced into equivalent sets of propositional clauses by adding equality axioms. In (Echenim et al, 2013, 2014) a much more efficient algorithm has been proposed, in which equality axioms are directly taken into account in the inference engine and redundancy pruning mechanism. From a practical point of view, the set $I_{\mathcal{A}}(S)$ can be very large, thus S' can also be viewed as a concise and suitable representation of such a set.
- The calculus $\mathcal{SAR}^{\prec}_{sel}$ restricts inferences on $\mathcal{A}$-flat literals to those that actually delete such literals, possibly by transferring them to the constraint part of the clauses (the $\mathcal{A}$-Assertion and $\mathcal{A}$-Reflection rules). From a practical point of view, this entails that these literals do not need to be considered anymore in the clausal part of the $\mathcal{A}$-clause: they can be transferred *systematically* in the constraints. This can reduce the number of generated clauses by an exponential factor, since a given $\mathcal{A}$-flat clause $l_1 \vee \ldots \vee l_n$ can be in principle represented by 2^n distinct $\mathcal{A}$-clauses depending on whether l_i is stored to the clausal or constraint part of the $\mathcal{A}$-clause (for instance $a \simeq b$ can be represented as $[a \simeq b \,|\, \emptyset]$ or $[\Box \,|\, a \not\simeq b]$). Furthermore, the number of applicable inferences is also drastically reduced, since the rules usually apply in many different ways on (selected) $\mathcal{A}$-literals, due to the fact that two $\mathcal{A}$-flat terms are always $\mathcal{A}$-unifiable and that the ordering $\succ_{\mathcal{A}}$ is empty when applied on terms in $\mathcal{A}$. For example the clauses $a \simeq b$ and $c \simeq d$ generate the $\mathcal{A}$-clauses

$$[d \simeq b \,|\, \{a \simeq c\}],\ [d \simeq a \,|\, \{b \simeq c\}],\ [c \simeq b \,|\, \{a \simeq d\}],\ [c \simeq a \,|\, \{b \simeq d\}],$$

  regardless of the ordering $\prec$.

The following example illustrates the differences between $\mathcal{SA}^{\prec}_{sel}$ and $\mathcal{SAR}^{\prec}_{sel}$.

Example 59 Let $S = \{f(a,b) \not\simeq f(c,d), g(x) \simeq 0 \vee x \simeq c, g(a) \not\simeq 0\}$, where $x \in \mathcal{V}$, $g(x) \succ a \succ b \succ c \succ d$ and $\mathcal{A} = \{a,b,c,d\}$. It is easy to check that $\mathcal{SAR}^{\prec}_{sel}$ generates the implicates $[\Box \mid \{a \simeq c, b \simeq d\}]$ (by $\mathcal{A}$-Reflection on

the first clause) and $[\square \mid \{a \not\simeq c\}]$ (by an application of the $\mathcal{A}$-Superposition rule from the second clause into the third one, followed by an application of the $\mathcal{A}$-Assertion rule). However, the implicate $[\square \mid \{b \simeq d\}]$ that is a logical consequence of the above $\mathcal{A}$-clauses is not generated. In contrast, it is possible to infer this implicate with $\mathcal{SA}^{\prec}_{sel}$: First the $\mathcal{A}$-Superposition rule generates as usual the clauses $a \simeq c$ and then $f(c,b) \not\simeq f(c,d)$ (the constraints are empty at this point since all the considered $\mathcal{A}$-unifiers are standard unifiers), and $[\square \mid \{b \simeq d\}]$ is inferred by applying $\mathcal{A}$-Reflection on the latter clause. Note that $\mathcal{SA}^{\prec}_{sel}$ has a larger search space than $\mathcal{SAR}^{\prec}_{sel}$. Consider for instance a clause $a \simeq b \vee c \simeq d$. $\mathcal{SAR}^{\prec}_{sel}$ simply reduces this clause into $[\square \mid \{a \not\simeq b, c \not\simeq d\}]$ and no further inference is applicable on it, while $\mathcal{SA}^{\prec}_{sel}$ also generates the $\mathcal{A}$-clauses $[a \simeq b \mid \{c \not\simeq d\}]$ and $[c \simeq d \mid \{a \not\simeq b\}]$, which in turn possibly enable other inferences.

It is possible to combine the two calculi $\mathcal{SA}^{\prec}_{sel}$ and $\mathcal{SAR}^{\prec}_{sel}$. This can be done in the following way.

- Starting from a set of clauses S, $\mathcal{SAR}^{\prec}_{sel}$ is first applied until saturation, yielding a new set S'. By Theorem 58 we have $\mathcal{C}_{\mathcal{A}}(S') \equiv I_{\mathcal{A}}(S)$.
- Then $\mathcal{SA}^{\prec}_{sel}(\mathfrak{P})$ is applied on $\mathcal{C}_{\mathcal{A}}(S')$ until saturation yielding a set S'', where $\mathfrak{P}$ denotes the set of clauses that logically entail at least one clause in $\mathcal{C}_{\mathcal{A}}(S')$. It is clear that this set of clauses is closed under subsumption, hence by Theorem 52, we eventually obtain a set of clauses $\mathcal{C}_{\mathcal{A}}(S'') \sqsubseteq I_{\mathcal{A}}(\mathcal{C}_{\mathcal{A}}(S')) \cap \mathfrak{P}$. But $I_{\mathcal{A}}(\mathcal{C}_{\mathcal{A}}(S')) \cap \mathfrak{P} \sqsubseteq \mathcal{C}_{\mathcal{A}}(S')$, hence $\mathcal{C}_{\mathcal{A}}(S'') \sqsubseteq \mathcal{C}_{\mathcal{A}}(S')$, and $\mathcal{C}_{\mathcal{A}}(S'') \equiv I_{\mathcal{A}}(S)$. The set of clauses $\mathcal{C}_{\mathcal{A}}(S'')$ can therefore be considered as a concise representation of $I_{\mathcal{A}}(S)$. This approach is appealing since $\mathcal{C}_{\mathcal{A}}(S'')$ is in general much smaller than $I_{\mathcal{A}}(S)$, and contrary to $\mathcal{C}_{\mathcal{A}}(S')$, this set is free of redundancies.

