University of Leeds
SCHOOL OF COMPUTING
RESEARCH REPORT SERIES
Report 2003.08

On the Analysis of Set-Sharing, Freeness and Linearity for Finite and Rational Tree Languages[1]

by

Patricia M. Hill[2] & Enea Zaffanella[3] & Roberto Bagnara[4]

May 2003

[1] The present work has been funded by MURST projects "Automatic Program Certification by Abstract Interpretation", "Abstract Interpretation, type systems and control-flow analysis", and "Automatic Aggregate- and Number-Reasoning for Computing: from Decision Algorithms to Constraint Programming with Multisets, Sets, and Maps"; by the Integrated Action Italy-Spain "Advanced Development Environments for Logic Programs"; by the University of Parma's FIL scientific research project (ex 60%) "Matematica pura e matematica applicata"; and by the UK's Engineering and Physical Sciences Research Council (EPSRC) under grant M05645.
[2] School of Computing, University of Leeds, UK, `hill@comp.leeds.ac.uk`
[3] Department of Mathematics, University of Parma, Italy, `zaffanella@cs.unipr.it`
[4] Department of Mathematics, University of Parma, Italy, `bagnara@cs.unipr.it`

Abstract

It is well-known that freeness and linearity information positively interact with aliasing information, allowing both the precision and the efficiency of the sharing analysis of logic programs to be improved. In this paper we present a novel combination of set-sharing with freeness and linearity information, which is characterized by an improved abstract unification operator. We provide a new abstraction function and prove the correctness of the analysis for both the finite tree and the rational tree cases. Moreover, we show that the same notion of redundant information as identified in (Bagnara et al. 2002; Zaffanella et al. 2002) also applies to this abstract domain combination: this allows for the implementation of an abstract unification operator running in polynomial time and achieving the same precision on all the considered observable properties.

1 Introduction

Even though the set-sharing domain is, in a sense, remarkably precise, more precision is attainable by combining it with other domains. In particular, freeness and linearity information has received much attention by the literature on sharing analysis (recall that a variable is said to be free if it is not bound to a non-variable term; it is linear if it is not bound to a term containing multiple occurrences of another variable).

As argued informally by Søndergaard (Søndergaard 1986), the mutual interaction between linearity and aliasing information can improve the accuracy of a sharing analysis. This observation has been formally applied in (Codish et al. 1991) to the specification of the abstract mgu operator for the domain ASub. In his PhD thesis (Langen 1990), Langen proposed a similar integration with linearity, but for the set-sharing domain. He has also shown how the aliasing information allows to compute freeness with a good degree of accuracy (however, freeness information was not exploited to improve aliasing). King (King 1994) has also shown how a more refined tracking of linearity allows for further precision improvements.

The synergy attainable from a bi-directional interaction between aliasing and freeness information was initially pointed out by Muthukumar and Hermenegildo (Muthukumar and Hermenegildo 1991; Muthukumar and Hermenegildo 1992). Since then, several authors considered the integration of set-sharing with freeness, sometimes also including additional explicit structural information (Codish et al. 1993; Codish et al. 1996; Filé 1994; King and Soper 1994).

Building on the results obtained in (Søndergaard 1986), (Codish et al. 1991) and (Muthukumar and Hermenegildo 1991), but independently from (Langen 1990), Hans and Winkler (Hans and Winkler 1992) proposed a combined integration of freeness and linearity information with set-sharing. Similar combinations have been proposed in (Bruynooghe and Codish 1993; Bruynooghe et al. 1994a; Bruynooghe et al. 1994b). From a more pragmatic point of view, Codish et al. (Codish et al. 1993; Codish et al. 1995) integrate the information captured by the domains of (Søndergaard 1986) and (Muthukumar and Hermenegildo 1991) by performing the analysis with both domains at the same time, exchanging information between the two components at each step.

Most of the above proposals differ in the carrier of the underlying abstract domain. Even when considering the simplest domain combinations where explicit structural information is ignored, there is no general consensus on the specification of the abstract unification procedure. From a theoretical point of view, once the abstract domain has been related to the concrete one by means of a Galois connection, it is always possible to specify the best correct approximation of each operator of the concrete semantics. However, empirical observations suggest that sub-optimal operators are likely to result in better complexity/precision trade-offs (Bagnara et al. 2000). As a consequence, it is almost impossible to identify "the right combination" of variable aliasing with freeness and linearity information, at least when practical issues, such as the complexity of the abstract unification procedure, are taken into account.

Given this state of affairs, we will now consider a domain combination whose carrier is essentially the same as specified by Langen (Langen 1990) and Hans and Winkler (Hans and Winkler 1992). (The same domain combination was also considered by Bruynooghe et al. (Bruynooghe et al. 1994a; Bruynooghe et al. 1994b), but with the addition of compoundness and explicit structural information.) The novelty of our proposal lies in the specification of an improved abstract unification procedure, better exploiting the interaction between sharing and linearity. As a matter of fact, we provide an example showing that all previous approaches to the combination of set-sharing with freeness and linearity are not uniformly more precise than the analysis based on the ASub domain (Codish et al. 1991; King 2000; Søndergaard 1986), whereas such a property is enjoyed by our proposal.

By extending the results of (Hill et al. 2002) to this combination, we provide a new abstraction function that can be applied to any logic language computing on domains of syntactic structures, with or without the occurs-check; by using this abstraction function, we also prove the correctness of the new abstract unification procedure. Moreover, we show that the same notion of redundant information as identified in (Bagnara et al. 2002; Zaffanella et al. 2002) also applies to this abstract domain combination. As a consequence, it is possible to implement an algorithm for abstract unification running in polynomial time and still obtain the same precision on all the considered observables: groundness, independence, freeness and linearity.

This paper is based on (Zaffanella 2001, Chapter 6), the PhD thesis of the second author. In Section 2, we define some notation and recall the basic concepts used later in the paper. In Section 3, we present the domain *SFL* that integrates set-sharing, freeness and linearity. In Section 4, we show that *SFL* is uniformly more precise than the domain ASub, whereas all the previous proposals for a domain integrating set-sharing and linearity fail to satisfy such a property. In Section 5, we show that the domain *SFL* can be simplified by removing some redundant information. In Section 6, we provide an experimental evaluation using the China analyzer (Bagnara 1997). In Section 7, we discuss some related work. Section 8 concludes with some final remarks. The proofs of the results stated in Sections 2, 3.1, 3.2, 4 and 5 are given in Appendix A, Appendix B, Appendix C, Appendix D and Appendix E, respectively.

2 Preliminaries

For a set S, $\wp(S)$ is the powerset of S. The cardinality of S is denoted by $\#S$ and the empty set is denoted by $\varnothing$. The notation $\wp_{\mathrm{f}}(S)$ stands for the set of all the *finite* subsets of S, while the notation $S \subseteq_{\mathrm{f}} T$ stands for $S \in \wp_{\mathrm{f}}(T)$. The set of all finite sequences of elements of S is denoted by S^*, the empty sequence by ϵ, and the concatenation of $s_1, s_2 \in S^*$ is denoted by $s_1 \,.\, s_2$.

2.1 Terms and Trees

Let *Sig* denote a possibly infinite set of function symbols, ranked over the set of natural numbers. Let *Vars* denote a denumerable set of variables, disjoint from *Sig*. Then *Terms* denotes the free algebra of all (possibly infinite) terms in the signature *Sig* having variables in *Vars*. Thus a term can be seen as an ordered labeled tree, possibly having some infinite paths and possibly containing variables: every inner node is labeled with a function symbol in *Sig* with a rank matching the number of the node's immediate descendants, whereas every leaf is labeled by either a variable in *Vars* or a function symbol in *Sig* having rank 0 (a constant). It is assumed that *Sig* contains at least two distinct function symbols, with one of them having rank 0.

If $t \in \mathit{Terms}$ then $\mathrm{vars}(t)$ and $\mathrm{mvars}(t)$ denote the set and the multiset of variables occurring in t, respectively. We will also write $\mathrm{vars}(o)$ to denote the set of variables occurring in an arbitrary syntactic object o.

Suppose $s, t \in \mathit{Terms}$: s and t are *independent* if $\mathrm{vars}(s) \cap \mathrm{vars}(t) = \varnothing$; we say that variable y *occurs linearly in* t, more briefly written using the predication $\mathrm{occ_lin}(y, t)$, if y occurs exactly once in $\mathrm{mvars}(t)$; t is said to be *ground* if $\mathrm{vars}(t) = \varnothing$; t is *free* if $t \in \mathit{Vars}$; t is *linear* if, for all $y \in \mathrm{vars}(t)$, we have $\mathrm{occ_lin}(y, t)$; finally, t is a *finite term* (or *Herbrand term*) if it contains a finite number of occurrences of function symbols. The sets of all ground, linear and finite terms are denoted by *GTerms*, *LTerms* and *HTerms*, respectively.

2.2 Substitutions

A *substitution* is a total function $\sigma\colon \mathit{Vars} \to \mathit{HTerms}$ that is the identity almost everywhere; in other words, the *domain* of σ,

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

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

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

If $t \in \mathit{HTerms}$, we write $t\sigma$ to denote $\sigma(t)$. Note that, for each substitution σ and each finite term $t \in \mathit{HTerms}$, if $t\sigma \in \mathit{Vars}$, then $t \in \mathit{Vars}$.

If $x \in \mathit{Vars}$ and $t \in \mathit{HTerms} \setminus \{x\}$, then $x \mapsto t$ is called a *binding*. The set of all bindings is denoted by *Bind*. Substitutions are denoted by the set of their bindings, thus a substitution σ is identified with the (finite) set

$$\big\{\, x \mapsto x\sigma \;\big|\; x \in \mathrm{dom}(\sigma) \,\big\}.$$

We denote by $\mathrm{vars}(\sigma)$ the set of variables occurring in the bindings of σ. We also define $\mathrm{range}(\sigma) \stackrel{\mathrm{def}}{=} \bigcup\big\{\, \mathrm{vars}(x\sigma) \;\big|\; x \in \mathrm{dom}(\sigma) \,\big\}$.

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

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

where $x_1, \ldots, x_n$ are distinct variables. 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 $t \in \mathit{Terms}$, we have $t\sigma\sigma = t\sigma$. Equivalently, σ is idempotent if and only if $\mathrm{dom}(\sigma) \cap \mathrm{range}(\sigma) = \varnothing$. The set of all idempotent substitutions is denoted by *ISubst* and $\mathit{ISubst} \subset \mathit{RSubst}$.

The composition of substitutions is defined in the usual way. Thus $\tau \circ \sigma$ is the substitution such that, for all terms $t \in \mathit{HTerms}$,

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

and has the formulation

$$\tau \circ \sigma = \big\{\, x \mapsto x\sigma\tau \;\big|\; x \in \mathrm{dom}(\sigma) \cup \mathrm{dom}(\tau), x \neq x\sigma\tau \,\big\}. \tag{1}$$

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

For each $\sigma \in \mathit{RSubst}$ and $s \in \mathit{HTerms}$, the sequence of finite terms

$$\sigma^0(s), \sigma^1(s), \sigma^2(s), \ldots$$

converges to a (possibly infinite) term, denoted $\sigma^\infty(s)$ (Intrigila and Zilli 1996; King 2000). Therefore, the function $\mathrm{rt}\colon \mathit{HTerms} \times \mathit{RSubst} \to \mathit{Terms}$ such that

$$\mathrm{rt}(s, \sigma) \stackrel{\mathrm{def}}{=} \sigma^\infty(s)$$

is well defined. Note that, in general, this function is not a substitution: while having a finite domain, its "bindings" $x \mapsto \mathrm{rt}(x, \sigma)$ can map a domain variable x into a term $\mathrm{rt}(x, \sigma) \in \mathit{Terms} \setminus \mathit{HTerms}$. However, as the name of the function suggests, the term $\mathrm{rt}(x, \sigma)$ is granted to be *rational*, meaning that it can only have a finite number of distinct subterms and hence, be finitely represented.

Example 1
Consider the substitutions

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

Note that there are substitutions, such as σ_2, that are not idempotent and nonetheless define finite trees only; namely, $\mathrm{rt}(x, \sigma_2) = f(a)$. Similarly, there are other substitutions, such as σ_4, whose bindings are not explicitly cyclic and nonetheless define rational trees that are infinite; namely, $\mathrm{rt}(x, \sigma_4) = f(f(f(\cdots)))$. Finally note that the 'rt' function is not defined on $\sigma_5 \notin \mathit{RSubst}$.

2.3 Equality Theories

An *equation* is of the form $s = t$ where $s, t \in \mathit{HTerms}$. *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 $\{\, s \mapsto t \mid (s = t) \in e \,\} \in \mathit{RSubst}$. 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). As is common in research work 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.

Let $\{r, s, t, s_1, \ldots, s_n, t_1, \ldots, t_n\} \subseteq \mathit{HTerms}$. We assume that any equality theory T over *Terms* 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, where f and g are distinct function symbols or $n \neq m$:

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

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

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), denoted $\mathcal{FT}$, 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

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

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

An alternative approach used in some implementations of logic programming systems, such as Prolog II, SICStus and Oz, does not require the occurs-check axioms. This approach is based on the theory of rational trees (Colmerauer 1982;

Colmerauer 1984), denoted $\mathcal{RT}$. 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 variable which is not in the domain, 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 expect. We refer the interested reader to (Jaffar et al. 1987; Keisu 1994; Maher 1988) for details on the subject.

In the sequel we use the expression "equality theory" to denote any consistent, decidable theory T satisfying the congruence axioms. We also use the expression "syntactic equality theory" to denote any equality theory T also satisfying the identity axioms.

We say that a substitution $\sigma \in \mathit{RSubst}$ is *satisfiable* in an equality theory T if, when interpreting σ as an equation system in rational solved form,

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

Let $e \in \wp_{\mathrm{f}}(\mathit{Eqs})$ be a set of equations in an equality theory T. A substitution $\sigma \in \mathit{RSubst}$ is called a *solution for e in T* if σ is satisfiable in T and $T \vdash \forall(\sigma \rightarrow e)$; we say that e is satisfiable if it has a solution. If $\mathrm{vars}(\sigma) \subseteq \mathrm{vars}(e)$, 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. When the theory T is clear from the context, the set of all the relevant most general solutions for e in T is denoted by $\mathrm{mgs}(e)$.

Example 2
Let $e = \big\{g(x) = g(f(y)), f(x) = y, z = g(w)\big\}$ and

$$\sigma = \big\{x \mapsto f(y), y \mapsto f(x), z \mapsto g(w)\big\}.$$

Then, for any syntactic equality theory T, we have $T \vdash \forall(\sigma \leftrightarrow e)$. Since $\sigma \in \mathit{RSubst}$, then σ and hence e is satisfiable in $\mathcal{RT}$. Intuitively, whatever rational tree t_w is assigned to the parameter variable w, there exist rational trees t_x, t_y and t_z that, when assigned to the domain variables x, y and z, will turn σ into a set of trivial identities; namely, let t_x and t_y be both equal to the infinite rational tree $f(f(f(\cdots)))$, which is usually denoted by f^ω, and let t_z be the rational tree $g(t_w)$. Thus σ is a relevant most general solution for e in $\mathcal{RT}$. In contrast,

$$\tau = \big\{x \mapsto f(y), y \mapsto f(x), z \mapsto g(f(a))\big\}$$

is just a relevant solution for e in $\mathcal{RT}$. Also observe that, for any equality theory T,

$$T \vdash \forall\Big(\sigma \rightarrow \big\{x = f(f(x))\big\}\Big)$$

so that σ does not satisfy the occurs-check axioms. Therefore, neither σ nor e are satisfiable in the Herbrand equality theory $\mathcal{FT}$. Intuitively, there is no finite tree t_x such that $t_x = f(f(t_x))$.

We have the following useful result regarding 'rt' and satisfiable substitutions that are equivalent with respect to any given syntactic equality theory.
Proposition 3
Let $\sigma, \tau \in \mathit{RSubst}$ be satisfiable in the syntactic equality theory T and suppose that $T \vdash \forall(\sigma \leftrightarrow \tau)$. Then

$$\mathrm{rt}(y,\sigma) \in \mathit{Vars} \iff \mathrm{rt}(y,\tau) \in \mathit{Vars}, \tag{9}$$
$$\mathrm{rt}(y,\sigma) \in \mathit{GTerms} \iff \mathrm{rt}(y,\tau) \in \mathit{GTerms}, \tag{10}$$
$$\mathrm{rt}(y,\sigma) \in \mathit{LTerms} \iff \mathrm{rt}(y,\tau) \in \mathit{LTerms}. \tag{11}$$

2.4 Galois Connections and uco's

Given two complete lattices $(C, \leq_C)$ and $(A, \leq_A)$, a *Galois connection* is a pair of monotonic functions $\alpha\colon C \to A$ and $\gamma\colon A \to C$ such that

$$\forall c \in C : c \leq_C \gamma\big(\alpha(c)\big), \qquad \forall a \in A : \alpha\big(\gamma(a)\big) \leq_A a.$$

The functions α and γ are said to be the abstraction and concretization functions, respectively. A *Galois insertion* is a Galois connection where the concretization function γ is injective.

An *upper closure operator* (uco) $\rho\colon C \to C$ on the complete lattice $(C, \leq_C)$ is a monotonic, idempotent and extensive[1] self-map. The set of all uco's on C, denoted by $\mathrm{uco}(C)$, is itself a complete lattice. For any $\rho \in \mathrm{uco}(C)$, the set $\rho(C)$, i.e., the image under ρ of the lattice carrier, is a complete lattice under the same partial order $\leq_C$ defined on C. Given a Galois connection, the function $\rho \stackrel{\mathrm{def}}{=} \gamma \circ \alpha$ is an element of $\mathrm{uco}(C)$. The presentation of abstract interpretation in terms of Galois connections can be rephrased by using uco's. In particular, the partial order $\sqsubseteq$ defined on $\mathrm{uco}(C)$ formalizes the intuition of an abstract domain being more precise than another one; moreover, given two elements $\rho_1, \rho_2 \in \mathrm{uco}(C)$, their reduced product (Cousot and Cousot 1979), denoted $\rho_1 \sqcap \rho_2$, is their glb on $\mathrm{uco}(C)$.

2.5 The Set-Sharing Domain

The set-sharing domain of Jacobs and Langen (Jacobs and Langen 1989), encodes both aliasing and groundness information. Let $\mathit{VI} \subseteq_{\mathrm{f}} \mathit{Vars}$ be a fixed and finite set of variables of interest. An element of the set-sharing domain (a *sharing set*) is a set of subsets of VI (the *sharing groups*). Note that the empty set is not a sharing group.

Definition 4
(The *set-sharing* lattice.) Let $\mathit{SG} \stackrel{\mathrm{def}}{=} \wp(\mathit{VI}) \setminus \{\varnothing\}$ be the set of *sharing groups*. The set-sharing lattice is defined as $\mathit{SH} \stackrel{\mathrm{def}}{=} \wp(\mathit{SG})$, ordered by subset inclusion.

The following operators on SH are needed for the specification of the abstract semantics.

[1] Namely, $c \leq_C \rho(c)$ for each $c \in C$.

Definition 5
(Auxiliary operators on *SH*.) For each $sh, sh_1, sh_2 \in SH$ and each $V \subseteq VI$, we define the following functions:
the *star-union* function $(\cdot)^\star$: $SH \to SH$, is defined as

$$sh^\star \stackrel{\text{def}}{=} \left\{ S \in SG \;\middle|\; \exists n \geq 1 \,.\, \exists S_1, \ldots, S_n \in sh \,.\, S = S_1 \cup \cdots \cup S_n \right\};$$

the extraction of the *relevant component of sh with respect to* V is encoded by rel: $\wp(VI) \times SH \to SH$ defined as

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

the *irrelevant component of sh with respect to* V is thus defined as

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

the *binary union* function bin: $SH \times SH \to SH$ is defined as

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

the *self-bin-union* operation on *SH* is defined as

$$sh^2 \stackrel{\text{def}}{=} \mathrm{bin}(sh, sh);$$

the *abstract existential quantification* function aexists: $SH \times \wp(VI) \to SH$ is defined as

$$\mathrm{aexists}(sh, V) \stackrel{\text{def}}{=} \left\{ S \setminus V \;\middle|\; S \in sh, S \setminus V \neq \varnothing \right\} \cup \left\{ \{x\} \;\middle|\; x \in V \right\}.$$

In (Bagnara et al. 1997; Bagnara et al. 2002) it was shown that the domain *SH* contains many elements that are redundant for the computation of the actual *observable* properties of the analysis, definite groundness and definite independence. The following formalization of these observables is a rewording of the definitions provided in (Zaffanella et al. 1999; Zaffanella et al. 2002).

Definition 6
(The observables of *SH*.) The *groundness* and *independence* observables (on *SH*) $\rho_{Con}, \rho_{PS} \in \mathrm{uco}(SH)$ are defined, for each $sh \in SH$, by

$$\rho_{Con}(sh) \stackrel{\text{def}}{=} \left\{ S \in SG \;\middle|\; S \subseteq \mathrm{vars}(sh) \right\},$$
$$\rho_{PS}(sh) \stackrel{\text{def}}{=} \left\{ S \in SG \;\middle|\; (P \subseteq S \wedge \#P = 2) \implies (\exists T \in sh \,.\, P \subseteq T) \right\}.$$

Note that, as usual in sharing analysis domains, definite groundness and definite independence are both represented by encoding possible non-groundness and possible pair-sharing information.

The abstract domain *PSD* (Bagnara et al. 2002; Zaffanella et al. 2002) is the simplest abstraction of the domain *SH* that still preserves the same precision on groundness and independence.

Definition 7
(The *pair-sharing dependency* lattice *PSD*.) The operator $\rho_{PSD} \in \mathrm{uco}(SH)$ is defined, for each $sh \in SH$, by

$$\rho_{PSD}(sh) \stackrel{\mathrm{def}}{=} \left\{ S \in SG \;\middle|\; \forall y \in S : S = \bigcup \{\, U \in sh \mid y \in U \subseteq S \,\} \right\}.$$

The *pair-sharing dependency* lattice is $PSD \stackrel{\mathrm{def}}{=} \rho_{PSD}(SH)$.

