Soundness, Idempotence and Commutativity of Set-Sharing

PATRICIA M. HILL*
School of Computer Studies, University of Leeds, Leeds, U.K.
(*e-mail:* `hill@scs.leeds.ac.uk`)

ROBERTO BAGNARA, ENEA ZAFFANELLA†
Department of Mathematics, University of Parma, Italy
(*e-mail:* `{bagnara,zaffanella}@cs.unipr.it`)

Abstract

It is important that practical data-flow analyzers are backed by reliably proven theoretical results. Abstract interpretation provides a sound mathematical framework and necessary generic properties for an abstract domain to be well-defined and sound with respect to the concrete semantics. In logic programming, the abstract domain Sharing is a standard choice for sharing analysis for both practical work and further theoretical study. In spite of this, we found that there were no satisfactory proofs for the key properties of commutativity and idempotence that are essential for Sharing to be well-defined and that published statements of the soundness of Sharing assume the occur-check. This paper provides a generalization of the abstraction function for Sharing that can be applied to any language, with or without the occur-check. Results for soundness, idempotence and commutativity for abstract unification using this abstraction function are proven.

1 Introduction

In abstract interpretation, the concrete semantics of a program is approximated by an abstract semantics; that is, the concrete domain is replaced by an abstract domain and each elementary operation on the concrete domain is replaced by a corresponding abstract operation on the abstract domain. Assuming the global abstract procedure mimics the concrete execution procedure, each basic operation on the elements of the abstract domain must produce a safe approximation of the corresponding operation on corresponding elements of the concrete domain. For logic programming, the key elementary operation is *unification* that computes a solution to a set of equations. This solution can be represented by means of a mapping (called a *substitution*) from variables to first-order terms in the language. For global soundness of the abstract semantics, there needs to be, therefore, a corresponding abstract operation, *aunify*, that is sound with respect to unification.

* This work was partly supported by EPSRC under grant GR/M05645.
† The work of the second and third authors has been partly supported by MURST project "Certificazione automatica di programmi mediante interpretazione astratta."

For parallelization and several other program optimizations, it is important to know before execution which variables may be bound to terms that share a common variable. Jacobs and Langen developed the abstract domain Sharing (Jacobs & Langen, 1989; Jacobs & Langen, 1992) for representing and propagating the sharing behavior of variables and this is now a standard choice for sharing analysis. Subsequent research then concentrated mainly on extending the domain to incorporate additional properties such as linearity, freeness and depth-k abstractions (Langen, 1990; Bruynooghe & Codish, 1993; Codish *et al.*, 1993; King, 1994; King & Soper, 1994; Muthukumar & Hermenegildo, 1992) or in reducing its complexity (Bagnara *et al.*, 1997; Bagnara *et al.*, 2000b). Key properties such as commutativity and soundness of this domain and its associated abstract operations such as abstract unification were normally assumed to hold. One reason for this was that (Jacobs & Langen, 1992) includes a proof of the soundness and refers to the Ph.D. thesis of Langen (Langen, 1990) for the proofs of commutativity and idempotence.[1] We discuss below why these results are inadequate.

1.1 Soundness of aunify

An important step in standard unification algorithms based on that of Robinson (Robinson, 1965) (such as the Martelli-Montanari algorithm (Martelli & Montanari, 1982)) is the *occur-check*, which avoids the generation of infinite (or cyclic) data structures. With such algorithms, the resulting solution is both unique and idempotent. However, in computational terms, the occur-check is expensive and the vast majority of Prolog implementations omit this test, although some Prolog implementations do offer unification with the occur-check as a separate built-in predicate (in ISO Prolog (ISO/IEC, 1995) the predicate is `unify_with_occurs_check/2`). In addition, if the unification algorithm is based on the Martelli-Montanari algorithm but without the occur-check step, then the resulting solution may be non-idempotent. Consider the following example.

Suppose we are given as input the equation $p\bigl(z, f(x,y)\bigr) = p\bigl(f(z,y), z\bigr)$ with an initial substitution that is empty. We apply the steps in the Martelli-Montanari procedure but without the occur-check:

	equations	substitution
1	$p(z, f(x,y)) = p(f(z,y), z)$	$\varnothing$
2	$z = f(z,y), f(x,y) = z$	$\varnothing$
3	$f(x,y) = f(z,y)$	$\bigl\{z \mapsto f(z,y)\bigr\}$
4	$x = z, y = y$	$\bigl\{z \mapsto f(z,y)\bigr\}$

[1] Even though the thesis of Langen has been published as a technical report of the University of Southern California, an extensive survey of the literature on Sharing indicates that the thesis has not been widely circulated even among researchers in the field. For instance, Langen is rarely credited as being the first person to integrate Sharing with linearity information, despite the fact that this is described in the thesis.

5	$y = y$	$\{z \mapsto f(z,y), x \mapsto z\}$
6	$\varnothing$	$\{z \mapsto f(z,y), x \mapsto z\}$

Then $\sigma = \{z \mapsto f(z,y), x \mapsto z\}$ is the computed substitution; it is not idempotent since, for example, $x\sigma = z$ and $x\sigma\sigma = f(z,y)$.

Non-standard equality theories and unification procedures are also available and used in many logic programming systems. In particular, there are theoretically coherent languages, such as Prolog III (Colmerauer, 1982), that employ an equality theory and unification algorithm based on a theory of *rational trees* (possibly infinite trees with a finite number of subtrees). As remarked in (Colmerauer, 1982), complete (i.e., always terminating) unification with the omission of the occur-check solves equations over rational trees. Complete unification is made available by several Prolog implementations. The substitutions computed by such systems are in *rational solved form* and therefore not necessarily idempotent. As an example, the substitution $\{x \mapsto f(x)\}$, which is clearly non-idempotent, is in rational solved form and could itself be computed by the above algorithms.

It is therefore important that theoretical work in data-flow analysis makes no assumption that the computed solutions are idempotent. In spite of this, most theoretical work on data-flow analysis of logic programming *and* of Prolog assume the occur-check is performed, thus allowing idempotent substitutions only. In particular, (Jacobs & Langen, 1992), (Langen, 1990), and, more recently, (Cortesi & Filé, 1999) make this assumption in their proofs of soundness. As a consequence, their results do not apply to the analysis of all Prolog programs. A recent exception to this is (King, 2000) where a soundness result is proved for a domain representing just the pair-sharing and linearity information. In this work it is assumed that a separate groundness analysis is performed and its results are used to recover from the precision losses incurred by the proposed domain. However, the problem of specifying a sound and precise groundness analysis when dealing with possibly non-idempotent substitutions is completely disregarded, so that the overall solution is incomplete. Moreover, the proposed abstraction function is based on a limit operation that, in the general case, is not finitely computable.

We have therefore addressed the problem of defining a sound and precise approximation of the sharing information contained in a substitution in rational solved form.

In particular, we observed that the Sharing domain is concerned with the set of variables occurring in a term, rather than with the term structure. We have therefore generalized the notion of idempotence to *variable-idempotence*. That is, if σ is a variable-idempotent substitution and t is any term, then any variable which is not in the domain of σ and occurs in $t\sigma$ also occurs in $t\sigma\sigma$. Clearly, as illustrated by the above example, substitutions generated by unification algorithms without the occur-check may not even be variable-idempotent. To resolve this, we have devised an algorithm that transforms any substitution in rational solved form to an equivalent (with respect to any equality theory) variable-idempotent substitution. For instance, in the example, it would transform σ to $\{z \mapsto f(z,y), x \mapsto f(z,y)\}$.

By suitably exploiting the properties enjoyed by variable-idempotent substitutions, we show that, for the domain Sharing, the abstract unification algorithm aunify is sound with respect to the actually implemented unification procedures for all logic programming languages. Moreover, we define a new abstraction function mapping any set of substitutions in rational solved form into the corresponding abstract descriptions so that there is no need for the analyser to compute the equivalent set of variable-idempotent substitutions. We note that this new abstraction function is carefully chosen so as to avoid any precision loss due to the possible non-idempotence of the substitution.

Note that both the notion of variable-idempotent substitution and the proven results relating it to arbitrary substitutions in rational solved form do not depend on the particular abstract domain considered. Indeed, we believe that this concept, perhaps with minor adjustments, can be usefully applied to other abstract domains when extending the soundness proofs devised for idempotent substitutions to the more general case of substitutions in rational solved form.

1.2 Commutativity and Idempotence of aunify

A substitution is defined as a *set* of bindings or equations between variables and other terms. Thus, for the concrete domain, the order and multiplicity of elements are irrelevant in both the computation and semantics of unification. It is therefore useful that the abstraction of the unification procedure should be unaffected by the order and multiplicity in which it abstracts the bindings that are present in the substitution. Furthermore, from a practical perspective, it is also useful if the global abstract procedure can proceed in a different order with respect to the concrete one without affecting the accuracy of the analysis results. On the other hand, as sharing is normally combined with linearity and freeness domains that are not idempotent or commutative (Langen, 1990; Bruynooghe & Codish, 1993; King, 1994), it may be asked why these properties are still important for sharing analysis. In answer to this, we observe that the order and multiplicity in which the bindings in a substitution are analyzed affects the accuracy of the linearity and freeness information. It is therefore a real advantage to be able to ignore these aspects as far as the sharing domain is concerned. Specifically, the order in which the bindings are analyzed can be chosen so as to improve the accuracy of linearity and freeness. We thus conclude that it is extremely desirable that aunify is also commutative and idempotent.

We found that there was no satisfactory proof of commutativity. In addition, for idempotence the only previous result was given in Theorem 32 of the thesis of Langen. However, his definition of abstract unification includes the renaming and projection operations and, in this case, only a weak form of idempotence holds. In fact, for the basic aunify operation as defined here and without projection and renaming, idempotence has never before been proven. We therefore provide here the first published proofs of these properties.

In summary, this paper, which is an extended and improved version of (Hill *et al.*, 1998), provides a generalization of the abstraction function for Sharing that can be applied to any logic programming language dealing with syntactic term structures.

The results for soundness, idempotence and commutativity for abstract unification using this abstraction function are proved.

The paper is organised as follows. In the next section, the notation and definitions needed for equality and substitutions in the concrete domain are given. In Section 3, we recall the definition of the domain Sharing and of the classical abstraction function defined for idempotent substitutions. We also show why this abstraction function cannot be applied, as is, to non-idempotent substitutions. In Section 4, we introduce *variable-idempotence* and provide a transformation that may be used to map any substitution in rational solved form to an equivalent, variable-idempontent one. In Section 5, we define a new abstraction function relating the Sharing domain to the domain of arbitrary substitutions in rational solved form. In Section 6, we recall the definition of the abstract unification for Sharing and state our main results. Section 7 concludes. For the convenience of the reader, throughout the paper all the proofs (apart from the simpler ones) of the stated results are appended to the end of the corresponding section.

2 Equations and Substitutions

In this section we introduce the notation and some terminology concerning equality and substitutions that will be used in the rest of the paper.

2.1 Notation

For a set S, $\wp(S)$ is the powerset of S, whereas $\wp_{\mathrm{f}}(S)$ is the set of all the *finite* subsets of S. The symbol *Vars* denotes a denumerable set of variables, whereas $\mathcal{T}_{Vars}$ denotes the set of first-order terms over *Vars* for some given set of function symbols. It is assumed that there are at least two distinct function symbols, one of which is a constant (i.e., of zero arity), in the given set. The set of variables occurring in a syntactic object o is denoted by $vars(o)$. To simplify the expressions in the paper, any variable in a formula that is not in the scope of a quantifier is assumed to be universally quantified. To prove the results in the paper, it is useful to assume a total ordering, denoted with '$\leq$', on *Vars*.

2.2 Substitutions

If $x \in Vars$ and $s \in \mathcal{T}_{Vars} \setminus \{x\}$, then $x \mapsto s$ is called a *binding*. The set of all bindings is denoted by *Bind*. A *substitution* is a total function $\sigma \colon Vars \rightarrow \mathcal{T}_{Vars}$ that is the identity almost everywhere; in other words, the *domain* of σ,

$$\mathrm{dom}(\sigma) \stackrel{\mathrm{def}}{=} \{\, x \in Vars \mid \sigma(x) \neq x \,\},$$

is finite. We also define the set of *parameter variables* of a substitution σ as

$$\mathrm{param}(\sigma) \stackrel{\mathrm{def}}{=} vars(\sigma) \setminus \mathrm{dom}(\sigma).$$

Given a substitution $\sigma\colon \mathit{Vars} \to \mathcal{T}_{\mathit{Vars}}$ we overload the symbol 'σ' so as to denote also the function $\sigma\colon \mathcal{T}_{\mathit{Vars}} \to \mathcal{T}_{\mathit{Vars}}$ defined as follows, for each term $t \in \mathcal{T}_{\mathit{Vars}}$:

$$\sigma(t) \stackrel{\mathrm{def}}{=} \begin{cases} t, & \text{if } t \text{ is a constant symbol;} \\ \sigma(t), & \text{if } t \in \mathit{Vars}; \\ f\bigl(\sigma(t_1), \ldots, \sigma(t_n)\bigr), & \text{if } t = f(t_1, \ldots, t_n). \end{cases}$$

Substitutions are denoted by the set of their non-trivial bindings, thus a substitution σ is identified with the (finite) set

$$\bigl\{\, x \mapsto \sigma(x) \bigm| x \in \mathrm{dom}(\sigma) \,\bigr\}.$$

If $t \in \mathcal{T}_{\mathit{Vars}}$, we write $t\sigma$ to denote $\sigma(t)$ and $t[x/s]$ to denote $t\{x \mapsto s\}$. The composition of substitutions is defined in the usual way. Thus $\tau \circ \sigma$ is the substitution such that, for all terms $t \in \mathcal{T}_{\mathit{Vars}}$,

$$(\tau \circ \sigma)(t) = \tau\bigl(\sigma(t)\bigr)$$

and has the formulation

$$\tau \circ \sigma = \bigl\{x \mapsto x\sigma\tau \bigm| x \in \mathrm{dom}(\sigma), x \neq x\sigma\tau\bigr\} \cup \bigl\{x \mapsto x\tau \bigm| x \in \mathrm{dom}(\tau) \setminus \mathrm{dom}(\sigma)\bigr\}. \tag{1}$$

As usual, σ^0 denotes the identity function (i.e., the empty substitution) and, when $i > 0$, σ^i denotes the substitution $(\sigma \circ \sigma^{i-1})$.

A substitution is said to be *circular* if, for $n > 1$, it has the form

$$\{x_1 \mapsto x_2, \ldots, x_{n-1} \mapsto x_n, x_n \mapsto x_1\}.$$

A substitution is in *rational solved form* if it has no circular subset. The set of all substitutions in rational solved form is denoted by *RSubst*. A substitution σ is *idempotent* if, for all $t \in \mathcal{T}_{\mathit{Vars}}$, we have $t\sigma\sigma = t\sigma$. The set of all idempotent substitutions is denoted by *ISubst* and $\mathit{ISubst} \subset \mathit{RSubst}$.

Example 1

The following hold:

$$\begin{aligned} \bigl\{x \mapsto y, y \mapsto a\bigr\} &\in \mathit{RSubst} \setminus \mathit{ISubst}, \\ \bigl\{x \mapsto a, y \mapsto a\bigr\} &\in \mathit{ISubst}, \\ \bigl\{x \mapsto y, y \mapsto g(y)\bigr\} &\in \mathit{RSubst} \setminus \mathit{ISubst}, \\ \bigl\{x \mapsto y, y \mapsto g(x)\bigr\} &\in \mathit{RSubst} \setminus \mathit{ISubst}, \\ \bigl\{x \mapsto y, y \mapsto x\bigr\} &\notin \mathit{RSubst}, \\ \bigl\{x \mapsto y, y \mapsto x, z \mapsto a\bigr\} &\notin \mathit{RSubst}. \end{aligned}$$

We have assumed that there is a total ordering '$\leq$' for *Vars*. We say that $\sigma \in \mathit{RSubst}$ is *ordered* (with respect to this ordering) if, for each binding $(v \mapsto w) \in \sigma$ such that $w \in \mathrm{param}(\sigma)$, we have $w < v$.

2.3 Equations

An *equation* is of the form $s = t$ where $s, t \in \mathcal{T}_{Vars}$. Eqs denotes the set of all equations. A substitution σ may be regarded as a finite set of equations, that is, as the set $\{\, x = t \mid x \mapsto t \in \sigma \,\}$. We say that a set of equations e is in *rational solved form* if $\bigl\{\, r \mapsto t \bigm| (r = t) \in e \,\bigr\} \in \mathit{RSubst}$. In the rest of the paper, we will often write a substitution $\sigma \in \mathit{RSubst}$ to denote a set of equations in rational solved form (and vice versa).

We assume that any equality theory T over $\mathcal{T}_{Vars}$ includes the *congruence axioms* denoted by the following schemata:

$$s = s, \tag{2}$$

$$s = t \leftrightarrow t = s, \tag{3}$$

$$r = s \wedge s = t \rightarrow r = t, \tag{4}$$

$$s_1 = t_1 \wedge \cdots \wedge s_n = t_n \rightarrow f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n). \tag{5}$$

In logic programming and most implementations of Prolog it is usual to assume an equality theory based on syntactic identity. This consists of the congruence axioms together with the *identity axioms* denoted by the following schemata:

$$f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n) \rightarrow s_1 = t_1 \wedge \cdots \wedge s_n = t_n, \tag{6}$$

$$\neg\bigl(f(s_1, \ldots, s_n) = g(t_1, \ldots, t_m)\bigr). \tag{7}$$

The axioms characterized by schemata (6) and (7) ensure the equality theory depends only on the syntax. The equality theory for a non-syntactic domain replaces these axioms by ones that depend instead on the semantics of the domain and, in particular, on the interpretation given to functor symbols.

The equality theory of Clark (Clark, 1978) on which pure logic programming is based, usually called the *Herbrand* equality theory, is given by the congruence axioms, the identity axioms, and the axiom schema

$$\forall z \in \mathit{Vars} : \forall t \in (\mathcal{T}_{Vars} \setminus \mathit{Vars}) : z \in \mathit{vars}(t) \rightarrow \neg(z = t). \tag{8}$$

Axioms characterized by the schema (8) are called the *occur-check axioms* and are an essential part of the standard unification procedure in SLD-resolution.

An alternative approach used in some implementations of Prolog, does not require the occur-check axioms. This approach is based on the theory of rational trees (Colmerauer, 1982; Colmerauer, 1984). It assumes the congruence axioms and the identity axioms together with a *uniqueness axiom* for each substitution in rational solved form. Informally speaking these state that, after assigning a ground rational tree to each parameter variable, the substitution uniquely defines a ground rational tree for each of its domain variables. Note that being in rational solved form is a very weak property. Indeed, unification algorithms returning a set of equations in rational solved form are allowed to be much more "lazy" than one would usually expect (e.g., see the first substitution in Example 1). We refer the interested reader to (Jaffar *et al.*, 1987; Keisu, 1994; Maher, 1988) for details on the subject.

In the sequel we will use the expression "equality theory" to denote any consistent, decidable theory T satisfying the congruence axioms. We will also use the expression "syntactic equality theory" to denote any equality theory T also satisfying the identity axioms.[2] When the equality theory T is clear from the context, it is convenient to adopt the notations $\sigma \implies \tau$ and $\sigma \iff \tau$, where σ, τ are sets of equations, to denote $T \vdash \forall(\sigma \to \tau)$ and $T \vdash \forall(\sigma \leftrightarrow \tau)$, respectively.

Given an equality theory T, and a set of equations in rational solved form σ, we say that σ is *satisfiable* in T if $T \vdash \forall\, \mathit{Vars} \setminus \mathrm{dom}(\sigma) : \exists\, \mathrm{dom}(\sigma) \mathrel{.} \sigma$. If T is a syntactic equality theory that also includes the occur-check axioms, and σ is satisfiable in T, then we say that σ is *Herbrand.*

Given a satisfiable set of equations $e \in \wp_{\mathrm{f}}(\mathrm{Eqs})$ in an equality theory T, then a substitution $\sigma \in \mathit{RSubst}$ is called a *solution for e in T* if σ is satisfiable in T and $T \vdash \forall(\sigma \to e)$. If $\mathit{vars}(\sigma) \subseteq \mathit{vars}(e)$, then σ is said to be a *relevant* solution for e. In addition, σ is a *most general solution for e in T* if $T \vdash \forall(\sigma \leftrightarrow e)$. In this paper, a most general solution is always a relevant solution of e.

Observe that, given an equality theory T, a set of equations in rational solved form may not be satisfiable in T. For example, $\exists x : \bigl\{x = f(x)\bigr\}$ is false in the Clark equality theory.

Lemma 1
Suppose T is an equality theory, $\sigma \in \mathit{RSubst}$ is satisfiable in T, $x \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$, and $a \in \mathcal{T}_{\varnothing}$. Then, if $\sigma' \stackrel{\mathrm{def}}{=} \sigma \cup \{x \mapsto a\} \in \mathit{RSubst}$, σ' is satisfiable in T.

Proof
As $x \notin \mathrm{dom}(\sigma)$ and $\sigma \in \mathit{RSubst}$, it follows that $\sigma' = \sigma \cup \{x \mapsto a\} \in \mathit{RSubst}$.

Since σ is satisfiable in T,

$$T \vdash \forall\, \mathit{Vars} \setminus \mathrm{dom}(\sigma) : \exists\, \mathrm{dom}(\sigma) \mathrel{.} \sigma.$$

Moreover, by the congruence axiom (2),

$$T \vdash \forall\, \mathit{Vars} \setminus \{x\} : \exists x \mathrel{.} \{x = a\}.$$

Hence,

$$T \vdash \forall\, \mathit{Vars} \setminus \bigl(\mathrm{dom}(\sigma) \cup \{x\}\bigr) : \exists \bigl(\mathrm{dom}(\sigma) \cup \{x\}\bigr) \mathrel{.} \sigma \cup \{x = a\}.$$

Thus $\sigma' = \sigma \cup \{x \mapsto a\}$ is satisfiable in T. $\square$

Syntactically we have shown that any substitution in *RSubst* may be regarded as a set of equations in rational solved form and vice versa. The next lemma shows the semantic relationship between them.