Another straightforward method to eliminate redundant literals from the clauses in $\mathcal{C}_{\mathcal{A}}(S')$ without having to explicitly compute the set $I_{\mathcal{A}}(S')$ is to test, for every clause $l \vee C \in \mathcal{C}_{\mathcal{A}}(S')$, whether the relation $\mathcal{C}_{\mathcal{A}}(S') \models C$, holds, in which case the literal l can be safely removed. The test can be performed by using any decision procedure for ground equational logic (see for instance Meir and Strichman, 2005; Dillig et al, 2010, for a similar approach). Note however that removing redundant literals is not sufficient to obtain prime implicates, as shown in the following example.

Example 60 Consider the clause set: $S \stackrel{\text{def}}{=} \{a \not\simeq c \vee b \not\simeq c \vee d \simeq e,\ a \simeq c \vee a \simeq f,\ b \simeq c \vee a \simeq f,\ f \not\simeq b\}$. It is easy to check that $a \not\simeq b \vee d \simeq e$ is an implicate of S and that this clause is strictly more general than $a \not\simeq c \vee b \not\simeq c \vee d \simeq e$. The calculus $\mathcal{SA}^{\prec}_{sel}$ computes the $\mathcal{A}$-clause $[\square \mid \{a \simeq b, d \simeq e\}]$, yielding the set of prime implicates: $S' \stackrel{\text{def}}{=} \{a \not\simeq b \vee d \simeq e, a \simeq c \vee a \simeq f, b \simeq c \vee a \simeq f, f \not\simeq b\}$. S' is equivalent to S and strictly smaller. In contrast, the approach devised by Dillig et al (2010) cannot simplify S since there is no useless literal.

6 Termination

We relate the termination behavior of $\mathcal{SA}^{\prec}_{sel}$ to that of the usual Superposition calculus. Our goal is to show that most existing termination results concerning the standard Superposition calculus (in particular those holding for theories of interest in program verification) can be carried over to $\mathcal{SA}^{\prec}_{sel}$, under some rather natural restrictions.

We first introduce restricted ordering and redundancy criteria. For all expressions (terms, atoms, literals or clauses) t and s, we write $t \rhd_{\mathcal{A}} s$ if $t' \succ s'$ holds for all expressions t', s' such that $t \sim_{\mathcal{A}} t'$ and $s \sim_{\mathcal{A}} s'$. Note that the ordering $\rhd_{\mathcal{A}}$ is stronger than $\succeq_{\mathcal{A}}$ (and also stronger than $\succeq$) because the constants in t and s can be rewritten independently of each other. Assume for instance that $\prec$ is such that $f(a) \prec g(a) \prec f(b) \prec g(b)$ with $\mathcal{A} = \{a, b\}$. Then it is easy to check that $g(a) \succeq_{\mathcal{A}} f(a)$ but $g(a) \not\rhd_{\mathcal{A}} f(a)$ since $g(a) \prec f(b) \sim_{\mathcal{A}} f(a)$. Also, let $sel_{\mathcal{A}}$ be the selection function defined from the function sel as follows: for every clause $l \vee C$, $l \in sel_{\mathcal{A}}(l \vee C)$ if there exists l', C' such that $l' \sim_{\mathcal{A}} l$, $C' \sim_{\mathcal{A}} C$ and $l' \in sel(l' \vee C')$. We show that most termination results for the calculus $\mathcal{SP}^{\trianglelefteq_{\mathcal{A}}}_{sel_{\mathcal{A}}}$ also apply to $\mathcal{SA}^{\prec}_{sel}$. To this purpose, we consider a restricted form of redundancy testing.

Definition 61 A standard clause C is *strongly redundant* in a set of standard clauses S iff for every clause $C' \sim_{\mathcal{A}} C$, C' is $\mathcal{A}$-redundant in S.

Definition 62 For every set of $\mathcal{A}$-clauses S and for every ground $\mathcal{A}$-set $\mathcal{Y}$, we denote by $\Gamma(S, \mathcal{Y})$ the set of standard clauses $C\sigma$, where $[C \,|\, \mathcal{X}] \in S$ and σ is an $\mathcal{X}$-pure substitution of domain $\mathrm{var}(\mathcal{X})$ such that $\sigma_{\downarrow \mathcal{Y}} = \sigma$ and $\mathcal{X}\sigma \subseteq \mathcal{Y}$.

The definition of $\Gamma(S, \mathcal{Y})$ is similar to that of $\Phi(S, \mathcal{Y})$ (see Section 4), except that: (i) only the variables occurring in $\mathcal{X}$ are instantiated; (ii) the clauses are not reduced with respect to the equations in the constraint part (but the constants replacing the variables in $\mathcal{X}$ are reduced).

Example 63 Let $S = \{[f(x, y) \simeq a \,|\, \{x \not\simeq b\}]\}$ with $\mathcal{A} = \{a, b, c\}$ and $a \succ b \succ c$. We have $\Gamma(S, \{a \not\simeq b, a \simeq c\}) = \{f(c, y) \simeq a\}$ and $\Gamma(S, \{a \not\simeq b, c \not\simeq b\}) = \{f(a, y) \simeq a, f(c, y) \simeq a\}$.