In the following example we provide an intuitive interpretation of the approximation induced by the three upper closure operators of Definitions 6 and 7.

Example 8
Let $VI = \{v, w, x, y, z\}$ and consider[2] $sh = \{vx, vy, xy, xyz\}$. Then

$$\begin{aligned}
\rho_{Con}(sh) &= \{v, vx, vxy, vxyz, vxz, vy, vyz, vz, x, xy, xyz, xz, y, yz, z\},\\
\rho_{PS}(sh) &= \{v, vx, vxy, vy, w, x, xy, xyz, xz, y, yz, z\},\\
\rho_{PSD}(sh) &= \{vx, vxy, vy, xy, xyz\}.
\end{aligned}$$

When observing $\rho_{Con}(sh)$, the only information available is that variable w does not occur in a sharing group; intuitively, this means that w is definitely ground. All the other information encoded in sh is lost; for instance, in sh variables v and z never occur in the *same* sharing group (i.e., they are definitely independent), while this happens in $\rho_{Con}(sh)$.

When observing $\rho_{PS}(sh)$, it should be noted that two distinct variables occur in the same sharing group if and only if they were also occurring together in a sharing group of sh, so that the definite independence information is preserved (e.g., v and z keep their independence). On the other hand, all the variables in VI occur as singletons in $\rho_{PS}(sh)$ whether or not they are known to be ground; for instance, $\{w\}$ occurs in $\rho_{PS}(sh)$ although w does not occur in any sharing group in sh.

By noting that $\rho_{PSD}(sh) \subset \rho_{Con}(sh) \cap \rho_{PS}(sh)$, it follows that $\rho_{PSD}(sh)$ preserves both the definite groundness and the definite independence information of sh; moreover, as the inclusion is strict, $\rho_{PSD}(sh)$ encodes other information, such as variable covering (the interested reader is referred to (Bagnara et al. 2002; Zaffanella et al. 2002) for a more formal discussion).

3 The Domain *SFL*

The abstract domain *SFL* is made up of three components, providing different kinds of sharing information regarding the set of variables of interest VI: the first component is the set-sharing domain SH of Jacobs and Langen (Jacobs and Langen 1989); the other two components provide freeness and linearity information, each represented by simply recording those variables of interest that are known to enjoy the corresponding property.

[2] In this and all the following examples, we will adopt a simplified notation for a set-sharing element sh, omitting inner braces. For instance, we will write $\{xy, xz, yz\}$ to denote $\{\{x, y\}, \{x, z\}, \{y, z\}\}$.

Definition 9

(The domain *SFL*.) Let $F \stackrel{\text{def}}{=} \wp(VI)$ and $L \stackrel{\text{def}}{=} \wp(VI)$ be partially ordered by reverse subset inclusion. The abstract domain *SFL* is defined as

$$SFL \stackrel{\text{def}}{=} \{\, \langle sh, f, l\rangle \mid sh \in SH, f \in F, l \in L \,\}$$

and is ordered by $\leq_S$, the component-wise extension of the orderings defined on the sub-domains. With this ordering, *SFL* is a complete lattice whose least upper bound operation is denoted by alub_S. The bottom element $\langle \varnothing, VI, VI\rangle$ will be denoted by $\bot_S$.

3.1 The Abstraction Function

When the concrete domain is based on the theory of finite trees, idempotent substitutions provide a finitely computable *strong normal form* for domain elements, meaning that different substitutions describe different sets of finite trees.[3] In contrast, when working on a concrete domain based on the theory of rational trees, substitutions in rational solved form, while being finitely computable, no longer satisfy this property: there can be an infinite set of substitutions in rational solved form all describing the same set of rational trees (i.e., the same element in the "intended" semantics). For instance, the substitutions

$$\sigma_n = \{x \mapsto \overbrace{f(\cdots f(}^{n} x) \cdots)\},$$

for $n = 1, 2, \ldots$, all map the variable x into the same infinite rational tree f^ω.

Ideally, a strong normal form for the set of rational trees described by a substitution $\sigma \in RSubst$ can be obtained by computing the limit σ^∞. The problem is that σ^∞ can map domain variables to infinite rational terms and may not be in *RSubst*.

This poses a non-trivial problem when trying to define "good" abstraction functions, since it would be really desirable for this function to map any two equivalent concrete elements to the same abstract element. As shown in (Hill et al. 2002), the classical abstraction function for set-sharing analysis (Cortesi and Filé 1999; Jacobs and Langen 1989), which was defined only for substitutions that are idempotent, does not enjoy this property when applied, as it is, to arbitrary substitutions in rational solved form. In (Hill et al. 1998; Hill et al. 2002), this problem is solved by replacing the sharing group operator 'sg' of (Jacobs and Langen 1989) by an occurrence operator, 'occ', defined by means of a fixpoint computation. However, to simplify the presentation, here we define 'occ' directly by exploiting the fact that the number of iterations needed to reach the fixpoint is bounded by the number of bindings in the substitution.

Definition 10

(Occurrence operator.) For each $\sigma \in RSubst$ and $v \in Vars$, the *occurrence*

[3] As usual, this is modulo the possible renaming of variables.

operator occ: $RSubst \times Vars \to \wp_{\mathrm{f}}(Vars)$ is defined as

$$\mathrm{occ}(\sigma, v) \stackrel{\mathrm{def}}{=} \{\, y \in Vars \;\big|\; n = \#\sigma, v \in \mathrm{vars}(y\sigma^n) \setminus \mathrm{dom}(\sigma) \,\}.$$

For each $\sigma \in RSubst$, the operator ssets: $RSubst \to SH$ is defined as

$$\mathrm{ssets}(\sigma) \stackrel{\mathrm{def}}{=} \{\, \mathrm{occ}(\sigma, v) \cap VI \;\big|\; v \in Vars \,\} \setminus \{\varnothing\}.$$

The operator 'ssets' is introduced for notational convenience only.

Example 11
Let

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

Then $\mathrm{dom}(\sigma) = \mathrm{dom}(\tau) = \{x_1, x_2, x_3\}$ so that $\mathrm{occ}(\sigma, x_i) = \mathrm{occ}(\tau, x_i) = \varnothing$, for $i = 1$, 2, 3 and $\mathrm{occ}(\sigma, x_4) = \mathrm{occ}(\tau, x_4) = \{x_1, x_2, x_3, x_4\}$. As a consequence, supposing that $VI = \{x_1, x_2, x_3, x_4\}$, we obtain $\mathrm{ssets}(\sigma) = \mathrm{ssets}(\tau) = \{VI\}$.

In a similar way, it is possible to define suitable operators for groundness, freeness and linearity. As all ground trees are linear, a knowledge of the definite groundness information can be useful for proving properties concerning the linearity abstraction. Groundness is already encoded in the abstraction for set-sharing provided in Definition 10; nonetheless, for both a simplified notation and a clearer intuitive reading, we now explicitly define the set of variables that are associated to ground trees by a substitution in *RSubst*.

Definition 12
(Groundness operator.) The *groundness operator* gvars: $RSubst \to \wp_{\mathrm{f}}(Vars)$ is defined, for each $\sigma \in RSubst$, by

$$\mathrm{gvars}(\sigma) \stackrel{\mathrm{def}}{=} \{\, y \in \mathrm{dom}(\sigma) \;\big|\; \forall v \in Vars : y \notin \mathrm{occ}(\sigma, v) \,\}.$$

Example 13
Consider $\sigma \in RSubst$ where

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

Then $\mathrm{gvars}(\sigma) = \{x_1, x_2, x_3, x_4\}$. Observe that $x_1 \in \mathrm{gvars}(\sigma)$ although $x_1\sigma \in Vars$. Also, $x_3 \in \mathrm{gvars}(\sigma)$ although $\mathrm{vars}(x_3\sigma^i) = \{x_2, x_4\} \neq \varnothing$ for all $i \geq 2$.

As for possible sharing, the definite freeness information can be extracted from a substitution in rational solved form by observing the result of a bounded number of applications of the substitution.

Definition 14
(Freeness operator.) The *freeness operator* fvars: $RSubst \to \wp(Vars)$ is defined, for each $\sigma \in RSubst$, by

$$\mathrm{fvars}(\sigma) \stackrel{\mathrm{def}}{=} \{\, y \in Vars \mid n = \#\sigma, y\sigma^n \in Vars \,\}.$$

As $\sigma \in RSubst$ has no circular subset, $y \in \mathrm{fvars}(\sigma)$ implies $y\sigma^n \in Vars \setminus \mathrm{dom}(\sigma)$.

Example 15
Let $VI = \{x_1, x_2, x_3, x_4, x_5\}$ and consider $\sigma \in RSubst$ where

$$\sigma = \big\{x_1 \mapsto x_2, x_2 \mapsto f(x_3), x_3 \mapsto x_4, x_4 \mapsto x_5\big\}.$$

Then $\mathrm{fvars}(\sigma) \cap VI = \{x_3, x_4, x_5\}$. Thus $x_1 \notin \mathrm{fvars}(\sigma)$ although $x_1\sigma \in Vars$. Also, $x_3 \in \mathrm{fvars}(\sigma)$ although $x_3\sigma \in \mathrm{dom}(\sigma)$.

As in previous cases, the definite linearity information can be extracted by observing the result of a bounded number of applications of the considered substitution.

Definition 16
(Linearity operator.) The *linearity operator* $\mathrm{lvars}\colon RSubst \to \wp(Vars)$ is defined, for each $\sigma \in RSubst$, by

$$\mathrm{lvars}(\sigma) \stackrel{\mathrm{def}}{=} \big\{\, y \in Vars \;\big|\; n = \#\sigma, \forall z \in \mathrm{vars}(y\sigma^n) \setminus \mathrm{dom}(\sigma) : \mathrm{occ_lin}(z, y\sigma^{2n}) \,\big\}.$$

In the next example we consider the extraction of linearity from two substitutions. The substitution σ shows that, in contrast with the case of set-sharing and freeness, for linearity we may need to compute up to $2n$ applications, where $n = \#\sigma$; the substitution τ shows that, when observing the term $y\tau^{2n}$, multiple occurrences of domain variables have to be disregarded.

Example 17
Let $VI = \{x_1, x_2, x_3, x_4\}$ and consider $\sigma \in RSubst$ where

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

Then $\mathrm{lvars}(\sigma) \cap VI = \{x_4\}$. Observe that $x_1 \notin \mathrm{lvars}(\sigma)$. This is because $x_4 \notin \mathrm{dom}(\sigma)$, $x_1\sigma^3 = f(x_1, x_4)$ so that $x_4 \in \mathrm{vars}(x_1\sigma^3)$ and $x_1\sigma^6 = f\big(f(x_1, x_4), x_4\big)$ so that $\mathrm{occ_lin}(x_4, x_1\sigma^6)$ does not hold. Note also that $\mathrm{occ_lin}(x_4, x_1\sigma^i)$ holds for $i = 3, 4, 5$.

Consider now $\tau \in RSubst$ where

$$\tau = \big\{x_1 \mapsto f(x_2, x_2), x_2 \mapsto f(x_2)\big\}.$$

Then $\mathrm{lvars}(\tau) \cap VI = VI$. Note that we have $x_1 \in \mathrm{lvars}(\tau)$ although, for all $i > 0$, $x_2 \in \mathrm{dom}(\tau)$ occurs more than once in the term $x_1\tau^i$.

The occurrence, groundness, freeness and linearity operators are invariant with respect to substitutions that are equivalent in the given syntactic equality theory.

Proposition 18
Let $\sigma, \tau \in RSubst$ be satisfiable in the syntactic equality theory T and suppose that $T \vdash \forall(\sigma \leftrightarrow \tau)$. Then

$$\mathrm{ssets}(\sigma) = \mathrm{ssets}(\tau), \tag{12}$$
$$\mathrm{gvars}(\sigma) = \mathrm{gvars}(\tau), \tag{13}$$
$$\mathrm{fvars}(\sigma) = \mathrm{fvars}(\tau), \tag{14}$$
$$\mathrm{lvars}(\sigma) = \mathrm{lvars}(\tau). \tag{15}$$

Moreover, these operators precisely capture the intended properties over the domain of rational trees.

Proposition 19
If $\sigma \in \mathit{RSubst}$ and $y, v \in \mathit{Vars}$ then

$$y \in \mathrm{occ}(\sigma, v) \iff v \in \mathrm{vars}\big(\mathrm{rt}(y, \sigma)\big), \tag{16}$$
$$y \in \mathrm{gvars}(\sigma) \iff \mathrm{rt}(y, \sigma) \in \mathit{GTerms}, \tag{17}$$
$$y \in \mathrm{fvars}(\sigma) \iff \mathrm{rt}(y, \sigma) \in \mathit{Vars}, \tag{18}$$
$$y \in \mathrm{lvars}(\sigma) \iff \mathrm{rt}(y, \sigma) \in \mathit{LTerms}. \tag{19}$$

It follows from (16) and (18) that any free variable necessarily shares (at least, with itself). Also, as $\mathit{Vars} \cup \mathit{GTerms} \subset \mathit{LTerms}$, it follows from (17), (18) and (19) that any variable that is either ground or free is also necessarily linear. Thus we have the following corollary.

Corollary 20
If $\sigma \in \mathit{RSubst}$, then

$$\mathrm{fvars}(\sigma) \subseteq \mathrm{vars}\big(\mathrm{ssets}(\sigma)\big),$$
$$\mathrm{fvars}(\sigma) \cup \mathrm{gvars}(\sigma) \subseteq \mathrm{lvars}(\sigma).$$

We are now in position to define the abstraction function mapping rational trees to elements of the domain *SFL*.

Definition 21
(The abstraction function for *SFL*.) For each substitution $\sigma \in \mathit{RSubst}$, the function $\alpha_s \colon \mathit{RSubst} \to \mathit{SFL}$ is defined by

$$\alpha_s(\sigma) \stackrel{\mathrm{def}}{=} \big\langle \mathrm{ssets}(\sigma), \mathrm{fvars}(\sigma) \cap \mathit{VI}, \mathrm{lvars}(\sigma) \cap \mathit{VI} \big\rangle,$$

The concrete domain $\wp(\mathit{RSubst})$ is related to *SFL* by means of the *abstraction function* $\alpha_s \colon \wp(\mathit{RSubst}) \to \mathit{SFL}$ such that, for each $\Sigma \in \wp(\mathit{RSubst})$,

$$\alpha_s(\Sigma) \stackrel{\mathrm{def}}{=} \mathrm{alub}_s\big\{\, \alpha_s(\sigma) \;\big|\; \sigma \in \Sigma \,\big\}.$$

Since the abstraction function α_s is additive, the concretization function is given by the adjoint (Cousot and Cousot 1977)

$$\gamma_s\big(\langle sh, f, l\rangle\big) \stackrel{\mathrm{def}}{=} \big\{\, \sigma \in \mathit{RSubst} \;\big|\; \mathrm{ssets}(\sigma) \subseteq sh, \mathrm{fvars}(\sigma) \supseteq f, \mathrm{lvars}(\sigma) \supseteq l \,\big\}.$$

With Definition 21 and Proposition 18, one of our objectives is fulfilled: substitutions in *RSubst* that are equivalent have the same abstraction.

Corollary 22
Let $\sigma, \tau \in \mathit{RSubst}$ be satisfiable in the syntactic equality theory T and suppose $T \vdash \forall(\sigma \leftrightarrow \tau)$. Then $\alpha_s(\sigma) = \alpha_s(\tau)$.

Observe that the Galois connection defined by the functions α_s and γ_s is not a Galois insertion since different abstract elements are mapped by γ_s to the same set of concrete computation states. To see this it is sufficient to observe that, by Corollary 20, any abstract element $d = \langle sh, f, l\rangle \in \mathit{SFL}$ such that $f \not\subseteq \mathrm{vars}(sh)$, as is the case for the bottom element $\bot_s$, satisfies $\gamma_s(d) = \gamma_s(\bot_s) = \varnothing$; thus, all such

d's will represent the semantics of those program fragments that have no successful computations. Similarly, by letting $V = \big(VI \setminus \mathrm{vars}(sh)\big) \cup f$, it can be seen that, for any l' such that $V \cup l = V \cup l'$, we have, again by Corollary 20, $\gamma_S(d) = \gamma_S\big(\langle sh, f, l' \rangle\big)$.

Of course, by taking the abstract domain as the subset of *SFL* that is the co-domain of α_S, we would have a Galois insertion. However, apart from the simple cases shown above, it is somehow difficult to *explicitly* characterize such a set. For instance, as observed in (Filé 1994), if

$$d = \big\langle \{xy, xz, yz\}, \{x, y, z\}, \{x, y, z\} \big\rangle \in SFL$$

we have $\gamma_S(d) = \gamma_S(\bot_S) = \varnothing$. It is worth stressing that these "spurious" elements do not compromise the correctness of the analysis and, although they can affect the precision of the analysis, they rarely occur in practice (Bagnara et al. 2000; Zaffanella 2001).

3.2 The Abstract Operators

The specification of the abstract unification operator on the domain *SFL* is rather complex, since it is based on a very detailed case analysis. To achieve some modularity, that will be also useful when proving its correctness, in the next definition we introduce several auxiliary abstract operators.

Definition 23
(Auxiliary operators in *SFL*.) Let $s, t \in HTerms$ be finite terms such that $\mathrm{vars}(s) \cup \mathrm{vars}(t) \subseteq VI$. For each $d = \langle sh, f, l \rangle \in SFL$ we define the following predicates:
s and t are *independent in* d if and only if $\mathrm{ind}_d \colon HTerms^2 \to Bool$ holds for (s, t), where

$$\mathrm{ind}_d(s, t) \stackrel{\mathrm{def}}{=} \Big(\mathrm{rel}\big(\mathrm{vars}(s), sh\big) \cap \mathrm{rel}\big(\mathrm{vars}(t), sh\big) = \varnothing\Big);$$

t is *ground in* d if and only if $\mathrm{ground}_d \colon HTerms \to Bool$ holds for t, where

$$\mathrm{ground}_d(t) \stackrel{\mathrm{def}}{=} \big(\mathrm{vars}(t) \subseteq VI \setminus \mathrm{vars}(sh)\big);$$

$y \in \mathrm{vars}(t)$ *occurs linearly (in* t*) in* d if and only if $\mathrm{occ_lin}_d \colon VI \times HTerms \to Bool$ holds for (y, t), where

$$\mathrm{occ_lin}_d(y, t) \stackrel{\mathrm{def}}{=} \mathrm{ground}_d(y) \vee \Big(\mathrm{occ_lin}(y, t) \wedge (y \in l) \\ \wedge \forall z \in \mathrm{vars}(t) : \big(y \neq z \implies \mathrm{ind}_d(y, z)\big)\Big);$$

t is *free in* d if and only if $\mathrm{free}_d \colon HTerms \to Bool$ holds for t, where

$$\mathrm{free}_d(t) \stackrel{\mathrm{def}}{=} (t \in f);$$

t is *linear in* d if and only if $\mathrm{lin}_d \colon HTerms \to Bool$ holds for t, where

$$\mathrm{lin}_d(t) \stackrel{\mathrm{def}}{=} \forall y \in \mathrm{vars}(t) : \mathrm{occ_lin}_d(y, t).$$

The function $\mathrm{share_with}_d \colon \mathit{HTerms} \to \wp(\mathit{VI})$ yields the set of variables of interest that may share with the given term. For each $t \in \mathit{HTerms}$,

$$\mathrm{share_with}_d(t) \stackrel{\mathrm{def}}{=} \mathrm{vars}\Big(\mathrm{rel}\big(\mathrm{vars}(t), \mathit{sh}\big)\Big).$$

The function $\mathrm{cyclic}_x^t \colon \mathit{SH} \to \mathit{SH}$ strengthens the sharing set sh by forcing the coupling of x with t. For each $\mathit{sh} \in \mathit{SH}$ and each $(x \mapsto t) \in \mathit{Bind}$,

$$\mathrm{cyclic}_x^t(\mathit{sh}) \stackrel{\mathrm{def}}{=} \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), \mathit{sh}\big) \cup \mathrm{rel}\big(\mathrm{vars}(t) \setminus \{x\}, \mathit{sh}\big).$$

As a first correctness result, we have that the auxiliary operators correctly approximate the corresponding concrete properties.

Theorem 24
Let $d \in \mathit{SFL}$, $\sigma \in \gamma_S(d)$ and $y \in \mathit{VI}$. Let also $s, t \in \mathit{HTerms}$ be two finite terms such that $\mathrm{vars}(s) \cup \mathrm{vars}(t) \subseteq \mathit{VI}$. Then

$$\mathrm{ind}_d(s,t) \implies \mathrm{vars}\big(\mathrm{rt}(s,\sigma)\big) \cap \mathrm{vars}\big(\mathrm{rt}(t,\sigma)\big) = \varnothing; \tag{20}$$

$$\mathrm{ind}_d(y,t) \iff y \notin \mathrm{share_with}_d(t); \tag{21}$$

$$\mathrm{free}_d(t) \implies \mathrm{rt}(t,\sigma) \in \mathit{Vars}; \tag{22}$$

$$\mathrm{ground}_d(t) \implies \mathrm{rt}(t,\sigma) \in \mathit{GTerms}; \tag{23}$$

$$\mathrm{lin}_d(t) \implies \mathrm{rt}(t,\sigma) \in \mathit{LTerms}. \tag{24}$$

Example 25
Let $\mathit{VI} = \{v, w, x, y, z\}$ and consider the abstract element $d = \langle \mathit{sh}, f, l \rangle \in \mathit{SFL}$, where

$$\mathit{sh} = \{v, wz, xz, z\}, \qquad f = \{v\}, \qquad l = \{v, x, y, z\}.$$

Then, by applying Definition 23, we obtain the following.