Lemma 2
If T is an equality theory and $\sigma \in \mathit{RSubst}$, then, for each $t \in \mathcal{T}_{\mathit{Vars}}$,

$$T \vdash \forall\bigl(\sigma \to (t = t\sigma)\bigr).$$

[2] Note that, as a consequence of axiom (7) and the assumption that there are at least two distinct function symbols in the language, one of which is a constant, there exist two terms $a_1, a_2 \in \mathcal{T}_{\varnothing}$ such that, for any syntactic equality theory T, we have $T \vdash a_1 \neq a_2$.

Proof
We assume the congruence axioms hold and prove that, for any $t \in \mathcal{T}_{Vars}$, we have $\sigma \implies \{t = t\sigma\}$. The proof is by induction on the depth of t.

Suppose, first that the depth of t is one. If t is a variable not in $\mathrm{dom}(\sigma)$ or a constant, then $t\sigma = t$ and the result follows from axiom (2). If $t \in \mathrm{dom}(\sigma)$, then, for some $r \in \mathcal{T}_{Vars}$, $(t \mapsto r) \in \sigma$. Thus $\sigma \implies \{t = t\sigma\}$.

If the depth of t is greater than one, then t has the form $f(s_1, \dots, s_n)$ where $s_1, \dots, s_n \in \mathcal{T}_{Vars}$ have depth less than the depth of t. By the inductive hypothesis, for each $i = 1, \dots, n$, we have $\sigma \implies \{s_i = s_i\sigma\}$. Therefore, applying axiom (5), we have $\sigma \implies \{t = t\sigma\}$. $\square$

As is common in papers involving equality, we overload the symbol '=' and use it to denote both equality and to represent syntactic identity. The context makes it clear what is intended.

3 The Set-Sharing Domain

In this section, we first recall the definition of the Sharing domain and present the (classical) abstraction function used for dealing with idempotent substitutions. We will then give evidence for the problems arising when applying this abstraction function to the more general case of substitutions in rational solved form.

3.1 The Sharing *Domain*

The Sharing domain is due to Jacobs and Langen (Jacobs & Langen, 1989). However, we use the definition as presented in (Bagnara *et al.*, 1997) where the set of variables of interest is given explicitly.

Definition 1
(The *set-sharing* lattice.) Let

$$SG \stackrel{\mathrm{def}}{=} \wp_{\mathrm{f}}(Vars) \setminus \{\varnothing\}$$

and let

$$SH \stackrel{\mathrm{def}}{=} \wp(SG).$$

The *set-sharing lattice* is given by the set

$$SS \stackrel{\mathrm{def}}{=} \bigl\{\, (sh, U) \bigm| sh \in SH, U \in \wp_{\mathrm{f}}(Vars), \forall S \in sh : S \subseteq U \,\bigr\} \cup \{\bot, \top\},$$

which is ordered by '$\preceq_{SS}$' defined as follows, for each $d, (sh_1, U_1), (sh_2, U_2) \in SS$:

$$\begin{aligned} \bot &\preceq_{SS} d, \\ d &\preceq_{SS} \top, \\ (sh_1, U_1) \preceq_{SS} (sh_2, U_2) \quad &\iff \quad (U_1 = U_2) \wedge (sh_1 \subseteq sh_2). \end{aligned}$$

It is straightforward to see that every subset of SS has a least upper bound with respect to $\preceq_{SS}$. Hence SS is a complete lattice.[3] The lub operator over SS will be denoted by '$\sqcup$'.

3.2 The Classical Abstraction Function for *ISubst*

An element sh of SH encodes the sharing information contained in an idempotent substitution σ. Namely, two variables x and y must be in the same set in sh if some variable occurs in both $x\sigma$ and $y\sigma$.

Definition 2
(Classical sg **and abstraction functions.)** $\mathrm{sg} \colon \mathit{ISubst} \times \mathit{Vars} \to \wp_{\mathrm{f}}(\mathit{Vars})$, called *sharing group function*, is defined, for each $\sigma \in \mathit{ISubst}$ and each $v \in \mathit{Vars}$, by

$$\mathrm{sg}(\sigma, v) \stackrel{\mathrm{def}}{=} \bigl\{\, y \in \mathit{Vars} \bigm| v \in \mathit{vars}(y\sigma) \,\bigr\}.$$

The concrete domain $\wp(\mathit{ISubst})$ is related to SS by means of the *abstraction function* $\alpha_I \colon \wp(\mathit{ISubst}) \times \wp_{\mathrm{f}}(\mathit{Vars}) \to SS$. For each $\Sigma \in \wp(\mathit{ISubst})$ and each $U \in \wp_{\mathrm{f}}(\mathit{Vars})$,

$$\alpha_I(\Sigma, U) \stackrel{\mathrm{def}}{=} \bigsqcup_{\sigma \in \Sigma} \alpha_I(\sigma, U),$$

where $\alpha_I \colon \mathit{ISubst} \times \wp_{\mathrm{f}}(\mathit{Vars}) \to SS$ is defined, for each substitution $\sigma \in \mathit{ISubst}$ and each $U \in \wp_{\mathrm{f}}(\mathit{Vars})$, by

$$\alpha_I(\sigma, U) \stackrel{\mathrm{def}}{=} \Bigl(\bigl\{\, \mathrm{sg}(\sigma, v) \cap U \bigm| v \in \mathit{Vars} \,\bigr\} \setminus \{\varnothing\}, U \Bigr).$$

The sharing group function sg was first defined by Jacobs and Langen (Jacobs & Langen, 1989) and used in their definition of a concretisation function for SH. The function α_I corresponds closely to the abstract counterpart of this concretisation function, but explicitly includes the set of variables of interest as a separate argument. It is identical to the abstraction function for Sharing defined by Cortesi and Filé (Cortesi & Filé, 1999).

In order to provide an intuitive reading of the sharing information encoded into an abstract element, we should stress that the analysis aims at capturing *possible* sharing. The corresponding *definite* information (e.g., definite groundness or independence) can be extracted by observing which sharing groups are *not* in the abstract element. As an example, if we observe that there is no sharing group containing a particular variable of U, then we can safely conclude that this variable is definitely ground (namely, it is bound to a term containing no variables). Similarly, if we observe that two variables never occur together in the same sharing group, then we can safely conclude that they are independent (namely, they are bound to terms that do not share a common variable). For a more detailed description of the information contained in an element of SS, we refer the interested reader to (Bagnara *et al.*, 1997; Bagnara *et al.*, 2000b).

[3] Notice that the only reason we have $\top \in SS$ is in order to turn SS into a lattice rather than a CPO.

Example 2
Assume $U = \{x_1, x_2, x_3, x_4\}$ and let

$$\sigma = \{x_1 \mapsto f(x_2, x_3), x_4 \mapsto a\},$$

so that its abstraction is given by

$$\alpha_I(\sigma, U) = \Bigl(\bigl\{\{x_1, x_2\}, \{x_1, x_3\}\bigr\}, U\Bigr).$$

From this abstraction we can safely conclude that variable x_4 is ground and variables x_2 and x_3 are independent.

3.3 *Towards an Abstraction Function for* RSubst

To help motivate the approach we have taken in adapting the classical abstraction function to non-idempotent substitutions, we now explain some of the problems that arise if we apply α_I, as it is defined on *ISubst*, to the non-idempotent substitutions in *RSubst*. Note that these problems are only partially due to allowing for non-Herbrand substitutions (that is substitutions that are not satisfiable in a syntactic equality theory containing the occur-check axioms). They are also due to the presence of non-idempotent but Herbrand substitutions that may arise because of the potential "laziness" of unification procedures based on the rational solved form.

We use the following substitutions to illustrate the problems, where it is assumed that the set of variables of interest is $U = \{x_1, x_2, x_3, x_4\}$. Let

$$\begin{aligned}
\sigma_1 &= \{x_1 \mapsto f(x_1)\},\\
\sigma_2 &= \{x_3 \mapsto x_4\},\\
\sigma_3 &= \{x_1 \mapsto x_2, x_2 \mapsto x_3, x_3 \mapsto x_4\},\\
\sigma_4 &= \{x_1 \mapsto x_4, x_2 \mapsto x_4, x_3 \mapsto x_4\}
\end{aligned}$$

so that we have

$$\begin{aligned}
\alpha_I(\varnothing, U) = \alpha_I(\sigma_1, U) &= \Bigl(\bigl\{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}\bigr\}, U\Bigr),\\
\alpha_I(\sigma_2, U) = \alpha_I(\sigma_3, U) &= \Bigl(\bigl\{\{x_1\}, \{x_2\}, \{x_3, x_4\}\bigr\}, U\Bigr),\\
\alpha_I(\sigma_4, U) &= \Bigl(\bigl\{\{x_1, x_2, x_3, x_4\}\bigr\}, U\Bigr).
\end{aligned}$$

The first problem is that the concrete equivalence classes induced by the classical abstraction function on *RSubst* are much coarser than one would expect and hence we have an unwanted loss of precision. For example, in all the sets of rational trees that are solutions for σ_1, the variable x_1 is ground. However, the computed abstract element fails to distinguish this situation from that resulting from the empty substitution, where all the variables are free and un-aliased. Similarly, we have the same abstract element for both σ_2 and σ_3 although, x_1, x_2 and x_3 are independent in σ_2 only.

The second problem is quite the opposite from the first in that the abstraction function distinguishes between substitutions that are equivalent (with respect to any equality theory). For example, σ_3 and σ_4 are equivalent although the abstract elements are distinct. Note that the two problems described here are completely orthogonal although they can interact and produce more complex situations.

4 Variable-Idempotence

In this section we define a new class of substitutions based on the concept of *variable-idempotence*. Variable-idempotent substitutions are then related to substitutions in rational solved form by means of an equivalence preserving rewriting relation.

4.1 Variable-Idempotent Substitutions

Recall that, for substitutions, the definition of idempotence requires that repeated applications of a substitution do not change the syntactic structure of a term. However, a sharing abstraction such as α_I is only interested in the variables and not in the structure that contains them. Thus, an obvious way to relax the definition of idempotence to allow for a non-Herbrand substitution is to ignore the structure and just require that its repeated application leaves the set of free variables in a term invariant.

Definition 3
(Variable-Idempotence.) A substitution σ is said to be *variable-idempotent* if $\sigma \in \mathit{RSubst}$ and, for each $t \in \mathcal{T}_{\mathit{Vars}}$,

$$\mathit{vars}(t\sigma\sigma) \setminus \mathrm{dom}(\sigma) = \mathit{vars}(t\sigma) \setminus \mathrm{dom}(\sigma).$$

The set of all variable-idempotent substitutions is denoted by VSubst.

Note that, as the condition $\mathit{vars}(t\sigma) \setminus \mathrm{dom}(\sigma) \subseteq \mathit{vars}(t\sigma\sigma)$ is trivial and holds for all substitutions, we have $\sigma \in \mathit{VSubst}$ if and only if $\sigma \in \mathit{RSubst}$ and

$$\mathit{vars}(t\sigma\sigma) \setminus \mathrm{dom}(\sigma) \subseteq \mathit{vars}(t\sigma). \tag{9}$$

Also note that any idempotent substitution is also variable-idempotent, so that $\mathit{ISubst} \subset \mathit{VSubst} \subset \mathit{RSubst}$.

Example 3
Consider the following substitutions which are all in RSubst.

$$\begin{aligned}
\sigma_1 &= \{x \mapsto f(x)\} && \in \mathit{VSubst} \setminus \mathit{ISubst},\\
\sigma_2 &= \{x \mapsto f(y), y \mapsto z\} && \notin \mathit{VSubst},\\
\sigma_3 &= \{x \mapsto f(z), y \mapsto z\} && \in \mathit{ISubst},\\
\sigma_4 &= \{x \mapsto z, y \mapsto f(x,y)\} && \notin \mathit{VSubst},\\
\sigma_5 &= \{x \mapsto z, y \mapsto f(z,y)\} && \in \mathit{VSubst} \setminus \mathit{ISubst}.
\end{aligned}$$

Note that σ_2 is equivalent (with respect to any equality theory) to the idempotent substitution σ_3; and σ_4 is equivalent (with respect to any equality theory) to the substitution σ_5 which is variable-idempotent but not idempotent.

The next result provides an alternative characterization of variable-idempotence.

Lemma 3
Suppose that $\sigma \in \mathit{RSubst}$. Then $\sigma \in \mathit{VSubst}$ if and only if, for all $(x \mapsto r) \in \sigma$,

$$\mathit{vars}(r\sigma) \setminus \mathrm{dom}(\sigma) = \mathit{vars}(r) \setminus \mathrm{dom}(\sigma).$$

Proof
Suppose first that $\sigma \in \mathit{VSubst}$ and that $(x \mapsto r) \in \sigma$. Then

$$\mathit{vars}(x\sigma\sigma) \setminus \mathrm{dom}(\sigma) = \mathit{vars}(x\sigma) \setminus \mathrm{dom}(\sigma)$$

and hence, $\mathit{vars}(r\sigma) \setminus \mathrm{dom}(\sigma) = \mathit{vars}(r) \setminus \mathrm{dom}(\sigma)$.

Next, suppose that for all $(x \mapsto r) \in \sigma$, $\mathit{vars}(r\sigma) \setminus \mathrm{dom}(\sigma) = \mathit{vars}(r) \setminus \mathrm{dom}(\sigma)$. Let $t \in \mathcal{T}_{\mathit{Vars}}$. We will show that $\mathit{vars}(t\sigma\sigma) \setminus \mathrm{dom}(\sigma) = \mathit{vars}(t\sigma) \setminus \mathrm{dom}(\sigma)$ by induction on the depth of t. If t is a constant or $t \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$, then the result follows from the fact that $t\sigma = t$. If $t \in \mathrm{dom}(\sigma)$, then the result follows from the hypothesis. Finally, if $t = f(t_1, \ldots, t_n)$, then, by the inductive hypothesis, $\mathit{vars}(t_i\sigma\sigma) \setminus \mathrm{dom}(\sigma) = \mathit{vars}(t_i\sigma) \setminus \mathrm{dom}(\sigma)$ for $i = 1, \ldots, n$. Therefore we have $\mathit{vars}(t\sigma\sigma) \setminus \mathrm{dom}(\sigma) = \mathit{vars}(t\sigma) \setminus \mathrm{dom}(\sigma)$. Thus, by Definition (3), as $\sigma \in \mathit{RSubst}$, $\sigma \in \mathit{VSubst}$. $\square$

Note that, as a consequence of Lemma 3, any substitution consisting of a single binding is variable-idempotent. Note though that we cannot assume that every subset of a variable-idempotent substitution is variable-idempotent.

Example 4
Let

$$\begin{aligned}
\sigma_1 &= \{x_1 \mapsto x_2, x_2 \mapsto g(x_3), x_3 \mapsto f(x_3)\},\\
\sigma_2 &= \{x_3 \mapsto f(x_3)\},\\
\sigma_3 &= \sigma_1 \setminus \sigma_2 = \{x_1 \mapsto x_2, x_2 \mapsto g(x_3)\}.
\end{aligned}$$

It can be observed that $\sigma_1, \sigma_2 \in \mathit{VSubst}$. Also note that $\sigma_3 \notin \mathit{VSubst}$, because we have $x_3 \in \mathit{vars}(x_1\sigma_3\sigma_3) \setminus \mathrm{dom}(\sigma_3)$ but $x_3 \notin \mathit{vars}(x_1\sigma_3) \setminus \mathrm{dom}(\sigma_3)$.

On the other hand, a variable-idempotent substitution does enjoy the following useful property with respect to its subsets.

Lemma 4
If $\sigma \in \mathit{VSubst}$ and $t \in \mathcal{T}_{\mathit{Vars}}$, then, for all $\sigma' \subseteq \sigma$,

$$\mathit{vars}(t\sigma\sigma') \setminus \mathrm{dom}(\sigma) = \mathit{vars}(t\sigma) \setminus \mathrm{dom}(\sigma).$$

Proof
Observe that, since $\sigma' \subseteq \sigma$, the relation $\mathit{vars}(t\sigma) \setminus \mathrm{dom}(\sigma) \subseteq \mathit{vars}(t\sigma\sigma')$ is trivial.

To prove the opposite relation, suppose that $y \in \mathit{vars}(t\sigma\sigma') \setminus \mathrm{dom}(\sigma)$. Then there exists $x \in \mathit{vars}(t\sigma)$ such that $y \in \mathit{vars}(x\sigma')$. Now, if $x \notin \mathrm{dom}(\sigma')$, then $x = y$ and $y \in \mathit{vars}(t\sigma)$. On the other hand, if $x \in \mathrm{dom}(\sigma')$, then $x\sigma' = x\sigma$ so that $y \in \mathit{vars}(t\sigma\sigma) \setminus \mathrm{dom}(\sigma)$ and hence, as $\sigma \in \mathit{VSubst}$, $y \in \mathit{vars}(t\sigma)$. $\square$

We note that this result depends on the definition of variable-idempotence ignoring the domain elements of the substitution.

Example 5
Let

$$\sigma = \bigl\{x \mapsto f(x,y), y \mapsto a\bigr\}.$$

Then $\sigma \in \mathit{VSubst}$ but

$$\begin{aligned} \mathit{vars}(x\sigma) &= \{x, y\}, \\ \mathit{vars}(x\sigma\sigma) &= \{x, y\}, \\ \mathit{vars}\bigl(x\sigma\{y \mapsto a\}\bigr) &= \{x\}. \end{aligned}$$

We now state three technical results that will be needed later in the paper. Note that, when proving these results at the end of this section, we require that the equality theory also satisfies the identity axioms. The first two results show that equivalent, ordered, variable-idempotent substitutions have the same domain and bind the domain variables to terms with the same set of parameter variables.

Lemma 5
Suppose that T is a syntactic equality theory, $\tau, \sigma \in \mathit{VSubst}$ are ordered and satisfiable in T and $T \vdash \forall(\tau \rightarrow \sigma)$. Then $\mathrm{dom}(\sigma) \subseteq \mathrm{dom}(\tau)$.

Lemma 6
Suppose that T is a syntactic equality theory, $\tau, \sigma \in \mathit{VSubst}$ are satisfiable in T and $T \vdash \forall(\tau \rightarrow \sigma)$. In addition, suppose $s, t \in \mathcal{T}_{\mathit{Vars}}$ are such that $T \vdash \forall\bigl(\tau \rightarrow (s = t)\bigr)$. Then, if $v \in \mathit{vars}(s) \setminus \mathrm{dom}(\tau)$, there exists a variable $z \in \mathit{vars}(t\sigma) \setminus \mathrm{dom}(\sigma)$ such that $v \in \mathit{vars}(z\tau)$.

Lemma 7
Let $\tau, \sigma \in \mathit{VSubst}$, where $\mathrm{dom}(\sigma) \cap \mathit{vars}(\tau) = \varnothing$. Then $\tau \circ \sigma$ has the following properties.

1. $T \vdash \forall\bigl((\tau \circ \sigma) \leftrightarrow (\tau \cup \sigma)\bigr)$, for some syntactic equality theory T;
2. $\mathrm{dom}(\tau \circ \sigma) = \mathrm{dom}(\tau \cup \sigma)$;
3. $\tau \circ \sigma \in \mathit{VSubst}$.

4.2 $\mathcal{S}$-transformations

A useful property of variable-idempotent substitutions is that any substitution can be transformed to an equivalent (with respect to any equality theory) variable-idempotent one.

Definition 4
($\mathcal{S}$-transformation.) The relation $\overset{\mathcal{S}}{\longmapsto} \subseteq \mathit{RSubst} \times \mathit{RSubst}$, called *$\mathcal{S}$-step*, is defined by

$$\frac{(x \mapsto t) \in \sigma \qquad (y \mapsto s) \in \sigma \qquad x \neq y}{\sigma \overset{\mathcal{S}}{\longmapsto} \bigl(\sigma \setminus \{y \mapsto s\}\bigr) \cup \{y \mapsto s[x/t]\}}.$$

If we have a finite sequence of $\mathcal{S}$-steps $\sigma_1 \overset{\mathcal{S}}{\longmapsto} \cdots \overset{\mathcal{S}}{\longmapsto} \sigma_n$ mapping σ_1 to σ_n, then we write $\sigma_1 \overset{\mathcal{S}}{\longmapsto}{}^{*} \sigma_n$ and say that σ_1 can be rewritten, by $\mathcal{S}$-transformation, to σ_n.

Example 6
Let

$$\begin{aligned}
\sigma_0 &= \bigl\{\underline{x_1 \mapsto f(x_2)}, x_2 \mapsto g(x_3, x_4), x_3 \mapsto x_1\bigr\}\\
\sigma_1 &= \bigl\{x_1 \mapsto f(x_2), \underline{x_2 \mapsto g(x_3, x_4)}, x_3 \mapsto f(x_2)\bigr\},\\
\sigma_2 &= \bigl\{x_1 \mapsto f(g(x_3, x_4)), x_2 \mapsto g(x_3, x_4), \underline{x_3 \mapsto f(g(x_3, x_4))}\bigr\},\\
\sigma_3 &= \bigl\{x_1 \mapsto f(g(f(g(x_3, x_4)), x_4)),\\
&\qquad x_2 \mapsto g(f(g(x_3, x_4)), x_4), x_3 \mapsto f(g(x_3, x_4))\bigr\}.
\end{aligned}$$

Then

$$\sigma_0 \overset{\mathcal{S}}{\longmapsto} \sigma_1 \overset{\mathcal{S}}{\longmapsto} \sigma_2 \overset{\mathcal{S}}{\longmapsto} \sigma_3.$$

Observe that σ_0 is not variable-idempotent since $\mathit{vars}(x_1\sigma_0) \setminus \{x_1, x_2, x_3\} = \varnothing$ but $\mathit{vars}(x_1\sigma_0\sigma_0) \setminus \{x_1, x_2, x_3\} = \{x_4\}$. Notice, however, that σ_3 is variable-idempotent and that $T \vdash \forall(\sigma_0 \leftrightarrow \sigma_3)$.