We need the following results:

Proposition 64 *If $(\sigma, \mathcal{E})$ is a most general $\mathcal{A}$-unifier of $t \simeq s$, then for all $\mathcal{X}$ such that $\mathcal{E} \subseteq \mathcal{X}$, $t_{\downarrow \mathcal{X}}$ and $s_{\downarrow \mathcal{X}}$ are unifiable, and $\sigma_{\downarrow \mathcal{X}}$ is a most general unifier of $t_{\downarrow \mathcal{X}} \simeq s_{\downarrow \mathcal{X}}$.*

Proof This is because if $(S, \theta, \mathcal{X})$ and $(S', \theta', \mathcal{X}')$ are $\mathcal{A}$-unification problems such that $(S, \theta, \mathcal{X}) \to (S', \theta', \mathcal{X}')$, then for all $\mathcal{A}$-sets $\mathcal{Y}$ such that $\mathcal{X}' \subseteq \mathcal{Y}$, we have $S_{\downarrow \mathcal{Y}} \to S'_{\downarrow \mathcal{Y}}$ for the standard unification rules. The proof follows by a straightforward induction.

Since terms that are $\sim_{\mathcal{A}}$-equivalent cannot be distinguished by $\trianglelefteq_{\mathcal{A}}$ and $sel_{\mathcal{A}}$, we have the following result.

Proposition 65 *Let $C = [t \bowtie s \vee D \mid \mathcal{X}]$ be an $\mathcal{A}$-clause, where $t \bowtie s \in sel(t \bowtie s \vee D)$ and $t \not\preceq_{\mathcal{A}} s$. Let σ be a ground $\mathcal{X}$-pure substitution of domain $\mathrm{var}(\mathcal{X})$. If $t\sigma \sim_{\mathcal{A}} t'$, then $t' \not\preceq_{\mathcal{A}} s\sigma$ and $(t' \bowtie s\sigma) \in sel_{\mathcal{A}}(t' \bowtie s\sigma \vee D\sigma)$.*

Proposition 66 *Let μ be an m.g.u. of $t \simeq s$.*

- *If μ_1, μ_2 are such that $\mathrm{dom}(\mu_1) \cap \mathrm{dom}(\mu_2) = \emptyset$ and $\mu = \mu_1 \uplus \mu_2$, then μ_2 is an m.g.u. of $t\mu_1 \simeq s\mu_1$.*
- *Let σ be a substitution such that $\mathrm{dom}(\sigma) \subseteq \mathrm{var}(t \simeq s)$ and $\mathrm{dom}(\sigma) \cap \mathrm{dom}(\mu) = \emptyset$. Then the restriction of $\mu\sigma$ to $\mathrm{dom}(\mu)$ is an m.g.u. of $t\sigma \simeq s\sigma$.*

Proof Since $t\mu_1\mu_2 = t\mu = s\mu = s\mu_1\mu_2$, it is clear that $t\mu_1$ and $s\mu_1$ are unifiable. If δ is a unifier of $t\mu_1 \simeq s\mu_1$, then $t\mu_1\delta = s\mu_1\delta$, hence $\mu_1\delta$ is a unifier of $t \simeq s$, and is therefore an instance of $\mu = \mu_1\mu_2$, thus δ is an instance of μ_2. This proves that μ_2 is an m.g.u. of $t\mu_1 \simeq s\mu_1$.

The second point is a consequence of the fact that for any unification problem, if $S \to S'$, then $S\sigma \to S'\sigma$. The result is proved by induction on the transformation of the unification problem $\{t \simeq s\}$.

Lemma 67 *Let S be a set of $\mathcal{A}$-clauses, E be an $\mathcal{A}$-clause and $\mathcal{U}$ be a ground $\mathcal{A}$-set.*

- *If E can be deduced from S by $\mathcal{SA}^{\prec}_{sel}$, then every clause in $\Gamma(E,\mathcal{U})$ can be deduced from $\Gamma(S,\mathcal{U}) \cup \mathcal{U}$ by $\mathcal{SP}^{\lhd_{\mathcal{A}}}_{sel_{\mathcal{A}}}$.*
- *If E is of the form $[E' \mid \mathcal{Z}]$ and there exists a clause $E'\gamma' \in \Gamma(S,\mathcal{U})$ that is strongly redundant in $\Gamma(S,\mathcal{U})$, then E is $\mathcal{A}$-redundant in S.*

Proof We prove that if the $\mathcal{A}$-Superposition rule applied to C, D generates E, then for all $E' \in \Gamma(E,\mathcal{U})$, there exists $C' \in \Gamma(C,\mathcal{U})$ and $D' \in \Gamma(D,\mathcal{U})$ such that E' can be derived from $C', D', \mathcal{U}$ by $\mathcal{SA}^{\prec}_{sel}$. The proof for the other inference rules is similar. We let

$$\begin{aligned} C &= [u \simeq v \vee C_1 \mid \mathcal{X}], \\ D &= [t \bowtie s \vee D_1 \mid \mathcal{Y}], \\ E &= [(t[v]_p \bowtie s \vee C_1 \vee D_1)\mu \mid \mathcal{Z}], \end{aligned}$$