- $\mathrm{ground}_d(x)$ does not hold whereas $\mathrm{ground}_d\big(h(y)\big)$ holds.
- $\mathrm{free}_d(v)$ holds but $\mathrm{free}_d\big(h(v)\big)$ does not hold.
- Both $\mathrm{ind}_d(w,x)$ and $\mathrm{ind}_d\big(f(w,y), f(x,y)\big)$ hold whereas $\mathrm{ind}_d(x,z)$ does not hold; note that, in the second case, the two arguments of the predicate do share y, but this does not affect the independence of the corresponding terms, because y is definitely ground in the abstract element d.
- Let $t = f(w,x,x,y,y,z)$; then $\mathrm{occ_lin}_d(w,t)$ does not hold because $w \notin l$; $\mathrm{occ_lin}_d(x,t)$ does not hold because x occurs more than once in t; $\mathrm{occ_lin}_d(y,t)$ holds, even though y occurs twice in t, because y is definitely ground in d; $\mathrm{occ_lin}_d(z,t)$ does not hold because both x and z occur in term t and, as observed in the point above, $\mathrm{ind}_d(x,z)$ does not hold.
- For the reasons given in the point above, $\mathrm{lin}_d(t)$ does not hold; in contrast, $\mathrm{lin}_d\big(f(y,y,z)\big)$ holds.
- $\mathrm{share_with}_d(w) = \{w, z\}$ and $\mathrm{share_with}_d(x) = \{x, z\}$; thus, both w and x may share one or more variables with z; since we observed that w and x are definitely independent in d, this means that the set of variables that w shares with z is disjoint from the set of variables that x shares with z.

- Let $t = f(w, z)$; then

$$\begin{aligned}\mathrm{cyclic}_z^t(sh) &= \overline{\mathrm{rel}}\big(\{w, z\}, sh\big) \cup \mathrm{rel}\big(\{w\}, sh\big)\\ &= \{v\} \cup \{wz\}\\ &= sh \setminus \{xz, z\}.\end{aligned}$$

  An intuitive explanation of the usefulness of this operator is deferred until after the introduction of the abstract mgu operator (see also Example 29).

We now introduce the abstract mgu operator, specifying how a single binding affects each component of the domain *SFL* in the context of a syntactic equality theory T.

Definition 26
(amgu_s.) The function amgu_s: $\mathit{SFL} \times \mathit{Bind} \to \mathit{SFL}$ captures the effects of a binding on an element of *SFL*. Let $d = \langle sh, f, l\rangle \in \mathit{SFL}$ and $(x \mapsto t) \in \mathit{Bind}$, where $\{x\} \cup \mathrm{vars}(t) \subseteq \mathit{VI}$. Let also

$$sh' \stackrel{\mathrm{def}}{=} \mathrm{cyclic}_x^t(sh_- \cup sh''),$$

where

$$\begin{aligned} sh_x &\stackrel{\mathrm{def}}{=} \mathrm{rel}\big(\{x\}, sh\big), & sh_t &\stackrel{\mathrm{def}}{=} \mathrm{rel}\big(\mathrm{vars}(t), sh\big),\\ sh_{xt} &\stackrel{\mathrm{def}}{=} sh_x \cap sh_t, & sh_- &\stackrel{\mathrm{def}}{=} \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), sh\big),\end{aligned}$$

$$sh'' \stackrel{\mathrm{def}}{=} \begin{cases} \mathrm{bin}(sh_x, sh_t), & \text{if } \mathrm{free}_d(x) \vee \mathrm{free}_d(t);\\ \mathrm{bin}\big(sh_x \cup \mathrm{bin}(sh_x, sh_{xt}^\star),\\ \qquad sh_t \cup \mathrm{bin}(sh_t, sh_{xt}^\star)\big), & \text{if } \mathrm{lin}_d(x) \wedge \mathrm{lin}_d(t);\\ \mathrm{bin}(sh_x^\star, sh_t), & \text{if } \mathrm{lin}_d(x);\\ \mathrm{bin}(sh_x, sh_t^\star), & \text{if } \mathrm{lin}_d(t);\\ \mathrm{bin}(sh_x^\star, sh_t^\star), & \text{otherwise.}\end{cases}$$

Letting $S_x \stackrel{\mathrm{def}}{=} \mathrm{share_with}_d(x)$ and $S_t \stackrel{\mathrm{def}}{=} \mathrm{share_with}_d(t)$, we also define

$$f' \stackrel{\mathrm{def}}{=} \begin{cases} f, & \text{if } \mathrm{free}_d(x) \wedge \mathrm{free}_d(t);\\ f \setminus S_x, & \text{if } \mathrm{free}_d(x);\\ f \setminus S_t, & \text{if } \mathrm{free}_d(t);\\ f \setminus (S_x \cup S_t), & \text{otherwise;}\end{cases}$$

$$l' \stackrel{\mathrm{def}}{=} \big(\mathit{VI} \setminus \mathrm{vars}(sh')\big) \cup f' \cup l'',$$

where

$$l'' \stackrel{\mathrm{def}}{=} \begin{cases} l \setminus (S_x \cap S_t), & \text{if } \mathrm{lin}_d(x) \wedge \mathrm{lin}_d(t);\\ l \setminus S_x, & \text{if } \mathrm{lin}_d(x);\\ l \setminus S_t, & \text{if } \mathrm{lin}_d(t);\\ l \setminus (S_x \cup S_t), & \text{otherwise.}\end{cases}$$

Then

$$\mathrm{amgu}_{_S}\big(d, x \mapsto t\big) \stackrel{\text{def}}{=} \begin{cases} \bot_{_S}, & \text{if } d = \bot_{_S} \vee \big(T = \mathcal{FT} \wedge x \in \mathrm{vars}(t)\big); \\ \langle sh', f', l' \rangle & \text{otherwise.} \end{cases}$$

The next result states that the abstract mgu operator is a correct approximation of the concrete one.

Theorem 27
Let $d \in \mathit{SFL}$ and $(x \mapsto t) \in \mathit{Bind}$, where $\{x\} \cup \mathrm{vars}(t) \subseteq \mathit{VI}$. Then, for all $\sigma \in \gamma_{_S}(d)$ and $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$ in the syntactic equality theory T, we have $\tau \in \gamma_{_S}\big(\mathrm{amgu}_{_S}(d, x \mapsto t)\big)$.

We now highlight the similarities and differences of the operator $\mathrm{amgu}_{_S}$ with respect to the corresponding ones defined in the "classical" proposals for the integration of set-sharing with freeness and linearity, such as (Bruynooghe et al. 1994a; Bruynooghe et al. 1995; Hans and Winkler 1992; Langen 1990). Note that, when comparing our domain with the proposal in (Bruynooghe et al. 1994a), we deliberately ignore all those enhancements that depend on properties that cannot be represented in *SFL* (i.e., compoundness and explicit structural information).

- In the computation of the set-sharing component, the main difference can be observed in the second, third and fourth cases of the definition of sh'': here we omit one of the star-unions even when the terms x and t possibly share. In contrast, in (Bruynooghe et al. 1994a; Hans and Winkler 1992; Langen 1990) the corresponding star-union is avoided only when $\mathrm{ind}_d(x, t)$ holds. Note that when $\mathrm{ind}_d(x, t)$ holds in the second case of sh'', then we have $sh_{xt} = \varnothing$; thus, the whole computation for this case reduces to $sh'' = \mathrm{bin}(sh_x, sh_t)$, as was the case in the previous proposals.
- Another improvement on the set-sharing component can be observed in the definition of sh': the cyclic_x^t operator allows the set-sharing description to be further enhanced when dealing with *explicitly cyclic bindings*, i.e., when $x \in \mathrm{vars}(t)$. This is the rewording of a similar enhancement proposed in (Bagnara 1997) for the domain *Pos* in the context of groundness analysis. Its net effect is to recover some groundness and sharing dependencies that would have been unnecessarily lost when using the standard operators. When $x \notin \mathrm{vars}(t)$, we have $\mathrm{cyclic}_x^t(sh_{-} \cup sh'') = sh_{-} \cup sh''$.
- The computation of the freeness component f' is the same as specified in (Bruynooghe et al. 1994a; Hans and Winkler 1992), and is more precise than the one defined in (Langen 1990).
- The computation of the linearity component l' is the same as specified in (Bruynooghe et al. 1994a), and is more precise than those defined in (Hans and Winkler 1992; Langen 1990).

In the following examples we show that the improvements in the abstract computation of the sharing component allow, in particular cases, to derive better information than that obtainable by using the classical abstract unification operators.

Example 28
Let $VI = \{x, x_1, x_2, y, y_1, y_2, z\}$ and $\sigma \in \mathit{RSubst}$ such that

$$\sigma \stackrel{\text{def}}{=} \big\{x \mapsto f(x_1, x_2, z), y \mapsto f(y_1, z, y_2)\big\}.$$

By Definition 21, we have $d \stackrel{\text{def}}{=} \alpha_s\big(\{\sigma\}\big) = \langle sh, f, l\rangle$, where

$$sh = \{xx_1, xx_2, xyz, yy_1, yy_2\}, \qquad f = VI \setminus \{x, y\}, \qquad l = VI.$$

Consider the binding $(x \mapsto y) \in \mathit{Bind}$. In the concrete domain, we compute (a substitution equivalent to) $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = y\}\big)$, where

$$\tau = \big\{x \mapsto f(y_1, y_2, y_2), y \mapsto f(y_1, y_2, y_2), x_1 \mapsto y_1, x_2 \mapsto y_2, z \mapsto y_2\big\}.$$

Note that $\alpha_s\big(\{\tau\}\big) = \langle sh_\tau, f_\tau, l_\tau\rangle$, where $sh_\tau = \{xx_1yy_1, xx_2yy_2z\}$, so that the pairs of variables $P_x = \{x_1, x_2\}$ and $P_y = \{y_1, y_2\}$ keep their independence.

When evaluating the sharing component of $\mathrm{amgu}_s(d, x \mapsto y)$, using the notation of Definition 26, we have

$$sh_x = \{xx_1, xx_2, xyz\}, \qquad sh_t = \{xyz, yy_1, yy_2\},$$
$$sh_{xt} = \{xyz\}, \qquad sh__ = \varnothing.$$

Since both $\mathrm{lin}_d(x)$ and $\mathrm{lin}_d(y)$ hold, we apply the second case of the definition of sh'' so that

$$\begin{aligned}
sh_x \cup \mathrm{bin}(sh_x, sh^\star_{xt}) &= \{xx_1, xx_1yz, xx_2, xx_2yz, xyz\},\\
sh_t \cup \mathrm{bin}(sh_t, sh^\star_{xt}) &= \{xyy_1z, xyy_2z, xyz, yy_1, yy_2\},\\
sh'' &= \mathrm{bin}\big(sh_x \cup \mathrm{bin}(sh_x, sh^\star_{xt}), sh_t \cup \mathrm{bin}(sh_t, sh^\star_{xt})\big)\\
&= \{xx_1yy_1, xx_1yy_1z, xx_1yy_2, xx_1yy_2z, xx_1yz,\\
&\quad xx_2yy_1, xx_2yy_1z, xx_2yy_2, xx_2yy_2z, xx_2yz,\\
&\quad xyy_1z, xyy_2z, xyz\}.
\end{aligned}$$

Finally, as the binding is not cyclic, we obtain $sh' = sh''$. Thus amgu_s captures the fact that pairs P_x and P_y keep their independence.

In contrast, since $\mathrm{ind}_d(x, y)$ does not hold, all of the classical definitions of abstract unification would have required the star-closure of both sh_x and sh_t, resulting in an abstract element including, among the others, the sharing group $S = \{x, x_1, x_2, y, y_1, y_2\}$. Since $P_x \cup P_y \subset S$, this independence information would have been unnecessarily lost.

Similar examples can be devised for the third and fourth cases of the definition of sh'', where only one side of the binding is known to be linear. The next example shows the precision improvements arising from the use of the cyclic^t_x operator.

Example 29
Let $VI = \{x, x_1, x_2, y\}$ and $\sigma \stackrel{\text{def}}{=} \big\{x \mapsto f(x_1, x_2)\big\}$. By Definition 21, we have $d \stackrel{\text{def}}{=} \alpha_s\big(\{\sigma\}\big) = \langle sh, f, l\rangle$, where

$$sh = \{xx_1, xx_2, y\}, \qquad f = VI \setminus \{x\}, \qquad l = VI.$$

Let $t = f(x, y)$ and consider the cyclic binding $(x \mapsto t) \in \mathit{Bind}$. In the concrete domain, we compute (a substitution equivalent to) $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$, where

$$\tau = \big\{x \mapsto f(x_1, x_2), x_1 \mapsto f(x_1, x_2), y \mapsto x_2, \big\}.$$

Note that if we further instantiate τ by grounding y, then variables x, x_1 and x_2 would become ground too. Formally we have $\alpha_s\big(\{\tau\}\big) = \langle sh_\tau, f_\tau, l_\tau\rangle$, where $sh_\tau = \{xx_1x_2y\}$. Thus, as observed above, y covers x, x_1 and x_2. When abstractly evaluating the binding, we compute

$$\begin{aligned} sh_x &= \{xx_1, xx_2\}, & sh_t &= \{xx_1, xx_2, y\},\\ sh_{xt} &= sh_x, & sh__ &= \varnothing. \end{aligned}$$

Since both $\mathrm{lin}_d(x)$ and $\mathrm{lin}_d(t)$ hold, we apply the second case of the definition of sh'', so that

$$\begin{aligned} sh_x \cup \mathrm{bin}(sh_x, sh^\star_{xt}) &= sh^\star_x = \{xx_1, xx_1x_2, xx_2\},\\ sh_t \cup \mathrm{bin}(sh_t, sh^\star_{xt}) &= \{xx_1, xx_1x_2, xx_1x_2y, xx_1y, xx_2, xx_2y, y\},\\ sh'' &= \mathrm{bin}\big(sh_x \cup \mathrm{bin}(sh_x, sh^\star_{xt}), sh_t \cup \mathrm{bin}(sh_t, sh^\star_{xt})\big)\\ &= \{xx_1, xx_1x_2, xx_1x_2y, xx_1y, xx_2, xx_2y\}. \end{aligned}$$

Thus, as $x \in \mathrm{vars}(t)$, we obtain

$$\begin{aligned} sh' &= \mathrm{cyclic}^t_x(sh__ \cup sh'')\\ &= \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), sh''\big) \cup \mathrm{rel}\big(\mathrm{vars}(t) \setminus \{x\}, sh''\big)\\ &= \varnothing \cup \mathrm{rel}\big(\{y\}, sh''\big)\\ &= \{xx_1x_2y, xx_1y, xx_2y\}. \end{aligned}$$

Note that, in the element $sh__ \cup sh'' = sh''$ (which is the abstract element that would have been computed when not exploiting the cyclic^t_x operator) variable y covers none of variables x, x_1 and x_2. Thus, by applying the cyclic^t_x operator, this covering information is restored.

The full abstract unification operator aunify_s, capturing the effect of a sequence of bindings on an abstract element, can now be specified by a straightforward inductive definition using the operator amgu_s.

Definition 30
(aunify_s.) The operator $\mathrm{aunify}_s\colon \mathit{SFL} \times \mathit{Bind}^* \to \mathit{SFL}$ is defined, for each $d \in \mathit{SFL}$ and each sequence of bindings $bs \in \mathit{Bind}^*$, by

$$\mathrm{aunify}_s(d, bs) \stackrel{\mathrm{def}}{=} \begin{cases} d, & \text{if } bs = \epsilon;\\ \mathrm{aunify}_s\big(\mathrm{amgu}_s(d, x \mapsto t), bs'\big), & \text{if } bs = (x \mapsto t)\,.\,bs'. \end{cases}$$

Note that the second argument of aunify_s is a *sequence* of bindings (i.e., it is not a substitution, which is a *set* of bindings), because amgu_s is neither commutative nor idempotent, so that the multiplicity and the actual order of application of the bindings can influence the overall result of the abstract computation. The correctness of the aunify_s operator is simply inherited from the correctness of the underlying

amgu$_s$ operator. In particular, any reordering of the bindings in the sequence bs still results in a correct implementation of aunify$_s$.

The 'merge-over-all-path' operator on the domain *SFL* is provided by alub$_s$ and is correct by definition. Finally, we define the abstract existential quantification operator for the domain *SFL*, whose correctness does not pose any problem.

Definition 31
(aexists$_s$.) The function aexists$_s$: $\mathit{SFL} \times \wp_{\mathrm{f}}(\mathit{VI}) \to \mathit{SFL}$ provides the *abstract existential quantification* of an element with respect to a subset of the variables of interest. For each $d \stackrel{\text{def}}{=} \langle sh, f, l\rangle \in \mathit{SFL}$ and $V \subseteq \mathit{VI}$,

$$\text{aexists}_s\big(\langle sh, f, l\rangle, V\big) \stackrel{\text{def}}{=} \big\langle \text{aexists}(sh, V), f \cup V, l \cup V\big\rangle.$$

The intuition behind the definition of the abstract operator aexists$_s$ is the following. As explained in Section 2, any substitution $\sigma \in \mathit{RSubst}$ can be interpreted, under the given equality theory T, as a first-order logical formula; thus, for each set of variables V, it is possible to consider the (concrete) existential quantification $\exists V \,.\, \sigma$. The goal of the abstract operator aexists$_s$ is to provide a correct approximation of such a quantification starting from any correct approximation for σ.

Example 32
Let $\mathit{VI} = \{x, y, z\}$ and $\sigma = \{x \mapsto f(v_1, v_2), y \mapsto g(v_2, v_3), z \mapsto f(v_1, v_1)\}$, so that, by Definition 21,

$$d = \alpha_s\big(\{\sigma\}\big) = \big\langle\{xy, xz, y\}, \varnothing, \{x, y\}\big\rangle.$$

Let $V = \{y, z\}$ and consider the concrete element corresponding to the logical formula $\exists V \,.\, \sigma$. Note that $T \vdash \forall(\tau \leftrightarrow \exists V \,.\, \sigma)$, where $\tau = \{x \mapsto f(v_1, v_2)\}$. By applying Definition 31, we obtain

$$\text{aexists}_s(d, V) = \big\langle\{x, y, z\}, \{y, z\}, \{x, y, z\}\big\rangle = \alpha_s\big(\{\tau\}\big).$$

It is worth stressing that such an operator does not affect the set VI of the variables of interest. In particular, the abstract element aexists$_s(d, V)$ still has to provide correct information about variables y and z. Intuitively, since all the occurrences of y and z in $\exists V \,.\, \sigma$ are bound by the existential quantifier, the two variables of interest are un-aliased, free and linear.

Note that an abstract *projection* operator, i.e., an operator that actually modifies the set of variables of interest, is easily specified by composing the operator aexists$_s$ with an operator that simply removes, from all the components of *SFL and* from the set of variables of interest VI, those variables that have to be projected out.

4 A Formal Comparison Between *SFL* and ASub

As we have already observed, Example 28 shows that the abstract domain *SFL*, when equipped with the abstract mgu operator introduced in Section 3.2, can yield results that are strictly more precise than all the classical combinations of set-sharing with freeness and linearity information. In this section we show that the same example has another interesting, unexpected consequence, since it can be used

to formally prove that all the classical combinations of set-sharing with freeness and linearity, including those presented in (Bagnara et al. 2000; Bruynooghe et al. 1994a; Hans and Winkler 1992; Langen 1990), are not *uniformly* more precise than the abstract domain ASub (Søndergaard 1986), which is based on pair-sharing.

To formalize the above observation, we now introduce the ASub domain and the corresponding abstract semantics operators as specified in (Codish et al. 1991). The elements of the abstract domain ASub have two components: the first one is a set of variables that are known to be definitely ground; the second one encodes both possible pair-sharing and possible non-linearity into a single relation defined on the set of variables. Intuitively, when $x \neq y$ and $(x, y) \in VI^2$ occurs in the second component, then x and y may share a variable; when $(x, x) \in VI^2$ occurs in the second component, then x may be non-linear. The second component always encodes a symmetric relation; thus, for notational convenience and without any loss of generality (King 2000), we will represent each pair (x, y) in such a relation as the sharing group $S = \{x, y\}$, which will have cardinality 1 or 2 depending on whether $x = y$ or not, respectively.

Definition 33
(The domain $\mathsf{ASub}_\perp$.) The abstract domain $\mathsf{ASub}_\perp$ is defined as $\mathsf{ASub}_\perp \stackrel{\text{def}}{=} \{\bot_{\mathsf{ASub}}\} \cup \mathsf{ASub}$, where

$$\mathsf{ASub} \stackrel{\text{def}}{=} \left\{ \langle G, R\rangle \in \wp(VI) \times SH \;\middle|\; \begin{array}{l} G \cap \mathrm{vars}(R) = \varnothing, \\ \forall S \in R : 1 \leq \#S \leq 2 \end{array} \right\}.$$

For $i \in \{1, 2\}$, let $\kappa_i = \langle G_i, R_i\rangle \in \mathsf{ASub}$. Then

$$\kappa_1 \preceq_{\mathsf{ASub}} \kappa_2 \quad \stackrel{\text{def}}{\iff} \quad G_1 \supseteq G_2 \wedge R_1 \subseteq R_2.$$

The partial order $\preceq_{\mathsf{ASub}}$ is extended on $\mathsf{ASub}_\perp$ by letting $\bot_{\mathsf{ASub}}$ be the bottom element.

Let $u, v \in VI$ and $\kappa = \langle G, R\rangle \in \mathsf{ASub}$. Then $u \stackrel{\kappa}{\longleftrightarrow} v$ is a shorthand for the condition $\{u, v\} \in R$, whereas $u \stackrel{\kappa}{\iff} v$ is a shorthand for $u = v \vee \{u, v\} \in R$.

It is well-known that the domain $\mathsf{ASub}_\perp$ can be obtained by a further abstraction of any domain such as *SFL* that is based on set-sharing and enhanced with linearity information. The following definition formalizes this abstraction.