The next two theorems, which are proved at the end of this section, show that we need only consider variable-idempotent substitutions.

Theorem 1
Suppose $\sigma \in \mathit{RSubst}$ and $\sigma \overset{\mathcal{S}}{\longmapsto}{}^{*} \sigma'$. Then $\sigma' \in \mathit{RSubst}$, $\mathrm{dom}(\sigma) = \mathrm{dom}(\sigma')$, $\mathit{vars}(\sigma) = \mathit{vars}(\sigma')$ and, if T is any equality theory, then $T \vdash \forall(\sigma \leftrightarrow \sigma')$.

Theorem 2
Suppose $\sigma \in \mathit{RSubst}$. Then there exists $\sigma' \in \mathit{VSubst}$ such that $\sigma \overset{\mathcal{S}}{\longmapsto}{}^{*} \sigma'$ and, for all $\tau \subseteq \sigma'$, $\tau \in \mathit{VSubst}$.

Note that as a consequence of this theorem, we can transform any substitution in rational solved form to a substitution for which it and all its subsets are variable-idempotent. In particular, by this transformation we can avoid having to consider a substitution such as that illustrated in Example 4.

4.3 The Abstraction Function for VSubst

With these results, it can be seen that we need to consider variable-idempotent substitutions only. Moreover, in this case, one of the causes of the problems outlined in Section 3.3, due to the possible "laziness" of the unification algorithm, is no longer present. As a consequence, it is now sufficient to address the potential loss in precision due to the non-Herbrand substitutions. The simple solution is to define a new abstraction function for *VSubst* which is the same as that in Definition 2 but where any sharing group generated by a variable in the domain of the substitution is disregarded. This new abstraction function works for variable-idempotent substitutions and no longer suffers the drawbacks outlined in Section 3.3.

Therefore, at least from a theoretical point of view, the problem of defining a sound and precise abstraction function for arbitrary substitutions in rational solved form would have been solved. Given a substitution in *RSubst*, we would proceed in two steps: we first transform it to an equivalent substitution in *VSubst* and then compute the corresponding description by using the modified abstraction function. However, from a practical point of view, we need to define an abstraction function that directly computes the description of a substitution in *RSubst* in a single step, thus avoiding the expensive computation of the intermediate variable-idempotent substitution. We present such an abstraction function in Section 5.

4.4 Proofs of Lemmas 5, 6 and 7 and Theorems 1 and 2

To prove Lemmas 5 and 6, it is useful to first establish the following two properties of variable-idempotent substitutions.

Lemma 8
Suppose that $\sigma \in \mathit{VSubst}$, $r \in \mathcal{T}_{\mathit{Vars}}$ and, for all $i \geq 0$, $r\sigma^i \in \mathit{Vars}$. Then we have $r\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$.

Proof
As σ has no circular subset and $\mathrm{dom}(\sigma)$ is finite, there exists a $j \geq 1$ such that $r\sigma^j = r\sigma^{j+1}$ and hence, $r\sigma^j \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. As σ is variable-idempotent, we have

$$\begin{aligned} \{r\sigma^j\} &= \mathit{vars}(r\sigma^j) \setminus \mathrm{dom}(\sigma) \\ &= \mathit{vars}(r\sigma) \setminus \mathrm{dom}(\sigma) \\ &= \{r\sigma\} \setminus \mathrm{dom}(\sigma). \end{aligned}$$

Hence $r\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. □

Lemma 9
Suppose that $\sigma \in \mathit{VSubst}$ and $v, r \in \mathcal{T}_{\mathit{Vars}}$ so that $v \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$ and, for any syntactic equality theory T, $T \vdash \forall\bigl(\sigma \rightarrow \{v = r\}\bigr)$. Then $v = r\sigma$.

Proof
We assume that the congruence and identity axioms hold. Let $a_1, a_2 \in \mathcal{T}_{\varnothing}$ have distinct outer-most symbols so that, by the identity axioms, $T \vdash a_1 \neq a_2$. By Lemma 8, either $r\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$ or, for some $j \geq 0$, $r\sigma^j \notin \mathit{Vars}$. We consider each case separately.

If, for some $j \geq 0$, $r\sigma^j \notin \mathit{Vars}$, then, as a_1 and a_2 have distinct outer-most symbols, there exists an $i \in \{1, 2\}$ such that a_i and $r\sigma^j$ have distinct outer-most symbols. Thus, by the identity axioms, $a_i \neq r\sigma^j$. Let $\sigma' = \sigma \cup \{v = a_i\}$. It follows from Lemma 1 that, as $v \notin \mathrm{dom}(\sigma)$ and σ is satisfiable, $\sigma' \in \mathit{RSubst}$ and is satisfiable. By Lemma 2 and the congruence axioms, $\sigma \implies \{v = r\sigma^j\}$. However, $\sigma' \implies \sigma$, so that $\sigma' \implies \{v = r\sigma^j, v = a_i\}$. Thus, by the congruence axioms, we have $\sigma' \implies \{a_i = r\sigma^j\}$, which is a contradiction.

Suppose then that $r\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. If $v \neq r\sigma$, then it follows from Lemma 1 that $\sigma' = \sigma \cup \{v = a_1, r\sigma = a_2\} \in \mathit{RSubst}$ and, as σ is satisfiable, σ' is satisfiable. By Lemma 2 and the congruence axioms, $\sigma \implies \{v = r\sigma\}$. However, $\sigma' \implies \sigma$, so that $\sigma' \implies \{v = r\sigma, v = a_1, r\sigma = a_2\}$. Thus, by the congruence axioms, we have $\sigma' \implies \{a_1 = a_2\}$, which is a contradiction. Hence $v = r\sigma$ as required. $\square$

Proof of Lemma 5.
We assume that the congruence and identity axioms hold. To prove the result, we suppose that there exists $v \in \mathrm{dom}(\sigma) \setminus \mathrm{dom}(\tau)$ and derive a contradiction.

By hypothesis, $\tau \implies \sigma$. Thus, using Lemma 2 and the congruence axioms, we have, for any $i \geq 0$, $\tau \implies \{v = v\sigma^i\}$. By Lemma 9, for all $i \geq 0$, $v = v\sigma^i\tau$ so that $v\sigma^i \in \mathit{Vars}$. By Lemma 8, $v\sigma \notin \mathrm{dom}(\sigma)$, so that, as σ is ordered and $v \in \mathrm{dom}(\sigma)$, $v\sigma < v$. In particular, $v\sigma \neq v$, so that as $v\sigma\tau = v$ and τ is ordered, we would have $v < v\sigma$, which is a contradiction. $\square$

Proof of Lemma 6.
We assume that the congruence and identity axioms hold. Note that, by the hypothesis, $\tau \implies \sigma$ and $\tau \implies \{s = t\}$ so that, using Lemma 2 and the congruence axioms, we have $\tau \implies \{s = t\sigma^j\}$ and $\tau \implies \{t\sigma\tau^k = s\}$, for all $j, k \geq 0$.

Let $v \in \mathit{vars}(s) \setminus \mathrm{dom}(\tau)$. We prove, by induction on the depth d of s, that there exists $z \in \mathit{vars}(t\sigma) \setminus \mathrm{dom}(\sigma)$ such that $v \in \mathit{vars}(z\tau)$. The base case is when $d = 1$ so that $s = v$. Now, for each $j \geq 0$, $\tau \implies \{v = t\sigma^j\}$ and hence, by Lemma 9 (as $v \notin \mathrm{dom}(\tau)$), $v = t\sigma^j\tau$. As a consequence, $t\sigma^j \in \mathit{Vars}$ for all $j \geq 0$ and $v = t\sigma\tau$. By Lemma 8, $t\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. Thus, we define $z = t\sigma$.

For the inductive step, we assume that $d > 1$ so that, for some $n \geq 1$, we have $s = f(s_1, \ldots, s_n)$ and, for some $i \in \{1, \ldots, n\}$, $v \in \mathit{vars}(s_i)$ and s_i has depth $d - 1$. By Lemma 8, either $t\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$ or there exists a $j \geq 0$ such that $t\sigma^j \notin \mathit{Vars}$.

First, suppose that $t\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. Now, $\tau \implies \{t\sigma\tau = s\}$ so that, as $s\tau \notin \mathit{Vars}$, by Lemma 9, we have $t\sigma\tau \notin \mathit{Vars} \setminus \mathrm{dom}(\tau)$. Thus, by Lemma 8, there exists $k > 1$ such that $t\sigma\tau^k \notin \mathit{Vars}$. Then, using the identity axioms, we have $t\sigma\tau^k = f(r_1, \ldots, r_n)$ and $\tau \implies \{s_i = r_i\}$. By the inductive hypothesis (letting σ be the empty substitution), we have $v \in \mathit{vars}(r_i\tau)$. However, $\mathit{vars}(r_i) \subseteq \mathit{vars}(t\sigma\tau^k)$ so that $v \in \mathit{vars}(t\sigma\tau^{k+1})$. As $\tau \in \mathit{VSubst}$ and $v \notin \mathrm{dom}(\tau)$, $v \in \mathit{vars}(t\sigma\tau)$. Thus, in this case, let $z = t\sigma$.

Secondly, suppose that there exists a $j \geq 0$ such that $t\sigma^j \notin \mathit{Vars}$. Then, as $\tau \implies \{s = t\sigma^j\}$, it follows from the identity axioms that $t\sigma^j = f(t_1, \ldots, t_n)$ and $\tau \implies \{s_i = t_i\}$. By the inductive hypothesis, there exists $z \in \mathit{vars}(t_i\sigma) \setminus \mathrm{dom}(\sigma)$ such that $v \in \mathit{vars}(z\tau)$. However, $\mathit{vars}(t_i\sigma) \subseteq \mathit{vars}(t\sigma^{j+1})$ so that we must have $z \in \mathit{vars}(t\sigma^{j+1}) \setminus \mathrm{dom}(\sigma)$. As $\sigma \in \mathit{VSubst}$, $z \in \mathit{vars}(t\sigma) \setminus \mathrm{dom}(\sigma)$ as required. $\square$

Proof of Lemma 7.
As consequences of the hypothesis, we have that $\mathrm{dom}(\sigma) \cap \mathrm{dom}(\tau) = \varnothing$ and that $(\tau \cup \sigma) \in \mathit{RSubst}$. It follows from Eq. (1) that $\tau \circ \sigma$ can be obtained from $(\tau \cup \sigma)$ by a sequence of $\mathcal{S}$-steps so that, by Theorem 1, we have Properties 1 and 2.

To prove Property 3, we suppose that, for some $v \in \mathrm{dom}(\tau \circ \sigma)$, there exist $w \in \mathit{vars}(v\sigma)$, $x \in \mathit{vars}(w\tau)$ and $y \in \mathit{vars}(x\sigma)$ such that $z \in \mathit{vars}(y\tau) \setminus \mathrm{dom}(\tau \circ \sigma)$. We need to prove that $z \in \mathit{vars}(v\sigma\tau)$.

It follows from Property 2, that $z \notin \mathrm{dom}(\sigma)$ and $z \notin \mathrm{dom}(\tau)$. Suppose first that

$x \notin \mathrm{dom}(\sigma)$. Then $y = x$ and hence $z \in \mathit{vars}(v\sigma\tau\tau)$. Therefore, as $\tau \in \mathit{VSubst}$ and $z \notin \mathrm{dom}(\tau)$, we can conclude $z \in \mathit{vars}(v\sigma\tau)$. Thus, we now assume that $x \in \mathrm{dom}(\sigma)$. As $\mathrm{dom}(\sigma) \cap \mathit{vars}(\tau) = \varnothing$, we have $x \notin \mathit{vars}(\tau)$, so that $x = w$ and hence, $y \in \mathit{vars}(v\sigma\sigma)$. If $y \notin \mathrm{dom}(\tau)$ we have $y = z$, so that $y \notin \mathrm{dom}(\sigma)$. On the other hand, if $y \in \mathrm{dom}(\tau)$ then, by the hypothesis, $y \notin \mathrm{dom}(\sigma)$. Thus, in both cases, as $\sigma \in \mathit{VSubst}$, we obtain $y \in \mathit{vars}(v\sigma)$ and hence $z \in \mathit{vars}(v\sigma\tau)$. It follows, using Eq. (9), that Property 3 holds. $\square$

To prove Theorem 1, we need to show that the result holds for a single $\mathcal{S}$-step.

Lemma 10
Let T be an equality theory and suppose that $\sigma \in \mathit{RSubst}$ and $\sigma \overset{\mathcal{S}}{\longmapsto} \sigma'$. Then $\sigma' \in \mathit{RSubst}$, $\mathrm{dom}(\sigma) = \mathrm{dom}(\sigma')$, $\mathit{vars}(\sigma) = \mathit{vars}(\sigma')$, and $T \vdash \forall(\sigma \leftrightarrow \sigma')$.

Proof
Since $\sigma \overset{\mathcal{S}}{\longmapsto} \sigma'$, there exists $x, y \in \mathrm{dom}(\sigma)$ with $x \neq y$ such that $(x \mapsto t), (y \mapsto s) \in \sigma$ and $\sigma' = \bigl(\sigma \setminus \{y \mapsto s\}\bigr) \cup \bigl\{y \mapsto s[x/t]\bigr\}$. If $x \notin \mathit{vars}(s)$, $\sigma = \sigma'$ and the result is trivial. Suppose now that $x \in \mathit{vars}(s)$. We define

$$\sigma_0 \stackrel{\mathrm{def}}{=} \sigma \setminus \{x = t, y = s\}.$$

Hence, as it is assumed that $x \neq y$,

$$\sigma = \sigma_0 \cup \{x \mapsto t, y \mapsto s\}, \tag{10}$$
$$\sigma' = \sigma_0 \cup \{x \mapsto t, y \mapsto s[x/t]\}. \tag{11}$$

We first show that $\sigma' \in \mathit{RSubst}$ and $\mathrm{dom}(\sigma) = \mathrm{dom}(\sigma')$. If $s \notin \mathit{Vars}$, then $s[x/t] \notin \mathit{Vars}$ so that $\mathrm{dom}(\sigma) = \mathrm{dom}(\sigma')$. Also, as σ has no circular subset, σ' has no circular subset and $\sigma' \in \mathit{RSubst}$. If $s \in \mathit{Vars}$, then $s = x$ and $s[x/t] = t$. Thus, as $\sigma = \sigma_0 \cup \{x \mapsto t, y \mapsto x\}$ has no circular subset, $t \neq y$ so that $\mathrm{dom}(\sigma) = \mathrm{dom}(\sigma')$. Moreover, neither $\sigma_0 \cup \{x \mapsto t\}$ nor $\sigma_0 \cup \{y \mapsto t\}$ have circular subsets. Hence σ' has no circular subset. Thus $\sigma' \in \mathit{RSubst}$.

Now, since

$$\bigl(\mathit{vars}(s) \cup \mathit{vars}(t)\bigr) \setminus \mathrm{dom}(\sigma) = \mathit{vars}\bigl(s[x/t] \cup \mathit{vars}(t)\bigr) \setminus \mathrm{dom}(\sigma),$$

it follows that $\mathit{vars}(\sigma) = \mathit{vars}(\sigma')$.

Therefore, it remains to show that, for any equality theory T, $T \vdash \forall(\sigma \leftrightarrow \sigma')$. To do this, we assume that the congruence axioms hold, and show that $\sigma \iff \sigma'$. By Lemma 2, we have

$$\{x = t\} \implies \{s = s[x/t]\}.$$

Thus, using the congruence axiom (4), we have

$$\begin{aligned} \{x = t, y = s\} &\implies \bigl\{x = t, y = s, s = s[x/t]\bigr\} \\ &\implies \bigl\{x = t, y = s[x/t]\bigr\}. \end{aligned}$$

Similarly, using congruence axioms (3) and (4), we have

$$\begin{aligned}\{x = t, y = s[x/t]\} &\implies \{x = t, y = s[x/t], s = s[x/t]\}\\ &\implies \{x = t, y = s\}.\end{aligned}$$

Thus

$$\{x = t, y = s\} \iff \{x = t, y = s[x/t]\}.$$

It therefore follows from (10) and (11) that $\sigma \iff \sigma'$. $\square$

The condition $x \neq y$ in the proof of Lemma 10 is necessary. For example, suppose $\sigma = \{x \mapsto f(x)\}$ and $\sigma' = \{x \mapsto f(f(x))\}$. Then we do not have $\sigma' \implies \sigma$. Note however that this implication will hold as soon as we enrich the equality theory T with either the occur-check axioms or the uniqueness axioms of the rational trees' theory.

Proof of Theorem 1.
The proof is by induction on the length of the sequence of $\mathcal{S}$-steps transforming σ to σ'. The base case is the empty sequence. For the inductive step, the sequence has length $n > 0$ and there exists σ_1 such that $\sigma \xmapsto{\mathcal{S}} \sigma_1 \xmapsto{\mathcal{S}}{}^* \sigma'$ and $\sigma_1 \xmapsto{\mathcal{S}}{}^* \sigma'$ has length $n-1$. By Lemma 10, $\sigma_1 \in \mathit{RSubst}$, $\operatorname{dom}(\sigma) = \operatorname{dom}(\sigma_1)$, $\mathit{vars}(\sigma) = \mathit{vars}(\sigma_1)$ and $T \vdash \forall(\sigma \leftrightarrow \sigma_1)$. By the inductive hypothesis, $\sigma' \in \mathit{RSubst}$, $\operatorname{dom}(\sigma_1) = \operatorname{dom}(\sigma')$, $\mathit{vars}(\sigma_1) = \mathit{vars}(\sigma')$ and $T \vdash \forall(\sigma_1 \leftrightarrow \sigma')$. Hence we have $\operatorname{dom}(\sigma) = \operatorname{dom}(\sigma')$, $\mathit{vars}(\sigma) = \mathit{vars}(\sigma')$, and $T \vdash \forall(\sigma \leftrightarrow \sigma')$. $\square$

Proof of Theorem 2.
To prove the theorem, we construct an $\mathcal{S}$-transformation and show that the resulting substitution has the required properties.

Suppose that $\{x_1, \dots, x_n\} = \operatorname{dom}(\sigma)$, $\sigma_0 = \sigma$ and, for each $j = 0, \dots, n$,

$$\sigma_j = \{x_1 \mapsto t_{1,j}, \dots, x_n \mapsto t_{n,j}\},$$

where, if $j > 0$, $t_{j,j} = t_{j,j-1}$ and, for each $i = 1, \dots, n$ with $i \neq j$, we have $t_{i,j} = t_{i,j-1}[x_j/t_{j,j}]$.

It follows from the definition of σ_j that, for $j = 1, \dots, n$, σ_j can be obtained from σ_{j-1} by two sequences of $\mathcal{S}$-steps of lengths $j-1$ and $n-j+1$:

$$\sigma_{j-1} = \sigma_{j-1}^0 \xmapsto{\mathcal{S}} \cdots \xmapsto{\mathcal{S}} \sigma_{j-1}^{j-1} = \sigma_{j-1}^j \xmapsto{\mathcal{S}} \cdots \xmapsto{\mathcal{S}} \sigma_{j-1}^n = \sigma_j,$$

where, for $i = 1, \dots, n$ with $i \neq j$,

$$\begin{aligned}\sigma_{j-1}^i &= \bigl(\sigma_{j-1}^{i-1} \setminus \{x_i \mapsto t_{i,j-1}\}\bigr) \cup \{x_i \mapsto t_{i,j-1}[x_j/t_{j,j}]\}\\ &= \{x_1 \mapsto t_{1,j}, \dots, x_i \mapsto t_{i,j}, x_{i+1} \mapsto t_{i+1,j-1}, \dots, x_n \mapsto t_{n,j-1}\}.\end{aligned}$$

Hence, by Theorem 1, $\sigma_1, \dots, \sigma_n \in \mathit{RSubst}$.

We next show, by induction on j, with $0 \leq j \leq n$, that, for each $i = 1, \dots, n$ and each $h = 1, \dots, j$, we have $\mathit{vars}(t_{i,j}) = \mathit{vars}\bigl(t_{i,j}[x_h/t_{h,j}]\bigr)$.