where $(\mu, \mathcal{E})$ is an $(\mathcal{X} \cup \mathcal{Y})$-pure most general $\mathcal{A}$-unifier of $u \simeq t|_p$ and $\mathcal{Z} = (\mathcal{X} \cup \mathcal{Y} \cup \mathcal{E})\mu$. Up to a renaming, we may assume that $\mathrm{var}(\mathcal{Z}) \subseteq \mathrm{var}(\mathcal{X} \cup \mathcal{Y})$, so that for all $x \in \mathrm{dom}(\mu) \cap \mathrm{var}(\mathcal{X} \cup \mathcal{Y})$, $x\mu \in \mathcal{A} \cup \mathrm{var}(\mathcal{X} \cup \mathcal{Y})$. Let $E' \in \Gamma(\{[E \mid \mathcal{Z}]\}, \mathcal{Z})$, and let σ be a $\mathcal{Z}$-pure substitution of domain $\mathrm{var}(\mathcal{Z})$ such that $\sigma_{\downarrow\mathcal{U}} = \sigma$, $\mathcal{Z}\sigma \subseteq \mathcal{U}$ and $E' = E\sigma$. We let $\mathcal{C} \stackrel{\text{def}}{=} \mathrm{dom}(\mu) \cap \mathrm{var}(\mathcal{X} \cup \mathcal{Y})$ and define μ_1 as the restriction of μ to $\mathcal{C}$ and μ_2 as the restriction of μ to $\mathrm{dom}(\mu) \setminus \mathcal{C}$, so that $\mu = \mu_1 \uplus \mu_2$. Consider the substitution $\delta \stackrel{\text{def}}{=} \mu_1\sigma$. It is clear that δ is a ground $(\mathcal{X} \cup \mathcal{Y})$-pure substitution of domain $\mathrm{var}(\mathcal{X} \cup \mathcal{Y})$, and that $\mathcal{X}\delta, \mathcal{Y}\delta \subseteq \mathcal{Z}\sigma \subseteq \mathcal{U}$; therefore, $C' \stackrel{\text{def}}{=} C\delta \in \Gamma(\{[C \mid \mathcal{X}]\}, \mathcal{U})$ and $D' \stackrel{\text{def}}{=} D\delta \in \Gamma(\{[D \mid \mathcal{Y}]\}, \mathcal{U})$. The clause C' is of the form $u_1 \simeq v_1 \vee C'_1$, and the clause D' of the form $t_1 \bowtie s_1 \vee D'_1$, where:

- $u_1 = u\delta$, $v_1 = v\delta$ and $C'_1 = C_1\delta$,

– $t_1 = t\delta$, $s_1 = s\delta$ and $D'_1 = D_1\delta$.

Let $t'_1 \stackrel{\text{def}}{=} t_{1\downarrow\mathcal{U}}|_p$ and $u'_1 \stackrel{\text{def}}{=} u_{1\downarrow\mathcal{U}}$. By Proposition 65, $t_1 \bowtie s_1$ and $u_1 \simeq v_1$ are selected in C' and D' respectively, and we have $t_1 \not\trianglelefteq_{\mathcal{A}} s_1$, $u_1 \not\trianglelefteq_{\mathcal{A}} v_1$. Thus, there is an $\mathcal{SA}^{\prec}_{sel}$-derivation from $\{C'\} \cup \mathcal{U}$ that generates $u'_1 \simeq v_1 \vee C'_1$, and an $\mathcal{SA}^{\prec}_{sel}$-derivation from $\{D'\} \cup \mathcal{U}$ that generates $t_1[t'_1]_p \bowtie s_1 \vee D'_1$: it suffices to use repeated applications of the Superposition rule from equations in $\mathcal{U}$ to replace every constant a occurring in u_1 or $t_1|_p$ by $a_{\downarrow\mathcal{U}}$. Note that $u'_1 \simeq v_1$ and $t_1[t'_1]_p \bowtie s_1$ are both selected and that $u'_1 \not\trianglelefteq_{\mathcal{A}} v_1$ and $t_1[t'_1]_p \not\trianglelefteq_{\mathcal{A}} s_1$.

We prove that t'_1 and u'_1 are unifiable. For $i = 1, 2$, let $\gamma_i \stackrel{\text{def}}{=} \mu_{i\downarrow\mathcal{U}}$, and let $\gamma \stackrel{\text{def}}{=} \gamma_1 \uplus \gamma_2$. By Proposition 64, since $(\mu, \mathcal{E})$ is a most general $\mathcal{A}$-unifier of $t|_p \simeq s$ and $\mathcal{E} \subseteq \mathcal{U}$, γ is a most general unifier of $(t|_p)_{\downarrow\mathcal{U}} \simeq u_{\downarrow\mathcal{U}}$. By Proposition 66, γ_2 is an m.g.u. of $(t|_p)_{\downarrow\mathcal{U}}\gamma_1 \simeq u_{\downarrow\mathcal{U}}\gamma_1$, and the restriction ν of $\gamma_2\sigma$ to $\mathrm{dom}(\gamma_2)$ is an m.g.u. of $(t|_p)_{\downarrow\mathcal{U}}\gamma_1\sigma \simeq u_{\downarrow\mathcal{U}}\gamma_1\sigma$. But we have

$$(t|_p)_{\downarrow\mathcal{U}}\gamma_1\sigma \;=\; (t\mu_1\sigma)|_{p\downarrow\mathcal{U}} \;=\; (t\delta)|_{p\downarrow\mathcal{U}} \;=\; t'_1,$$

and similarly, $u_{\downarrow\mathcal{U}}\gamma_1\sigma = u'_1$. Since t'_1 and u'_1 are unifiable with m.g.u. ν, the Superposition rule applied to $u'_1 \simeq v_1 \vee C'_1$ and $t_1[t'_1]_p \bowtie s_1 \vee D'_1$ generates the clause $F \stackrel{\text{def}}{=} (t_1[v_1]_p \bowtie s_1 \vee C'_1 \vee D'_1)\nu$, and:

$$\begin{aligned} F &= (t_1[v_1]_p \bowtie s_1 \vee C'_1 \vee D'_1)\nu \\ &= (t[v]_p\delta \bowtie s\delta \vee C_1\delta \vee D_1\delta)\nu \\ &= (t[v]_p \bowtie s \vee C_1 \vee D_1)\mu_1\sigma\nu. \end{aligned}$$