Definition 34
($\alpha_{\mathsf{ASub}} \colon SFL \to \mathsf{ASub}_\perp$.) Let $d = \langle sh, f, l\rangle \in SFL$. Then

$$\alpha_{\mathsf{ASub}}(d) \stackrel{\text{def}}{=} \begin{cases} \bot_{\mathsf{ASub}}, & \text{if } d = \bot_S; \\ \langle G, R\rangle, & \text{otherwise;} \end{cases}$$

where

$$\begin{aligned} G &\stackrel{\text{def}}{=} \{\, x \in VI \mid x \notin \mathrm{vars}(sh) \,\}, \\ R &\stackrel{\text{def}}{=} \{\, \{x\} \subseteq VI \mid x \in \mathrm{vars}(sh) \wedge x \notin l \,\} \\ &\qquad \cup \{\, \{x, y\} \subseteq VI \mid x \neq y \wedge \exists S \in sh \,.\, \{x, y\} \subseteq S \,\}. \end{aligned}$$

The definition of abstract unification in (Codish et al. 1991) is based on a few auxiliary operators. The first of these introduces the concept of abstract multiplicity for a term under a given abstract substitution, therefore modeling the notion of definite groundness and definite linearity.

Definition 35

(Abstract multiplicity.) Let $\kappa = \langle G, R\rangle \in \mathsf{ASub}$ and let $t \in \mathit{HTerms}$ be a term such that $\mathrm{vars}(t) \subseteq \mathit{VI}$. We say that $y \in \mathrm{vars}(t)$ *occurs linearly (in t) in κ* if and only if $\mathrm{occ_lin}_\kappa \colon \mathit{VI} \times \mathit{HTerms} \to \mathit{Bool}$ holds for (y, t), where

$$\mathrm{occ_lin}_\kappa(y,t) \stackrel{\text{def}}{=} y \in G \vee \big(\mathrm{occ_lin}(y,t) \wedge \forall z \in \mathrm{vars}(t) : \{y,z\} \notin R\big).$$

We say that t *has abstract multiplicity m in κ* if and only if $\chi_\kappa(t) = m$, where $\chi_\kappa \colon \mathit{HTerms} \to \{0,1,2\}$ is defined as follows

$$\chi_\kappa(t) \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if } \mathrm{vars}(t) \subseteq G; \\ 1, & \text{if } \forall y \in \mathrm{vars}(t) : \mathrm{occ_lin}_\kappa(y,t); \\ 2, & \text{otherwise.} \end{cases}$$

For any binding $x \mapsto t$, the function $\chi_\kappa \colon \mathit{Bind} \to \{0\} \cup \{1,2\}^2$ is defined as follows

$$\chi_\kappa(x \mapsto t) \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if } \chi_\kappa(x) = 0 \text{ or } \chi_\kappa(t) = 0; \\ \big(\chi_\kappa(x), \chi_\kappa(t)\big), & \text{otherwise;} \end{cases}$$

It is worth noting that, modulo a few insignificant differences in notation, the multiplicity operator χ_κ defined above corresponds to the abstract multiplicity operator $\chi^{\mathcal{A}}$, which was introduced in (Codish et al. 1991, Definition 3.4) and provided with an executable specification in (King 2000, Definition 4.3). Similarly, the next definition corresponds to (Codish et al. 1991, Definition 4.3).

Definition 36

(Sharing caused by an abstract equation.) For each $\kappa \in \mathsf{ASub}$ and $(x \mapsto t) \in \mathit{Bind}$, where $V_x = \{x\}$ and $V_t = \mathrm{vars}(t)$ are such that $V_x \cup V_t \subseteq \mathit{VI}$, the function $\mathrm{soln}\colon \mathsf{ASub} \times \mathit{Bind} \to \mathsf{ASub}$ is defined as follows

$$\mathrm{soln}(\kappa, x \mapsto t) \stackrel{\text{def}}{=} \begin{cases} \langle V_x \cup V_t, \varnothing\rangle, & \text{if } \chi_\kappa(x \mapsto t) = 0; \\ \big\langle \varnothing, \mathrm{bin}(V_x, V_t)\big\rangle, & \text{if } \chi_\kappa(x \mapsto t) = (1,1); \\ \big\langle \varnothing, \mathrm{bin}(V_x, V_x \cup V_t)\big\rangle, & \text{if } \chi_\kappa(x \mapsto t) = (1,2); \\ \big\langle \varnothing, \mathrm{bin}(V_x \cup V_t, V_t)\big\rangle, & \text{if } \chi_\kappa(x \mapsto t) = (2,1); \\ \big\langle \varnothing, \mathrm{bin}(V_x \cup V_t, V_x \cup V_t)\big\rangle, & \text{if } \chi_\kappa(x \mapsto t) = (2,2); \end{cases}$$

where the function $\mathrm{bin}\colon \wp(\mathit{VI})^2 \to \mathit{SH}$, for each $V, W \subseteq \mathit{VI}$, is defined as follows

$$\mathrm{bin}(V,W) \stackrel{\text{def}}{=} \big\{\, \{v,w\} \subseteq \mathit{VI} \;\big|\; v \in V, w \in W \,\big\}.$$

The next definition corresponds to (Codish et al. 1991, Definition 4.5).

Definition 37

(Abstract composition.) Let $\kappa, \kappa' \in \mathsf{ASub}$, where $\kappa = \langle G, R\rangle$ and $\kappa' = \langle G', R'\rangle$. Then $\kappa \circ \kappa' \stackrel{\text{def}}{=} \langle G'', R''\rangle$, where

$$G'' \stackrel{\text{def}}{=} G \cup G',$$

$$R'' \stackrel{\text{def}}{=} \left\{ \{u, v\} \in SH \;\middle|\; \begin{array}{l} \{u, v\} \cap G'' = \varnothing, \\ (u \stackrel{\kappa}{\longleftrightarrow} v) \vee \left(\exists x, y \,.\, u \stackrel{\kappa}{\Longleftrightarrow} x \stackrel{\kappa'}{\longleftrightarrow} y \stackrel{\kappa}{\Longleftrightarrow} v\right) \end{array} \right\}.$$

We are now ready to define the abstract mgu operator for the domain $\mathsf{ASub}_\perp$. This operator can be viewed as a specialization of (Codish et al. 1991, Definition 4.6) for the case when we have to abstract a single binding.

Definition 38
(Abstract mgu for $\mathsf{ASub}_\perp$.) Let $\kappa \in \mathsf{ASub}_\perp$ and $(x \mapsto t) \in \mathit{Bind}$, where $\{x\} \cup \mathrm{vars}(t) \subseteq \mathit{VI}$. Then

$$\mathrm{amgu}_{\mathsf{ASub}}(\kappa, x \mapsto t) \stackrel{\text{def}}{=} \begin{cases} \perp_{\mathsf{ASub}}, & \text{if } \kappa = \perp_{\mathsf{ASub}}; \\ \kappa \circ \mathrm{soln}(\kappa, x \mapsto t), & \text{otherwise;} \end{cases}$$

By repeating the abstract computation of Example 28 on the domain ASub, we provide a formal proof that all the classical approaches based on set-sharing are not uniformly more precise than the pair-sharing domain ASub.

Example 39
Consider the substitutions $\sigma, \tau \in \mathit{RSubst}$ and the abstract element $d \in \mathit{SFL}$ as introduced in Example 28.

By Definition 34, we obtain $\kappa = \alpha_{\mathsf{ASub}}(d) = \langle \varnothing, R\rangle$, where

$$R = \{xx_1, xx_2, xy, xz, yy_1, yy_2, yz\},$$

When abstractly evaluating the binding $x \mapsto y$ according to Definition 38, we compute the following:

$$\begin{aligned} \chi_\kappa(x \mapsto y) &= (1, 1), \\ \mathrm{soln}(\kappa, x \mapsto y) &= \langle \varnothing, \{xy\}\rangle, \\ \mathrm{amgu}_{\mathsf{ASub}}(\kappa, x \mapsto y) &= \kappa \circ \mathrm{soln}(\kappa, x \mapsto y) = \langle \varnothing, R''\rangle, \end{aligned}$$

where

$$R'' = R \cup \{x, xy_1, xy_2, x_1y, x_1y_1, x_1y_2, x_1z, x_2y, x_2y_1, x_2y_2, x_2z, y, y_1z, y_2z, z\}.$$

Note that $\{x_1, x_2\} \notin R''$ and $\{y_1, y_2\} \notin R''$, so that these pairs of variables keep their independence. In contrast, as observed in Example 28, the operators in (Bagnara et al. 2000; Bruynooghe et al. 1994a; Hans and Winkler 1992; Langen 1990) will fail to preserve the independence of these pairs.

We now show that the abstract domain *SFL*, when equipped with the operators introduced in Section 3.2, is uniformly more precise than the domain ASub. In particular, the following theorem states that the abstract operator $\mathrm{amgu}_{\mathit{S}}$ of Definition 26 is uniformly more precise than the abstract operator $\mathrm{amgu}_{\mathsf{ASub}}$.

Theorem 40
Let $d \in \mathit{SFL}$ and $\kappa \in \mathsf{ASub}_\bot$ be such that $\alpha_{\mathsf{ASub}}(d) \preceq_{\mathsf{ASub}} \kappa$. Let also $(x \mapsto t) \in \mathit{Bind}$, where $\{x\} \cup \mathrm{vars}(t) \subseteq \mathit{VI}$. Then

$$\alpha_{\mathsf{ASub}}\big(\mathrm{amgu}_{\mathit{S}}(d, x \mapsto t)\big) \preceq_{\mathsf{ASub}} \mathrm{amgu}_{\mathsf{ASub}}(\kappa, x \mapsto t).$$

Similar results can be stated for the other abstract operators, such as the abstract existential quantification $\mathrm{aexists}_{\mathit{S}}$ and the merge-over-all-path operator $\mathrm{alub}_{\mathit{S}}$. It is worth stressing that, when sequences of bindings come into play, the specification provided in (Codish et al. 1991, Definition 4.7) requires that the *grounding* bindings (i.e., those bindings such that $\chi_\kappa(x \mapsto t) = 0$) are evaluated before the non-grounding ones. Clearly, if we want to lift the result of Theorem 40 so that it also applies to the operator $\mathrm{aunify}_{\mathit{S}}$, the same evaluation strategy has to be adopted when computing on the domain *SFL*; this improvement is well-known (Langen 1990, pp. 66-67) and already exploited in most implementations of sharing analysis (Bagnara et al. 2000).

5 SFL_2: Eliminating Redundancies

As done in (Bagnara et al. 2002; Zaffanella et al. 2002) for the plain set-sharing domain *SH*, even when considering the richer domain *SFL* it is natural to question whether it contains redundancies with respect to the computation of the observable properties.

It is worth stressing that the results presented in (Bagnara et al. 2002) and (Zaffanella et al. 2002) cannot be simply inherited by the new domain. The concept of "redundancy" depends on both the starting domain and the given observables: in the *SFL* domain both of these have changed. First of all, as can be seen by looking at the definition of $\mathrm{amgu}_{\mathit{S}}$, freeness and linearity positively interact in the computation of sharing information: *a priori* it is an open issue whether or not the "redundant" sharing groups can play a role in such an interaction. Secondly, since freeness and linearity information can be themselves usefully exploited in a number of applications of static analysis (e.g., in the optimized implementation of concrete unification or in occurs-check reduction), these properties have to be included in the observables.

We will now show that the domain *SFL* can be simplified by applying the same notion of redundancy as identified in (Bagnara et al. 2002). Namely, in the definition of *SFL* it is possible to replace the set-sharing component *SH* by *PSD* without affecting the precision on groundness, independence, freeness and linearity. In order to prove such a claim, we now formalize the new observable properties.

Definition 41
(The observables of *SFL*.) The (overloaded) *groundness* and *independence* observables $\rho_{\mathit{Con}}, \rho_{\mathit{PS}} \in \mathrm{uco}(\mathit{SFL})$ are defined, for each $\langle sh, f, l\rangle \in \mathit{SFL}$, by

$$\rho_{\mathit{Con}}\big(\langle sh, f, l\rangle\big) \stackrel{\text{def}}{=} \big\langle \rho_{\mathit{Con}}(sh), \varnothing, \varnothing\big\rangle,$$
$$\rho_{\mathit{PS}}\big(\langle sh, f, l\rangle\big) \stackrel{\text{def}}{=} \big\langle \rho_{\mathit{PS}}(sh), \varnothing, \varnothing\big\rangle;$$

the *freeness* and *linearity* observables $\rho_F, \rho_L \in \mathrm{uco}(SFL)$ are defined, for each $\langle sh, f, l\rangle \in SFL$, by

$$\rho_F\big(\langle sh, f, l\rangle\big) \stackrel{\mathrm{def}}{=} \langle SG, f, \varnothing\rangle,$$
$$\rho_L\big(\langle sh, f, l\rangle\big) \stackrel{\mathrm{def}}{=} \langle SG, \varnothing, l\rangle.$$

The overloading of ρ_{PSD} working on the domain SFL is the straightforward extension of the corresponding operator on SH: in particular, the freeness and linearity components are left untouched.

Definition 42
(Non-redundant *SFL*.) For each $\langle sh, f, l\rangle \in SFL$, the operator $\rho_{PSD} \in \mathrm{uco}(SFL)$ is defined by

$$\rho_{PSD}\big(\langle sh, f, l\rangle\big) \stackrel{\mathrm{def}}{=} \big\langle \rho_{PSD}(sh), f, l\big\rangle.$$

This operator induces the lattice $SFL_2 \stackrel{\mathrm{def}}{=} \rho_{PSD}(SFL)$.

As proved in (Zaffanella et al. 2002), we have that $\rho_{PSD} \sqsubseteq (\rho_{Con} \sqcap \rho_{PS})$; by the above definitions, it is also clear that $\rho_{PSD} \sqsubseteq (\rho_F \sqcap \rho_L)$; thus, ρ_{PSD} is more precise than the reduced product $(\rho_{Con} \sqcap \rho_{PS} \sqcap \rho_F \sqcap \rho_L)$. Informally, this means that the domain SFL_2 is able to *represent* all of our observable properties without precision losses.

The next theorem shows that ρ_{PSD} is a congruence with respect to the aunify_S, alub_S and $\mathrm{aexists}_S$ operators. This means that the domain SFL_2 is able to *propagate* the information on the observables as precisely as SFL, therefore providing a completeness result.

Theorem 43
Let $d_1, d_2 \in SFL$ be such that $\rho_{PSD}(d_1) = \rho_{PSD}(d_2)$. Then, for each sequence of bindings $bs \in Bind^*$, for each $d' \in SFL$ and $V \in \wp(VI)$,

$$\rho_{PSD}\big(\mathrm{aunify}_S(d_1, bs)\big) = \rho_{PSD}\big(\mathrm{aunify}_S(d_2, bs)\big),$$
$$\rho_{PSD}\big(\mathrm{alub}_S(d_1, d')\big) = \rho_{PSD}\big(\mathrm{alub}_S(d_2, d')\big),$$
$$\rho_{PSD}\big(\mathrm{aexists}_S(d_1, V)\big) = \rho_{PSD}\big(\mathrm{aexists}_S(d_2, V)\big).$$

Finally, by providing the minimality result, we show that the domain SFL_2 is indeed the generalized quotient (Cortesi et al. 1998; Giacobazzi et al. 1998) of SFL with respect to the reduced product $(\rho_{Con} \sqcap \rho_{PS} \sqcap \rho_F \sqcap \rho_L)$.

Theorem 44
For each $i \in \{1, 2\}$, let $d_i = \langle sh_i, f_i, l_i\rangle \in SFL$ be such that $\rho_{PSD}(d_1) \neq \rho_{PSD}(d_2)$. Then there exist a sequence of bindings $bs \in Bind^*$ and an observable property $\rho \in \{\rho_{Con}, \rho_{PS}, \rho_F, \rho_L\}$ such that

$$\rho\big(\mathrm{aunify}_S(d_1, bs)\big) \neq \rho\big(\mathrm{aunify}_S(d_2, bs)\big).$$

As far as the implementation is concerned, the results proved in (Bagnara et al. 2002) for the domain *PSD* can also be applied to SFL_2. In particular, in the definition of amgu_S every occurrence of the star-union operator can be safely replaced

by the self-bin-union operator. As a consequence, it is possible to provide an implementation where the time complexity of the amgu_s operator is bounded by a polynomial in the number of sharing groups of the set-sharing component.

The following result provides another optimization that can be applied when both terms x and t are definitely linear, but none of them is definitely free (i.e., when we compute sh'' by the second case stated in Definition 26).

Theorem 45
Let $sh \in SH$ and $(x \mapsto t) \in Bind$, where $\{x\} \cup \mathrm{vars}(t) \subseteq VI$. Let $sh_- \stackrel{\mathrm{def}}{=} \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), sh\big)$, $sh_x \stackrel{\mathrm{def}}{=} \mathrm{rel}\big(\{x\}, sh\big)$, $sh_t \stackrel{\mathrm{def}}{=} \mathrm{rel}\big(\mathrm{vars}(t), sh\big)$, $sh_{xt} \stackrel{\mathrm{def}}{=} sh_x \cap sh_t$, $sh_W \stackrel{\mathrm{def}}{=} \mathrm{rel}(W, sh)$, where $W = \mathrm{vars}(t) \setminus \{x\}$, and

$$sh^\diamond \stackrel{\mathrm{def}}{=} \mathrm{bin}\big(sh_x \cup \mathrm{bin}(sh_x, sh^\star_{xt}), sh_t \cup \mathrm{bin}(sh_t, sh^\star_{xt})\big).$$

Then it holds

$$\rho_{PSD}\big(\mathrm{cyclic}^t_x(sh_- \cup sh^\diamond)\big) = \begin{cases} \rho_{PSD}\big(sh_- \cup \mathrm{bin}(sh_x, sh_t)\big), & \text{if } x \notin \mathrm{vars}(t); \\ \rho_{PSD}\big(sh_- \cup \mathrm{bin}(sh^2_x, sh_W)\big), & \text{otherwise.} \end{cases}$$

Therefore, even when terms x and t possibly share (i.e., when $sh_{xt} \neq \varnothing$), by using SFL_2 we can avoid the expensive computation of at least one of the two inner binary unions in the expression for $sh^\diamond$.

6 Experimental Evaluation

Example 28 shows that an analysis based on the new abstract unification operator can be strictly more precise than one based on the classical proposal. However, that example is artificial and leaves open the question as to whether or not such a phenomenon actually happens during the analysis of real programs and, if so, how often. This was the motivation for the experimental evaluation we describe in this section. We consider the abstract domain $Pos \times SFL_2$ (Bagnara et al. 2001), where the non-redundant version SFL_2 of the domain SFL is further combined, as described in (Bagnara et al. 2001, Section 4), with the definite groundness information computed by Pos and compare the results using the (classical) abstract unification operator of (Bagnara et al. 2001, Definition 4) with the (new) operator amgu_s given in Definition 26. Taking this as a starting point, we experimentally evaluate eight variants of the analysis arising from all possible combinations of the following options:

1. the analysis can be goal independent or goal dependent;
2. the set-sharing component may or may not have widening enabled (Zaffanella et al. 1999);
3. the abstract domain may or may not be upgraded with structural information using the Pattern($\cdot$) operator (see (Bagnara et al. 2000) and (Bagnara et al. 2001, Section 5)).

The experiments have been conducted using the China analyzer (Bagnara 1997) on a GNU/Linux PC system. China is a data-flow analyzer for (constraint) logic

programs performing bottom-up analysis and deriving information on both call-patterns and success-patterns by means of program transformations and optimized fixpoint computation techniques. An abstract description is computed for the call- and success-patterns for each predicate defined in the program. The benchmark suite, which is composed of 372 logic programs of various sizes and complexity, can be considered representative.

The precision results for the goal independent comparisons are summarized in Table 1. For each benchmark, precision is measured by counting the number of independent pairs as well as the numbers of definitely ground, free and linear variables detected. For each variant of the analysis, these numbers are then compared by computing the relative precision improvements and expressing them using percentages. The benchmark suite is then partitioned into several precision equivalence classes and the cardinalities of these classes are shown in Table 1. For example, when considering a goal independent analysis without structural information and without widenings, the value 5 found at the intersection of the row labeled '$0 < p \leq 2$' with the column labeled 'I' should be read: "for five benchmarks there has been a (positive) increase in the number of independent pairs of variables which is less than or equal to two percent." Note that we only report on independence and linearity (in the columns labelled 'I' and 'L', respectively), because no differences have been observed for groundness and freeness. The precision class labeled 'unknown' identifies those benchmarks for which the analyses timed-out (the time-out threshold was fixed at 600 seconds). Hence, for goal independent analyses, a precision improvement affects from 1.6% to 3% of the benchmarks, depending on the considered variant.

When considering the goal dependent analyses, we obtain a single, small improvement, so that no comparison tables are included here: the improvement, affecting linearity information, can be observed when the abstract domain includes structural information.

With respect to differences in the efficiency, the introduction of the new abstract unification operator has no significant effect on the computation time: small differences (usually improvements) are observed on as many as 6% of the benchmarks for the goal independent analysis without structural information and without widenings; other combinations register even less differences.

We note that it is not surprising that the precision and efficiency improvements occur very rarely since the abstract unification operators behave the same except under very specific conditions: the two terms being unified must not only be definitely linear, but also possibly non-free and share a variable.