For the base case when $j = 0$ there is nothing to prove. Suppose, therefore, that

$1 \leq j \leq n$ and that, for each $i = 1, \ldots, n$ and $h = 1, \ldots, j-1$,

$$\mathit{vars}(t_{i,j-1}) = \mathit{vars}\bigl(t_{i,j-1}[x_h/t_{h,j-1}]\bigr).$$

Now by the definition of $t_{k,j}$ where $1 \leq k \leq n$, $k \neq j$, we have

$$\mathit{vars}(t_{k,j}) = \mathit{vars}\bigl(t_{k,j-1}[x_j/t_{j,j}]\bigr). \tag{12}$$

Also, since a substitution consisting of a single binding is variable-idempotent,

$$\mathit{vars}(t_{j,j}) = \mathit{vars}\bigl(t_{j,j}[x_j/t_{j,j}]\bigr)$$

so that, as $t_{j,j} = t_{j,j-1}$,

$$\mathit{vars}(t_{j,j}) = \mathit{vars}\bigl(t_{j,j-1}[x_j/t_{j,j}]\bigr). \tag{13}$$

Thus, by (12) and (13), for all k such that $1 \leq k \leq n$, we have

$$\mathit{vars}(t_{k,j}) = \mathit{vars}\bigl(t_{k,j-1}[x_j/t_{j,j}]\bigr). \tag{14}$$

Therefore, for each $i = 1, \ldots, n$ and $h = 1, \ldots, j$, using (14) and the inductive hypothesis, we have

$$\begin{aligned}
\mathit{vars}\bigl(t_{i,j}[x_h/t_{h,j}]\bigr) &= \mathit{vars}\Bigl(t_{i,j-1}[x_j/t_{j,j}]\bigl[x_h/t_{h,j-1}[x_j/t_{j,j}]\bigr]\Bigr) \\
&= \mathit{vars}\bigl(t_{i,j-1}[x_h/t_{h,j-1}][x_j/t_{j,j}]\bigr) \\
&= \mathit{vars}\bigl(t_{i,j-1}[x_j/t_{j,j}]\bigr) \\
&= \mathit{vars}(t_{i,j}).
\end{aligned}$$

Letting $j = n$ we obtain, for each $i, h = 1, \ldots, n$,

$$\mathit{vars}\bigl(t_{i,n}[x_h/t_{h,n}]\bigr) = \mathit{vars}(t_{i,n}).$$

Therefore, for all $\tau \subseteq \sigma_n$ and each $i = 1, \ldots, n$,

$$\mathit{vars}(t_{i,n}\tau) = \mathit{vars}(t_{i,n}).$$

Thus, by Lemma 3, for all $\tau \subseteq \sigma_n$, $\tau \in \mathit{VSubst}$. The result follows by taking $\sigma' = \sigma_n$. $\square$

5 The Abstraction Function for *RSubst*

In this section we define a new abstraction function mapping arbitrary substitutions in rational solved form into their abstract descriptions. This abstraction function is based on a new definition for the notion of *occurrence*. The new occurrence operator occ is defined on *RSubst* so that it does not require the explicit computation of intermediate variable-idempotent substitutions. To this end, it is given as the fixed point of a sequence of *occurrence functions*. The occ operator generalises the sg operator, defined for *ISubst*, coinciding with it when applied to idempotent substitutions.

Definition 5
(Occurrence functions.) For each $n \in \mathbb{N}$, $\mathrm{occ}_n \colon \mathit{RSubst} \times \mathit{Vars} \to \wp_{\mathrm{f}}(\mathit{Vars})$, called *occurrence function*, is defined, for each $\sigma \in \mathit{RSubst}$ and each $v \in \mathit{Vars}$, by

$$\mathrm{occ}_0(\sigma, v) \stackrel{\mathrm{def}}{=} \{v\} \setminus \mathrm{dom}(\sigma)$$

and, for $n > 0$, by

$$\mathrm{occ}_n(\sigma, v) \stackrel{\mathrm{def}}{=} \bigl\{\, y \in \mathit{Vars} \bigm| \mathit{vars}(y\sigma) \cap \mathrm{occ}_{n-1}(\sigma, v) \neq \varnothing \,\bigr\}.$$

The following monotonicity property for occ_n is proved at the end of this section.

Lemma 11
If $n > 0$, then, for each $\sigma \in \mathit{RSubst}$ and each $v \in \mathit{Vars}$,

$$\mathrm{occ}_{n-1}(\sigma, v) \subseteq \mathrm{occ}_n(\sigma, v).$$

Note that, by considering the substitution $\{u \mapsto v, v \mapsto w\}$, it can be seen that, if we had not excluded the domain variables in the definition of occ_0, then this monotonicity property would not have held.

For any n, the set $\mathrm{occ}_n(\sigma, v)$ is restricted to the set $\{v\} \cup \mathit{vars}(\sigma)$. Thus, it follows from Lemma 11, that there is an $\ell = \ell(\sigma, v) \in \mathbb{N}$ such that $\mathrm{occ}_\ell(\sigma, v) = \mathrm{occ}_n(\sigma, v)$ for all $n \geq \ell$.

Definition 6
(Occurrence operator.) For each $\sigma \in \mathit{RSubst}$ and $v \in \mathit{Vars}$, the *occurrence operator* $\mathrm{occ} \colon \mathit{RSubst} \times \mathit{Vars} \to \wp_{\mathrm{f}}(\mathit{Vars})$ is given by

$$\mathrm{occ}(\sigma, v) \stackrel{\mathrm{def}}{=} \mathrm{occ}_\ell(\sigma, v)$$

where $\ell \in \mathbb{N}$ is such that $\mathrm{occ}_\ell(\sigma, v) = \mathrm{occ}_n(\sigma, v)$ for all $n \geq \ell$.

Note that, by combining Definitions 5 and 6, we obtain

$$\mathrm{occ}(\sigma, v) = \bigl\{\, y \in \mathit{Vars} \bigm| \mathit{vars}(y\sigma) \cap \mathrm{occ}(\sigma, v) \neq \varnothing \,\bigr\}. \tag{15}$$

The following simpler characterisations for occ can be used when the variable is in the domain of the substitution, the substitution is variable-idempotent or the substitution is idempotent.

Lemma 12
If $\sigma \in \mathit{RSubst}$ and $v \in \mathrm{dom}(\sigma)$, then $\mathrm{occ}(\sigma, v) = \varnothing$.

Lemma 13
If $\sigma \in \mathit{VSubst}$ then, for each $v \in \mathit{Vars}$,

$$\begin{aligned} \mathrm{occ}(\sigma, v) &= \mathrm{occ}_1(\sigma, v) \\ &= \bigl\{\, y \in \mathit{Vars} \bigm| v \in \mathit{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) \,\bigr\}. \end{aligned}$$

Lemma 14
If $\sigma \in \mathit{ISubst}$ and $v \in \mathit{Vars}$ then $\mathrm{occ}(\sigma, v) = \mathrm{sg}(\sigma, v)$.

These results are proved at the end of this section.

Example 7
Consider again Example 6. Then, for all $i \geq 0$, $\operatorname{dom}(\sigma_i) = \{x_1, x_2, x_3\}$ so that

$$\operatorname{occ}(\sigma_i, x_1) = \operatorname{occ}(\sigma_i, x_2) = \operatorname{occ}(\sigma_i, x_3) = \varnothing.$$

However,

$$\begin{aligned}
&\operatorname{occ}_0(\sigma_0, x_4) = \{x_4\},\\
&\operatorname{occ}_1(\sigma_0, x_4) = \{x_2, x_4\},\\
&\operatorname{occ}_2(\sigma_0, x_4) = \{x_1, x_2, x_4\},\\
&\operatorname{occ}_3(\sigma_0, x_4) = \{x_1, x_2, x_3, x_4\} = \operatorname{occ}(\sigma_0, x_4).
\end{aligned}$$

Also, note that

$$\operatorname{occ}_1(\sigma_3, x_4) = \{x_1, x_2, x_3, x_4\} = \operatorname{occ}(\sigma_3, x_4).$$

The definition of abstraction is based on the occurrence operator, occ.

Definition 7
(Abstraction.) The concrete domain $\wp(\mathit{RSubst})$ is related to SS by means of the *abstraction function* $\alpha\colon \wp(\mathit{RSubst}) \times \wp_{\mathrm{f}}(\mathit{Vars}) \to \mathit{SS}$. For each $\Sigma \in \wp(\mathit{RSubst})$ and each $U \in \wp_{\mathrm{f}}(\mathit{Vars})$,

$$\alpha(\Sigma, U) \stackrel{\mathrm{def}}{=} \bigsqcup_{\sigma \in \Sigma} \alpha(\sigma, U)$$

where $\alpha\colon \mathit{RSubst} \times \wp_{\mathrm{f}}(\mathit{Vars}) \to \mathit{SS}$ is defined, for each substitution $\sigma \in \mathit{RSubst}$ and each $U \in \wp_{\mathrm{f}}(\mathit{Vars})$, by

$$\alpha(\sigma, U) \stackrel{\mathrm{def}}{=} \Bigl(\bigl\{\, \operatorname{occ}(\sigma, v) \cap U \bigm| v \in \mathit{Vars} \,\bigr\} \setminus \{\varnothing\}, U\Bigr).$$

Example 8
Let us consider Examples 6 and 7 once more. Then, assuming $U = \{x_1, x_2, x_3, x_4\}$,

$$\alpha(\sigma_0, U) = \Bigl(\bigl\{\operatorname{occ}(\sigma_0, x_4)\bigr\}, U\Bigr) = \Bigl(\bigl\{\{x_1, x_2, x_3, x_4\}\bigr\}, U\Bigr).$$

As a second example, consider the substitution

$$\sigma = \bigl\{x_1 \mapsto f(x_1), x_2 \mapsto x_1, x_3 \mapsto x_1, x_4 \mapsto x_2\bigr\}.$$

Then

$$\operatorname{occ}(\sigma, x_1) = \operatorname{occ}(\sigma, x_2) = \operatorname{occ}(\sigma, x_3) = \operatorname{occ}(\sigma, x_4) = \varnothing$$

so that, if we again assume $U = \{x_1, x_2, x_3, x_4\}$,

$$\alpha\bigl(\sigma, U\bigr) = \bigl(\varnothing, U\bigr).$$

Any substitution in rational solved form is equivalent, with respect to any equality theory, to a variable-idempotent substitution having the same abstraction.

Theorem 3
If T is an equality theory and $\sigma \in \mathit{RSubst}$ is satisfiable in T, then there exists a substitution $\sigma' \in \mathit{VSubst}$ such that $\tau \in \mathit{VSubst}$, for all $\tau \subseteq \sigma'$, $T \vdash \forall(\sigma \leftrightarrow \sigma')$, $\mathit{vars}(\sigma) = \mathit{vars}(\sigma')$ and $\alpha(\sigma, U) = \alpha(\sigma', U)$, for any $U \in \wp_{\mathrm{f}}(\mathit{Vars})$.

Equivalent substitutions in rational solved form have the same abstraction. We note that this property is essential for the implementation of the SS domain.

Theorem 4
If T is a syntactic equality theory and $\sigma, \sigma' \in \mathit{RSubst}$ are satisfiable in T and such that $T \vdash \forall(\sigma \leftrightarrow \sigma')$, then $\alpha(\sigma, U) = \alpha(\sigma', U)$, for any $U \in \wp_{\mathrm{f}}(\mathit{Vars})$.

5.1 Proofs of Lemmas 11, 12, 13 and 14 and Theorems 3 and 4

Proof of Lemma 11.
The proof is by induction on n. For the base case (when $n = 1$), if $\mathit{occ}_0(\sigma, v) \neq \varnothing$, then $v \notin \mathrm{dom}(\sigma)$ and $\mathit{occ}_0(\sigma, v) = \{v\}$. Thus, $v = v\sigma$ so that, by Definition 5, $v \in \mathrm{occ}_1(\sigma, v)$. Suppose $n > 1$. Then, if $y \in \mathrm{occ}_{n-1}(\sigma, v)$, we have, by Definition 5, $\mathit{vars}(y\sigma) \cap \mathrm{occ}_{n-2}(\sigma, v) \neq \varnothing$. By the induction hypothesis,

$$\mathrm{occ}_{n-2}(\sigma, v) \subseteq \mathrm{occ}_{n-1}(\sigma, v)$$

so that $\mathit{vars}(y\sigma) \cap \mathrm{occ}_{n-1}(\sigma, v) \neq \varnothing$ and thus $y \in \mathrm{occ}_n(\sigma, v)$. $\square$

Proof of Lemma 12.
By Definition 5, $\mathrm{occ}_0(\sigma, v) = \varnothing$ and, for all $n > 0$, we have $\mathrm{occ}_n(\sigma, v) = \varnothing$ if $\mathrm{occ}_{n-1}(\sigma, v) = \varnothing$. Thus, $\mathrm{occ}_n(\sigma, v) = \varnothing$, for all $n \geq 0$, so that, by Definition 6, $\mathrm{occ}(\sigma, v) = \varnothing$. $\square$

Proof of Lemma 13.
Suppose first that $v \in \mathrm{dom}(\sigma)$. Then

$$\{\, y \in \mathit{Vars} \mid v \in \mathit{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) \,\} = \varnothing.$$

Also, by Lemma 12, $\mathrm{occ}_1(\sigma, v) = \mathrm{occ}(\sigma, v) = \varnothing$.

Suppose next that $v \notin \mathrm{dom}(\sigma)$. It follows from Definition 5, that

$$\begin{aligned}
\mathrm{occ}_0(\sigma, v) &= \{v\}, \\
\mathrm{occ}_1(\sigma, v) &= \{\, y \in \mathit{Vars} \mid \mathit{vars}(y\sigma) \cap \{v\} \neq \varnothing \,\} \\
&= \{\, y \in \mathit{Vars} \mid v \in \mathit{vars}(y\sigma) \,\},
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{occ}_2(\sigma, v) &= \Bigl\{\, y \in \mathit{Vars} \Bigm| \mathit{vars}(y\sigma) \cap \{\, y_1 \in \mathit{Vars} \mid v \in \mathit{vars}(y_1\sigma) \,\} \neq \varnothing \,\Bigr\} \\
&= \{\, y \in \mathit{Vars} \mid v \in \mathit{vars}(y\sigma^2) \,\}.
\end{aligned}$$

However, as $\sigma \in \mathit{VSubst}$, we have $\mathit{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) = \mathit{vars}(y\sigma^2) \setminus \mathrm{dom}(\sigma)$. Thus, as $v \notin \mathrm{dom}(\sigma)$, $\mathrm{occ}_1(\sigma, v) = \mathrm{occ}_2(\sigma, v)$ and hence, by Definition 5, we have also $\mathrm{occ}_n(\sigma, v) = \mathrm{occ}_1(\sigma, v)$, for all $n \geq 1$. Therefore, by Definition 6,

$$\mathrm{occ}(\sigma, v) = \mathrm{occ}_1(\sigma, v) = \{\, y \in \mathit{Vars} \mid v \in \mathit{vars}(y\sigma) \,\}. \quad \square$$

Proof of Lemma 14.
As $\sigma \in \mathit{ISubst}$ we have, for all $y \in \mathit{Vars}$, $\mathit{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) = \mathit{vars}(y\sigma)$. Also, as $\sigma \in \mathit{VSubst}$, we can apply Lemma 13 so that

$$\begin{aligned}\mathrm{occ}(\sigma, v) &= \{\, y \in \mathit{Vars} \mid v \in \mathit{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) \,\} \\ &= \{\, y \in \mathit{Vars} \mid v \in \mathit{vars}(y\sigma) \,\} \\ &= \mathrm{sg}(\sigma, v).\end{aligned}$$

□

To prove Theorem 3, we need to show that the abstraction function α is invariant with respect to $\mathcal{S}$-transformation.

Lemma 15
Let $\sigma, \sigma' \in \mathit{RSubst}$ where $\sigma \stackrel{\mathcal{S}}{\longmapsto}{}^{*} \sigma'$ and $U \in \wp_{\mathrm{f}}(\mathit{Vars})$. Then $\alpha(\sigma, U) = \alpha(\sigma', U)$.

Proof
Suppose first that $\sigma \stackrel{\mathcal{S}}{\longmapsto} \sigma'$. Thus we assume that $(x \mapsto t), (y \mapsto s) \in \sigma$, where $x \neq y$, and that

$$\sigma' = \bigl(\sigma \setminus \{y \mapsto s\}\bigr) \cup \bigl\{y \mapsto s[x/t]\bigr\}. \tag{16}$$

Suppose $v \in \mathit{Vars}$. Then we show that $\mathrm{occ}(\sigma, v) = \mathrm{occ}(\sigma', v)$.

If $x \notin \mathit{vars}(s)$, then $\sigma' = \sigma$ and there is nothing to prove. Also, if $v \in \mathrm{dom}(\sigma)$ then, by Theorem 1, $v \in \mathrm{dom}(\sigma')$ so that by Lemma 12, $\mathrm{occ}(\sigma, v) = \mathrm{occ}(\sigma', v) = \varnothing$.

We now assume that $x \in \mathit{vars}(s)$ and $v = v\sigma = v\sigma'$. We first prove that, for each $m \geq 0$,

$$\mathrm{occ}_m(\sigma, v) \subseteq \mathrm{occ}(\sigma', v). \tag{17}$$

The proof is by induction on m. By Definition 5, we have that

$$\mathrm{occ}_0(\sigma, v) = \mathrm{occ}_0(\sigma', v) = \{v\},$$

so that (17) holds for $m = 0$. Suppose then that $m > 0$ and that $v_m \in \mathrm{occ}_m(\sigma, v)$. Then, to prove (17), we must show that $v_m \in \mathrm{occ}(\sigma', v)$. By Definition 5, there exists

$$v_{m-1} \in \mathit{vars}(v_m\sigma) \cap \mathrm{occ}_{m-1}(\sigma, v). \tag{18}$$

Hence, by the inductive hypothesis, $v_{m-1} \in \mathrm{occ}(\sigma', v)$. If $v_{m-1} \in \mathit{vars}(v_m\sigma')$, then, by Eq. (15), $v_m \in \mathrm{occ}(\sigma', v)$. Suppose now that $v_{m-1} \notin \mathit{vars}(v_m\sigma')$. Since, by (18), we have that $v_{m-1} \in \mathit{vars}(v_m\sigma)$, it follows, using (16), that $v_m = y$ and $v_{m-1} = x$. However, by assumption, $v \in \mathrm{dom}(\sigma)$, so that $x \neq v$ and $m > 1$. Thus, by Definition 5, there exists

$$v_{m-2} \in \mathit{vars}(x\sigma) \cap \mathrm{occ}_{m-2}(\sigma, v). \tag{19}$$

However, $x\sigma = t$ and $x \in \mathit{vars}(s)$ so that, by (19), we have $v_{m-2} \in \mathit{vars}\bigl(s[x/t]\bigr)$. Since, by Eq. (16), $\bigl(y \mapsto s[x/t]\bigr) \in \sigma'$, we have also $v_{m-2} \in \mathit{vars}(y\sigma')$. Moreover, by (19), $v_{m-2} \in \mathrm{occ}_{m-2}(\sigma, v)$ so that, by the inductive hypothesis, we have that $v_{m-2} \in \mathrm{occ}(\sigma', v)$. Thus, by Eq. (15), as $v_m = y$, $v_m \in \mathrm{occ}(\sigma', v)$.

Conversely, we now prove that, for all $m \geq 0$,

$$\mathrm{occ}_m(\sigma', v) \subseteq \mathrm{occ}(\sigma, v). \tag{20}$$

The proof is again by induction on m. As before, $\mathrm{occ}_0(\sigma', v) = \mathrm{occ}_0(\sigma, v) = \{v\}$ so that (20) holds for $m = 0$. Suppose then that $m > 0$ and $v_m \in \mathrm{occ}_m(\sigma', v)$. Then, to prove (20), we must show that $v_m \in \mathrm{occ}(\sigma, v)$. By Definition 5, there exists

$$v_{m-1} \in \mathit{vars}(v_m \sigma') \cap \mathrm{occ}_{m-1}(\sigma', v). \tag{21}$$

Hence, by the inductive hypothesis, $v_{m-1} \in \mathrm{occ}(\sigma, v)$. If $v_{m-1} \in \mathit{vars}(v_m \sigma)$ then, by Eq. (15), we have $v_m \in \mathrm{occ}(\sigma, v)$. Suppose now that $v_{m-1} \notin \mathit{vars}(v_m \sigma)$. Since, by (21), we have $v_{m-1} \in \mathit{vars}(v_m \sigma')$, it follows, using Eq. (16), that $v_m = y$ and $v_{m-1} \in \mathit{vars}(t) = \mathit{vars}(x\sigma)$. Hence, since $v_{m-1} \in \mathrm{occ}(\sigma, v)$, by Eq. (15), we have also $x \in \mathrm{occ}(\sigma, v)$. Furthermore, $x \in \mathit{vars}(y\sigma)$ so again, by Eq. (15), as $v_m = y$, $v_m \in \mathrm{occ}(\sigma, v)$.

Combining (17) and (20) we obtain the result that, if σ' is obtained from σ by a single $\mathcal{S}$-step, then $\mathrm{occ}(\sigma, v) = \mathrm{occ}(\sigma', v)$. Thus, as $v \in \mathit{Vars}$ was arbitrary, $\alpha(\sigma, U) = \alpha(\sigma', U)$.

Suppose now that $\sigma = \sigma_1 \overset{\mathcal{S}}{\longmapsto} \cdots \overset{\mathcal{S}}{\longmapsto} \sigma_n = \sigma'$. If $n = 1$, then $\sigma = \sigma'$. If $n > 1$, we have by the first part of the proof that, for each $i = 2, \ldots, n$, $\alpha(\sigma_{i-1}, U) = \alpha(\sigma_i, U)$, and hence the required result. □

Proof of Theorem 3.
By Theorem 2, there exists $\sigma' \in \mathit{VSubst}$ such that $\sigma \overset{\mathcal{S}}{\longmapsto}{}^{*} \sigma'$ and, for any $\tau \subseteq \sigma'$, $\tau \in \mathit{VSubst}$. Moreover, by Theorem 1, $\mathit{vars}(\sigma) = \mathit{vars}(\sigma')$ and $T \vdash \forall(\sigma \leftrightarrow \sigma')$. Thus, by Lemma 15, $\alpha(\sigma, U) = \alpha(\sigma', U)$. □

To prove Theorem 4, we need to show that the abstraction function α is invariant when we exchange equivalent variables to obtain an ordered substitution.