We now prove that for any variable x, we have $x\mu_1\sigma\nu = x\mu\sigma$. First assume that $x \notin \mathrm{dom}(\mu_1)$. If $x \in \mathrm{var}(\mathcal{X} \cup \mathcal{Y})$, then necessarily $x \in \mathrm{var}(\mathcal{Z})$, and therefore, $x\mu = x$ and $x \in \mathrm{dom}(\sigma)$. Thus, $x\mu_1\sigma\nu = x\sigma\nu = x\sigma = x\mu\sigma$. Otherwise, since $\mathrm{dom}(\sigma) \subseteq \mathrm{var}(\mathcal{X} \cup \mathcal{Y})$, necessarily $x\sigma = x$ and $x\mu_1\sigma\nu = x\sigma\nu = x\nu$. If $x \in \mathrm{dom}(\nu)$ then $x\nu = x\mu\sigma$ by definition of ν, otherwise, since $x \notin \mathrm{dom}(\mu_1) \uplus \mathrm{dom}(\mu_2) = \mathrm{dom}(\mu)$, we deduce that $x\nu = x = x\mu = x\mu\sigma$. Now assume that $x \in \mathrm{dom}(\mu_1)$. Then $x\mu_1 = x\mu$, and if $x\mu \in \mathcal{A}$, then $x\mu_1\sigma\nu = x\mu = x\mu\sigma$. Otherwise $x\mu \in \mathrm{var}(\mathcal{Z}) = \mathrm{dom}(\sigma)$, hence $x\mu\sigma\gamma = x\mu\sigma$. This proves that $F = E\sigma = E'$.

For the second part of the lemma, let $E \stackrel{\text{def}}{=} [E' \,|\, \mathcal{Z}]$ and suppose that $\Gamma(E,\mathcal{U})$ contains a clause $E'\gamma'$ (with $\gamma'_{\downarrow\mathcal{U}} = \gamma'$) that is strongly redundant in $\Gamma(S,\mathcal{U})$. Let σ be a ground substitution of the variables in $[E' \,|\, \mathcal{Z}]$ such that $\mathcal{Z}\sigma \subseteq \mathcal{U}$. We show that $[E' \,|\, \mathcal{Z}]\sigma$ is $\mathcal{A}$-redundant in S. We assume, w.l.o.g., that $\sigma = \sigma_{\downarrow\mathcal{Z}}$. Let γ and θ be the restrictions of σ to $\mathrm{var}(\mathcal{Z})$ and $\mathrm{dom}(\sigma) \setminus \mathrm{var}(\mathcal{Z})$ respectively. By definition we have $\mathrm{dom}(\gamma) = \mathrm{var}(\mathcal{Z}) = \mathrm{dom}(\gamma')$, hence $E'\gamma' \sim_{\mathcal{A}} E\gamma$. Since $E'\gamma'$ is strongly redundant in $\Gamma(S,\mathcal{U})$ we deduce that $E'\gamma\theta = E'\sigma$ is $\mathcal{A}$-redundant in S, and therefore that $E\sigma = [E' \,|\, \mathcal{Z}]\sigma$ is also $\mathcal{A}$-redundant in S.

We denote by $\mathcal{U}_{\mathcal{A}}$ the set of all unit clauses of the form $p(a_1, \ldots, a_n) \bowtie \top$ or $a \bowtie b$, with $a_1, \ldots, a_n, a, b \in \mathcal{A}$. For any set of clauses S, we denote by $S^\star$ the set of clauses inductively defined as follows.

- $S \subseteq S^\star$.
- If C is not strongly redundant in S and is deducible from $S^\star \cup \mathcal{U}_\mathcal{A}$ by applying the rules in $\mathcal{SP}^{\lhd_\mathcal{A}}_{sel_\mathcal{A}}$ (in one step), then $C \in S^\star$.

Lemma 67 immediately entails the following:

Corollary 68 *Let S be a set of clauses. If $S^\star$ is finite then $\mathcal{SA}^{\prec}_{sel}$ terminates on S (up to redundancy).*

Proof Since $S^\star$ is finite, so is the set $\overline{S}$ of clauses obtained from S^* by the application of substitutions mapping every variable in its domain to a variable or an element in $\mathcal{A}$ (up to a renaming of variables). By Lemma 67, for every $\mathcal{A}$-clause $[E \,|\, \mathcal{Z}]$ generated from S that is not $\mathcal{A}$-redundant in S and for every $\mathcal{Z}$-pure substitution γ, we have $E\gamma \in \overline{S}$, hence the result.

In order to prove that $\mathcal{SA}^{\prec}_{sel}$ terminates on some class of clause sets $\mathfrak{S}$, it suffices to prove that $S^\star$ is finite, for every $S \in \mathfrak{S}$. Since all the clauses in $S^\star$ are generated from S by Superposition, this entails that if the Superposition calculus terminates on some class of clauses $\mathfrak{S}$, and if $\mathfrak{S}$ contains all equations between constants (the set $\mathcal{U}_\mathcal{A}$ above) then $\mathcal{SA}^{\prec}_{sel}$ also terminates on $\mathfrak{S}$. The complexity results, however, do not necessarily carry over, because the set of unit flat clauses $\mathcal{U}$ in Lemma 67 can be arbitrary, and these sets may be generated as constraints using $\mathcal{SA}^{\prec}_{sel}$.

To be precise, it is important to emphasize that the calculus $\mathcal{SP}^{\lhd_\mathcal{A}}_{sel}$ is actually slightly less restrictive than the usual Superposition calculus $\mathcal{SP}^{\prec}_{sel}$, since $\lhd_\mathcal{A}$ is a stronger relation than $\prec$. However, most of the usual termination results for the Superposition calculus still hold for $\mathcal{SP}^{\lhd_\mathcal{A}}_{sel}$, because they are closed under the addition of equalities between constants and do not depend on the order of $\sim_\mathcal{A}$-equivalent terms. Similarly, redundancy testing is usually restricted to subsumption and tautology detection. In particular, all the termination results described by Armando, Ranise, and Rusinowitch (2003) are preserved (it is easy to check that $S^\star$ is finite for the considered sets of axioms).