7 Related work

Sharing information has been shown to be important for finite-tree analysis (Bagnara et al. 2001; Bagnara et al. 2001). This aims at identifying those program variables that, at a particular program point, cannot be bound to an infinite rational tree (in other words, they are necessarily bound to acyclic terms). This novel analysis is irrelevant for those logic languages computing over a domain of finite

Goal	Without Widening				With Widening			
Independent	w/o SI		with SI		w/o SI		with SI	
Prec. class	I	L	I	L	I	L	I	L
$5 < p \leq 10$	—	2	—	2	—	2	—	2
$2 < p \leq 5$	—	—	—	—	—	—	—	1
$0 < p \leq 2$	5	5	9	6	6	6	12	8
same precision	357	355	337	338	366	364	360	361
unknown	10	10	26	26	—	—	—	—

Table 1. *Classical Pos* $\times$ *SFL*$_2$ *versus enhanced one: precision.*

trees, while having several applications for those (constraint) logic languages that are explicitly designed to compute over a domain including rational trees, such as Prolog II and its successors (Colmerauer 1982; Colmerauer 1990), SICStus Prolog (Swedish Institute of Computer Science, Programming Systems Group 1995), and Oz (Smolka and Treinen 1994). The analysis specified in (Bagnara et al. 2001) is based on a parametric abstract domain $H \times P$, where the H component (the Herbrand component) is a set of variables that are known to be bound to finite terms, while the parametric component P can be any domain capturing aliasing, groundness, freeness and linearity information that is useful to compute finite-tree information. An obvious choice for such a parameter is the domain combination *SFL*. It is worth noting that, in (Bagnara et al. 2001), the correctness of the finite-tree analysis is proved by *assuming* the correctness of the underlying analysis on the parameter P. Thus, thanks to the results shown in this paper, the proof for the domain $H \times SFL$ can now be considered complete.

Codish et al. (Codish et al. 2000) describe an algebraic approach to the sharing analysis of logic programs that is based on *set logic programs*. A set logic program is a logic program in which the terms are sets of variables and standard unification is replaced by a suitable unification for sets, called *ACI1-unification* (unification in the presence of an associative, commutative, and idempotent equality theory with a unit element). The authors show that the domain of *set-substitutions*, with a few modifications, can be used as an abstract domain for sharing analysis. They also provide an isomorphism between this domain and the set-sharing domain *SH* of Jacobs and Langen. The approach using set logic programs is also generalized to include linearity information, by suitably annotating the set-substitutions, and the authors formally state the optimality of the corresponding abstract unification operator *lin-mgu*$_{ACI1}$ (Lemma A.10 in the Appendix of (Codish et al. 2000)). However, this operator is very similar to the classical combinations of set-sharing with linearity (Bruynooghe et al. 1994a; Hans and Winkler 1992; Langen 1990): in particular, the precision improvements arising from this enhancement are only exploited when the two terms being unified are definitely independent. As we have seen in this paper, such a choice results in a sub-optimal abstract unification operator, so that the optimality result cannot hold. By looking at the proof of Lemma A.10 in (Codish et al. 2000), it can be seen that the case when the two terms possibly share a vari-

able is dealt with by referring to an example:[4] this one is supposed to show that all the possible sharing groups can be generated. However, even our improved operator correctly characterizes the given example, so that the proof is wrong. It should be stressed that the amgu_s operator presented in this paper, though remarkably precise, is not meant to subsume all of the proposals for an improved sharing analysis that appeared in the recent literature (for a thorough experimental evaluation of many of these proposals, the reader is referred to (Bagnara et al. 2000; Zaffanella 2001)). In particular, it is not difficult to show that our operator is not the optimal approximation of concrete unification.

In a very recent paper (Howe and King 2003), J. Howe and A. King consider the domain *SFL* and propose three optimizations to improve both the precision and the efficiency of the (classical) abstract unification operator. The first optimization is based on the same observation we have made in this paper, namely that the independence check between the two terms being unified is not necessary for ensuring the correctness of the analysis. However, the proposed enhancement does not fully exploit this observation, so that the resulting operator is strictly less precise than our amgu_s operator (even when the operator cyclic_x^t does not come into play). In fact, the first optimization of (Howe and King 2003) is not uniformly more precise than the classical proposals. The following example illustrates this point.

Example 46
Let $VI = \{x, y, z_1, z_2, z_3\}$, $(x \mapsto y) \in \mathit{Bind}$ and $d \stackrel{\text{def}}{=} \langle sh, \varnothing, VI \rangle$, where $sh = \{xz_1, xz_2, xz_3, yz_1, yz_2, yz_3\}$.

Since x and y are linear and independent, amgu_s as well as all the classical abstract unification operators will compute $d_1 = \big\langle sh_1, \varnothing, \{x, y\} \big\rangle$, where

$$sh_1 \stackrel{\text{def}}{=} \text{bin}(sh_x, sh_y) = \{xyz_1, xyz_1z_2, xyz_1z_3, xyz_2, xyz_2z_3, xyz_3\}.$$

In contrast, a computation based on (Howe and King 2003, Definition 3.2), results in the less precise abstract element $d_2 = \big\langle sh_2, \varnothing, \{x, y\} \big\rangle$, where

$$sh_2 \stackrel{\text{def}}{=} \text{bin}(sh_x^\star, sh_y) \cap \text{bin}(sh_x, sh_y^\star) = sh_1 \cup \{xyz_1z_2z_3\}.$$

The second optimization shown in (Howe and King 2003) is based on the enhanced combination of set-sharing and freeness information, which was originally proposed in (Filé 1994). In particular, the authors propose a slightly different precision enhancement, less powerful as far as precision is concerned, which however seems to be amenable for an efficient implementation. The third optimization in (Howe and King 2003) exploits the combination of the domain *SFL* with the groundness domain *Pos*.

8 Conclusion

In this paper we have introduced the abstract domain *SFL*, combining the set-sharing domain *SH* with freeness and linearity information. While the carrier of *SFL*

[4] The proof refers to Example 8, which however has nothing to do with the possibility that the two terms share; we believe that Example 2 was intended.

can be considered standard, we have provided the specification of a new abstract unification operator, showing examples where this operator achieves more precision than the classical proposals. The main contributions of this paper are the following:

- we have defined a precise abstraction function, mapping arbitrary substitutions in rational solved form into their *most precise* approximation on *SFL*;
- using this abstraction function, we have provided the mandatory proof of *correctness* for the new abstract unification operator, *for both finite-tree and rational-tree languages*;
- we have formally shown that the domain *SFL* is *uniformly* more precise than the domain ASub; we have also provided an example showing that all the classical approaches to the combinations of set-sharing with freeness and linearity fail to satisfy this property;
- we have shown that, in the definition of *SFL*, we can replace the set-sharing domain *SH* by its non-redundant version *PSD*. As a consequence, it is possible to implement an algorithm for abstract unification running in *polynomial time* and still obtain the same precision on all the considered observables, that is groundness, independence, freeness and linearity.

Acknowledgment

We recognize the hard work required to review technical papers such as this one and would like to express our real gratitude to the referees of *Theory and Practice of Logic Programming* for their critical reading and constructive suggestions for preparing this improved version.

References

Bagnara, R. 1997. Data-flow analysis for constraint logic-based languages. Ph.D. thesis, Dipartimento di Informatica, Università di Pisa, Pisa, Italy. Printed as Report TD-1/97.

Bagnara, R., Gori, R., Hill, P. M., and Zaffanella, E. 2001. Finite-tree analysis for constraint logic-based languages. In *Static Analysis: 8th International Symposium, SAS 2001*, P. Cousot, Ed. Lecture Notes in Computer Science, vol. 2126. Springer-Verlag, Berlin, Paris, France, 165–184.

Bagnara, R., Hill, P. M., and Zaffanella, E. 1997. Set-sharing is redundant for pair-sharing. In *Static Analysis: Proceedings of the 4th International Symposium*, P. Van Hentenryck, Ed. Lecture Notes in Computer Science, vol. 1302. Springer-Verlag, Berlin, Paris, France, 53–67.

Bagnara, R., Hill, P. M., and Zaffanella, E. 2000. Efficient structural information analysis for real CLP languages. In *Proceedings of the 7th International Conference on Logic for Programming and Automated Reasoning (LPAR 2000)*, M. Parigot and A. Voronkov, Eds. Lecture Notes in Artificial Intelligence, vol. 1955. Springer-Verlag, Berlin, Réunion Island, France, 189–206.

Bagnara, R., Hill, P. M., and Zaffanella, E. 2002. Set-sharing is redundant for pair-sharing. *Theoretical Computer Science 277*, 1-2, 3–46.

Bagnara, R., Zaffanella, E., Gori, R., and Hill, P. M. 2001. Boolean functions for finite-tree dependencies. In *Proceedings of the 8th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR 2001)*, R. Nieuwenhuis and

A. Voronkov, Eds. Lecture Notes in Artificial Intelligence, vol. 2250. Springer-Verlag, Berlin, Havana, Cuba, 579–594.

Bagnara, R., Zaffanella, E., and Hill, P. M. 2000. Enhanced sharing analysis techniques: A comprehensive evaluation. In *Proceedings of the 2nd International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming*, M. Gabbrielli and F. Pfenning, Eds. Association for Computing Machinery, Montreal, Canada, 103–114.

Bagnara, R., Zaffanella, E., and Hill, P. M. 2001. Enhanced sharing analysis techniques: A comprehensive evaluation. Submitted for publication. Available at `http://www.cs.unipr.it/~bagnara/`.

Bruynooghe, M. and Codish, M. 1993. Freeness, sharing, linearity and correctness — All at once. In *Static Analysis, Proceedings of the Third International Workshop*, P. Cousot, M. Falaschi, G. Filé, and A. Rauzy, Eds. Lecture Notes in Computer Science, vol. 724. Springer-Verlag, Berlin, Padova, Italy, 153–164. An extended version is available as Technical Report CW 179, Department of Computer Science, K.U. Leuven, September 1993.

Bruynooghe, M., Codish, M., and Mulkers, A. 1994a. Abstract unification for a composite domain deriving sharing and freeness properties of program variables. In *Verification and Analysis of Logic Languages, Proceedings of the W2 Post-Conference Workshop, International Conference on Logic Programming*, F. S. de Boer and M. Gabbrielli, Eds. Santa Margherita Ligure, Italy, 213–230.

Bruynooghe, M., Codish, M., and Mulkers, A. 1994b. A composite domain for freeness, sharing, and compoundness analysis of logic programs. Technical Report CW 196, Department of Computer Science, K.U. Leuven, Belgium. July.

Bruynooghe, M., Codish, M., and Mulkers, A. 1995. Abstracting unification: A key step in the design of logic program analyses. In *Computer Science Today: Recent Trends and Developments*, J. van Leeuwen, Ed. Lecture Notes in Computer Science, vol. 1000. Springer-Verlag, Berlin, 406–425.

Clark, K. L. 1978. Negation as failure. In *Logic and Databases*, H. Gallaire and J. Minker, Eds. Plenum Press, Toulouse, France, 293–322.

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

Codish, M., Dams, D., Filé, G., and Bruynooghe, M. 1996. On the design of a correct freeness analysis for logic programs. *Journal of Logic Programming 28,* 3, 181–206.

Codish, M., Dams, D., and Yardeni, E. 1991. Derivation and safety of an abstract unification algorithm for groundness and aliasing analysis. See Furukawa (1991), 79–93.

Codish, M., Lagoon, V., and Bueno, F. 2000. An algebraic approach to sharing analysis of logic programs. *Journal of Logic Programming 42,* 2, 111–149.

Codish, M., Mulkers, A., Bruynooghe, M., Garcìa de la Banda, M., and Hermenegildo, M. 1993. Improving abstract interpretations by combining domains. In *Proceedings of the ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation.* ACM Press, Copenhagen, Denmark, 194–205. Also available as Technical Report CW 162, Department of Computer Science, K.U. Leuven, December 1992.

Codish, M., Mulkers, A., Bruynooghe, M., Garcìa de la Banda, M., and Her-

MENEGILDO, M. 1995. Improving abstract interpretations by combining domains. *ACM Transactions on Programming Languages and Systems 17,* 1 (Jan.), 28–44.

COLMERAUER, A. 1982. Prolog and infinite trees. In *Logic Programming, APIC Studies in Data Processing*, K. L. Clark and S. Å. Tärnlund, Eds. Vol. 16. Academic Press, New York, 231–251.

COLMERAUER, A. 1984. Equations and inequations on finite and infinite trees. In *Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'84).* ICOT, Tokyo, Japan, 85–99.

COLMERAUER, A. 1990. An introduction to Prolog-III. *Communications of the ACM 33,* 7, 69–90.

CORTESI, A. AND FILÉ, G. 1999. Sharing is optimal. *Journal of Logic Programming 38,* 3, 371–386.

CORTESI, A., FILÉ, G., AND WINSBOROUGH, W. 1998. The quotient of an abstract interpretation for comparing static analyses. *Theoretical Computer Science 202,* 1&2, 163–192.

COUSOT, P. AND COUSOT, R. 1977. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the Fourth Annual ACM Symposium on Principles of Programming Languages.* ACM Press, New York, 238–252.

COUSOT, P. AND COUSOT, R. 1979. Systematic design of program analysis frameworks. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Programming Languages.* ACM Press, New York, 269–282.

FILÉ, G. 1994. Share × Free: Simple and correct. Tech. Rep. 15, Dipartimento di Matematica, Università di Padova. Dec.

FURUKAWA, K., Ed. 1991. *Logic Programming: Proceedings of the Eighth International Conference on Logic Programming.* MIT Press Series in Logic Programming. The MIT Press, Paris, France.

GIACOBAZZI, R., RANZATO, F., AND SCOZZARI, F. 1998. Complete abstract interpretations made constructive. In *Proceedings of 23rd International Symposium on Mathematical Foundations of Computer Science (MFCS'98)*, J. Gruska and J. Zlatuska, Eds. Lecture Notes in Computer Science, vol. 1450. Springer-Verlag, Berlin, 366–377.

HANS, W. AND WINKLER, S. 1992. Aliasing and groundness analysis of logic programs through abstract interpretation and its safety. Tech. Rep. 92–27, Technical University of Aachen (RWTH Aachen).

HILL, P. M., BAGNARA, R., AND ZAFFANELLA, E. 1998. The correctness of set-sharing. In *Static Analysis: Proceedings of the 5th International Symposium*, G. Levi, Ed. Lecture Notes in Computer Science, vol. 1503. Springer-Verlag, Berlin, Pisa, Italy, 99–114.

HILL, P. M., BAGNARA, R., AND ZAFFANELLA, E. 2002. Soundness, idempotence and commutativity of set-sharing. *Theory and Practice of Logic Programming 2,* 2, 155–201.

HILL, P. M., BAGNARA, R., AND ZAFFANELLA, E. 2003. On the analysis of set-sharing, freeness and linearity for finite and rational tree languages. Tech. Rep. 2003.08, School of Computing, University of Leeds. Available at `http://www.comp.leeds.ac.uk/research/pubs/reports.shtml`.

HOWE, J. M. AND KING, A. 2003. Three optimisations for sharing. *Theory and Practice of Logic Programming 3,* 2, 243–257.

INTRIGILA, B. AND ZILLI, M. V. 1996. A remark on infinite matching vs infinite unification. *Journal of Symbolic Computation 21,* 3, 2289–2292.

JACOBS, D. AND LANGEN, A. 1989. Accurate and efficient approximation of variable

aliasing in logic programs. In *Logic Programming: Proceedings of the North American Conference*, E. L. Lusk and R. A. Overbeek, Eds. MIT Press Series in Logic Programming. The MIT Press, Cleveland, Ohio, USA, 154–165.

Jaffar, J., Lassez, J.-L., and Maher, M. J. 1987. Prolog-II as an instance of the logic programming scheme. In *Formal Descriptions of Programming Concepts III*, M. Wirsing, Ed. North-Holland, Amsterdam, 275–299.

Keisu, T. 1994. 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. In *Proceedings of the Fifth European Symposium on Programming*, D. Sannella, Ed. Lecture Notes in Computer Science, vol. 788. Springer-Verlag, Berlin, Edinburgh, UK, 363–378.

King, A. 2000. Pair-sharing over rational trees. *Journal of Logic Programming 46,* 1–2, 139–155.

King, A. and Soper, P. 1994. Depth-k sharing and freeness. In *Logic Programming: Proceedings of the Eleventh International Conference on Logic Programming*, P. Van Hentenryck, Ed. MIT Press Series in Logic Programming. The MIT Press, Santa Margherita Ligure, Italy, 553–568.

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. In *Proceedings, Third Annual Symposium on Logic in Computer Science*. IEEE Computer Society, Edinburgh, Scotland, 348–357.

Muthukumar, K. and Hermenegildo, M. 1991. Combined determination of sharing and freeness of program variables through abstract interpretation. See Furukawa (1991), 49–63. An extended version appeared in (Muthukumar and Hermenegildo 1992).

Muthukumar, K. and Hermenegildo, M. 1992. Compile-time derivation of variable dependency using abstract interpretation. *Journal of Logic Programming 13,* 2&3, 315–347.

Smolka, G. and Treinen, R. 1994. Records for logic programming. *Journal of Logic Programming 18,* 3, 229–258.

Søndergaard, H. 1986. An application of abstract interpretation of logic programs: Occur check reduction. In *Proceedings of the 1986 European Symposium on Programming*, B. Robinet and R. Wilhelm, Eds. Lecture Notes in Computer Science, vol. 213. Springer-Verlag, Berlin, 327–338.

Swedish Institute of Computer Science, Programming Systems Group 1995. *SICStus Prolog User's Manual*, release 3 #0 ed. Swedish Institute of Computer Science, Programming Systems Group.

Zaffanella, E. 2001. Correctness, precision and efficiency in the sharing analysis of real logic languages. Ph.D. thesis, School of Computing, University of Leeds, Leeds, U.K. Available at `http://www.cs.unipr.it/~zaffanella/`.

Zaffanella, E., Bagnara, R., and Hill, P. M. 1999. Widening Sharing. In *Principles and Practice of Declarative Programming*, G. Nadathur, Ed. Lecture Notes in Computer Science, vol. 1702. Springer-Verlag, Berlin, Paris, France, 414–431.

Zaffanella, E., Hill, P. M., and Bagnara, R. 1999. Decomposing non-redundant sharing by complementation. In *Static Analysis: Proceedings of the 6th International Symposium*, A. Cortesi and G. Filé, Eds. Lecture Notes in Computer Science, vol. 1694. Springer-Verlag, Berlin, Venice, Italy, 69–84.

Zaffanella, E., Hill, P. M., and Bagnara, R. 2002. Decomposing non-redundant sharing by complementation. *Theory and Practice of Logic Programming 2,* 2, 233–261.

Appendix A Proofs of the Results of Section 2

The proof of Proposition 3 requires the concept of a *path* $p \in (\mathbb{N} \setminus \{0\})^*$ which is any finite sequence of (non-zero) natural numbers. Given a path p and a (possibly infinite) term $t \in \mathit{Terms}$, we denote by $t[p]$ the subterm of t found by following path p. Formally,

$$t[p] = \begin{cases} t & \text{if } p = \epsilon; \\ t_i[q] & \text{if } p = i \,.\, q \wedge (1 \leq i \leq n) \wedge t = f(t_1, \ldots, t_n). \end{cases}$$

Note that $t[p]$ is only defined for those paths p actually corresponding to subterms of t.

The following two lemmas, proved in (Hill et al. 2002), state some basic properties of equality theories.

Lemma 47
((Hill et al. 2002, Lemma 1).) Let $\sigma \in \mathit{RSubst}$ be satisfiable in the equality theory T and consider $(x \mapsto t) \in \mathit{Bind}$ such that $x \notin \mathrm{dom}(\sigma)$ and $t \in \mathit{GTerms} \cap \mathit{HTerms}$. Then, $\sigma' \stackrel{\mathrm{def}}{=} \sigma \cup \{x \mapsto t\} \in \mathit{RSubst}$ and σ' is satisfiable in T.

Lemma 48
((Hill et al. 2002, Lemma 2).) Let T be an equality theory, $\sigma \in \mathit{RSubst}$ and $t \in \mathit{HTerms}$. Then

$$T \vdash \forall\big(\sigma \rightarrow (t = t\sigma)\big).$$

Note that, by repeated applications of Lemma 48 and the congruence axioms, we also obtain that $T \vdash \forall\big(\sigma \rightarrow (t = t\sigma^i)\big)$, for all $i \geq 0$.

The next lemma provides a link between syntactic equality theories and the function 'rt'.

Lemma 49
Let $\sigma \in \mathit{RSubst}$ be satisfiable in the syntactic equality theory T. Suppose $s, t \in \mathit{HTerms}$ are such that $T \vdash \forall\big(\sigma \rightarrow (s = t)\big)$. Then $\mathrm{rt}(s, \sigma) = \mathrm{rt}(t, \sigma)$.

Proof
We suppose, toward a contradiction, that $\mathrm{rt}(s, \sigma) \neq \mathrm{rt}(t, \sigma)$. Then, there must exist a finite path p such that:

a. $x = \mathrm{rt}(s, \sigma)[p] \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$, $y = \mathrm{rt}(t, \sigma)[p] \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$ and $x \neq y$; or
b. $x = \mathrm{rt}(s, \sigma)[p] \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$ and $r = \mathrm{rt}(t, \sigma)[p] \notin \mathit{Vars}$ or, symmetrically, we have $r = \mathrm{rt}(s, \sigma)[p] \notin \mathit{Vars}$ and $x = \mathrm{rt}(t, \sigma)[p] \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$; or
c. $r_1 = \mathrm{rt}(s, \sigma)[p] \notin \mathit{Vars}$, $r_2 = \mathrm{rt}(t, \sigma)[p] \notin \mathit{Vars}$ and r_1 and r_2 have different principal functors.

Then, by definition of 'rt', there must exists an index $i \in \mathbb{N}$ such that one of these hold:

1. $x = s\sigma^i[p] \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$, $y = t\sigma^i[p] \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$ and $x \neq y$; or
2. $x = s\sigma^i[p] \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$ and $r = t\sigma^i[p] \notin \mathit{Vars}$ or, symmetrically, we have $r = s\sigma^i[p] \notin \mathit{Vars}$ and $x = t\sigma^i[p] \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$; or
3. $r_1 = s\sigma^i[p] \notin \mathit{Vars}$ and $r_2 = t\sigma^i[p] \notin \mathit{Vars}$ have different principal functors.