Lemma 16
Suppose $\sigma \in \mathit{VSubst}$, $v, w \in \mathit{Vars}$ and $(v \mapsto w) \in \sigma$. Let $\rho = \{v \mapsto w, w \mapsto v\}$ be a (circular) substitution and define $\sigma' = \rho \circ \sigma = \{x\rho \mapsto t\rho \mid x \mapsto t \in \sigma\}$. Then

1. $\sigma' \in \mathit{VSubst}$,
2. $\mathit{vars}(\sigma) = \mathit{vars}(\sigma')$,
3. $\alpha(\sigma, U) = \alpha(\sigma', U)$, for all $U \in \wp_{\mathrm{f}}(\mathit{Vars})$, and
4. $T \vdash \forall(\sigma \leftrightarrow \sigma')$, for any equality theory T.

Proof
Since σ' is obtained from σ by renaming variables and $\sigma \in \mathit{VSubst}$, we have also that $\sigma' \in \mathit{VSubst}$. In addition, $\mathit{vars}(\sigma) \setminus \{v, w\} = \mathit{vars}(\sigma') \setminus \{v, w\}$ so that, since $(v \mapsto w) \in \mathit{vars}(\sigma)$ and $(w \mapsto v) \in \mathit{vars}(\sigma')$, we have $\mathit{vars}(\sigma) = \mathit{vars}(\sigma')$.

To prove 16.3, we have to show that, if

$$\alpha(\sigma, U) \overset{\mathrm{def}}{=} (\mathit{sh}, U) \quad \text{and} \quad \alpha(\sigma', U) \overset{\mathrm{def}}{=} (\mathit{sh}', U),$$

then $\mathit{sh} = \mathit{sh}'$. By the hypothesis, for all $y \in \mathit{Vars}$ we have $x \in \mathit{vars}(y\sigma)$ if and only if $x\rho \in \mathit{vars}(y\sigma')$. As $\sigma, \sigma' \in \mathit{VSubst}$, we can use the alternative characterisation of occ given by Lemma 13 and conclude that, for each $x \in \mathit{Vars}$, $\mathrm{occ}(\sigma, x) = \mathrm{occ}(\sigma', x\rho)$. Therefore $sh \subseteq sh'$. The reverse inclusion follows by symmetry so that $sh = sh'$.

To prove 16.4, we first show by induction on the depth of $r \in \mathcal{T}_{\mathit{Vars}}$ that

$$T \vdash \forall\bigl((v = w) \rightarrow (r = r\rho)\bigr). \tag{22}$$

For the base case, r has depth 1. If r is a constant or a variable other than v or w, then $r = r\rho$. If $r = v$, then $r\rho = w$ and $T \vdash \forall\bigl((v = w) \rightarrow (v = w)\bigr)$. Finally, if $r = w$, then $r\rho = v$ and we have, using the congruence axioms, that $T \vdash \forall\bigl((v = w) \rightarrow (w = v)\bigr)$. For the inductive step, let $r = f(r_1, \ldots, r_n)$. Then $r\rho = f(r_1\rho, \ldots, r_n\rho)$. Thus, using the inductive hypothesis, for each $i = 1, \ldots, n$, $T \vdash \forall\bigl((v = w) \rightarrow (r_i = r_i\rho)\bigr)$. Hence, by the congruence axioms, (22) holds.

Note that $(v \mapsto w) \in \sigma$. Thus, it follows from (22) that, for each $(x \mapsto t) \in \sigma$, $T \vdash \forall\bigl(\sigma \rightarrow \{x = t, x = x\rho, t = t\rho\}\bigr)$ and hence, using the congruence axioms, $T \vdash \forall\bigl(\sigma \rightarrow \{x\rho = t\rho\}\bigr)$. Thus, $T \vdash \forall(\sigma \rightarrow \sigma')$. Since $(w \mapsto v) \in \sigma'$, the reverse implication follows by symmetry so that $T \vdash \forall(\sigma' \leftrightarrow \sigma)$. □

Lemma 17

Suppose $\sigma \in \mathit{VSubst}$. Then there exists $\sigma' \in \mathit{VSubst}$ that is ordered such that $\mathit{vars}(\sigma) = \mathit{vars}(\sigma')$, $\alpha(\sigma, U) = \alpha(\sigma', U)$, for all $U \in \wp_{\mathrm{f}}(\mathit{Vars})$, and $T \vdash \forall(\sigma \leftrightarrow \sigma')$, for any equality theory T.

Proof

The proof is by induction on the number $b \geq 0$ of the bindings $(v \mapsto w) \in \sigma$ such that $w \in \mathrm{param}(\sigma)$ and $w > v$ (the number of *unordered* bindings). For the base case, when $b = 0$, σ is ordered and the result holds by taking $\sigma' = \sigma$.

For the inductive case, when $b > 0$, let $(v \mapsto w) \in \sigma$ be an unordered binding and define $\rho = \{v \mapsto w, w \mapsto v\}$. Then, by Lemma 16, we have $\rho \circ \sigma \in \mathit{VSubst}$, $\mathit{vars}(\sigma) = \mathit{vars}(\rho \circ \sigma)$, $\alpha(\sigma, U) = \alpha(\rho \circ \sigma, U)$, for all $U \in \wp_{\mathrm{f}}(\mathit{Vars})$, and, finally, $T \vdash \forall(\sigma \leftrightarrow \rho \circ \sigma)$, for any equality theory T. In order to apply the inductive hypothesis to $\rho \circ \sigma$, we must show that the number of unordered bindings in $\rho \circ \sigma$ is less than b. To this end, roughly speaking, we start showing that any ordered binding in σ is mapped by ρ into another ordered binding in $\rho \circ \sigma$, therefore proving that the number of unordered bindings is not increasing. There are three cases. First, any ordered binding $(y \mapsto t) \in \sigma$ such that $t \notin \mathit{Vars}$ is mapped by ρ into the binding $(y\rho \mapsto t\rho) \in (\rho \circ \sigma)$ which is clearly ordered, since $t\rho \notin \mathit{Vars}$. Second, consider any ordered binding $(y \mapsto z) \in \sigma$ such that $z \in \mathrm{dom}(\sigma)$. Since $w \in \mathrm{param}(\sigma)$, we have $z \neq w$. If also $z \neq v$ then we have $z\rho = z$ and $z \in \mathrm{dom}(\rho \circ \sigma)$; otherwise $z = v$ so that $z\rho = w$ and, as $(w \mapsto v) \in (\rho \circ \sigma)$, $z\rho \in \mathrm{dom}(\rho \circ \sigma)$. Thus, in either case, such a binding is mapped by ρ into the binding $(y\rho \mapsto z\rho) \in (\rho \circ \sigma)$ which is ordered since $z\rho \in \mathrm{dom}(\rho \circ \sigma)$. Third, consider any ordered binding $(y \mapsto z) \in \sigma$ such that $z \in \mathrm{param}(\sigma)$ and $z < y$. The ordering relation implies $y \neq v$ and we also have $y \neq w$, since $w \in \mathrm{param}(\sigma)$. Hence, we obtain $y\rho = y$. Now, as $z \in \mathrm{param}(\sigma)$, $z \neq v$. If $z \neq w$, then $z\rho = z$. On the other hand, if $z = w$, then $z\rho = v$ so that $z\rho < z$. Thus, in both cases, as $z < y$, $z\rho < y$. and hence, $(y\rho \mapsto z\rho) \in (\rho \circ \sigma)$ is ordered. Finally, to show that the number of unordered bindings is strictly decreasing, we note that the unordered binding $(v \mapsto w) \in \sigma$ is mapped by ρ into the binding $(w \mapsto v) \in (\rho \circ \sigma)$, which is ordered.

Therefore, by applying the inductive hypothesis, there exists a substitution σ' such that $\sigma' \in \mathit{VSubst}$ is ordered, $\mathit{vars}(\rho \circ \sigma) = \mathit{vars}(\sigma')$, $\alpha(\rho \circ \sigma, U) = \alpha(\sigma', U)$, for all $U \in \wp_{\mathrm{f}}(\mathit{Vars})$, and $T \vdash \forall(\rho \circ \sigma \leftrightarrow \sigma')$, for any equality theory T. Then the required result follows by transitivity. $\square$

Proof of Theorem 4.
By Theorem 3, we can assume that $\sigma, \sigma' \in \mathit{VSubst}$, $T \vdash \forall(\sigma \leftrightarrow \sigma')$ and, for any $U \in \wp_{\mathrm{f}}(\mathit{Vars})$, $\alpha(\sigma, U) = \alpha(\sigma', U)$. By Lemma 17, we can assume that σ, σ' are also ordered substitutions so that, by Lemma 5, $\mathrm{dom}(\sigma') = \mathrm{dom}(\sigma)$.

To prove the result we need to show that, for all $v \in \mathit{Vars}$, we have both $\mathrm{occ}(\sigma, v) \subseteq \mathrm{occ}(\sigma', v)$ and $\mathrm{occ}(\sigma', v) \subseteq \mathrm{occ}(\sigma, v)$. We just prove the first of these as the other case is symmetric.

Suppose that $w \in \mathit{Vars}$ and that $v \in \mathit{vars}(w\sigma) \setminus \mathrm{dom}(\sigma)$. Then, using the alternative characterisation of occ for variable-idempotent substitutions given by Lemma 13, we just have to show that $v \in \mathit{vars}(w\sigma') \setminus \mathrm{dom}(\sigma')$.

By Lemma 6 (replacing τ by σ, σ by σ' and $s = t$ by $w = w$), we have that there exists $z \in \mathit{vars}(w\sigma') \setminus \mathrm{dom}(\sigma')$ such that $v \in \mathit{vars}(z\sigma)$. Thus as $\mathrm{dom}(\sigma') = \mathrm{dom}(\sigma)$, $z \notin \mathrm{dom}(\sigma)$, and hence, $v = z$ so that $v \in \mathit{vars}(w\sigma') \setminus \mathrm{dom}(\sigma')$, as required. $\square$

6 Abstract Unification

The operations of abstract unification together with statements of the main results are presented here in three stages. In the first two stages, we consider substitutions containing just a single binding. For the first, it is assumed that the set of variables of interest is fixed so that the definition is based on the *SH* domain. Then, in the second, using the *SS* domain, the definition is extended to allow for the introduction of new variables in the binding. The final stage extends this definition further to deal with arbitrary substitutions.

6.1 Abstract Operations for Sharing Sets

The abstract unifier amgu abstracts the effect of a single binding on an element of the *SH* domain. For this we need some ancillary definitions.

Definition 8
(Auxiliary functions.) The *closure under union* function (also called *star-union*), $(\cdot)^\star \colon SH \to SH$, is given, for each $sh \in SH$, by

$$sh^\star \stackrel{\mathrm{def}}{=} \{\, S \in SG \mid \exists n \geq 1 \,.\, \exists S_1, \ldots, S_n \in sh \,.\, S = S_1 \cup \cdots \cup S_n \,\}.$$

For each $sh \in SH$ and each $V \in \wp_{\mathrm{f}}(\mathit{Vars})$, the extraction of the *relevant component of sh with respect to V* is encoded by $\mathrm{rel} \colon \wp_{\mathrm{f}}(\mathit{Vars}) \times SH \to SH$ defined as

$$\mathrm{rel}(V, sh) \stackrel{\mathrm{def}}{=} \{\, S \in sh \mid S \cap V \neq \varnothing \,\}.$$

For each $sh_1, sh_2 \in SH$, the *binary union* function $\mathrm{bin}\colon SH \times SH \to SH$ is given by

$$\mathrm{bin}(sh_1, sh_2) \stackrel{\mathrm{def}}{=} \{\, S_1 \cup S_2 \mid S_1 \in sh_1, S_2 \in sh_2 \,\}.$$

Definition 9
(amgu.) The function $\mathrm{amgu}\colon SH \times \mathit{Bind} \to SH$ captures the effects of a binding on an SH element. Suppose $x \in \mathit{Vars}$, $r \in \mathcal{T}_{\mathit{Vars}}$, and $sh \in SH$. Let

$$A \stackrel{\mathrm{def}}{=} \mathrm{rel}\bigl(\{x\}, sh\bigr),$$
$$B \stackrel{\mathrm{def}}{=} \mathrm{rel}\bigl(\mathit{vars}(r), sh\bigr).$$

Then

$$\mathrm{amgu}(sh, x \mapsto r) \stackrel{\mathrm{def}}{=} \bigl(sh \setminus (A \cup B)\bigr) \cup \mathrm{bin}(A^\star, B^\star).$$

The following soundness result for amgu is proved in Section 6.4.

Theorem 5
Let T be a syntactic equality theory, $(sh, U) \in SS$ an abstract description and $\{x \mapsto r\}, \sigma \in \mathit{RSubst}$ such that $\mathit{vars}(x \mapsto r) \cup \mathit{vars}(\sigma) \subseteq U$. Suppose that there exists a most general solution μ for $\bigl(\{x = r\} \cup \sigma\bigr)$ in T. Then

$$\alpha(\sigma, U) \preceq_{ss} (sh, U) \implies \alpha(\mu, U) \preceq_{ss} \bigl(\mathrm{amgu}(sh, x \mapsto r), U\bigr).$$

The following theorems, proved in Section 6.4, show that amgu is idempotent and commutative.

Theorem 6
Let $sh \in SH$ and $(x \mapsto r) \in \mathit{Bind}$. Then

$$\mathrm{amgu}(sh, x \mapsto r) = \mathrm{amgu}\bigl(\mathrm{amgu}(sh, x \mapsto r), x \mapsto r\bigr).$$

Theorem 7
Let $sh \in SH$ and $(x \mapsto r), (y \mapsto t) \in \mathit{Bind}$. Then

$$\mathrm{amgu}\bigl(\mathrm{amgu}(sh, x \mapsto r), y \mapsto t\bigr) = \mathrm{amgu}\bigl(\mathrm{amgu}(sh, y \mapsto t), x \mapsto r\bigr).$$

6.2 Abstract Operations for Sharing Domains

The definitions and results of Section 6.1 can be lifted to apply to the proper set-sharing domain.

Definition 10
(Amgu.) The operation $\mathrm{Amgu}\colon SS \times \mathit{Bind} \to SS$ extends the SS description it takes as an argument to the set of variables occurring in the binding it is given as the second argument. Then it applies amgu. Formally:

$$U' \stackrel{\mathrm{def}}{=} \mathit{vars}(x \mapsto r) \setminus U,$$
$$\mathrm{Amgu}\bigl((sh, U), x \mapsto r\bigr) \stackrel{\mathrm{def}}{=} \Bigl(\mathrm{amgu}\Bigl(sh \cup \{\, \{u\} \mid u \in U' \,\}, x \mapsto r\Bigr), U \cup U'\Bigr).$$

The results for amgu can easily be extended to apply to Amgu giving us the following corollaries.

Corollary 1
Let T be a syntactic equality theory, $(sh, U) \in SS$ and $\{x \mapsto r\}, \sigma \in RSubst$ such that $vars(\sigma) \subseteq U$. Suppose there exists a most general solution μ for $\bigl(\{x = r\} \cup \sigma\bigr)$ in T. Then

$$\alpha(\sigma, U) \preceq_{SS} (sh, U) \implies \alpha\bigl(\mu, U \cup vars(x \mapsto r)\bigr) \preceq_{SS} \mathrm{Amgu}\bigl((sh, U), x \mapsto r\bigr).$$

Corollary 2
Let $sh \in SH$ and $(x \mapsto r) \in Bind$. Then

$$\mathrm{Amgu}\bigl((sh, U), x \mapsto r\bigr) = \mathrm{Amgu}\Bigl(\mathrm{Amgu}\bigl((sh, U), x \mapsto r\bigr), x \mapsto r\Bigr).$$

Corollary 3
Let $sh \in SH$ and $(x \mapsto r), (y \mapsto t) \in Bind$. Then

$$\begin{aligned}\mathrm{Amgu}\Bigl(\mathrm{Amgu}\bigl((sh, U), x \mapsto r\bigr), y \mapsto t\Bigr)& \\ = \mathrm{Amgu}\Bigl(\mathrm{Amgu}\bigl((sh, U), y \mapsto t\bigr), x \mapsto r\Bigr)&.\end{aligned}$$

6.3 Abstract Unifiers for Sharing

We now extend the above definitions and results for a single binding to any substitution.

Definition 11
(aunify.**)** The function aunify: $SS \times RSubst \to SS$ generalizes Amgu to any substitution $\mu \in RSubst$ in the context of some syntactic equality theory T: If we have $(sh, U) \in SS$, then

$$\mathrm{aunify}\bigl((sh, U), \varnothing\bigr) \stackrel{\mathrm{def}}{=} (sh, U);$$

if μ holds in T and $(x \mapsto r) \in \mu$,

$$\mathrm{aunify}\bigl((sh, U), \mu\bigr) \stackrel{\mathrm{def}}{=} \mathrm{aunify}\Bigl(\bigl(\mathrm{Amgu}(sh, U), x \mapsto r\bigr), \mu \setminus \{x \mapsto r\}\Bigr);$$

and, if μ does not hold in T,

$$\mathrm{aunify}\bigl((sh, U), \mu\bigr) \stackrel{\mathrm{def}}{=} \bot.$$

For the distinguished elements $\bot$ and $\top$ of SS,

$$\mathrm{aunify}(\bot, \mu) \stackrel{\mathrm{def}}{=} \bot,$$
$$\mathrm{aunify}(\top, \mu) \stackrel{\mathrm{def}}{=} \top.$$

As a result of Corollary 3, Amgu and aunify commute.

Lemma 18
Let $(sh, U) \in SS$, $\nu \in RSubst$ and $(y \mapsto t) \in Bind$. Then

$$\operatorname{aunify}\Bigl(\operatorname{Amgu}\bigl((sh, U), y \mapsto t\bigr), \nu\Bigr) = \operatorname{Amgu}\Bigl(\operatorname{aunify}\bigl((sh, U), \nu\bigr), y \mapsto t\Bigr).$$

As a consequence of this and Corollaries 1, 2 and 3, we have the following soundness, idempotence and commutativity results required for aunify to be sound and well-defined.

Theorem 8
Let T be a syntactic equality theory, $(sh, U) \in SS$ and $\sigma, \nu \in RSubst$ such that $vars(\sigma) \subseteq U$. Suppose also that there exists a most general solution μ for $(\nu \cup \sigma)$ in T. Then

$$\alpha(\sigma, U) \preceq_{SS} (sh, U) \implies \alpha\bigl(\mu, U \cup vars(\nu)\bigr) \preceq_{SS} \operatorname{aunify}\bigl((sh, U), \mu\bigr).$$

This theorem shows also that it is safe for the analyzer to perform part or all of the concrete unification algorithm before computing aunify.

Theorem 9
Let $(sh, U) \in SS$ and $\nu \in RSubst$. Then

$$\operatorname{aunify}\bigl((sh, U), \nu\bigr) = \operatorname{aunify}\Bigl(\operatorname{aunify}\bigl((sh, U), \nu\bigr), \nu\Bigr).$$

Theorem 10
Let $(sh, U) \in SS$ and $\nu_1, \nu_2 \in RSubst$. Then

$$\operatorname{aunify}\Bigl(\operatorname{aunify}\bigl((sh, U), \nu_1\bigr), \nu_2\Bigr) = \operatorname{aunify}\Bigl(\operatorname{aunify}\bigl((sh, U), \nu_2\bigr), \nu_1\Bigr).$$

The proofs of all these results are in Section 6.5.

6.4 Proofs of Results for Sharing-Sets

In the proofs we use the fact that $(\cdot)^\star$ and rel are monotonic so that

$$sh_1 \subseteq sh_2 \implies sh_1^\star \subseteq sh_2^\star, \tag{23}$$

$$sh_1 \subseteq sh_2 \implies \operatorname{rel}(sh_1, U) \subseteq \operatorname{rel}(sh_2, U). \tag{24}$$

We will also use the fact that $(\cdot)^\star$ is idempotent.

Let $t_1, \ldots, t_n$ be terms. For the sake of brevity we will use the notation $v_{t_1 \cdots t_n}$ to denote $\bigcup_{i=1}^n vars(t_i)$. In particular, if x and y are variables, and r and t are terms, we will use the following definitions:

$$\begin{aligned} v_x &\stackrel{\mathrm{def}}{=} \{x\}, & v_y &\stackrel{\mathrm{def}}{=} \{y\}, \\ v_r &\stackrel{\mathrm{def}}{=} vars(r), & v_t &\stackrel{\mathrm{def}}{=} vars(t), \\ v_{xr} &\stackrel{\mathrm{def}}{=} v_x \cup v_r, & v_{yt} &\stackrel{\mathrm{def}}{=} v_y \cup v_t. \end{aligned}$$

Definition 12
($\overline{\mathrm{rel}}$.) Suppose $V \in \wp_{\mathrm{f}}(Vars)$ and $sh \in SH$. Then

$$\overline{\operatorname{rel}}(V, sh) \stackrel{\mathrm{def}}{=} sh \setminus \operatorname{rel}(V, sh).$$

Notice that if $S \in \overline{\mathrm{rel}}(V, sh)$ then $S \cap V = \varnothing$. Conversely, if $S \in sh$ and $S \cap V = \varnothing$ then $S \in \overline{\mathrm{rel}}(V, sh)$. The following definition of amgu is clearly equivalent to the one given in Definition 9: for each variable x, each term r, and each $sh \in SH$,

$$\mathrm{amgu}(sh, x \mapsto r) \stackrel{\mathrm{def}}{=} \overline{\mathrm{rel}}(v_{xr}, sh) \cup \mathrm{bin}\bigl(\mathrm{rel}(v_x, sh)^\star, \mathrm{rel}(v_r, sh)^\star\bigr). \tag{25}$$

Proof of Theorem 5.
We first prove the result under the assumption that $\alpha(\sigma, U) = (sh, U)$. We do this in two parts. In the first, we partition σ into two substitutions one of which, called σ^-, is the same as σ when σ and μ are idempotent. We construct a new substitution ν which, in the case that σ and μ are idempotent, is a most general solution for $x\sigma = r\sigma$. Finally we compose ν with σ^- to define a substitution that has the same abstraction as μ but with a number of useful properties including that of variable-idempotence. In the second part, we use this composed substitution in place of μ to prove the result.