An interesting continuation of the present work would be to devise formal (automated) proofs of the termination of $\mathcal{SA}^{\prec}_{sel}$ on the usual theories of interest in program verification, enriched by arbitrary ground clauses. This could be done by using existing schematic calculi (see, e.g., Lynch and Morawska, 2002; Lynch et al, 2011; Tushkanova et al, 2013) to compute a symbolic representation of the set of $\mathcal{A}$-clauses $S^\star$.

7 Conclusion and Discussion

Although the Superposition calculus is not deductive-complete in general, we have shown that it can be adapted in order to make it able to generate all implicates defined over a given *finite* set of *ground* terms denoted by constant symbols, using a finite set of predicate symbols including the equality predicate. Furthermore, this is done in such a way that the usual termination

properties of the calculus are preserved. By duality, the procedure can be used to generate abductive explanations of first-order formulæ.

A major restriction of our approach is that it cannot handle built-in theories such as arithmetics which play an essential role in verification. Axiomatizing these theories in first-order logic is often infeasible or inefficient. A natural follow-up of this work is therefore to make the procedure able to cooperate with external decision procedures. This can be done for instance by combining our approach with existing techniques for fusing the Superposition calculus and external reasoning tools (Bachmair et al, 1994; Althaus et al, 2009; Baumgartner and Waldmann, 2013; Baumgartner et al, 2014). These techniques, based on the use of constrained Superposition together with an abstraction of the terms of the considered theory, should be easy to combine with $\mathcal{A}$-Superposition. Note that our calculus has many common points with the above-mentioned constrained Superposition calculi, however in our case the constraint and clausal parts are not defined over disjoint signatures: in contrast the $\mathcal{A}$-unification and Assertion rules allow one to transfer literals from the clausal part to the constraints. In other approaches (Bachmair et al, 1994) the constraints are used to store formulæ that cannot be handled by the Superposition calculus, whereas in our case they are used to store properties that are *asserted* instead of being proved.

Another obvious drawback with the calculi $\mathcal{SA}^{\prec}_{sel}$ and $\mathcal{SAR}^{\prec}_{sel}$ is that the user has to explicitly declare the set of abducible terms (i.e., the constants in $\mathcal{A}$). This set must be finite and must contain all built-in constants. Note that, thanks to the results in Section 5, unsatisfiable or irrelevant implicates (such as $0 \simeq 1$) can be easily detected and discarded on the fly during proof search. Handling infinite (but recursive) sets of terms is possible from a theoretical point of view: it suffices to add an inference rule generating clauses of the form $a \simeq t$, where t is an abducible ground terms and a is a fresh abducible constant symbol. It is easy to see that completeness is preserved, but of course termination is lost. A way to recover termination is to develop additional techniques to restrict the application of this rule by selecting the terms t. This could be done either statically, from the initial set of clauses, or dynamically, from the information deduced during proof search.

Another possible extension would be to generate "mixed" implicates, containing both abducible and non-abducible terms, which would avoid having to declare built-in constants as abducible. An alternative approach consists in avoiding to have to explicitly declare abducible terms, by adding rules for generating them symbolically (as the $\mathcal{A}$-Substitutivity rule does for predicate symbols). For termination, additional conditions should be added to ensure that the set of abducible terms is finite (using, e.g., sort constraints).

Another restriction is that our method does not handle non-ground abducible terms, hence cannot generate quantified formulæ. We are currently investigating these issues.

References

Althaus E, Kruglov E, Weidenbach C (2009) Superposition modulo linear arithmetic sup(la). In: Ghilardi S, Sebastiani R (eds) FroCoS 2009, Springer, LNCS, vol 5749, pp 84–99

Armando A, Ranise S, Rusinowitch M (2003) A rewriting approach to satisfiability procedures. Information and Computation 183(2):140–164

Armando A, Bonacina MP, Ranise S, Schulz S (2009) New results on rewrite-based satisfiability procedures. ACM Transactions on Computational Logic 10(1):129–179

Baader F, Nipkow T (1998) Term Rewriting and All That. Cambridge University Press

Bachmair L, Ganzinger H (1994) Rewrite-based equational theorem proving with selection and simplification. Journal of Logic and Computation 3(4):217–247

Bachmair L, Ganzinger H, Waldmann U (1994) Refutational theorem proving for hierarchic first-order theories. Applicable Algebra in Engineering, Communication and Computing 5(3):193–212

Baumgartner P, Waldmann U (2013) Hierarchic superposition with weak abstraction. In: Bonacina (2013), pp 39–57

Baumgartner P, Bax J, Waldmann U (2014) Finite quantification in hierarchic theorem proving. In: Demri S, Kapur D, Weidenbach C (eds) IJCAR, Springer, Lecture Notes in Computer Science, vol 8562, pp 152–167

Bonacina MP (ed) (2013) Automated Deduction - CADE-24 - 24th International Conference on Automated Deduction, Lake Placid, NY, USA, June 9-14, 2013. Proceedings, Lecture Notes in Computer Science, vol 7898, Springer

Bouton T, de Oliveira DCB, Déharbe D, Fontaine P (2009) verit: an open, trustable and efficient smt-solver. In: Schmidt RA (ed) Proc. Conference on Automated Deduction (CADE), Springer-Verlag, Lecture Notes in Computer Science

Caferra R, Leitsch A, Peltier N (2004) Automated Model Building, Applied Logic Series, vol 31. Kluwer Academic Publishers

De Kleer J (1992) An improved incremental algorithm for generating prime implicates. In: Proceedings of the National Conference on Artificial Intelligence, John Wiley & Sons ltd, pp 780–780

Dillig I, Dillig T, Aiken A (2010) Small formulas for large programs: On-line constraint simplification in scalable static analysis. In: Cousot R, Martel M (eds) SAS, Springer, Lecture Notes in Computer Science, vol 6337, pp 236–252

Dillig I, Dillig T, McMillan KL, Aiken A (2012) Minimum satisfying assignments for smt. In: Madhusudan P, Seshia SA (eds) CAV, Springer, Lecture Notes in Computer Science, vol 7358, pp 394–409

Dutertre D, de Moura L (2006) The YICES SMT-solver. In SMT-COMP: Satisfiability Modulo Theories Competition. Available at http://yices.csl.sri.com.