By Lemma 48, we have $T \vdash \forall\big(\sigma \to (s\sigma^i = t\sigma^i)\big)$; from this, by the identity axioms, we obtain that

$$T \vdash \forall\Big(\sigma \to \big(s\sigma^i[p] = t\sigma^i[p]\big)\Big). \tag{A1}$$

We now prove that each case leads to a contradiction.

Consider case 1. Let $r_1, r_2 \in \mathit{GTerms} \cap \mathit{HTerms}$ be two ground and finite terms having different principal functors, so that $T \vdash \forall(r_1 \neq r_2)$. By Lemma 47, we have that $\sigma' = \sigma \cup \{x \mapsto r_1, y \mapsto r_2\} \in \mathit{RSubst}$ is satisfiable; moreover, $T \vdash \forall(\sigma' \to \sigma)$, $T \vdash \forall\big(\sigma' \to (x = r_1)\big)$ and $T \vdash \forall\big(\sigma' \to (y = r_2)\big)$. This is a contradiction, since, by (A1), we have $T \vdash \forall\big(\sigma \to (x = y)\big)$.

Consider case 2. Without loss of generality, consider the first subcase, where $x = s\sigma^i$ and $r = t\sigma^i[p] \notin \mathit{Vars}$. Let $r' \in \mathit{GTerms} \cap \mathit{HTerms}$ be such that r and r' have different principal functors, so that $T \vdash \forall(r \neq r')$. By Lemma 47, $\sigma' = \sigma \cup \{x \mapsto r'\} \in \mathit{RSubst}$ is satisfiable; we also have $T \vdash \forall(\sigma' \to \sigma)$ and $T \vdash \forall\big(\sigma' \to (x = r')\big)$. This is a contradiction, since, by (A1), $T \vdash \forall\big(\sigma \to (x = r)\big)$.

Finally, consider case 3. In this case $T \vdash \forall(r_1 \neq r_2)$. This immediately leads to a contradiction, since, by (A1), $T \vdash \forall\big(\sigma \to (r_1 = r_2)\big)$. □

Proof of Proposition 3 on page 7 For each stated equivalence, we will prove only one implication since the other one will follow by symmetry.

Consider (9). Reasoning by contraposition, suppose $\mathrm{rt}(y, \sigma) \notin \mathit{Vars}$. Then there exists an index $i \geq 0$ such that $y\sigma^i \notin \mathit{Vars}$. Since $T \vdash \forall(\tau \to \sigma)$, by Lemma 48 we have $T \vdash \forall\big(\tau \to (y = y\sigma^i)\big)$. By Lemma 49, we obtain $\mathrm{rt}(y, \tau) = \mathrm{rt}(y\sigma^i, \tau)$, so that $\mathrm{rt}(y, \tau) \notin \mathit{Vars}$.

Consider (10). We suppose, toward a contradiction, that $\mathrm{rt}(y, \sigma) \in \mathit{GTerms}$ but $\mathrm{rt}(y, \tau) \notin \mathit{GTerms}$. Then, there must exist a finite path p such that:

a. $r = \mathrm{rt}(y, \sigma)[p] \in \mathit{GTerms}$ and $x = \mathrm{rt}(y, \tau)[p] \in \mathit{Vars} \setminus \mathrm{dom}(\tau)$; or
b. $r_1 = \mathrm{rt}(y, \sigma)[p] \notin \mathit{Vars}$, $r_2 = \mathrm{rt}(y, \tau)[p] \notin \mathit{Vars}$ and r_1 and r_2 have different principal functors.

Then, by definition of 'rt', there must exists an index $i \in \mathbb{N}$ such that one of these holds:

1. $r = y\sigma^i[p] \notin \mathit{Vars}$ and $x = y\tau^i[p] \in \mathit{Vars} \setminus \mathrm{dom}(\tau)$; or
2. $r_1 = y\sigma^i[p] \notin \mathit{Vars}$ and $r_2 = y\tau^i[p] \notin \mathit{Vars}$ have different principal functors.

By Lemma 48, we have $T \vdash \forall\big(\sigma \to (y\sigma^i = y\tau^i)\big)$; from this, by the identity axioms, we obtain that

$$T \vdash \forall\Big(\sigma \to \big(y\sigma^i[p] = y\tau^i[p]\big)\Big). \tag{A2}$$

We now prove that both cases lead to a contradiction.

Consider case 1. Let $r' \in \mathit{GTerms} \cap \mathit{HTerms}$ be such that r and r' have different principal functors, so that $T \vdash \forall(r \neq r')$. By Lemma 47, $\tau' = \tau \cup \{x \mapsto r'\} \in \mathit{RSubst}$ is satisfiable; we also have $T \vdash \forall(\tau' \rightarrow \tau)$ and $T \vdash \forall\big(\tau' \rightarrow (x = r')\big)$. This is a contradiction, since, by (A2), $T \vdash \forall\big(\tau \rightarrow (x = r)\big)$.

Finally, consider case 2. In this case $T \vdash \forall(r_1 \neq r_2)$. This immediately leads to a contradiction, since, by (A2), $T \vdash \forall\big(\sigma \rightarrow (r_1 = r_2)\big)$.

Consider (11). Reasoning by contraposition, suppose that $\mathrm{rt}(y,\tau) \notin \mathit{LTerms}$, so that there exists $v \in \mathrm{vars}\big(\mathrm{rt}(y,\tau)\big)$ such that $\mathrm{occ_lin}\big(v, \mathrm{rt}(y,\tau)\big)$ does not hold. By definition of 'rt', there exists an index $i \geq 0$ such that $v \in \mathrm{vars}(y\tau^i)$ and $\mathrm{occ_lin}(v, y\tau^i)$ does not hold. Thus, as $v \notin \mathrm{dom}(\tau)$, so that $\mathrm{rt}(v,\tau) = v \in \mathit{Vars}$. By (9), $\mathrm{rt}(v,\sigma) = w \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. Therefore there exists $j \geq 0$ such that $w = v\sigma^j$. Hence, we obtain that $w \in \mathrm{vars}\big(\mathrm{rt}(y\tau^i\sigma^j, \sigma)\big)$ and $\mathrm{occ_lin}\big(w, \mathrm{rt}(y\tau^i\sigma^j, \sigma)\big)$ does not hold, so that $\mathrm{rt}(y\tau^i\sigma^j, \sigma) \notin \mathit{LTerms}$. Since $T \vdash \forall(\sigma \rightarrow \tau)$, by Lemma 48 we have $T \vdash \forall\big(\sigma \rightarrow (y = y\tau^i\sigma^j)\big)$. By Lemma 49, we obtain $\mathrm{rt}(y,\sigma) = \mathrm{rt}(y\tau^i\sigma^j, \sigma)$, so that $\mathrm{rt}(y,\sigma) \notin \mathit{LTerms}$.

Appendix B Proofs of the Results of Subsection 3.1.

The definition of idempotence requires that repeated applications of a substitution do not change the syntactic structure of a term. However, several abstractions of terms, such as the ones commonly used for sharing analysis, are 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 variables in a term invariant.

Definition 50
(Variable-idempotence.) A substitution $\sigma \in \mathit{RSubst}$ is *variable-idempotent* if and only if for all $t \in \mathit{HTerms}$ we have

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

The set of variable-idempotent substitutions is denoted *VSubst*.

Note that idempotent substitutions are also variable-idempotent so that $\mathit{ISubst} \subset \mathit{VSubst} \subset \mathit{RSubst}$. Definition 50, which is the same as that originally provided in (Hill et al. 1998), is slightly stronger than the one adopted in (Hill et al. 2002) (*weak* variable-idempotence), which disregarded the domain variables of the substitution. Since variable-idempotent substitutions are weak variable-idempotent, many results in (Hill et al. 2002) also hold when weak variable-idempotence is replaced by variable-idempotence as given in Definition 50.[5]

[5] The adoption of this stronger definition allows for some simplifications in the correctness proofs for freeness and linearity.

Example 51
Consider the following substitutions which are all in *RSubst*.

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

The next result provides a sufficient condition for a variable-idempotent substitution so that all of its subsets are variable-idempotent too.

Lemma 52
Let $\sigma \in \mathit{VSubst}$ be such that for all $y \in \mathrm{range}(\sigma)$, $y \in \mathrm{vars}(y\sigma)$. Then, for all $\sigma' \subseteq \sigma$, $\sigma' \in \mathit{VSubst}$.

Proof
Suppose $t \in \mathit{HTerms}$. We need to show that $\mathrm{vars}(t\sigma') = \mathrm{vars}(t\sigma'\sigma')$.

To prove that $\mathrm{vars}(t\sigma') \subseteq \mathrm{vars}(t\sigma'\sigma')$, let $y \in \mathrm{vars}(t\sigma')$. If $y \notin \mathrm{range}(\sigma')$ then, as $y \in \mathrm{vars}(t\sigma')$, $y \notin \mathrm{dom}(\sigma')$ so that $y = y\sigma'$. If $y \in \mathrm{range}(\sigma')$ then, as $\sigma' \subseteq \sigma$, $y \in \mathrm{range}(\sigma)$; hence, by hypothesis, $y \in \mathrm{vars}(y\sigma)$; and hence, as $\sigma' \subseteq \sigma$, $y \in \mathrm{vars}(y\sigma')$. Thus, for all $y \in \mathrm{vars}(t\sigma')$, $y \in \mathrm{vars}(y\sigma')$ so that $y \in \mathrm{vars}(t\sigma'\sigma')$.

To prove that $\mathrm{vars}(t\sigma'\sigma') \subseteq \mathrm{vars}(t\sigma')$, let $y \in \mathrm{vars}(t\sigma'\sigma')$. Then there exists $v \in \mathrm{vars}(t\sigma')$ such that $y \in \mathrm{vars}(v\sigma')$ and $w \in \mathrm{vars}(t)$ such that $v \in \mathrm{vars}(w\sigma')$. If $y = v$ or $v = w$, then $y \in \mathrm{vars}(t\sigma')$. If $y \neq v$ and $v \neq w$, then $v, w \in \mathrm{dom}(\sigma')$. As $\sigma' \subseteq \sigma$, $v\sigma' = v\sigma$ and $w\sigma' = w\sigma$ so that $y \in \mathrm{vars}(w\sigma\sigma)$. As $\sigma \in \mathit{VSubst}$, $y \in \mathrm{vars}(w\sigma)$; so that, as $w\sigma' = w\sigma$, $y \in \mathrm{vars}(w\sigma')$. Therefore $y \in \mathrm{vars}(t\sigma')$. □

The following result concerns the composition of variable idempotent substitutions and shows that, under the right conditions, the composition of variable idempotent substitutions is also variable idempotent.

Lemma 53
If $\sigma, \tau \in \mathit{VSubst}$ and $\mathrm{dom}(\sigma) \cap \mathrm{vars}(\tau) = \varnothing$, then $\tau \circ \sigma \in \mathit{VSubst}$.

Proof
We will show that, for all terms $t \in \mathit{HTerms}$,

$$\mathrm{vars}(t\sigma\tau) = \mathrm{vars}(t\sigma\tau\sigma\tau).$$

To prove the inclusion $\mathrm{vars}(t\sigma\tau) \subseteq \mathrm{vars}(t\sigma\tau\sigma\tau)$, let $z \in \mathrm{vars}(t\sigma\tau)$. First note that, if $z \notin \mathrm{dom}(\sigma) \cup \mathrm{vars}(\tau)$, then the result is trivial.

Suppose $z \in \mathrm{dom}(\sigma)$. By hypothesis, $z \notin \mathrm{vars}(\tau)$ so that $z \in \mathrm{vars}(t\sigma)$. Since σ is variable-idempotent, $z \in \mathrm{vars}(t\sigma\sigma)$. Thus there exists $v \in \mathrm{vars}(t\sigma)$ such that $z \in \mathrm{vars}(v\sigma)$. As $z \in \mathrm{dom}(\sigma)$, $v \in \mathrm{dom}(\sigma)$. By hypothesis, $\mathrm{dom}(\sigma) \cap \mathrm{vars}(\tau) = \varnothing$ so that $v \notin \mathrm{vars}(\tau)$. Thus $v \in \mathrm{vars}(t\sigma\tau)$. Therefore $z \in \mathrm{vars}(t\sigma\tau\sigma)$ and, since $z \notin \mathrm{vars}(\tau)$, we can conclude $z \in \mathrm{vars}(t\sigma\tau\sigma\tau)$.

Otherwise, let $z \in \mathrm{vars}(\tau)$. As $z \in \mathrm{vars}(t\sigma\tau)$, there exists $v \in \mathrm{vars}(t\sigma)$ such that $z \in \mathrm{vars}(v\tau)$. Since τ is variable-idempotent, $z \in \mathrm{vars}(v\tau\tau)$ so that there exists

$w \in \mathrm{vars}(v\tau) \subseteq \mathrm{vars}(t\sigma\tau)$ such that $z \in \mathrm{vars}(w\tau)$. Since $z \in \mathrm{vars}(\tau)$, $w \in \mathrm{vars}(\tau)$. By hypothesis, $\mathrm{dom}(\sigma) \cap \mathrm{vars}(\tau) = \varnothing$ so that $w \notin \mathrm{dom}(\sigma)$. Thus $w \in \mathrm{vars}(t\sigma\tau\sigma)$. Therefore we can conclude $z \in \mathrm{vars}(t\sigma\tau\sigma\tau)$.

To prove the other inclusion, let $z \in \mathrm{vars}(t\sigma\tau\sigma\tau)$, so that there exists $v \in \mathrm{vars}(t\sigma\tau\sigma)$ such that $z \in \mathrm{vars}(v\tau)$. Similarly, there exists $w \in \mathrm{vars}(t\sigma\tau)$ such that $v \in \mathrm{vars}(w\sigma)$.

Suppose $v \neq w$. Then $w \in \mathrm{dom}(\sigma)$, so that, by hypothesis, $w \notin \mathrm{vars}(\tau)$. As a consequence, $w \in \mathrm{vars}(t\sigma)$ and hence, as $v \in \mathrm{vars}(w\sigma)$, $v \in \mathrm{vars}(t\sigma\sigma)$. Thus, as $\sigma \in \mathit{VSubst}$, $v \in \mathrm{vars}(t\sigma)$. Therefore, as $z \in \mathrm{vars}(v\tau)$, we obtain $z \in \mathrm{vars}(t\sigma\tau)$.

Otherwise, if $v = w$, then $z \in \mathrm{vars}(w\tau)$. As $w \in \mathrm{vars}(t\sigma\tau)$, $z \in \mathrm{vars}(t\sigma\tau\tau)$. Thus, as $\tau \in \mathit{VSubst}$, we obtain $z \in \mathrm{vars}(t\sigma\tau)$. □

The following result shows that there is no loss in generality in considering variable-idempotent substitutions only.

Proposition 54
Suppose T is an equality theory and $\sigma \in \mathit{RSubst}$. Then there exists $\sigma' \in \mathit{VSubst}$ such that $\mathrm{dom}(\sigma) = \mathrm{dom}(\sigma')$, $\mathrm{vars}(\sigma) = \mathrm{vars}(\sigma')$ and $T \vdash \forall(\sigma \leftrightarrow \sigma')$; also, for all $y \in \mathrm{range}(\sigma')$, $y \in \mathrm{vars}(y\sigma')$.

Proof
The proof is the same as the proofs of (Hill et al. 2002, Theorems 1 and 2) where weaker properties were stated. □

Note that the proof of the above proposition is constructive, i.e., it shows a systematic way of building such a $\sigma' \in \mathit{VSubst}$ starting from an arbitrary $\sigma \in \mathit{RSubst}$. For an example of this construction process, we refer the interested reader to (Hill et al. 2002, Example 6).

The next useful result applies to any substitution in rational solved form.

Lemma 55
If $\sigma \in \mathit{RSubst}$, $n = \#\sigma$ and $t \in \mathit{HTerms}$, then $t\sigma^n \notin \mathrm{dom}(\sigma)$.

Proof
Suppose $t\sigma^n \in \mathit{Vars}$. Then, $t\sigma^i \in \mathit{Vars}$ for all $i = 1, \ldots, n$, so that, as $\sigma \in \mathit{RSubst}$ has no circular subsets, we have $t\sigma^n \notin \mathrm{dom}(\sigma)$. □

For variable-idempotent substitutions, the following simplified characterizations for the operators occ, fvars, gvars and lvars can be used.

Proposition 56
For each $\sigma \in \mathit{VSubst}$ and $v \in \mathit{Vars}$, we have

$$\mathrm{occ}(\sigma, v) = \{\, y \in \mathit{Vars} \mid v \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) \,\}, \tag{B1}$$

$$\mathrm{gvars}(\sigma) = \{\, y \in \mathit{Vars} \mid \mathrm{vars}(y\sigma) \subseteq \mathrm{dom}(\sigma) \,\}, \tag{B2}$$

$$\mathrm{fvars}(\sigma) = \{\, y \in \mathit{Vars} \mid y\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma) \,\}, \tag{B3}$$

$$\mathrm{lvars}(\sigma) = \left\{ y \in \mathit{Vars} \;\middle|\; \begin{array}{l} \forall z \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) : \mathrm{occ_lin}(z, y\sigma), \\ \forall z \in \mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma) : z \in \mathrm{gvars}(\sigma) \end{array} \right\}. \tag{B4}$$

Proof
We prove each equation separately.

(B1). This has been proved in (Hill et al. 2002, Lemma 13).

(B2). By Definition 12, $y \in \mathrm{gvars}(\sigma)$ if and only if, for all $v \in \mathit{Vars}$, we have $y \notin \mathrm{occ}(\sigma, v)$. By equation (B1) proved above, this holds if and only if there does not exist $v \in \mathit{Vars}$ such that $v \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma)$, i.e., if and only if $\mathrm{vars}(y\sigma) \subseteq \mathrm{dom}(\sigma)$.

In order to prove the remaining equations, let $n = \#\sigma$.

(B3). By Definition 14, $y \in \mathrm{fvars}(\sigma)$ if and only if $y\sigma^n \in \mathit{Vars}$. Thus we need to show that $y\sigma^n \in \mathit{Vars}$ if and only if $y\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. If $y\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$, then $y\sigma = y\sigma^n \in \mathit{Vars}$. If $y\sigma^n \in \mathit{Vars}$, then $y\sigma \in \mathit{Vars}$ so that, as $\sigma \in \mathit{VSubst}$, $\{y\sigma\} = \{y\sigma^n\}$ and hence, by Lemma 55, $y\sigma = y\sigma^n \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$.

(B4). First, suppose that $y \in \mathrm{lvars}(\sigma)$. If $z \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma)$ then $z \in \mathrm{vars}(y\sigma^n) \setminus \mathrm{dom}(\sigma)$ and hence, by Definition 16, $\mathrm{occ_lin}(z, y\sigma^{2n})$ holds so that $\mathrm{occ_lin}(z, y\sigma)$ holds. Suppose next $z \in \mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma)$ so that $n \geq 1$ and, as $\sigma \in \mathit{VSubst}$, $z \in \mathrm{vars}(y\sigma^n)$. If there exists $v \in \mathrm{vars}(z\sigma) \setminus \mathrm{dom}(\sigma)$, then $\{v, z\} \subseteq \mathrm{vars}(y\sigma^n)$ so that $\mathrm{occ_lin}(v, y\sigma^{2n})$ does not hold, contradicting Definition 16. Thus $\mathrm{vars}(z\sigma) \subseteq \mathrm{dom}(\sigma)$ so that, by equation (B2) proved above, $z \in \mathrm{gvars}(\sigma)$.

Secondly, suppose that $y \notin \mathrm{lvars}(\sigma)$ so that $n \geq 1$. Then, by Definition 16, there exists $z \in \mathrm{vars}(y\sigma^n) \setminus \mathrm{dom}(\sigma)$ such that $\mathrm{occ_lin}(z, y\sigma^{2n})$ does not hold. Thus, as $\sigma \in \mathit{VSubst}$, $z \in \mathrm{vars}(y\sigma)$. Also, if $\mathrm{occ_lin}(z, y\sigma)$ holds, there must exist $v \in \mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma)$ and $z \in \mathrm{vars}(v\sigma^{2n-1})$. Thus, as $\sigma \in \mathit{VSubst}$, $z \in \mathrm{vars}(v\sigma)$ and $\mathrm{vars}(v\sigma) \setminus \mathrm{dom}(\sigma) \neq \varnothing$ and hence, by equation (B2), $z \notin \mathrm{gvars}(\sigma)$. □

The following proposition shows that, for a substitution $\sigma \in \mathit{VSubst}$, the occurrence, groundness, freeness and linearity operators precisely capture the intended properties.

Proposition 57
Let $\sigma \in \mathit{VSubst}$, $y \in \mathit{VI}$ and $v \in \mathit{Vars}$. Then:

$$y \in \mathrm{occ}(\sigma, v) \iff v \in \mathrm{vars}\big(\mathrm{rt}(y, \sigma)\big), \tag{B5}$$

$$y \in \mathrm{gvars}(\sigma) \iff \mathrm{rt}(y, \sigma) \in \mathit{GTerms}, \tag{B6}$$

$$y \in \mathrm{fvars}(\sigma) \iff \mathrm{rt}(y, \sigma) \in \mathit{Vars}, \tag{B7}$$

$$y \in \mathrm{lvars}(\sigma) \iff \mathrm{rt}(y, \sigma) \in \mathit{LTerms}. \tag{B8}$$

Proof
We prove each item separately.

(B5). By Proposition 56, $y \in \mathrm{occ}(\sigma, v)$ if and only if $v \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma)$. To prove the first implication ($\Rightarrow$), let $v \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma)$. Then, for all $i > 0$, we have $v \in \mathrm{vars}(y\sigma^i) \setminus \mathrm{dom}(\sigma)$, so that $v \in \mathrm{vars}\big(\mathrm{rt}(y, \sigma)\big)$. To prove the other implication ($\Leftarrow$), assume that $v \in \mathrm{vars}\big(\mathrm{rt}(y, \sigma)\big)$. Then, by definition of rt, $v \notin \mathrm{dom}(\sigma)$ and $v \in \mathrm{vars}(y\sigma^i)$ for some $i \geq 1$. Assume that $i \geq 1$ is minimal such that $v \in \mathrm{vars}(y\sigma^i)$. If $i > 1$, then we have $v \notin \mathrm{vars}(y\sigma^{i-1})$ which contradicts the assumption that $\sigma \in \mathit{VSubst}$. Thus $i = 1$ and $v \in \mathrm{vars}(y\sigma)$.