Part 1. By Theorem 3, we can assume that

$$\sigma \in \mathit{VSubst} \tag{26}$$

and that all subsets of σ are in VSubst. Let $\sigma^\circ, \sigma^- \in \mathit{RSubst}$ be defined such that

$$\sigma^- \cup \sigma^\circ = \sigma, \tag{27}$$

$$\mathrm{dom}(\sigma^\circ) = \mathrm{dom}(\sigma) \cap \bigcup_{i \geq 1} \mathit{vars}(x\sigma^i = r\sigma^i), \tag{28}$$

$$\mathrm{dom}(\sigma^-) \cap \mathrm{dom}(\sigma^\circ) = \varnothing. \tag{29}$$

Then, it follows from the above assumption on subsets of σ that

$$\sigma^- \in \mathit{VSubst}, \quad \sigma^\circ \in \mathit{VSubst}. \tag{30}$$

Now, suppose $z \in \mathit{vars}(\sigma^\circ) \setminus \mathrm{dom}(\sigma^\circ)$. Then $z \in \mathit{vars}(y\sigma^\circ)$ for some $y \in \mathrm{dom}(\sigma^\circ)$. Thus, by (28), for some $j \geq 2$, $z \in \mathit{vars}(x\sigma^j = r\sigma^j) \setminus \mathrm{dom}(\sigma^\circ)$ and, again by (28), $z \notin \mathrm{dom}(\sigma)$ so that, by (26), $z \in \mathit{vars}(x\sigma = r\sigma)$. Therefore, as z was an arbitrary variable in $\mathit{vars}(\sigma^\circ) \setminus \mathrm{dom}(\sigma^\circ)$,

$$\mathit{vars}(\sigma^\circ) \subseteq \bigl(\mathit{vars}(x\sigma = r\sigma) \cup \mathrm{dom}(\sigma^\circ)\bigr). \tag{31}$$

It follows from (28) that $\mathrm{dom}(\sigma) \cap \mathit{vars}(x\sigma = r\sigma) \subseteq \mathrm{dom}(\sigma^\circ)$ so that, by (29)

$$\mathrm{dom}(\sigma^-) \cap \mathit{vars}(x\sigma = r\sigma) = \varnothing. \tag{32}$$

Hence, by (29) and (31), we have

$$\mathrm{dom}(\sigma^-) \cap \mathit{vars}(\sigma^\circ) = \varnothing. \tag{33}$$

Let $\nu \in \mathit{RSubst}$ be a most general solution for $\{x\sigma = r\sigma\} \cup \sigma^\circ$ in T so that

$$T \vdash \forall\bigl(\nu \leftrightarrow \{x\sigma = r\sigma\} \cup \sigma^\circ\bigr), \tag{34}$$

$$\mathit{vars}(\nu) \subseteq \bigl(\mathit{vars}(x\sigma = r\sigma) \cup \mathit{vars}(\sigma^\circ)\bigr). \tag{35}$$

By Theorem 3, we can assume that

$$\nu \in \mathit{VSubst}. \tag{36}$$

By (32), (33), and (35), we have

$$\mathrm{dom}(\sigma^-) \cap \mathit{vars}(\nu) = \varnothing. \tag{37}$$

Therefore, as $\sigma^-, \nu \in \mathit{VSubst}$ (by (30) and (36)), we can use Lemma 7 to obtain the following properties for $\nu \circ \sigma^-$.

$$T \vdash \forall\bigl((\nu \circ \sigma^-) \leftrightarrow (\nu \cup \sigma^-)\bigr), \tag{38}$$

$$\mathrm{dom}(\nu \circ \sigma^-) = \mathrm{dom}(\nu \cup \sigma^-), \tag{39}$$

$$\nu \circ \sigma^- \in \mathit{VSubst}. \tag{40}$$

Now we have

$$T \vdash \forall\bigl(\mu \leftrightarrow \{x = r\} \cup \sigma\bigr)$$

[by hypothesis]

$$T \vdash \forall\bigl(\mu \leftrightarrow \{x\sigma = r\sigma\} \cup \sigma\bigr)$$

[by Lemma 2 and the congruence axioms]

$$T \vdash \forall\bigl(\mu \leftrightarrow \nu \cup \sigma^-\bigr)$$

[by (27) and (34)]

$$T \vdash \forall\bigl(\mu \leftrightarrow \nu \circ \sigma^-\bigr) \tag{41}$$

[by (38)].

Therefore, by Theorem 4,

$$\alpha(\mu, U) = \alpha(\nu \circ \sigma^-, U). \tag{42}$$

Part 2. To prove the result under the assumption that $\alpha(\sigma, U) = (\mathit{sh}, U)$, we define $\mathit{sh}' \in \mathit{SH}$ so that

$$\alpha(\mu, U) = (\mathit{sh}', U). \tag{43}$$

Then, by (42), $\alpha(\nu \circ \sigma^-, U) = (\mathit{sh}', U)$. We show that $\mathit{sh}' \subseteq \mathrm{amgu}(\mathit{sh}, x \mapsto r)$. If $\mathit{sh}' = \varnothing$, there is nothing to prove. Therefore, we assume that there exists $S \in \mathit{sh}'$ so that $S \neq \varnothing$ and, for some $v \in \mathit{Vars}$,

$$v \notin \mathrm{dom}(\nu \circ \sigma^-), \tag{44}$$

$$S \stackrel{\mathrm{def}}{=} \mathrm{occ}(\nu \circ \sigma^-, v). \tag{45}$$

Note that (39) and (44) imply that

$$v \notin \mathrm{dom}(\nu), \qquad v \notin \mathrm{dom}(\sigma^-). \tag{46}$$

Let

$$S' \stackrel{\mathrm{def}}{=} \bigcup\bigl\{\, \mathrm{occ}(\sigma, y) \bigm| y \in \mathrm{occ}(\nu, v) \,\bigr\}. \tag{47}$$

We show that

$$S = S'. \tag{48}$$

By (26), (36) and (40), $\sigma, \nu, \nu \circ \sigma^- \in \mathit{VSubst}$ and, by (44) and (46), $v \notin \mathrm{dom}(\nu \circ \sigma^-)$ and $v \notin \mathrm{dom}(\nu)$. Thus, it follows from Lemma 13 with (45) and (47), that it suffices to show that, for each $w \in \mathit{Vars}$, $v \in \mathit{vars}(w\sigma^-\nu)$ if and only if there exists $z \in \mathit{vars}(w\sigma) \setminus \mathrm{dom}(\sigma)$ such that $v \in \mathit{vars}(z\nu)$.

First, we suppose that $v \in \mathit{vars}(w\sigma^-\nu)$. Thus, there exists $y \in \mathit{vars}(w\sigma^-)$ such that $v \in \mathit{vars}(y\nu)$. Since $\sigma^\circ, \nu \in \mathit{VSubst}$ (by (30) and (36)), $T \vdash \forall(\nu \rightarrow \sigma^\circ)$ (by (34)), $v \notin \mathrm{dom}(\nu)$ (by (46)) and $T \vdash \forall\bigl(\nu \rightarrow (y\nu = y)\bigr)$ (using Lemma 2), we can apply Lemma 6 (replacing τ by ν, σ by σ° and $s = t$ by $y\nu = y$) so that there exists $z \in \mathit{vars}(y\sigma^\circ) \setminus \mathrm{dom}(\sigma^\circ)$ such that $v \in \mathit{vars}(z\nu)$. We want to show that $z \in \mathit{vars}(w\sigma) \setminus \mathrm{dom}(\sigma)$. Now either $z \in \mathrm{dom}(\nu)$ or $z = v$ so that, by (37) (if $z \in \mathrm{dom}(\nu)$) or (46) (if $z = v$), $z \notin \mathrm{dom}(\sigma^-)$. However, $z \notin \mathrm{dom}(\sigma^\circ)$, so that, by (27), $z \notin \mathrm{dom}(\sigma)$. Thus, it remains to prove that $z \in \mathit{vars}(w\sigma)$. Now, as $y \in \mathit{vars}(w\sigma^-)$ and $z \in \mathit{vars}(y\sigma^\circ)$, we have $z \in \mathit{vars}(w\sigma^-\sigma^\circ)$. So we must show that $\mathit{vars}(w\sigma^-\sigma^\circ) \setminus \mathrm{dom}(\sigma) \subseteq \mathit{vars}(w\sigma)$. To see this note that, if $w \notin \mathrm{dom}(\sigma^-)$, then $w\sigma^- = w$ and, by (27), $w\sigma^\circ = w\sigma$ so that $w\sigma^-\sigma^\circ = w\sigma$. On the other hand, if $w \in \mathrm{dom}(\sigma^-)$, then, by (27), $w\sigma^- = w\sigma$ so that $w\sigma^-\sigma^\circ = w\sigma\sigma^\circ$ Now, as $\sigma \in \mathit{VSubst}$ and $\sigma^\circ \subseteq \sigma$ (by (26) and (27)), we can apply Lemma 4 so that $\mathit{vars}(w\sigma\sigma^\circ) \setminus \mathrm{dom}(\sigma) \subseteq \mathit{vars}(w\sigma)$. Hence, $\mathit{vars}(w\sigma^-\sigma^\circ) \setminus \mathrm{dom}(\sigma) \subseteq \mathit{vars}(w\sigma)$.

Secondly, suppose there exists $z \in \mathit{vars}(w\sigma) \setminus \mathrm{dom}(\sigma)$ such that $v \in \mathit{vars}(z\nu)$. Then $v \in \mathit{vars}(w\sigma\nu)$. We need to show that $v \in \mathit{vars}(w\sigma^-\nu)$. By Eq. (27), if $w \in \mathrm{dom}(\sigma^-)$, then $w\sigma^-\nu = w\sigma\nu$ so that $v \in \mathit{vars}(w\sigma^-\nu)$. On the other hand, if $w \notin \mathrm{dom}(\sigma^-)$, then again, by (27), $v \in \mathit{vars}(w\sigma^\circ\nu)$. Moreover, $w = w\sigma^-$ so that, by (34) and Lemma 2 with the congruence axioms, $T \vdash \forall\bigl(\nu \rightarrow (w\sigma^\circ\nu = w\sigma^-)\bigr)$. Hence, since $\nu \in \mathit{VSubst}$ (by (36)) and $v \notin \mathrm{dom}(\nu)$ (by (46)), we can apply Lemma 6 (replacing τ by ν, σ by the empty substitution and $s = t$ by $w\sigma^\circ\nu = w\sigma^-$) and obtain $v \in \mathit{vars}(w\sigma^-\nu)$.

Therefore, as a consequence of the previous two paragraphs, for each $w \in \mathit{Vars}$, we have $v \in \mathit{vars}(w\sigma^-\nu)$ if and only if there exists $z \in \mathit{vars}(w\sigma) \setminus \mathrm{dom}(\sigma)$ such that $v \in \mathit{vars}(z\nu)$. It therefore follows that Eq. (48) holds.

Let

$$S_x \stackrel{\mathrm{def}}{=} \bigcup\Bigl(\bigl\{\, \mathrm{occ}(\sigma, y) \bigm| y \in \mathrm{occ}(\nu, v) \,\bigr\} \cap \mathrm{rel}(v_x, \mathit{sh})\Bigr), \tag{49}$$

$$S_r \stackrel{\mathrm{def}}{=} \bigcup\Bigl(\bigl\{\, \mathrm{occ}(\sigma, y) \bigm| y \in \mathrm{occ}(\nu, v) \,\bigr\} \cap \mathrm{rel}(v_r, \mathit{sh})\Bigr), \tag{50}$$

$$S_0 \stackrel{\mathrm{def}}{=} \bigcup\Bigl(\bigl\{\, \mathrm{occ}(\sigma, y) \bigm| y \in \mathrm{occ}(\nu, v) \,\bigr\} \cap \overline{\mathrm{rel}}(v_{xr}, \mathit{sh})\Bigr). \tag{51}$$

Note that by (47), (48) and the fact that

$$\overline{\mathrm{rel}}(v_{xr}, \mathit{sh}) = \mathit{sh} \setminus \bigl(\mathrm{rel}(v_x, \mathit{sh}) \cup \mathrm{rel}(v_r, \mathit{sh})\bigr),$$

we have

$$S_0 = S \setminus (S_x \cup S_r). \tag{52}$$

We now consider the two cases $S_0 \neq \varnothing$ and $S_0 = \varnothing$ separately.

Consider first the case when $S_0 \neq \varnothing$. Then, by (51), for some $y \in \mathit{Vars}$,

$$y \in \mathrm{occ}(\nu, v), \tag{53}$$

$$\mathrm{occ}(\sigma, y) \in \overline{\mathrm{rel}}(v_{xr}, sh). \tag{54}$$

Thus, by Lemma 12, $y \notin \mathrm{dom}(\sigma)$ and hence, by (27), $y \notin \mathrm{dom}(\sigma^\circ)$. Also, by (54), $\mathrm{occ}(\sigma, y) \cap v_{xr} = \varnothing$. Thus as $\sigma \in \mathit{VSubst}$ (by (26)) we can use Lemma 13 to see that, for each $w \in v_{xr}$, $y \notin \mathit{vars}(w\sigma)$ and hence, $y \notin \mathit{vars}(x\sigma = r\sigma)$. Therefore, by (31) and (35), $y \notin \mathit{vars}(\nu)$. As $\nu \in \mathit{VSubst}$ (by (36)), we can apply Lemma 13 to both $\mathrm{occ}(\nu, y)$ and $\mathrm{occ}(\nu, v)$. Thus, as $y \notin \mathit{vars}(\nu)$, $\mathrm{occ}(\nu, y) = \{y\}$ and also (using (53)) $v = y$ so that $\mathrm{occ}(\nu, v) = \{v\}$. It therefore follows from (47) and (48) that $S = \mathrm{occ}(\sigma, v)$ and hence from (54), that

$$S \in \overline{\mathrm{rel}}(v_{xr}, sh). \tag{55}$$

Now consider the case when $S_0 = \varnothing$. By (52), and the assumption that $S \neq \varnothing$,

$$S = S_x \cup S_r \neq \varnothing. \tag{56}$$

As a consequence of (49) and (50),

$$S_x \in \mathrm{rel}(v_x, sh)^\star \cup \varnothing, \tag{57}$$

$$S_r \in \mathrm{rel}(v_r, sh)^\star \cup \varnothing. \tag{58}$$

Now, by (56) either $S_x \neq \varnothing$ or $S_r \neq \varnothing$. We will show that both $S_x \neq \varnothing$ and $S_r \neq \varnothing$. Suppose first that $S_x \neq \varnothing$. Then, by (57), $x \in S_x$. Hence, by (56), $x \in S$. By (45), $x \in \mathrm{occ}(\nu \circ \sigma^-, v)$. However, $\nu \circ \sigma^- \in \mathit{VSubst}$ (by (40)) so that we can apply Lemma 13 to $\mathrm{occ}(\nu \circ \sigma^-, v)$ and obtain that $v \in \mathit{vars}(x\sigma^-\nu)$. By the definition of μ in the hypothesis and (41), $T \vdash \forall\bigl(\nu \circ \sigma^- \rightarrow (x = r)\bigr)$ and hence, by Lemma 2 with the congruence axioms, $T \vdash \forall\bigl(\nu \circ \sigma^- \rightarrow (x\sigma^-\nu = r)\bigr)$. Thus, as $\nu \circ \sigma^- \in \mathit{VSubst}$ (by (40)) and $v \notin \mathrm{dom}(\nu \circ \sigma^-)$ (by (44)), we have, by Lemma 6 (replacing τ by $\nu \circ \sigma^-$, σ by the empty substitution and $s = t$ by $x\sigma^-\nu = r$), $v \in \mathit{vars}(r\sigma^-\nu)$. By re-applying Lemma 13 to $\mathrm{occ}(\nu \circ \sigma^-, v)$, it can be seen that, as $v \notin \mathrm{dom}(\nu)$ (by (44)), $v_r \cap \mathrm{occ}(\nu \circ \sigma^-, v) \neq \varnothing$. Hence, by (45), $S \cap v_r \neq \varnothing$. Thus, by (47) and (48), there exists a $y \in \mathrm{occ}(\nu, v)$ such that $\mathrm{occ}(\sigma, y) \cap v_r \neq \varnothing$. Therefore, by (50), $S_r \cap v_r \neq \varnothing$ and so $S_r \neq \varnothing$. Secondly, by a similar argument, if $S_r \neq \varnothing$ then we have $S_x \neq \varnothing$. Hence $S_x \neq \varnothing$ and $S_r \neq \varnothing$. So that, by (57) and (58), $S_x \in \mathrm{rel}(v_x, sh)^\star$ and $S_r \in \mathrm{rel}(v_r, sh)^\star$. Therefore, we have, by (56),

$$S \in \mathrm{bin}\bigl(\mathrm{rel}(v_x, sh)^\star, \mathrm{rel}(v_r, sh)^\star\bigr). \tag{59}$$

Combining (55) when $S_0 \neq \varnothing$ and (59) when $S_0 = \varnothing$ we obtain

$$S \in \overline{\mathrm{rel}}(v_{xr}, sh) \cup \mathrm{bin}\bigl(\mathrm{rel}(v_x, sh)^\star, \mathrm{rel}(v_r, sh)^\star\bigr)$$

and therefore, by (25),

$$S \in \mathrm{amgu}(sh, x \mapsto r).$$

As a consequence, since S was any set in sh', we have $sh' \subseteq \mathrm{amgu}(sh, x \mapsto r)$ and hence, by (43),

$$\alpha(\mu, U) \preceq_{ss} \bigl(\mathrm{amgu}(sh, x \mapsto r), U\bigr). \tag{60}$$

We now drop the assumption that $\alpha(\sigma, U) = (sh, U)$ and just assume the hypothesis of the theorem that $\alpha(\sigma, U) \preceq_{ss} (sh, U)$. Suppose $\alpha(\sigma, U) = (sh_1, U)$. Then $sh_1 \subseteq sh$. It follows from Definition 9 that amgu is monotonic on its first argument so that

$$\mathrm{amgu}(sh_1, x \mapsto r) \subseteq \mathrm{amgu}(sh, x \mapsto r).$$

Thus, by (60) (replacing sh by sh_1), we obtain the required result

$$\alpha(\mu, U) \preceq_{ss} \bigl(\mathrm{amgu}(sh, x \mapsto r), U\bigr). \quad \square$$

Proof of Theorem 6.
Let

$$\begin{aligned} sh_- &\stackrel{\mathrm{def}}{=} \overline{\mathrm{rel}}(v_{xr}, sh), \\ sh_{xr} &\stackrel{\mathrm{def}}{=} \mathrm{bin}\bigl(\mathrm{rel}(v_x, sh)^\star, \mathrm{rel}(v_r, sh)^\star\bigr). \end{aligned}$$

Then,

$$sh_{xr}^\star = sh_{xr}, \qquad \mathrm{bin}(sh_{xr}, sh_{xr}) = sh_{xr}.$$

Moreover,

$$\begin{aligned} \mathrm{rel}(v_x, sh_{xr}) &= sh_{xr}, & \mathrm{rel}(v_x, sh_-) &= \varnothing, \\ \mathrm{rel}(v_r, sh_{xr}) &= sh_{xr}, & \mathrm{rel}(v_r, sh_-) &= \varnothing, \\ \overline{\mathrm{rel}}(v_{xr}, sh_{xr}) &= \varnothing, & \overline{\mathrm{rel}}(v_{xr}, sh_-) &= sh_-. \end{aligned}$$

Hence, we have

$$\begin{aligned} \mathrm{rel}(v_x, sh_- \cup sh_{xr}) &= sh_{xr}, \\ \mathrm{rel}(v_r, sh_- \cup sh_{xr}) &= sh_{xr}, \\ \overline{\mathrm{rel}}(v_{xr}, sh_- \cup sh_{xr}) &= sh_-. \end{aligned}$$

Now, by (25),

$$\begin{aligned} &\mathrm{amgu}\bigl(\mathrm{amgu}(sh, x \mapsto r), x \mapsto r\bigr) \\ &\qquad = \overline{\mathrm{rel}}(v_{xr}, sh_- \cup sh_{xr}) \cup \mathrm{bin}\bigl(\mathrm{rel}(v_x, sh_- \cup sh_{xr})^\star, \mathrm{rel}(v_r, sh_- \cup sh_{xr})^\star\bigr) \\ &\qquad = sh_- \cup sh_{xr} \\ &\qquad = \mathrm{amgu}(sh, x \mapsto r). \quad \square \end{aligned}$$

For the proof of commutativity, we require the following auxiliary results.

Lemma 19
For each $V \in \wp_{\mathrm{f}}(\mathit{Vars})$ and $sh \in SH$ we have

$$\overline{\mathrm{rel}}(V, sh^\star) = \overline{\mathrm{rel}}(V, sh)^\star.$$

Proof
Let $S \in SG$. Then $S \in \overline{\mathrm{rel}}(V, sh^\star)$ means $S \in sh^\star$ and $S \cap V = \varnothing$. In other words, there exist $S_1, \ldots, S_n \in sh$ such that $S = \bigcup_{i=1}^n S_i$ and, for each $i = 1, \ldots, n$, we have $S_i \cap V = \varnothing$. This amounts to saying that there exist $S_1, \ldots, S_n \in \overline{\mathrm{rel}}(V, sh)$ such that $S = \bigcup_{i=1}^n S_i$, which is equivalent to $S \in \overline{\mathrm{rel}}(V, sh)^\star$. $\square$

The auxiliary function rel possesses a weaker property.