Echenim M, Peltier N (2012) A Calculus for Generating Ground Explanations. In: Proceedings of the International Joint Conference on Automated Reasoning (IJCAR'12), Springer LNCS, vol 7364, pp 194–209

Echenim M, Peltier N, Tourret S (2013) An approach to abductive reasoning in equational logic. In: Proceedings of IJCAI'13 (International Conference on Artificial Intelligence), AAAI, pp 3–9

Echenim M, Peltier N, Tourret S (2014) A Rewriting Strategy to Generate Prime Implicates in Equational Logic. In: Proceedings of the International Joint Conference on Automated Reasoning (IJCAR'14), Springer

Henocque L (2002) The prime normal form of boolean formulas. Technical report at http://wwwIsisorg/fichephp

Jackson P, Pais J (1990) Computing prime implicants. In: 10th International Conference on Automated Deduction, Springer, pp 543–557

Kean A, Tsiknis G (1990) An incremental method for generating prime implicants/implicates. Journal of Symbolic Computation 9(2):185–206

Knill E, Cox P, Pietrzykowski T (1992) Equality and abductive residua for horn clauses. Theoretical Computer Science 120:1–44

Leitsch A (1997) The resolution calculus. Springer. Texts in Theoretical Computer Science

Lynch C, Morawska B (2002) Automatic Decidability. In: Proc. of 17th IEEE Symposium on Logic in Computer Science (LICS'2002), IEEE Computer Society, Copenhagen, Denmark, pp 7–16

Lynch C, Ranise S, Ringeissen C, Tran DK (2011) Automatic decidability and combinability. Information and Computation 209(7):1026 – 1047

Marquis P (1991) Extending abduction from propositional to first-order logic. In: Jorrand P, Kelemen J (eds) FAIR, Springer, Lecture Notes in Computer Science, vol 535, pp 141–155

Matusiewicz A, Murray N, Rosenthal E (2009) Prime implicate tries. Automated Reasoning with Analytic Tableaux and Related Methods pp 250–264

Matusiewicz A, Murray N, Rosenthal E (2011) Tri-based set operations and selective computation of prime implicates. Foundations of Intelligent Systems pp 203–213

Mayer MC, Pirri F (1993) First order abduction via tableau and sequent calculi. Logic Journal of the IGPL 1(1):99–117

McCarthy J (1962) Computer programs for checking mathematical proofs. In: Recursive Function Theory, Proc. of Symposia in Pure Mathematics, Volume 5, American Mathematical Society, Providence, Rhode Island, pp 219–228

Meir O, Strichman O (2005) Yet another decision procedure for equality logic. In: Proceedings of the 17th International Conference on Computer Aided Verification, Springer-Verlag, Berlin, Heidelberg, CAV'05, pp 307–320

de Moura LM, Bjørner N (2008) Z3: An Efficient SMT Solver. In: Ramakrishnan CR, Rehof J (eds) TACAS, Springer, LNCS, vol 4963, pp 337–340

Nieuwenhuis R, Rubio A (2001) Paramodulation-based theorem proving. In: Robinson JA, Voronkov A (eds) Handbook of Automated Reasoning, Elsevier and MIT Press, pp 371–443

(T)	$(t \simeq t \cup S, \theta, \mathcal{Z})$	$\rightarrow$	$(S, \theta, \mathcal{Z})$
(E)	$(a \simeq b \cup S, \theta, \mathcal{Z})$	$\rightarrow$	$(S, \theta, \mathcal{Z} \cup \{a \simeq b\})$ if $a, b \in \mathcal{A}$
(C)	$(\{f(t_1, \ldots, t_n) \simeq g(s_1, \ldots, s_m)\} \cup S, \theta, \mathcal{Z})$	$\rightarrow$	$\bot$ if $f \neq g$
(O)	$(\{x \simeq t[x]_p\} \cup S, \theta, \mathcal{Z})$	$\rightarrow$	$\bot$
(R)	$(\{x \simeq t\} \cup S, \theta, \mathcal{Z})$	$\rightarrow$	$(S\{x \mapsto t\}, \theta \cup \{x \mapsto t\}, \mathcal{Z})$
(D)	$(\{f(t_1, \ldots, t_n) \simeq f(s_1, \ldots, s_n)\} \cup S, \theta, \mathcal{Z})$	$\rightarrow$	$(\bigcup_{i=1}^{n} \{t_i \simeq s_i\} \cup S, \theta, \mathcal{Z})$

Fig. 2: $\mathcal{A}$-unification rules

Simon L, Del Val A (2001) Efficient consequence finding. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence, pp 359–370

Sofronie-Stokkermans V (2010) Hierarchical reasoning for the verification of parametric systems. In: Giesl J, Hähnle R (eds) IJCAR, Springer, LNCS, vol 6173, pp 171–187

Sofronie-Stokkermans V (2013) Hierarchical reasoning and model generation for the verification of parametric hybrid systems. In: Bonacina (2013), pp 360–376

Tison P (1967) Generalization of consensus theory and application to the minimization of boolean functions. Electronic Computers, IEEE Transactions on 4:446–456

Tran DK, Ringeissen C, Ranise S, Kirchner H (2010) Combination of convex theories: Modularity, deduction completeness, and explanation. J Symb Comput 45(2):261–286

Tushkanova E, Ringeissen C, Giorgetti A, Kouchnarenko O (2013) Automatic decidability: A schematic calculus for theories with counting operators. In: van Raamsdonk F (ed) RTA, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, LIPIcs, vol 21, pp 303–318