(B6). By Definition 12, we have $y \in \mathrm{gvars}(\sigma)$ if and only if $y \notin \mathrm{occ}(\sigma, v)$, for all

$v \in \mathit{Vars}$. By case (B5) proved above, this is equivalent to $v \notin \mathrm{vars}\big(\mathrm{rt}(y,\sigma)\big)$, for all $v \in \mathit{Vars}$. Thus, $\mathrm{vars}\big(\mathrm{rt}(y,\sigma)\big) = \varnothing$ and $\mathrm{rt}(y,\sigma) \in \mathit{GTerms}$.

(B7). By Proposition 56, $y \in \mathrm{fvars}(\sigma)$ if and only if $y\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. To prove the first implication ($\Rightarrow$), let $y\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. Then, for all $i > 0$, $y\sigma^i = y\sigma$. Thus $\mathrm{rt}(y,\sigma) = y\sigma \in \mathit{Vars}$. To prove the other implication ($\Leftarrow$), assume that $\mathrm{rt}(y,\sigma) \in \mathit{Vars}$. We prove by contradiction that $y\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. In fact, assume that $y\sigma \notin \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. We have two cases:

1. if $y\sigma \notin \mathit{Vars}$ then, by definition, $\mathrm{rt}(y,\sigma) \notin \mathit{Vars}$.
2. otherwise, let $y\sigma \in \mathrm{dom}(\sigma)$. Thus, we have $y \neq y\sigma \neq y\sigma\sigma$, so that $(y \mapsto y\sigma) \in \sigma$ and $(y\sigma \mapsto y\sigma\sigma) \in \sigma$. Since $\sigma \in \mathit{VSubst}$, we also have $\{y\sigma\} = \mathrm{vars}(y\sigma) = \mathrm{vars}(y\sigma\sigma)$. Therefore, $y\sigma\sigma \notin \mathit{Vars}$. Hence, by definition, $\mathrm{rt}(y,\sigma) \notin \mathit{Vars}$.

(B8). In order to prove the first implication ($\Rightarrow$), assume $y \in \mathrm{lvars}(\sigma)$ so that, by Proposition 56, we have

$$\forall z \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) : \mathrm{occ_lin}(z, y\sigma), \tag{B9}$$

$$\forall z \in \mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma) : \mathrm{vars}(z\sigma) \subseteq \mathrm{dom}(\sigma). \tag{B10}$$

We need to show that $\mathrm{rt}(y,\sigma) \in \mathit{LTerms}$. Let $v \in \mathrm{vars}\big(\mathrm{rt}(y,\sigma)\big)$. Then $v \notin \mathrm{dom}(\sigma)$ and there exists an index $j > 1$ such that $v \in \mathrm{vars}(y\sigma^j)$. As $\sigma \in \mathit{VSubst}$, $v \in \mathrm{vars}(y\sigma^i)$ for all $i > 0$. As $v \notin \mathrm{dom}(\sigma)$, by (B9), $\mathrm{occ_lin}(v, y\sigma)$ holds. Also, by (B10), for all $w \in \mathrm{vars}(y\sigma) \setminus \{v\}$, $v \notin \mathrm{vars}(w\sigma)$. As $\sigma \in \mathit{VSubst}$, $v \notin \mathrm{vars}(w\sigma^i)$ for all $i > 0$. Thus $\mathrm{occ_lin}(v, y\sigma^i)$ holds for all $i > 0$. Therefore $\mathrm{occ_lin}\big(v, \mathrm{rt}(y,\sigma)\big)$ holds. As v was an arbitrary variable in $\mathrm{vars}\big(\mathrm{rt}(y,\sigma)\big)$, we obtain $\mathrm{rt}(y,\sigma) \in \mathit{LTerms}$.

To prove the other implication ($\Leftarrow$), assume $\mathrm{rt}(y,\sigma) \in \mathit{LTerms}$, so that, by definition, we have $\mathrm{occ_lin}\big(z, \mathrm{rt}(y,\sigma)\big)$, for all $z \in \mathrm{vars}\big(\mathrm{rt}(y,\sigma)\big)$. We need to show that both (B9) and (B10) hold. We proceed by contradiction, negating the conclusion. There are two cases.

1. Assume that (B9) does not hold, i.e., there exists $z \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma)$ such that $\mathrm{occ_lin}(z, y\sigma)$ does not hold. Then, for all $i > 0$, we have that $z \in \mathrm{vars}(y\sigma^i)$, but $\mathrm{occ_lin}(z, y\sigma^i)$ does not hold. Hence, $\mathrm{occ_lin}\big(z, \mathrm{rt}(y,\sigma)\big)$ does not hold and $\mathrm{rt}(y,\sigma) \notin \mathit{LTerms}$, obtaining the contradiction.
2. Assume now (B10) does not hold, i.e., there exists $z \in \mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma)$ such that $\mathrm{vars}(z\sigma) \not\subseteq \mathrm{dom}(\sigma)$. Thus, let $v \in \mathrm{vars}(z\sigma) \setminus \mathrm{dom}(\sigma)$. Since $\sigma \in \mathit{VSubst}$ and $v \in \mathrm{vars}(y\sigma\sigma)$, then $v \in \mathrm{vars}(y\sigma)$. Then, since $z \in \mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma)$, $\mathrm{occ_lin}(v, y\sigma\sigma)$ does not hold. Also, since $v \notin \mathrm{dom}(\sigma)$, for all $i \geq 2$, $\mathrm{occ_lin}(v, y\sigma^i)$ does not hold. By definition, $\mathrm{occ_lin}\big(v, \mathrm{rt}(y,\sigma)\big)$ does not hold and $\mathrm{rt}(y,\sigma) \notin \mathit{LTerms}$, obtaining the contradiction.

□

In order to simplify the proof of Proposition 18, we first prove some preliminary results.

Lemma 58
Let T be a syntactic equality theory and $\sigma, \tau \in \mathit{RSubst}$ be satisfiable substitutions such that $T \vdash \forall(\sigma \leftrightarrow \tau)$. Let $m = \#\sigma$, $n = \#\tau$ and $s, t \in \mathit{HTerms}$, where $T \vdash \forall\big(\sigma \rightarrow (s = t)\big)$. Then $s\sigma^m \in \mathit{Vars}$ if and only if $t\tau^n \in \mathit{Vars}$.

Proof
We prove only one implication since the other follows by symmetry. Suppose $s\sigma^m \in \mathit{Vars}$. Then, by Lemma 55, $s\sigma^m \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$ so that $\mathrm{rt}(s\sigma^m, \sigma) = s\sigma^m$. Moreover, by Lemma 48, $T \vdash \forall\big(\sigma \rightarrow (s\sigma^m = t\tau^n)\big)$ so that, by Lemma 49, $\mathrm{rt}(t\tau^n, \sigma) = s\sigma^m \in \mathit{Vars}$. Therefore $t\tau^n \in \mathit{Vars}$. □

Lemma 59
Let T be a syntactic equality theory and $\sigma, \tau \in \mathit{RSubst}$ be satisfiable substitutions such that $T \vdash \forall(\sigma \leftrightarrow \tau)$. Let also $n = \#\tau$ and $s, t \in \mathit{HTerms}$ be such that $T \vdash \forall\big(\sigma \rightarrow (s = t)\big)$. If $z \in \mathrm{vars}(s) \setminus \mathrm{dom}(\sigma)$, then there exists $z' \in \mathrm{vars}(t\tau^n) \setminus \mathrm{dom}(\tau)$ and $T \vdash \forall\big(\tau \rightarrow (z = z')\big)$.

Proof
The proof is by induction on the structure of the term s.

Since $z \in \mathrm{vars}(s)$, then s cannot be a functor of arity 0, so that the only base case is when $s = z$. Then, for all $i \geq 0$, $s\sigma^i \in \mathit{Vars}$. Thus, by Lemma 58, $t\tau^n \in \mathit{Vars}$ and hence, by Lemma 55, $t\tau^n \in \mathit{Vars} \setminus \mathrm{dom}(\tau)$. By Lemma 48 and the hypothesis, we obtain $T \vdash \forall\big(\tau \rightarrow (z = t\tau^n)\big)$. Thus, in this case let $z' = t\tau^n$.

For the inductive step, suppose that $s = f(s_1, \ldots, s_m)$ and $z \in \mathrm{vars}(s_j)$, for some $j \in \{1, \ldots, m\}$. Then, by Lemma 58, we obtain $t\tau^n \notin \mathit{Vars}$. By the satisfiability hypothesis and the identity axioms, there exist $t_1, \ldots, t_m \in \mathit{HTerms}$ such that $t\tau^n = f(t_1, \ldots, t_m)$ and, in particular, $T \vdash \forall\big(\sigma \rightarrow (s_j = t_j)\big)$. By the inductive hypothesis, there exists $z' \in \mathrm{vars}(t_j\tau^n) \setminus \mathrm{dom}(\tau)$ such that $T \vdash \forall\big(\tau \rightarrow (z = z')\big)$. It follows that $z' \in \mathrm{vars}(t\tau^{2n})$ and, since $n = \#\tau$, $z' \in \mathrm{vars}(t\tau^n)$. □

Lemma 60
Let T be a syntactic equality theory and $\sigma, \tau \in \mathit{RSubst}$ be satisfiable substitutions such that $T \vdash \forall(\sigma \leftrightarrow \tau)$. Let also $n = \#\tau$ and $s, t \in \mathit{HTerms}$ be such that $T \vdash \forall\big(\sigma \rightarrow (s = t)\big)$. If there exists $z \in \mathrm{vars}(s) \setminus \mathrm{dom}(\sigma)$ such that $\mathrm{occ_lin}(z, s)$ does not hold, then there exists $z' \in \mathrm{vars}(t\tau^n) \setminus \mathrm{dom}(\tau)$ such that $\mathrm{occ_lin}(z', t\tau^{2n})$ does not hold and $T \vdash \forall\big(\tau \rightarrow (z = z')\big)$.

Proof
The proof is by induction on the structure of the term s.

Since $z \in \mathrm{vars}(s)$, then s cannot be a functor of arity 0, so that the only base case is when $s = z$. Thus there is nothing to prove, as $\mathrm{occ_lin}(z, s)$ necessarily holds.

For the inductive step, suppose that $s = f(s_1, \ldots, s_m)$ and there exists $z \in \mathrm{vars}(s) \setminus \mathrm{dom}(\sigma)$ such that $\mathrm{occ_lin}(z, s)$ does not hold. For all $i \geq 0$, we have $s\sigma^i \notin \mathit{Vars}$ so that, by Lemma 58, $t\tau^n \notin \mathit{Vars}$. By the satisfiability hypothesis and the identity axioms, there exist $t_1, \ldots, t_m \in \mathit{HTerms}$ such that $t\tau^n = f(t_1, \ldots, t_m)$ and $T \vdash \forall\big(\sigma \rightarrow (s_i = t_i)\big)$, for all $i \in \{1, \ldots, m\}$. Since $\mathrm{occ_lin}(z, s)$ does not hold, there are two cases.

1. There exist $j, k \in \{1, \ldots, m\}$ such that $j \neq k$ and $z \in \mathrm{vars}(s_j) \cap \mathrm{vars}(s_k)$. By applying Lemma 59 twice, there exist $z' \in \mathrm{vars}(t_j\tau^n) \setminus \mathrm{dom}(\tau)$ and $z'' \in$

$\mathrm{vars}(t_k\tau^n) \setminus \mathrm{dom}(\tau)$ such that $T \vdash \forall\big(\tau \to (z = z')\big)$ and $T \vdash \forall\big(\tau \to (z = z'')\big)$. Thus $T \vdash \forall\big(\tau \to (z' = z'')\big)$ and, since $z', z'' \notin \mathrm{dom}(\tau)$, by Lemma 49, we have that $z' = z''$. Hence, $\mathrm{occ_lin}(z', t\tau^{2n})$ does not hold.

2. There exists $j \in \{1, \ldots, m\}$ such that $\mathrm{occ_lin}(z, s_j)$ does not hold. Then, by the inductive hypothesis, there exists $z' \in \mathrm{vars}(t_j\tau^n) \setminus \mathrm{dom}(\tau)$ such that $\mathrm{occ_lin}(z', t_j\tau^{2n})$ does not hold and $T \vdash \forall\big(\tau \to (z = z')\big)$. Thus $\mathrm{occ_lin}(z', t\tau^{3n})$ does not hold. We now show that $\mathrm{occ_lin}(z', t\tau^{2n})$ does not hold. As $n = \#\tau$, $z' \in \mathrm{vars}(t\tau^n)$. Clearly, if $\mathrm{occ_lin}(z', t\tau^n)$ does not hold, then $\mathrm{occ_lin}(z', t\tau^{2n})$ does not hold. On the other hand, if $\mathrm{occ_lin}(z', t\tau^n)$ holds, then there exists $y \in \mathrm{vars}(t\tau^n)$ such that $y \neq z'$ and $z' \in \mathrm{vars}(y\tau^{2n})$. Again, as $n = \#\tau$, $z' \in \mathrm{vars}(y\tau^n)$, so that $\mathrm{occ_lin}(z', t\tau^{2n})$ does not hold.

□

Proof of Proposition 18 on page 12. As the operator 'ssets' basically corresponds to the abstraction function defined in (Hill et al. 2002), equation (12) follows from (Hill et al. 2002, Theorem 4). Also, by Definition 12, equation (13) is a simple consequence of equation (12).

To prove (14) and (15), let $m = \#\sigma$ and $n = \#\tau$. We will prove one inclusion only, as the other one follows by a symmetric reasoning.

Consider (14) and let $y \in \mathrm{fvars}(\sigma)$. Then, by Definition 14, $y\sigma^m \in \mathit{Vars}$. so that, by Lemma 58, $y\tau^n \in \mathit{Vars}$. Hence, by Definition 14, $y \in \mathrm{fvars}(\tau)$.

Consider (15) and let $y \notin \mathrm{lvars}(\sigma)$. Then, by Definition 16, there exists $z \in \mathrm{vars}(y\sigma^m) \setminus \mathrm{dom}(\sigma)$ such that $\mathrm{occ_lin}(z, y\sigma^{2m})$ does not hold. By Lemma 48, $T \vdash \forall\big(\sigma \to (y\sigma^{2m} = y)\big)$ so that, by Lemma 60, there exists $z' \in \mathrm{vars}(y\tau^n) \setminus \mathrm{dom}(\tau)$ such that $\mathrm{occ_lin}(z', y\tau^{2n})$ does not hold. Therefore, by Definition 16, $y \notin \mathrm{lvars}(\tau)$.

Lemma 61
Let T be a syntactic equality theory and $\sigma, \tau \in \mathit{RSubst}$ be satisfiable substitutions such that $\mathrm{dom}(\sigma) = \mathrm{dom}(\tau)$ and $T \vdash \forall(\sigma \leftrightarrow \tau)$. Then $\mathrm{vars}\big(\mathrm{rt}(y, \sigma)\big) = \mathrm{vars}\big(\mathrm{rt}(y, \tau)\big)$.

Proof
We only prove $\mathrm{vars}\big(\mathrm{rt}(y, \sigma)\big) \subseteq \mathrm{vars}\big(\mathrm{rt}(y, \tau)\big)$, since the other direction follows by symmetry. Suppose $v \in \mathrm{vars}\big(\mathrm{rt}(y, \sigma)\big)$. Then $v \notin \mathrm{dom}(\sigma)$ and hence $v \notin \mathrm{dom}(\tau)$. Moreover, there exists an index $i \geq 0$ such that $v \in \mathrm{vars}(y\sigma^i)$. Hence we have $v \in \mathrm{vars}\big(\mathrm{rt}(y\sigma^i, \tau)\big)$. By Lemma 48 and the hypothesis, we have $T \vdash \forall\big(\tau \to (y = y\sigma^i)\big)$ so that, by Lemma 49, $\mathrm{rt}(y, \tau) = \mathrm{rt}(y\sigma^i, \tau)$. It follows that $v \in \mathrm{vars}\big(\mathrm{rt}(y, \tau)\big)$. □

Proof of Proposition 19 on page 13. By Proposition 54, there exists $\tau \in \mathit{VSubst}$ such that $\mathrm{dom}(\sigma) = \mathrm{dom}(\tau)$ and $T \vdash \forall(\sigma \leftrightarrow \tau)$. By Proposition 18, we have $\mathrm{ssets}(\sigma) = \mathrm{ssets}(\tau)$, $\mathrm{gvars}(\sigma) = \mathrm{gvars}(\tau)$, $\mathrm{fvars}(\sigma) = \mathrm{fvars}(\tau)$ and $\mathrm{lvars}(\sigma) = \mathrm{lvars}(\tau)$. From all of the above, by Proposition 57, we obtain

$$y \in \mathrm{occ}(\sigma, v) \quad \Longleftrightarrow \quad v \in \mathrm{vars}\big(\mathrm{rt}(y, \tau)\big),$$

$$
\begin{aligned}
y \in \mathrm{gvars}(\sigma) &\iff \mathrm{rt}(y,\tau) \in \mathit{GTerms},\\
y \in \mathrm{fvars}(\sigma) &\iff \mathrm{rt}(y,\tau) \in \mathit{Vars},\\
y \in \mathrm{lvars}(\sigma) &\iff \mathrm{rt}(y,\tau) \in \mathit{LTerms}.
\end{aligned}
$$

Thus (16) follows from Lemma 61 while (17), (18) and (19) follow from Proposition 3.

Appendix C Proofs of the Results of Subsection 3.2

In this section we prove that the abstract unification operator defined on the domain *SFL* is a correct approximation of the concrete unification procedure.

The section is composed of several subsections. In Subsection C.1, we recall the definition of the abstract mgu operator on *SH* and the corresponding correctness result, as provided in (Hill et al. 2002). Then, in Subsection C.2, we prove the correctness of the auxiliary operations given in Definition 23. Subsection C.3, proves some general results and introduces some notation that helps to simplify the main correctness results. In the next three Subsections C.4, C.5 and C.6, we prove the correctness of the abstract unification operator on set-sharing, freeness and linearity, respectively. Then, finally, in Subsection C.7, we combine these results to establish the correctness of amgu_s in the proof of Theorem 27.

C.1 Abstract unification for SH

Definition 62

(amgu and aunify.) The function amgu: $\mathit{SH} \times \mathit{Bind} \to \mathit{SH}$ captures the effects of a binding on an *SH* element. For each $sh \in \mathit{SH}$ and $(x \mapsto t) \in \mathit{Bind}$ such that $\{x\} \cup \mathrm{vars}(t) \subseteq \mathit{VI}$, let $sh_x \stackrel{\mathrm{def}}{=} \mathrm{rel}\big(\{x\}, sh\big)$, $sh_t \stackrel{\mathrm{def}}{=} \mathrm{rel}\big(\mathrm{vars}(t), sh\big)$ and $sh_{_} \stackrel{\mathrm{def}}{=} \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), sh\big)$. Then

$$\mathrm{amgu}(sh, x \mapsto t) \stackrel{\mathrm{def}}{=} sh_{_} \cup \mathrm{bin}(sh_x^\star, sh_t^\star).$$

The operator aunify: $\mathit{SH} \times \mathit{RSubst} \to \mathit{SH}$ is defined, for each $sh \in \mathit{SH}$ and each substitution $\sigma \in \mathit{RSubst}$ such that $\mathrm{vars}(\sigma) \subseteq \mathit{VI}$, by

$$\mathrm{aunify}(sh, \sigma) \stackrel{\mathrm{def}}{=} \begin{cases} sh, & \text{if } \sigma = \varnothing;\\ \mathrm{aunify}\big(\mathrm{amgu}(sh, x \mapsto t), \sigma \setminus \{x \mapsto t\}\big), & \text{if } (x \mapsto t) \in \sigma. \end{cases}$$

Theorem 63

((Hill et al. 2002, Theorem 5).) Let $sh \in \mathit{SH}$ and $(x \mapsto t) \in \mathit{Bind}$, where $\{x\} \cup \mathrm{vars}(t) \subseteq \mathit{VI}$. Let $\sigma \in \mathit{RSubst}$ be such that $\mathrm{ssets}(\sigma) \subseteq sh$ and suppose that $\tau \in \mathrm{mgs}\big(\{x = t\} \cup \sigma\big)$ in the syntactic equality theory T. Then

$$\mathrm{ssets}(\tau) \subseteq \mathrm{amgu}(sh, x \mapsto t).$$

C.2 The Correctness of the auxiliary operations

Proof of Theorem 24 on page 15. Let $d = \langle sh, f, l\rangle$, $V_s = \mathrm{vars}(s)$ and $V_t = \mathrm{vars}(t)$. By Definition 21, we have $\forall v \in \mathit{Vars} : \mathrm{occ}(\sigma, v) \cap \mathit{VI} \in sh \cup \{\varnothing\}$, $f \subseteq \mathrm{fvars}(\sigma)$, and $l \subseteq \mathrm{lvars}(\sigma)$.