Lemma 20
For each $V \in \wp_{\mathrm{f}}(\mathit{Vars})$ and $\mathit{sh} \in \mathit{SH}$ we have

$$\mathrm{rel}(V, \mathit{sh}^\star) \supseteq \mathrm{rel}(V, \mathit{sh})^\star.$$

Proof
Let $S \in \mathit{SG}$. Then $S \in \mathrm{rel}(V, \mathit{sh})^\star$ means that there exist $S_1, \dots, S_n \in \mathit{sh}$ such that $S_i \cap V \neq \varnothing$, for each $i = 1, \dots, n$, and $S = \bigcup_{i=1}^n S_i$. Thus $S \cap V \neq \varnothing$ and $S \in \mathrm{rel}(V, \mathit{sh}^\star)$. Hence, $\mathrm{rel}(V, \mathit{sh}^\star) \supseteq \mathrm{rel}(V, \mathit{sh})^\star$. $\square$

Lemma 21
For each $V \in \wp_{\mathrm{f}}(\mathit{Vars})$, $\mathit{sh}_1, \mathit{sh}_2 \in \mathit{SH}$, and $S \in \wp_{\mathrm{f}}(\mathit{Vars})$ we have

$$\begin{aligned} &S \in \mathrm{rel}(V, \mathit{sh}_1 \cup \mathit{sh}_2)^\star \cup \{\varnothing\} \\ &\qquad \iff \exists S_1 \in \mathrm{rel}(V, \mathit{sh}_1)^\star \cup \{\varnothing\} \mathrel{.} \exists S_2 \in \mathrm{rel}(V, \mathit{sh}_2)^\star \cup \{\varnothing\} \mathrel{.} S = S_1 \cup S_2. \end{aligned}$$

Proof
If $S = \varnothing$ the statement is trivial.

Suppose $S \in \mathrm{rel}(V, \mathit{sh}_1 \cup \mathit{sh}_2)^\star$. Then, for some $n \in \mathbb{N}$, there exists n sets $R_1, \dots, R_n \in (\mathit{sh}_1 \cup \mathit{sh}_2)$ such that $R_i \cap V \neq \varnothing$ for each $i = 1, \dots, n$, and $S = \bigcup_{i=1}^n R_i$. Suppose $S_j = \bigcup\{\, R_i \in \mathit{sh}_j \mid 1 \leq i \leq n \,\}$ for $j = 1, 2$. Thus we have $S_1 \in \mathrm{rel}(V, \mathit{sh}_1)^\star \cup \{\varnothing\}$, $S_2 \in \mathrm{rel}(V, \mathit{sh}_2)^\star \cup \{\varnothing\}$, and $S = S_1 \cup S_2$.

Suppose

$$\exists S_1 \in \mathrm{rel}(V, \mathit{sh}_1)^\star \cup \{\varnothing\} \mathrel{.} \exists S_2 \in \mathrm{rel}(V, \mathit{sh}_2)^\star \cup \{\varnothing\} \mathrel{.} S = S_1 \cup S_2,$$

with S_1 and S_2 not both empty. Then, for some $m \geq 0$ and $n \geq 0$, there exist $R_1, \dots, R_m \in \mathrm{rel}(V, \mathit{sh}_1)$ and $T_1, \dots, T_n \in \mathrm{rel}(V, \mathit{sh}_2)$ such that $S_1 = \bigcup_{i=1}^m R_i$ and $S_2 = \bigcup_{i=1}^n T_i$. Then $R_1, \dots, R_m, T_1, \dots, T_n \in \mathrm{rel}(V, \mathit{sh}_1 \cup \mathit{sh}_2)$ and

$$S = \Bigl(\bigcup_{i=1}^m R_i\Bigr) \cup \Bigl(\bigcup_{i=1}^n T_i\Bigr).$$

Thus $S \in \mathrm{rel}(V, \mathit{sh}_1 \cup \mathit{sh}_2)^\star$. $\square$

Lemma 22
For each $V_1, V_2 \in \wp_{\mathrm{f}}(\mathit{Vars})$ and $\mathit{sh} \in \mathit{SH}$ we have

$$\mathrm{rel}\bigl(V_1, \overline{\mathrm{rel}}(V_2, \mathit{sh})\bigr) = \overline{\mathrm{rel}}\bigl(V_2, \mathrm{rel}(V_1, \mathit{sh})\bigr).$$

Proof
Suppose $S \in \mathit{SG}$. Then $S \in \mathrm{rel}\bigl(V_1, \overline{\mathrm{rel}}(V_2, \mathit{sh})\bigr)$ means $S \cap V_1 \neq \varnothing$ and $S \cap V_2 = \varnothing$. Similarly, $S \in \overline{\mathrm{rel}}\bigl(V_2, \mathrm{rel}(V_1, \mathit{sh})\bigr)$ means that $S \cap V_2 = \varnothing$ and $S \cap V_1 \neq \varnothing$. $\square$

Lemma 23
For each $\mathit{sh}_1, \mathit{sh}_2 \in \mathit{SH}$, we have

$$\mathrm{bin}(\mathit{sh}_1, \mathit{sh}_2)^\star = \mathrm{bin}(\mathit{sh}_1^\star, \mathit{sh}_2^\star).$$

Proof
Suppose $S \in SG$. Then $S \in \mathrm{bin}(sh_1, sh_2)^\star$ means that, for some $n \in \mathbb{N}$, there exist sets $R_1, \dots, R_n \in sh_1$ and $T_1, \dots, T_n \in sh_2$ such that $S = (R_1 \cup T_1) \cup \cdots \cup (R_n \cup T_n)$. Thus $S = (R_1 \cup \cdots \cup R_n) \cup (T_1 \cup \cdots \cup T_n)$. However $R_1 \cup \cdots \cup R_n \in sh_1^\star$ and $T_1 \cup \cdots \cup T_n \in sh_2^\star$. Thus $S \in \mathrm{bin}(sh_1^\star, sh_2^\star)$.

On the other hand, $S \in \mathrm{bin}(sh_1^\star, sh_2^\star)$ means that $S = R \cup T$ where, for some $k, l \in \mathbb{N}$, $R_1, \dots, R_k \in sh_1$, and $T_1, \dots, T_l \in sh_2$, we have $R = R_1 \cup \cdots \cup R_k$ and $T = T_1 \cup \cdots \cup T_l$. Let n be the maximum of $\{k, l\}$ and suppose that, for each $i, j \in \mathbb{N}$ where $k + 1 \leq i \leq n$ and $l + 1 \leq j \leq n$, we define $R_i \stackrel{\mathrm{def}}{=} R_k$ and $T_j \stackrel{\mathrm{def}}{=} T_l$. Then, $S = (R_1 \cup T_1) \cup \cdots \cup (R_n \cup T_n)$. However, for $1 \leq i \leq n$, $R_i \cup T_i \in \mathrm{bin}(sh_1, sh_2)$. Thus $S \in \mathrm{bin}(sh_1, sh_2)^\star$. $\square$

Proof of Theorem 7.
We let R, S, T, and U (possibly subscripted) denote elements of $sh^\star$. The subscripts reflect certain properties of the sets. In particular, subscripts x, r, xr, y, t, yt indicate sets of variables that definitely have a variable in common with the subscripted set. For example, R_x is a set in $sh^\star$ that has a common element with v_x and T_{xt} is a set in $sh^\star$ that has common elements with v_x and v_t. In contrast, the subscript '$-$' indicates that the subscripted set does not share with one of the sets v_{xr} or v_{yt}. Of course, in the proof, each set is formally defined as needed.

Suppose that

$$S \in \mathrm{amgu}\bigl(\mathrm{amgu}(sh, x \mapsto r), y \mapsto t\bigr).$$

We will show that

$$S \in \mathrm{amgu}\bigl(\mathrm{amgu}(sh, y \mapsto t), x \mapsto r\bigr).$$

The converse then holds by simply exchanging x and y, and r and t.

There are two cases due to the two components of the definition of amgu in Eq. (25).

Case 1. Assume

$$S \in \overline{\mathrm{rel}}\bigl(v_{yt}, \mathrm{amgu}(sh, x \mapsto r)\bigr).$$

Then $S \in \mathrm{amgu}(sh, x \mapsto r)$ and $S \cap v_{yt} = \varnothing$. Again there are two possibilities.

Subcase 1a. Suppose first that

$$S \in \overline{\mathrm{rel}}(v_{xr}, sh).$$

Thus $S \in sh$, and, since in this case we have $S \cap v_{yt} = \varnothing$,

$$S \in \overline{\mathrm{rel}}(v_{yt}, sh).$$

The alternative definition of amgu, (25), implies $\overline{\mathrm{rel}}(v_{yt}, sh) \subseteq \mathrm{amgu}(sh, y \mapsto t)$ and thus we have also

$$S \in \mathrm{amgu}(sh, y \mapsto t).$$

Now, since the hypothesis of this subcase implies $S \cap v_{xr} = \varnothing$, we obtain

$$S \in \overline{\mathrm{rel}}\bigl(v_{xr}, \mathrm{amgu}(sh, y \mapsto t)\bigr).$$

Hence, again by (25), we can conclude that

$$S \in \mathrm{amgu}\bigl(\mathrm{amgu}(sh, y \mapsto t), x \mapsto r\bigr).$$

Subcase 1b. Suppose now that

$$S \in \mathrm{bin}\bigl(\mathrm{rel}(v_x, sh)^\star, \mathrm{rel}(v_r, sh)^\star\bigr).$$

Then, there exist $S_x, S_r \in SG$ such that $S = S_x \cup S_r$, where

$$S_x \in \mathrm{rel}(v_x, sh)^\star, \qquad S_r \in \mathrm{rel}(v_r, sh)^\star.$$

By the hypothesis for this case we have $S \cap v_{yt} = \varnothing$ and thus $S_x \cap v_{yt} = \varnothing$ and $S_r \cap v_{yt} = \varnothing$. This allows to state that

$$S_x \in \overline{\mathrm{rel}}\bigl(v_{yt}, \mathrm{rel}(v_x, sh)^\star\bigr), \qquad S_r \in \overline{\mathrm{rel}}\bigl(v_{yt}, \mathrm{rel}(v_r, sh)^\star\bigr),$$

and hence, by Lemma 19,

$$S_x \in \overline{\mathrm{rel}}\bigl(v_{yt}, \mathrm{rel}(v_x, sh)\bigr)^\star, \qquad S_r \in \overline{\mathrm{rel}}\bigl(v_{yt}, \mathrm{rel}(v_r, sh)\bigr)^\star,$$

Thus, by Lemma 22,

$$S_x \in \mathrm{rel}\bigl(v_x, \overline{\mathrm{rel}}(v_{yt}, sh)\bigr)^\star, \qquad S_r \in \mathrm{rel}\bigl(v_r, \overline{\mathrm{rel}}(v_{yt}, sh)\bigr)^\star,$$

so that, by (23), (24), and (25),

$$S_x \in \mathrm{rel}\bigl(v_x, \mathrm{amgu}(sh, y \mapsto t)\bigr)^\star, \qquad S_r \in \mathrm{rel}\bigl(v_r, \mathrm{amgu}(sh, y \mapsto t)\bigr)^\star.$$

Therefore,

$$S_x \cup S_r \in \mathrm{bin}\Bigl(\mathrm{rel}\bigl(v_x, \mathrm{amgu}(sh, y \mapsto t)\bigr)^\star, \mathrm{rel}\bigl(v_r, \mathrm{amgu}(sh, y \mapsto t)\bigr)^\star\Bigr)$$

so that, as $S_x \cup S_r = S$, it follows from (25) that

$$S \in \mathrm{amgu}\bigl(\mathrm{amgu}(sh, y \mapsto t), x \mapsto r\bigr).$$

Case 2. Assume

$$S \in \mathrm{bin}\Bigl(\mathrm{rel}\bigl(v_y, \mathrm{amgu}(sh, x \mapsto r)\bigr)^\star, \mathrm{rel}\bigl(v_t, \mathrm{amgu}(sh, x \mapsto r)\bigr)^\star\Bigr).$$

Then there exist $S_y, S_t \in SG$ such that

$$S = S_y \cup S_t \tag{61}$$

where

$$\begin{aligned} S_y &\in \mathrm{rel}\bigl(v_y, \mathrm{amgu}(sh, x \mapsto r)\bigr)^\star, \\ S_t &\in \mathrm{rel}\bigl(v_t, \mathrm{amgu}(sh, x \mapsto r)\bigr)^\star. \end{aligned} \tag{62}$$

Then, by Lemma 20,

$$S_y \cap v_y \neq \varnothing, \qquad S_t \cap v_t \neq \varnothing. \tag{63}$$

By (25) and Lemma 21, there exist R_-, R_{xr}, T_-, and T_{xr} such that

$$S_y = R_- \cup R_{xr}, \qquad S_t = T_- \cup T_{xr} \tag{64}$$

where

$$\begin{aligned}
R_- &\in \operatorname{rel}\bigl(v_y, \overline{\operatorname{rel}}(v_{xr}, sh)\bigr)^\star \cup \{\varnothing\},\\
R_{xr} &\in \operatorname{rel}\Bigl(v_y, \operatorname{bin}\bigl(\operatorname{rel}(v_x, sh)^\star, \operatorname{rel}(v_r, sh)^\star\bigr)\Bigr)^\star \cup \{\varnothing\},\\
T_- &\in \operatorname{rel}\bigl(v_t, \overline{\operatorname{rel}}(v_{xr}, sh)\bigr)^\star \cup \{\varnothing\},\\
T_{xr} &\in \operatorname{rel}\Bigl(v_t, \operatorname{bin}\bigl(\operatorname{rel}(v_x, sh)^\star, \operatorname{rel}(v_r, sh)^\star\bigr)\Bigr)^\star \cup \{\varnothing\}.
\end{aligned} \tag{65}$$

Then, by Lemmas 22 and 19,

$$\begin{aligned}
R_- &\in \overline{\operatorname{rel}}\bigl(v_{xr}, \operatorname{rel}(v_y, sh)^\star\bigr) \cup \{\varnothing\},\\
T_- &\in \overline{\operatorname{rel}}\bigl(v_{xr}, \operatorname{rel}(v_t, sh)^\star\bigr) \cup \{\varnothing\}.
\end{aligned} \tag{66}$$

Also, using Lemmas 20, 23, and then the idempotence of $(\cdot)^\star$,

$$\begin{aligned}
R_{xr} &\in \operatorname{rel}\Bigl(v_y, \operatorname{bin}\bigl(\operatorname{rel}(v_x, sh)^\star, \operatorname{rel}(v_r, sh)^\star\bigr)\Bigr) \cup \{\varnothing\},\\
T_{xr} &\in \operatorname{rel}\Bigl(v_t, \operatorname{bin}\bigl(\operatorname{rel}(v_x, sh)^\star, \operatorname{rel}(v_r, sh)^\star\bigr)\Bigr) \cup \{\varnothing\}.
\end{aligned} \tag{67}$$

Subcase 2a. Suppose $R_{xr} = T_{xr} = \varnothing$. Then, by (64),

$$S_y = R_-, \quad S_t = T_-. \tag{68}$$

By (63), $R_-, T_- \neq \varnothing$ and hence, using (66),

$$R_- \cup T_- \in \operatorname{bin}\bigl(\operatorname{rel}(v_y, sh)^\star, \operatorname{rel}(v_t, sh)^\star\bigr),$$

so that, by (25),

$$R_- \cup T_- \in \operatorname{amgu}(sh, y \mapsto t).$$

Also, it follows from (66) that $R_- \cap v_{xr} = \varnothing$ and $T_- \cap v_{xr} = \varnothing$, so that

$$R_- \cup T_- \in \overline{\operatorname{rel}}\bigl(v_{xr}, \operatorname{amgu}(sh, y \mapsto t)\bigr).$$

However, by (61) and (68), $S = R_- \cup T_-$ so that, by (25),

$$S \in \operatorname{amgu}\bigl(\operatorname{amgu}(sh, y \mapsto t), x \mapsto r\bigr).$$

Subcase 2b. Suppose $R_{xr} \cup T_{xr} \neq \varnothing$. Then, by (67),

$$(R_{xr} \cup T_{xr}) \cap v_{yt} \neq \varnothing. \tag{69}$$

The proof of this subcase is in two parts. In the first part we divide R_{xr} and T_{xr} into a number of subsets. In the second part, these subsets will be reassembled so as to prove the required result.

First, by (67), there exist $R_x, R_r, T_x, T_r \in \wp_{\mathrm{f}}(\mathit{Vars})$ such that

$$R_{xr} = R_x \cup R_r, \qquad T_{xr} = T_x \cup T_r, \tag{70}$$

where either $R_x = R_r = \varnothing$ or

$$R_x \in \mathrm{rel}(v_x, sh)^\star, \qquad R_r \in \mathrm{rel}(v_r, sh)^\star,$$

and either $T_x = T_r = \varnothing$ or

$$T_x \in \mathrm{rel}(v_x, sh)^\star, \qquad T_r \in \mathrm{rel}(v_r, sh)^\star.$$

Thus, if either $R_x \cup T_x = \varnothing$ or $R_r \cup T_r = \varnothing$, it follows that

$$R_{xr} \cup T_{xr} = (R_x \cup R_r) \cup (T_x \cup T_r) = \varnothing.$$

However, by (69), $R_{xr} \cup T_{xr} \neq \varnothing$, so that we have

$$R_x \cup T_x \neq \varnothing, \qquad R_r \cup T_r \neq \varnothing. \tag{71}$$

We now subdivide the sets R_x, T_x, R_r, and T_r further. First note that

$$\begin{aligned}
sh &= \overline{\mathrm{rel}}(v_{yt}, sh) \cup \mathrm{rel}(v_y, sh) \cup \overline{\mathrm{rel}}\bigl(v_y, \mathrm{rel}(v_t, sh)\bigr),\\
sh &= \overline{\mathrm{rel}}(v_{yt}, sh) \cup \overline{\mathrm{rel}}\bigl(v_t, \mathrm{rel}(v_y, sh)\bigr) \cup \mathrm{rel}(v_t, sh).
\end{aligned}$$

Hence, by Lemma 21, sets R_{x-}, R_{xy}, R_{xt}, R_{r-}, R_{ry}, R_{rt}, T_{x-}, T_{xy}, T_{xt}, T_{r-}, T_{ry}, $T_{rt} \in \wp_{\mathrm{f}}(\mathit{Vars})$ exist such that

$$\begin{aligned}
R_x &= R_{x-} \cup R_{xy} \cup R_{xt}, \qquad & T_x &= T_{x-} \cup T_{xy} \cup T_{xt},\\
R_r &= R_{r-} \cup R_{ry} \cup R_{rt}, \qquad & T_r &= T_{r-} \cup T_{ry} \cup T_{rt},
\end{aligned} \tag{72}$$

where

$$\begin{aligned}
R_{x-}, T_{x-} &\in \mathrm{rel}\bigl(v_x, \overline{\mathrm{rel}}(v_{yt}, sh)\bigr)^\star \cup \{\varnothing\},\\
R_{r-}, T_{r-} &\in \mathrm{rel}\bigl(v_r, \overline{\mathrm{rel}}(v_{yt}, sh)\bigr)^\star \cup \{\varnothing\},
\end{aligned} \tag{73}$$

and

$$\begin{aligned}
R_{xy}, T_{xy} &\in \mathrm{rel}\bigl(v_x, \mathrm{rel}(v_y, sh)\bigr)^\star \cup \{\varnothing\},\\
R_{ry}, T_{ry} &\in \mathrm{rel}\bigl(v_r, \mathrm{rel}(v_y, sh)\bigr)^\star \cup \{\varnothing\},\\
R_{xt}, T_{xt} &\in \mathrm{rel}\bigl(v_x, \mathrm{rel}(v_t, sh)\bigr)^\star \cup \{\varnothing\},\\
R_{rt}, T_{rt} &\in \mathrm{rel}\bigl(v_r, \mathrm{rel}(v_t, sh)\bigr)^\star \cup \{\varnothing\},
\end{aligned} \tag{74}$$

and also

$$\begin{aligned}
(R_x \setminus R_{xy}) \cap v_y &= \varnothing, \qquad & (T_x \setminus T_{xt}) \cap v_t &= \varnothing,\\
(R_r \setminus R_{ry}) \cap v_y &= \varnothing, \qquad & (T_r \setminus T_{rt}) \cap v_t &= \varnothing.
\end{aligned} \tag{75}$$

We note a few simple but useful consequences of these definitions. First, it follows from (73) using (23), (24), and (25), that

$$\begin{aligned} R_{x-}, T_{x-} &\in \mathrm{rel}\bigl(v_x, \mathrm{amgu}(sh, y \mapsto t)\bigr)^\star \cup \{\varnothing\}, \\ R_{r-}, T_{r-} &\in \mathrm{rel}\bigl(v_r, \mathrm{amgu}(sh, y \mapsto t)\bigr)^\star \cup \{\varnothing\}. \end{aligned} \tag{76}$$

Secondly, using (73) with Lemma 20, we have

$$R_{x-}, T_{x-}, R_{r-}, T_{r-} \in \overline{\mathrm{rel}}(v_{yt}, sh)^\star \cup \{\varnothing\}, \tag{77}$$

and then, using this with (69), (70), and (72), it follows that

$$R_{xy} \cup T_{xy} \cup R_{ry} \cup T_{ry} \cup R_{xt} \cup T_{xt} \cup R_{rt} \cup T_{rt} \neq \varnothing. \tag{78}$$