Wernhard C (2013) Abduction in logic programming as second-order quantifier elimination. In: Fontaine P, Ringeissen C, Schmidt R (eds) Frontiers of Combining Systems, Lecture Notes in Computer Science, vol 8152, Springer Berlin Heidelberg, pp 103–119

A $\mathcal{A}$-Unification

Definition 69 An *$\mathcal{A}$-unification problem* is either $\bot$ or a triple $(S, \theta, \mathcal{Z})$ where S is a set of equations, θ is a substitution such that $x\theta = x$ for every variable x occurring in S and $\mathcal{Z}$ is a positive $\mathcal{A}$-set. A pair $(\sigma, \mathcal{Z})$ is a *solution* of an $\mathcal{A}$-unification problem $(S, \theta, \mathcal{Z}')$ iff $(\theta, \mathcal{Z}') \geq_{\mathcal{A}} (\sigma, \mathcal{Z})$ and $(\sigma, \mathcal{Z})$ is an $\mathcal{A}$-unifier of every $t \simeq s \in S$. An $\mathcal{A}$-unification problem is *satisfiable* if it has a solution.

The set of *$\mathcal{A}$-unification rules* is the set of rules depicted in Figure 2. They are almost identical to the standard unification rules, except that equations of the form $a \simeq b$ where $a \neq b$ do not lead to failure but are instead stored in $\mathcal{Z}$. We assume that the rules are applied in the specified order, i.e., a rule applies only if the previous rules do not apply. Note that, following our convention, $\mathcal{Z} \cup \{a \simeq b\}$ actually denotes the smallest $\mathcal{A}$-set containing $\mathcal{Z}$ and $a \simeq b$ (obtained by transitive closure from $\mathcal{Z} \cup \{a \simeq b\}$).

Lemma 70 *The $\mathcal{A}$-unification rules preserve the set of solutions of the considered problem.*

Proof The proof is by an easy inspection of each rule (see Figure 2 for the notations):

(T) We have $t\sigma \sim_{\mathcal{A}}^{\mathcal{Z}'} t\sigma$, for all $t, \sigma, \mathcal{Z}'$; hence removing the equation $t \simeq t$ does not affect the set of solutions.

(E) Since (T) is not applicable, a and b are distinct. Thus we have $a\sigma \sim_{\mathcal{A}}^{\mathcal{Z}'} b\sigma$ iff $a \simeq b \in \mathcal{Z}'$. Consequently, adding the equation $a \simeq b$ to the last component of the problem does not affect the set of solutions. Furthermore, every $\mathcal{A}$-substitution $(\sigma, \mathcal{Z}')$ such that $\mathcal{Z} \cup \{a \simeq b\} \subseteq \mathcal{Z}'$ is an $\mathcal{A}$-unifier of a and b, thus the equation can be removed from the first component of the problem once it has been added to the last component.

(C) Due to the ordering of the rules, f and g must be distinct and cannot both occur in $\mathcal{A}$, since otherwise, one of (T) or (E) would apply first. Thus, the problem has no solution, since by definition of the relation $\sim_{\mathcal{A}}^{\mathcal{Z}'}$, we have $f(t_1, \ldots, t_n)\sigma \not\sim_{\mathcal{A}}^{\mathcal{Z}'} g(s_1, \ldots, s_m)\sigma$, for all $\sigma, \mathcal{Z}'$.

(O) Due to the ordering of the rules, p cannot be empty, since otherwise (T) would apply first, thus $x\sigma$ contains strictly less positions than $t\sigma$. Hence $x\sigma \not\sim_{\mathcal{A}}^{\mathcal{Z}} t\sigma$, for all $\sigma, \mathcal{Z}$, and the problem has no solution.

(R) If an $\mathcal{A}$-substitution $(\sigma, \mathcal{Z}')$ is a unifier of x and t then necessarily $x\sigma \sim_{\mathcal{A}}^{\mathcal{Z}'} t\sigma$. Thus adding the mapping $x \mapsto t$ to the substitution and replacing x by t does not affect the set of solutions. Afterwards the equation $x \simeq t$ becomes trivial and can be removed.

(D) If is clear that $f(t_1, \ldots, t_n)\sigma \sim_{\mathcal{A}}^{\mathcal{Z}} f(s_1, \ldots, s_n)\sigma$ holds iff for all $i \in [1, n]$, $t_i\sigma \sim_{\mathcal{A}}^{\mathcal{Z}} s_i\sigma$ holds. Thus the replacement of the equation $f(t_1, \ldots, t_n) \simeq f(s_1, \ldots, s_n)$ by the set $\{t_i \simeq s_i \mid i \in [1, n]\}$ preserves the set of solutions.

Corollary 71 *Every satisfiable $\mathcal{A}$-unification problem has a most general $\mathcal{A}$-unifier, which is unique up to $\sim_{\mathcal{A}}$-equivalence.*

Proof It is easy to check that the $\mathcal{A}$-unification rules terminate: all the rules strictly decrease the size of the first component of the problem, except for (R), which strictly decreases the number of variables occurring in the first component (moreover, no rule can increase this number of variables). Furthermore, irreducible problems are either $\bot$ or of the form $(\emptyset, \sigma, \mathcal{Z})$. In the former case the problem has no solution and in the latter, $(\sigma, \mathcal{Z})$ is a most general solution. Also, if $(\sigma, \mathcal{Z})$ and $(\sigma', \mathcal{Z}')$ are two most general solutions then by definition we have $(\sigma, \mathcal{Z}) \geq_{\mathcal{A}} (\sigma', \mathcal{Z}')$ and $(\sigma', \mathcal{Z}') \geq_{\mathcal{A}} (\sigma, \mathcal{Z})$, thus $(\sigma, \mathcal{Z}) \sim_{\mathcal{A}} (\sigma', \mathcal{Z}')$.

Note that the proposed algorithm is exponential w.r.t. the size of the initial problem, however it can be easily transformed into a polynomial algorithm by using structure sharing (thus avoiding any duplication of terms).