In the second part of the proof for this subcase, the component subsets of S are reassembled in an order that proves the required result. First, let

$$\begin{aligned} U_y &\stackrel{\mathrm{def}}{=} R_- \cup R_{xy} \cup R_{ry} \cup T_{xy} \cup T_{ry}, \\ U_t &\stackrel{\mathrm{def}}{=} T_- \cup R_{xt} \cup R_{rt} \cup T_{xt} \cup T_{rt}, \end{aligned} \tag{79}$$

and

$$U \stackrel{\mathrm{def}}{=} U_y \cup U_t. \tag{80}$$

By relations (65) and (74) (with Lemma 20), each component set in the definition of U_y is in $\mathrm{rel}(v_y, sh)^\star \cup \{\varnothing\}$ and each component set in the definition of U_t is in $\mathrm{rel}(v_t, sh)^\star \cup \{\varnothing\}$. Thus, by the definition of $(\cdot)^\star$,

$$\begin{aligned} U_y &\in \mathrm{rel}(v_y, sh)^\star \cup \{\varnothing\}, \\ U_t &\in \mathrm{rel}(v_t, sh)^\star \cup \{\varnothing\}. \end{aligned} \tag{81}$$

By (70) and (75) we have

$$\bigl(R_{xr} \setminus (R_{xy} \cup R_{ry})\bigr) \cap v_y = \varnothing$$

and hence, by (64), we have also that

$$\bigl(S_y \setminus (R_{xy} \cup R_{ry} \cup R_-)\bigr) \cap v_y = \varnothing.$$

By (63), $S_y \cap v_y \neq \varnothing$. Thus, $R_{xy} \cup R_{ry} \cup R_- \neq \varnothing$ and, as a consequence of (79), $U_y \neq \varnothing$. For similar reasons, $U_t \neq \varnothing$. Hence, by (80),

$$U \in \mathrm{bin}\bigl(\mathrm{rel}(v_y, sh)^\star, \mathrm{rel}(v_t, sh)^\star\bigr),$$

and therefore, using (25), it follows that

$$U \in \mathrm{amgu}(sh, y \mapsto t). \tag{82}$$

Now, by (78), at least one of the following two inequalities holds:

$$\begin{aligned} R_{xy} \cup T_{xy} \cup R_{xt} \cup T_{xt} &\neq \varnothing, \\ R_{ry} \cup T_{ry} \cup R_{rt} \cup T_{rt} &\neq \varnothing. \end{aligned} \tag{83}$$

$R_{xy} \cup T_{xy} \cup R_{xt} \cup T_{xt} = \varnothing$ and $R_{ry} \cup T_{ry} \cup R_{rt} \cup T_{rt} \neq \varnothing$. Then, using (71) and (72) with the first of these,

$$R_{x-} \cup T_{x-} \neq \varnothing.$$

Also, using (74) with the second, we have $(R_{ry} \cup R_{rt} \cup T_{ry} \cup T_{rt}) \cap v_r \neq \varnothing$ and therefore it follows from (79) and (80), that

$$U \cap v_r \neq \varnothing.$$

Hence, by (76) and (82),

$$\begin{aligned} R_{x-} \cup T_{x-} &\in \operatorname{rel}\bigl(v_x, \operatorname{amgu}(sh, y \mapsto t)\bigr)^{\star}, \\ U \cup R_{r-} \cup T_{r-} &\in \operatorname{rel}\bigl(v_r, \operatorname{amgu}(sh, y \mapsto t)\bigr)^{\star}. \end{aligned} \tag{84}$$

Similarly, assuming $R_{xy} \cup T_{xy} \cup R_{xt} \cup T_{xt} \neq \varnothing$ and $R_{ry} \cup T_{ry} \cup R_{rt} \cup T_{rt} = \varnothing$ it follows that

$$\begin{aligned} R_{r-} \cup T_{r-} &\in \operatorname{rel}\bigl(v_r, \operatorname{amgu}(sh, y \mapsto t)\bigr)^{\star}, \\ R_{x-} \cup T_{x-} \cup U &\in \operatorname{rel}\bigl(v_x, \operatorname{amgu}(sh, y \mapsto t)\bigr)^{\star}. \end{aligned} \tag{85}$$

Finally, assuming $R_{xy} \cup T_{xy} \cup R_{xt} \cup T_{xt} \neq \varnothing$ and $R_{ry} \cup T_{ry} \cup R_{rt} \cup T_{rt} \neq \varnothing$ it follows from (74) that $U \cap v_x \neq \varnothing$ and $U \cap v_r \neq \varnothing$, and hence

$$\begin{aligned} R_{x-} \cup T_{x-} \cup U &\in \operatorname{rel}\bigl(v_x, \operatorname{amgu}(sh, y \mapsto t)\bigr)^{\star}, \\ U \cup R_{r-} \cup T_{r-} &\in \operatorname{rel}\bigl(v_r, \operatorname{amgu}(sh, y \mapsto t)\bigr)^{\star}. \end{aligned} \tag{86}$$

Thus, as one of the inequalities in (83) holds, one of (84), (85) or (86) holds so that

$$\begin{aligned} &R_{x-} \cup T_{x-} \cup U \cup R_{r-} \cup T_{r-} \\ &\qquad \in \operatorname{bin}\Bigl(\operatorname{rel}\bigl(v_x, \operatorname{amgu}(sh, y \mapsto t)\bigr)^{\star}, \operatorname{rel}\bigl(v_r, \operatorname{amgu}(sh, y \mapsto t)\bigr)^{\star}\Bigr). \end{aligned}$$

However, since

$$S = R_{x-} \cup T_{x-} \cup U \cup R_{r-} \cup T_{r-},$$

we have

$$S \in \operatorname{bin}\Bigl(\operatorname{rel}\bigl(v_x, \operatorname{amgu}(sh, y \mapsto t)\bigr)^{\star}, \operatorname{rel}\bigl(v_r, \operatorname{amgu}(sh, y \mapsto t)\bigr)^{\star}\Bigr).$$

Hence, by (25),

$$S \in \operatorname{amgu}\bigl(\operatorname{amgu}(sh, y \mapsto t), x \mapsto r\bigr). \quad \square$$

6.5 Proofs of Results for Sharing Domains

We prove all the results in this section by induction on the cardinality of a substitution ν. For each result, the proof is obvious if ν is empty or does not unify. Thus, in the following proofs, we assume that ν unifies and is non-empty. We suppose that $(x \mapsto r) \in \nu$ and let $\nu' \stackrel{\mathrm{def}}{=} \nu \setminus \{x \mapsto r\}$.

Proof of Lemma 18.
We have

$$\begin{aligned}
&\mathrm{aunify}\Bigl(\mathrm{Amgu}\bigl((sh,U), y \mapsto t\bigr), \nu\Bigr) \\
&\qquad = \mathrm{aunify}\biggl(\mathrm{Amgu}\Bigl(\mathrm{Amgu}\bigl((sh,U), y \mapsto t\bigr), x \mapsto r\Bigr), \nu'\biggr) && \text{[Def. 11]} \\
&\qquad = \mathrm{aunify}\biggl(\mathrm{Amgu}\Bigl(\mathrm{Amgu}\bigl((sh,U), x \mapsto r\bigr), y \mapsto t\Bigr), \nu'\biggr) && \text{[Cor. 3]} \\
&\qquad = \mathrm{Amgu}\biggl(\mathrm{aunify}\Bigl(\mathrm{Amgu}\bigl((sh,U), x \mapsto r\bigr), \nu'\Bigr), y \mapsto t\biggr) && \text{[induction]} \\
&\qquad = \mathrm{Amgu}\Bigl(\mathrm{aunify}\bigl((sh,U), \nu\bigr), y \mapsto t\Bigr) && \text{[Def. 11].} \quad \square
\end{aligned}$$

Proof of Theorem 8.
Let μ' be a most general solution for $(\nu' \cup \sigma)$. Then

$$\begin{aligned}
&\alpha(\sigma, U) \preceq_{ss} (sh, U) \\
&\quad \Longrightarrow \alpha\bigl(\mu', U \cup \mathit{vars}(\nu')\bigr) \\
&\qquad\qquad \preceq_{ss} \mathrm{aunify}\bigl((sh,U), \nu'\bigr) && \text{[induction]} \\
&\quad \Longrightarrow \alpha\bigl(\mu, U \cup \mathit{vars}(\nu)\bigr) \\
&\qquad\qquad \preceq_{ss} \mathrm{Amgu}\Bigl(\mathrm{aunify}\bigl((sh,U), \nu'\bigr), x \mapsto r\Bigr) && \text{[Cor. 1]} \\
&\quad \Longrightarrow \alpha\bigl(\mu, U \cup \mathit{vars}(\nu)\bigr) \\
&\qquad\qquad \preceq_{ss} \mathrm{aunify}\Bigl(\mathrm{Amgu}\bigl((sh,U), x \mapsto r\bigr), \nu'\Bigr) && \text{[Lem. 18]} \\
&\quad \Longrightarrow \alpha\bigl(\mu, U \cup \mathit{vars}(\nu)\bigr) \\
&\qquad\qquad \preceq_{ss} \mathrm{aunify}\bigl((sh,U), \nu\bigr) && \text{[Def. 11].} \quad \square
\end{aligned}$$

Proof of Theorem 9.
We have

$$\begin{aligned}
&\mathrm{aunify}\Bigl(\mathrm{aunify}\bigl((sh,U), \nu\bigr), \nu\Bigr) \\
&\quad = \mathrm{aunify}\biggl(\mathrm{Amgu}\Bigl(\mathrm{aunify}\bigl(\mathrm{Amgu}((sh,U), x \mapsto r), \nu'\bigr), x \mapsto r\Bigr), \nu'\biggr) && \text{[Def. 11]} \\
&\quad = \mathrm{aunify}\biggl(\mathrm{aunify}\Bigl(\mathrm{Amgu}\bigl(\mathrm{Amgu}((sh,U), x \mapsto r), x \mapsto r\bigr), \nu'\Bigr), \nu'\biggr) && \text{[Lem. 18]} \\
&\quad = \mathrm{aunify}\Bigl(\mathrm{Amgu}\bigl(\mathrm{Amgu}((sh,U), x \mapsto r), x \mapsto r\bigr), \nu'\Bigr) && \text{[induction]} \\
&\quad = \mathrm{aunify}\Bigl(\mathrm{Amgu}\bigl((sh,U), x \mapsto r\bigr), \nu'\Bigr) && \text{[Cor. 2]} \\
&\quad = \mathrm{aunify}\bigl((sh,U), \nu\bigr) && \text{[Def. 11].} \quad \square
\end{aligned}$$

Proof of Theorem 10.
In this theorem the induction is on the set of equations ν_1. The comments at the start of this section apply therefore to ν_1 instead of ν and thus we let $\nu_1' \stackrel{\mathrm{def}}{=} \nu_1 \setminus \{x \mapsto$

$r\}$ so that we have

$$
\begin{aligned}
&\operatorname{aunify}\Bigl(\operatorname{aunify}\bigl((sh,U),\nu_1\bigr),\nu_2\Bigr)\\
&\qquad= \operatorname{aunify}\biggl(\operatorname{aunify}\Bigl(\operatorname{Amgu}\bigl((sh,U),x\mapsto r\bigr),\nu_1'\Bigr),\nu_2\biggr) && \text{[Def. 11]}\\
&\qquad= \operatorname{aunify}\biggl(\operatorname{aunify}\Bigl(\operatorname{Amgu}\bigl((sh,U),x\mapsto r\bigr),\nu_2\Bigr),\nu_1'\biggr) && \text{[induction]}\\
&\qquad= \operatorname{aunify}\biggl(\operatorname{Amgu}\Bigl(\operatorname{aunify}\bigl((sh,U),\nu_2\bigr),x\mapsto r\Bigr),\nu_1'\biggr) && \text{[Lem. 18]}\\
&\qquad= \operatorname{aunify}\Bigl(\operatorname{aunify}\bigl((sh,U),\nu_2\bigr),\nu_1\Bigr) && \text{[Def. 11].} \quad \square
\end{aligned}
$$

7 Conclusion

The Sharing domain, which was defined in (Jacobs & Langen, 1989; Langen, 1990), is considered to be the principle abstract domain for sharing analysis of logic programs in both practical work and theoretical study. For many years, this domain was accepted and implemented as it was. However, in (Bagnara *et al.*, 1997), we proved that Sharing is, in fact, redundant for pair-sharing and we identified the weakest abstraction of Sharing that can capture pair-sharing with the same degree of precision. One notable advantage of this abstraction is that the costly star-union operator is no longer necessary. The question of whether the abstract operations for Sharing were complete or optimal was studied by Cortesi and Filé (Cortesi & Filé, 1999). Here it is proved that although the '$\sqcup$' and projection operations are complete (and hence, optimal), aunify is optimal but not complete. The problem of scalability of Sharing, still retaining as much precision as possible, was tackled in (Zaffanella *et al.*, 1999b), where a family of widenings is presented that allow the desired goal to be achieved. In (Zaffanella *et al.*, 1999a), the decomposition of Sharing and its non-redundant counterpart via complementation is studied. This shows the close relationship between these domains and *PS* (the usual domain for pair-sharing) and *Def* (the domain of definite Boolean functions). Many sharing analysis techniques and/or enhancements have been advocated to have potential for improving the precision of the sharing information over and above that obtainable using the classical combination of Sharing with the usual domains for linearity and freeness. Moreover, these enhancements had been circulating for years without an adequate supporting experimental evaluation. Thus we investigated these techniques to see if and by how much they could improve precision. Using the China analyzer (Bagnara, 1997) for the experimental part of the work, we discovered that, apart from the enhancement that upgrades Sharing with structural information, these techniques had little impact on precision (Bagnara *et al.*, 2000a).

In this paper, we have defined a new abstraction function mapping a set of substitutions in rational solved form into their corresponding sharing abstraction. The new function is a generalisation of the classical abstraction function of (Jacobs & Langen, 1989), which was defined for idempotent substitutions only. Using our new abstraction function, we have proved the soundness of the classical abstract unification operator aunify. Other contributions of our work are the formal proofs of the commutativity and idempotence of the aunify operator on the Sharing domain. Even if commutativity was a known property, the corresponding proof in (Langen, 1990) was not satisfactory. As far as idempotence is concerned, our result differs from that given in (Langen, 1990), which was based on a composite abstract unification operator performing also the renaming of variables. It is our opinion that our main result, the soundness of the aunify operator, is really valuable as it allows for the safe application of sharing analysis based on Sharing to any constraint logic language supporting syntactic term structures, based on either finite trees or rational trees. This happens because our result does not rely on the presence (or even the absence) of the occur-check in the concrete unification procedure implemented by the analysed language. Furthermore, as the groundness domain *Def* is included in Sharing, our main soundness result also shows that *Def* is sound for non-idempotent substitutions.

From a technical point of view, we have introduced a new class of concrete substitutions based on the notion of *variable-idempotence*, generalizing the classical concept of idempotence. We have shown that any substitution is equivalent to a variable-idempotent one, providing a finite sequence of transformations for its construction. This result assumes an arbitrary equality theory and is therefore applicable to the study of any abstract property which is preserved by logical equivalence. Our application of this idea to the study of the soundness of abstract unification for Sharing has shown that it is particularly suitable for data-flow analyzers where the corresponding abstraction function only depends on the set of variables occurring in a term. However, we believe that this concept can be usefully exploited in a more general context. Possible applications include the proofs of optimality and completeness of abstract operators with respect to the corresponding concrete operators defined on a domain of substitutions in rational solved form.

References

Bagnara, R. 1997 (Mar.). *Data-flow analysis for constraint logic-based languages.* Ph.D. thesis, Dipartimento di Informatica, Università di Pisa, Corso Italia 40, I-56125 Pisa, Italy. Printed as Report TD-1/97.

Bagnara, R., Hill, P. M., & Zaffanella, E. (1997). Set-sharing is redundant for pair-sharing. *Pages 53–67 of:* Van Hentenryck, P. (ed), *Static analysis: Proceedings of the 4th international symposium.* Lecture Notes in Computer Science, vol. 1302. Paris, France: Springer-Verlag, Berlin.

Bagnara, R., Zaffanella, E., & Hill, P. M. (2000a). Enhanced sharing analysis techniques: A comprehensive evaluation. Gabbrielli, M., & Pfenning, F. (eds), *Proceedings of the acm sigplan 2nd international conference on principles and practice of declarative programming.* Lecture Notes in Computer Science. Montreal, Canada: Springer-Verlag, Berlin. To appear.

Bagnara, R., Hill, P. M., & Zaffanella, E. (2000b). Set-sharing is redundant for pair-sharing. *Theoretical computer science.* To appear.

Bruynooghe, M., & Codish, M. (1993). Freeness, sharing, linearity and correctness — All at once. *Pages 153–164 of:* Cousot, P., Falaschi, M., Filé, G., & Rauzy, A. (eds), *Static analysis, proceedings of the third international workshop.* Lecture Notes in Computer Science, vol. 724. Padova, Italy: Springer-Verlag, Berlin. An extended version is available as Technical Report CW 179, Department of Computer Science, K.U. Leuven, September 1993.

Clark, K. L. (1978). Negation as failure. *Pages 293–322 of:* Gallaire, H., & Minker, J. (eds), *Logic and databases*. Toulouse, France: Plenum Press.

Codish, M., Dams, D., Filé, G., & Bruynooghe, M. (1993). Freeness analysis for logic programs — and correctness? *Pages 116–131 of:* Warren, D. S. (ed), *Logic programming: Proceedings of the tenth international conference on logic programming*. MIT Press Series in Logic Programming. Budapest, Hungary: The MIT Press. An extended version is available as Technical Report CW 161, Department of Computer Science, K.U. Leuven, December 1992.

Colmerauer, A. (1982). Prolog and infinite trees. *Pages 231–251 of:* Clark, K. L., & Tärnlund, S. Å. (eds), *Logic programming, apic studies in data processing*, vol. 16. Academic Press, New York.

Colmerauer, A. (1984). Equations and inequations on finite and infinite trees. *Pages 85–99 of: Proceedings of the international conference on fifth generation computer systems (fgcs'84)*. Tokyo, Japan: ICOT.

Cortesi, A., & Filé, G. (1999). Sharing is optimal. *Journal of logic programming*, **38**(3), 371–386.

Hill, P. M., Bagnara, R., & Zaffanella, E. (1998). The correctness of set-sharing. *Pages 99–114 of:* Levi, G. (ed), *Static analysis: Proceedings of the 5th international symposium*. Lecture Notes in Computer Science, vol. 1503. Pisa, Italy: Springer-Verlag, Berlin.

ISO/IEC. (1995). *ISO/IEC 13211-1: 1995 Information technology — Programming languages — Prolog — Part 1: General core*. International Standard Organization.

Jacobs, D., & Langen, A. (1989). Accurate and efficient approximation of variable aliasing in logic programs. *Pages 154–165 of:* Lusk, E. L., & Overbeek, R. A. (eds), *Logic programming: Proceedings of the north american conference*. MIT Press Series in Logic Programming. Cleveland, Ohio, USA: The MIT Press.

Jacobs, D., & Langen, A. (1992). Static analysis of logic programs for independent AND parallelism. *Journal of logic programming*, **13**(2&3), 291–314.

Jaffar, J., Lassez, J-L., & Maher, M. J. (1987). Prolog-II as an instance of the logic programming scheme. *Pages 275–299 of:* Wirsing, M. (ed), *Formal descriptions of programming concepts III*. North-Holland.

Keisu, T. 1994 (May). *Tree constraints*. Ph.D. thesis, The Royal Institute of Technology, Stockholm, Sweden. Also available in the SICS Dissertation Series: SICS/D–16–SE.

King, A. (1994). A synergistic analysis for sharing and groundness which traces linearity. *Pages 363–378 of:* Sannella, D. (ed), *Proceedings of the fifth european symposium on programming*. Lecture Notes in Computer Science, vol. 788. Edinburgh, UK: Springer-Verlag, Berlin.

King, A. (2000). Pair-sharing over rational trees. *Journal of logic programming*.

King, A., & Soper, P. (1994). Depth-k sharing and freeness. *Pages 553–568 of:* Van Hentenryck, P. (ed), *Logic programming: Proceedings of the eleventh international conference on logic programming*. MIT Press Series in Logic Programming. Santa Margherita Ligure, Italy: The MIT Press.

Langen, A. (1990). *Advanced techniques for approximating variable aliasing in logic programs*. Ph.D. thesis, Computer Science Department, University of Southern California. Printed as Report TR 91-05.

Maher, M. J. (1988). Complete axiomatizations of the algebras of finite, rational and infinite trees. *Pages 348–357 of: Proceedings, third annual symposium on logic in computer science*. IEEE Computer Society, Edinburgh, Scotland.

Martelli, A., & Montanari, U. (1982). An efficient unification algorithm. *Acm transactions on programming languages and systems*, **4**(2), 258–282.

Muthukumar, K., & Hermenegildo, M. (1992). Compile-time derivation of variable dependency using abstract interpretation. *Journal of logic programming*, **13**(2&3), 315–347.

Robinson, J. A. (1965). A machine-oriented logic based on the resolution principle. *Journal of the acm*, **12**(1), 23–41.

Zaffanella, E., Hill, P. M., & Bagnara, R. (1999a). Decomposing non-redundant sharing by complementation. *Pages 69–84 of:* Cortesi, A., & Filé, G. (eds), *Static analysis: Proceedings of the 6th international symposium*. Lecture Notes in Computer Science, vol. 1694. Venice, Italy: Springer-Verlag, Berlin.

Zaffanella, E., Bagnara, R., & Hill, P. M. (1999b). Widening Sharing. *Pages 414–431 of:* Nadathur, G. (ed), *Principles and practice of declarative programming*. Lecture Notes in Computer Science, vol. 1702. Paris, France: Springer-Verlag, Berlin.