Consider the implication (20). By Definition 23, the hypothesis and Proposition 19, we have

$$\begin{aligned}
\mathrm{ind}_d(s,t) &\iff \mathrm{rel}(V_s, sh) \cap \mathrm{rel}(V_t, sh) = \varnothing \\
&\iff \forall S \in sh, w_1 \in V_s, w_2 \in V_t : \{w_1, w_2\} \not\subseteq S \\
&\implies \forall v \in \mathit{Vars}, w_1 \in V_s, w_2 \in V_t : \{w_1, w_2\} \not\subseteq \mathrm{occ}(\sigma, v) \\
&\iff \forall w_1 \in V_s, w_2 \in V_t : \mathrm{vars}\big(\mathrm{rt}(w_1, \sigma)\big) \cap \mathrm{vars}\big(\mathrm{rt}(w_2, \sigma)\big) = \varnothing \\
&\iff \mathrm{vars}\big(\mathrm{rt}(s, \sigma)\big) \cap \mathrm{vars}\big(\mathrm{rt}(t, \sigma)\big) = \varnothing.
\end{aligned}$$

Consider the equivalence (21). By Definition 23, we have

$$\begin{aligned}
\mathrm{ind}_d(y,t) &\iff \mathrm{rel}\big(\{y\}, sh\big) \cap \mathrm{rel}(V_t, sh) = \varnothing \\
&\iff \forall S \in \mathrm{rel}(V_t, sh) : y \notin S \\
&\iff y \notin \mathrm{share_with}_d(t).
\end{aligned}$$

Consider now the implication (22). By Definition 23, the hypothesis and Proposition 19, we have

$$\begin{aligned}
\mathrm{free}_d(t) &\iff t \in f \\
&\implies t \in \mathrm{fvars}(\sigma) \\
&\iff \mathrm{rt}(t, \sigma) \in \mathit{Vars}.
\end{aligned}$$

Consider now the implication (23). By Definition 23, the hypothesis, Definition 12, and Proposition 19, we have

$$\begin{aligned}
\mathrm{ground}_d(t) &\iff V_t \subseteq \mathit{VI} \setminus \mathrm{vars}(sh) \\
&\iff \forall w \in V_t, S \in sh : w \notin S \\
&\implies \forall w \in V_t, v \in \mathit{Vars} : w \notin \mathrm{occ}(\sigma, v) \\
&\iff \forall w \in V_t : w \in \mathrm{gvars}(\sigma) \\
&\iff \forall w \in V_t : \mathrm{rt}(w, \sigma) \in \mathit{GTerms} \\
&\iff \mathrm{rt}(t, \sigma) \in \mathit{GTerms}.
\end{aligned}$$

Finally consider the implication (24). By Definition 23, the hypothesis, the above results and Proposition 19, we have

$$\begin{aligned}
\mathrm{lin}_d(t) &\iff \forall y, z \in \mathrm{vars}(t) : \mathrm{ground}_d(y) \\
&\qquad \vee \Big((y \in l) \wedge \mathrm{occ_lin}(y, t) \wedge \big(y \neq z \implies \mathrm{ind}_d(y, z)\big) \Big) \\
&\implies \forall y, z \in \mathrm{vars}(t) : \mathrm{ground}_d(y) \\
&\qquad \vee \Big(\big(y \in \mathrm{lvars}(\sigma)\big) \wedge \mathrm{occ_lin}(y, t) \wedge \big(y \neq z \implies \mathrm{ind}_d(y, z)\big) \Big)
\end{aligned}$$

$$
\begin{aligned}
&\Longrightarrow \forall y, z \in \mathrm{vars}(t) : \mathrm{rt}(y,\sigma) \in \mathit{GTerms} \\
&\qquad \vee \Big(\big(\mathrm{rt}(y,\sigma) \in \mathit{LTerms}\big) \wedge \mathrm{occ_lin}(y,t) \\
&\qquad\qquad \wedge \big(y \neq z \implies \mathrm{vars}(\mathrm{rt}(y,\sigma)) \cap \mathrm{vars}(\mathrm{rt}(z,\sigma)) = \varnothing\big)\Big) \\
&\Longleftrightarrow \mathrm{rt}(t,\sigma) \in \mathit{LTerms}.
\end{aligned}
$$

C.3 Additional results and notation used for the correctness results

Lemma 64
Assume T is an equality theory and $\sigma \in \mathit{RSubst}$. Then, for each $s, t \in \mathit{HTerms}$,

$$\mathrm{mgs}\big(\sigma \cup \{s = t\}\big) = \mathrm{mgs}\big(\sigma \cup \{s = t\sigma\}\big).$$

Proof
First, note, using the congruence axioms (3) and (4), that, for any terms $p, q, r \in \mathit{HTerms}$,

$$T \vdash \forall(p = q \wedge q = r) \leftrightarrow \forall(p = r \wedge q = r).$$

Secondly note that, using Lemma 48, for any substitution $\tau \in \mathit{RSubst}$ and term $r \in \mathit{HTerms}$, $T \vdash \forall\big(\tau \rightarrow (r = r\tau)\big)$, so that

$$T \vdash \forall\big(\tau \leftrightarrow \tau \cup \{r = r\tau\}\big).$$

Using these results,

$$
\begin{aligned}
&T \vdash \forall\big(\sigma \cup \{s = t\} \leftrightarrow \sigma \cup \{s = t, t = t\sigma\}\big), \\
&T \vdash \forall\big(\sigma \cup \{s = t\} \leftrightarrow \sigma \cup \{s = t\sigma, t = t\sigma\}\big), \\
&T \vdash \forall\big(\sigma \cup \{s = t\} \leftrightarrow \sigma \cup \{s = t\sigma\}\big).
\end{aligned}
$$

The thesis follows by the definition of mgs. □

The following rather technical result is required more than once in the proofs in this section.

Lemma 65
((Hill et al. 2002, Lemma 6).) Let $\tau, \sigma \in \mathit{VSubst}$ be satisfiable in the syntactic equality theory T and suppose $T \vdash \forall(\tau \rightarrow \sigma)$. In addition, let $s, t \in \mathit{HTerms}$ be such that $T \vdash \forall\big(\tau \rightarrow (s = t)\big)$ and $v \in \mathrm{vars}(s) \setminus \mathrm{dom}(\tau)$. Then there exists a variable $z \in \mathrm{vars}(t\sigma) \setminus \mathrm{dom}(\sigma)$ such that $v \in \mathrm{vars}(z\tau)$.

The following lemma, which will be used several times in the following proofs without an explicit reference to it, states the well-known result that groundness is closed by entailment.

Lemma 66
Let $\sigma, \tau \in \mathit{RSubst}$ be satisfiable in the syntactic equality theory T and such that $T \vdash \forall(\tau \rightarrow \sigma)$. Then $\mathrm{gvars}(\sigma) \subseteq \mathrm{gvars}(\tau)$.

Proof

We prove the result by showing that $x \notin \text{gvars}(\tau)$ implies $x \notin \text{gvars}(\sigma)$.

By Proposition 54, we can assume there exist $\sigma', \tau' \in \mathit{VSubst}$ such that $T \vdash \forall(\sigma \leftrightarrow \sigma')$ and $T \vdash \forall(\tau \leftrightarrow \tau')$, so that $T \vdash \forall(\tau' \rightarrow \sigma')$. Also, by Proposition 18, we have $\text{gvars}(\sigma) = \text{gvars}(\sigma')$ and $\text{gvars}(\tau) = \text{gvars}(\tau')$. Therefore, it is sufficient to prove that $x \notin \text{gvars}(\tau')$ implies $x \notin \text{gvars}(\sigma')$.

Assume $x \notin \text{gvars}(\tau')$. By Definition 12, there exists $v \in \mathit{Vars}$ such that $x \in \text{occ}(\tau', v)$. By Proposition 56, $v \in \text{vars}(x\tau') \setminus \text{dom}(\tau')$. Also, by Lemma 48, $T \vdash \forall(\tau' \rightarrow x\tau' = x)$. Therefore, by Lemma 65 (taking $s = x\tau'$ and $t = x$) there exists $z \in \text{vars}(x\sigma') \setminus \text{dom}(\sigma')$ such that $v \in \text{vars}(z\tau')$. By Definition 10, we have $x \in \text{occ}(\sigma', z)$ so that, by Definition 12, $x \notin \text{gvars}(\sigma')$. □

Since all substitutions in $\text{mgs}(e)$ are equivalent, they also have the same sharing groups, ground the same variables and have the same sets of free and linear variables.

Lemma 67

Let $e \subseteq \mathit{Eqs}$ be satisfiable in the syntactic equality theory T. If $\sigma, \tau \in \text{mgs}(e)$, then $\text{ssets}(\sigma) = \text{ssets}(\tau)$, $\text{gvars}(\sigma) = \text{gvars}(\tau)$, $\text{fvars}(\sigma) = \text{fvars}(\tau)$ and $\text{lvars}(\sigma) = \text{lvars}(\tau)$.

Proof

By definition of mgs, we have $\sigma, \tau \in \mathit{RSubst}$ and $T \vdash \forall(\sigma \leftrightarrow e \leftrightarrow \tau)$. Thus, all the stated equivalences follow from Proposition 18. □

We now introduce a bit of terminology that helps simplify the notation used in the next and later proofs. Given $V \subseteq \mathit{Vars}$, we say that $t \in \mathit{HTerms}$ is V-linear if $\text{occ_lin}(v, t)$ holds for all variables $v \in \text{vars}(t) \cap V$. Note that if a term is V-linear, then it is also W-linear, for all $W \subseteq V$. This terminology also applies to n-tuples of terms, by simply regarding the n-tuple construction as a term functor of arity n. Moreover, if $\bar{s}, \bar{t} \in \mathit{HTerms}^n$ are such that $\text{mgs}(\bar{s} = \bar{t}) \neq \varnothing$, then we write $\text{gvars}(\bar{s} = \bar{t})$ to denote the set $\text{gvars}(\mu)$, where $\mu \in \text{mgs}(\bar{s} = \bar{t})$. Note that, by Lemma 67, this notation is not ambiguous.

The next proposition shows under what conditions the properties of linearity and independence are preserved.

Proposition 68

Suppose $\bar{s}, \bar{t} \in \mathit{HTerms}^n$ and $\text{mgs}(\bar{s} = \bar{t}) \neq \varnothing$. Let $G \stackrel{\text{def}}{=} \text{gvars}(\bar{s} = \bar{t})$ and suppose also that $\bar{s}$ is $(\mathit{Vars} \setminus G)$-linear. Then for all $\mu \in \text{mgs}(\bar{s} = \bar{t}) \cap \mathit{VSubst}$ we have

$$\mathit{Vars} \setminus \text{vars}(\bar{s}) \subseteq \text{lvars}(\mu), \tag{C1}$$

$$\forall z, z' \in \mathit{Vars} \setminus \text{vars}(\bar{s}) : \quad z \neq z' \implies \text{rel}\big(\{z\}, \text{ssets}(\mu)\big) \cap \text{rel}\big(\{z'\}, \text{ssets}(\mu)\big) = \varnothing. \tag{C2}$$

Proof
By Propositions 18 and 54, we just have to show that *there exists* $\mu \in \mathit{VSubst}$ such that (C1) and (C2) hold.

Observe that (C1) and (C2) will hold if and only if, for each $z, z' \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s})$:

$$z\mu \text{ is } (\mathit{Vars} \setminus G)\text{-linear;} \tag{C3}$$

$$\mathrm{vars}(z\mu) \cap \mathrm{dom}(\mu) \subseteq G; \tag{C4}$$

$$z \neq z' \implies \mathrm{vars}(z\mu) \cap \mathrm{vars}(z'\mu) \subseteq G. \tag{C5}$$

To see this note first that, by case (B4) of Proposition 56, if $z \in \mathit{Vars}$ then $z \in \mathrm{lvars}(\mu)$ if and only if (C4) holds and $z\mu$ is $\big(\mathit{Vars} \setminus \mathrm{dom}(\mu)\big)$-linear. However, as $G \subseteq \mathrm{dom}(\mu)$, if (C4) holds, then $\mathrm{vars}(z\mu) \setminus \mathrm{dom}(\mu) = \mathrm{vars}(z\mu) \setminus G$ so that $z\mu$ is $\big(\mathit{Vars} \setminus \mathrm{dom}(\mu)\big)$-linear if and only if (C3) holds. Moreover, by case (B1) of Proposition 56, if $z, z' \in \mathit{Vars}$ where $z \neq z'$, then $\mathrm{vars}(z\mu) \cap \mathrm{vars}(z'\mu) \subseteq \mathrm{dom}(\mu)$ holds if and only if $\mathrm{rel}\big(\{z\}, \mathrm{ssets}(\mu)\big) \cap \mathrm{rel}\big(\{z'\}, \mathrm{ssets}(\mu)\big) = \varnothing$. However, as $G \subseteq \mathrm{dom}(\mu)$, if (C4) holds then, for any $z'' \in \{z, z'\}$, $\mathrm{vars}(z''\mu) \cap \mathrm{dom}(\mu) = \mathrm{vars}(z''\mu) \cap G$ so that $\mathrm{vars}(z\mu) \cap \mathrm{vars}(z'\mu) \subseteq \mathrm{dom}(\mu)$ if and only if $\mathrm{vars}(z\mu) \cap \mathrm{vars}(z'\mu) \subseteq G$ and hence $z \neq z'$ implies $\mathrm{rel}\big(\{z\}, \mathrm{ssets}(\mu)\big) \cap \mathrm{rel}\big(\{z'\}, \mathrm{ssets}(\mu)\big) = \varnothing$ if and only if (C5) holds.

Let $\bar{s} = (s_1, \ldots, s_n)$, and $\bar{t} = (t_1, \ldots, t_n)$, $W = \mathrm{vars}(\bar{s}) \cup \mathrm{vars}(\bar{t})$ and $V = \mathit{Vars} \setminus G$. We assume that $\bar{s}$ is V-linear and show, by induction on the number of variables in W, that there exists $\mu \in \mathrm{mgs}(\bar{s} = \bar{t}) \cap \mathit{VSubst}$ such that, for each $z \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s})$, properties (C3), (C4) and (C5) hold.

Suppose first that, for some $i = 1, \ldots, n$, we have $s_i = f(r_1, \ldots, r_m)$ and $t_i = f(u_1, \ldots, u_m)$, where $m \geq 0$. Let

$$\bar{s}_i \stackrel{\text{def}}{=} (s_1, \ldots, s_{i-1}, r_1, \ldots, r_m, s_{i+1}, \ldots, s_n),$$
$$\bar{t}_i \stackrel{\text{def}}{=} (t_1, \ldots, t_{i-1}, u_1, \ldots, u_m, t_{i+1}, \ldots, t_n).$$

Then $\mathrm{mvars}(\bar{s}_i) = \mathrm{mvars}(\bar{s})$ and $\mathrm{mvars}(\bar{t}_i) = \mathrm{mvars}(\bar{t})$ so that, since $\bar{s}$ is V-linear, $\bar{s}_i$ is V-linear. Moreover, by the congruence axiom (5), we have $\mathrm{mgs}(\bar{s}_i = \bar{t}_i) = \mathrm{mgs}(\bar{s} = \bar{t})$. (Note that in the case that s_i and t_i are identical constants, the equation $s_i = t_i$ is just removed.) Thus, as $\bar{s}$ and $\bar{t}$ are finite sequences of finite terms, we can assume that $s_i \in \mathit{Vars}$ or $t_i \in \mathit{Vars}$, for all $i = 1, \ldots, n$.

Secondly, suppose that for some $i = 1, \ldots, n$, $s_i = t_i$. By the previous paragraph, we can assume that $s_i \in \mathit{Vars}$. Let

$$\bar{s}_i \stackrel{\text{def}}{=} (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n),$$
$$\bar{t}_i \stackrel{\text{def}}{=} (t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n).$$

Then $\mathrm{mvars}(\bar{s}_i) \cup \{s_i\} = \mathrm{mvars}(\bar{s})$ and $\mathrm{mvars}(\bar{t}_i) \cup \{s_i\} = \mathrm{mvars}(\bar{t})$ so that, as $\bar{s}$ is V-linear, $\bar{s}_i$ is V-linear. Furthermore, by the congruence axiom (2), $\mathrm{mgs}(\bar{s}_i = \bar{t}_i) = \mathrm{mgs}(\bar{s} = \bar{t})$. Thus, as $\bar{s}$ and $\bar{t}$ are sequences of finite length n, we can assume that $s_i \neq t_i$, for all $i = 1, \ldots, n$.

Therefore, for the rest of the proof, we will assume that $s_i \neq t_i$ and $s_i \in \mathit{Vars}$ or $t_i \in \mathit{Vars}$, for all $i = 1, \ldots, n$.

The base case is when $W = \varnothing$, so that we have $\mathrm{vars}(\mu) = \varnothing$ for all $\mu \in \mathrm{mgs}(\bar{s} = \bar{t})$. Thus all three properties hold trivially.

To prove the inductive step, we assume that $W \neq \varnothing$, so that $n > 0$.

As the order of equations is irrelevant, we choose an ordering so that, if there exists $i = 1, \ldots, n$ such that $V \cap \mathrm{vars}(s_i) \cap \mathrm{vars}(t_i) = \varnothing$, then $V \cap \mathrm{vars}(s_1) \cap \mathrm{vars}(t_1) = \varnothing$. Thus we can identify three different cases:

a. for all $i = 1, \ldots, n$, $V \cap \mathrm{vars}(s_i) \cap \mathrm{vars}(t_i) \neq \varnothing$;
b. $s_1 \in V \setminus \mathrm{vars}(t_1)$;
c. $t_1 \in V \setminus \mathrm{vars}(s_1)$ and $s_1 \notin \mathit{Vars}$.

Case a. For all $i = 1, \ldots, n$, $V \cap \mathrm{vars}(s_i) \cap \mathrm{vars}(t_i) \neq \varnothing$.

For each $i = 1, \ldots, n$, we are assuming that $s_i \in V$ or $t_i \in V$. Therefore, for each $i = 1, \ldots, n$, $s_i \in \mathrm{vars}(t_i)$ or $t_i \in \mathrm{vars}(s_i)$ so that, without loss of generality, we can assume, for some k where $0 \leq k \leq n$, $s_i \in V$ if $1 \leq i \leq k$ and $t_i \in V$ if $k + 1 \leq i \leq n$. Let $\mu' \subseteq \mathit{Eqs}$ be defined as

$$\mu' \stackrel{\text{def}}{=} \{s_1 = t_1, \ldots, s_k = t_k\} \cup \{t_{k+1} = s_{k+1}, \ldots, t_n = s_n\}.$$

We show that $\mu' \in \mathrm{mgs}(\bar{s} = \bar{t})$. First we must show that $\mu' \in \mathit{RSubst}$. As $\bar{s}$ is V-linear, $(s_1, \ldots, s_k)$ is linear; $(t_{k+1}, \ldots, t_n)$ is also linear, because $\bar{s}$ is V-linear and $t_i \in V \cap \mathrm{vars}(s_i)$ if $k + 1 \leq i \leq n$; moreover, for the same reasons, $\{s_1, \ldots, s_k\} \cap \{t_{k+1}, \ldots, t_n\} = \varnothing$. As we are assuming that, for all $i = 1, \ldots, n$, $s_i \neq t_i$ and $V \cap \mathrm{vars}(s_i) \cap \mathrm{vars}(t_i) \neq \varnothing$, it follows that $t_i \notin \mathit{Vars}$ when $1 \leq i \leq k$ and $s_i \notin \mathit{Vars}$ when $k + 1 \leq i \leq n$, so that each equation in μ' is a binding and μ' has no circular subsets. Thus $\mu' \in \mathit{RSubst}$ and hence, by the congruence axiom (3), $\mu' \in \mathrm{mgs}(\bar{s} = \bar{t})$.

By Proposition 54, there exists $\mu \in \mathit{VSubst}$ such that $\mathrm{dom}(\mu) = \mathrm{dom}(\mu')$ and $\mu \in \mathrm{mgs}(\bar{s} = \bar{t})$. As $\{t_{k+1}, \ldots, t_n\} \subseteq \mathrm{vars}\big((s_{k+1}, \ldots, s_n)\big)$, we have $\mathrm{dom}(\mu) = \mathrm{dom}(\mu') \subseteq \mathrm{vars}(\bar{s})$ so that the required result for μ holds trivially.

Case b. Suppose $s_1 \in V \setminus \mathrm{vars}(t_1)$ and let

$$\bar{s}_1 \stackrel{\text{def}}{=} (s_2, \ldots, s_n),$$
$$\bar{t}_1 \stackrel{\text{def}}{=} \big(t_2\{s_1 \mapsto t_1\}, \ldots, t_n\{s_1 \mapsto t_1\}\big).$$

As $\bar{s}$ is V-linear, $\bar{s}_1$ is V-linear and $s_1 \notin \mathrm{vars}(\bar{s}_1)$. Also, all occurrences of s_1 in $\bar{t}$ are replaced in $\bar{t}_1$ by t_1 so that, as $s_1 \notin \mathrm{vars}(t_1)$ (by the assumption for this case), $s_1 \notin \mathrm{vars}(\bar{t}_1)$. Thus,

$$s_1 \notin W_1 \stackrel{\text{def}}{=} \mathrm{vars}(\bar{s}_1) \cup \mathrm{vars}(\bar{t}_1), \tag{C6}$$

so that $W_1 \subset W$. Let $G_1 \stackrel{\text{def}}{=} \mathrm{gvars}(\bar{s}_1 = \bar{t}_1)$ and $V_1 \stackrel{\text{def}}{=} \mathit{Vars} \setminus G_1$. Note that $G \setminus \{s_1\} \subseteq G_1 \subseteq G$ and, by the assumption for this case, $s_1 \in V$, so that $s_1 \notin G$. As a consequence, $G_1 = G$, $V_1 = V$ and $\bar{s}_1$ is V_1-linear, so that the inductive hypothesis applies to $\bar{s}_1$ and $\bar{t}_1$. Thus, there exists $\mu_1 \in \mathrm{mgs}(\bar{s}_1 = \bar{t}_1) \cap \mathit{VSubst}$ such that, for each $z, z' \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$, the three inductive properties hold.

Let $\mu \subseteq \mathit{Eqs}$ be defined as

$$\mu \stackrel{\text{def}}{=} \{s_1 = t_1\mu_1\} \cup \mu_1.$$

We show that $\mu \in \mathit{VSubst}$. Note that $\nu \stackrel{\text{def}}{=} \{s_1 \mapsto t_1\} \in \mathit{VSubst}$ and, by (C6), we have $\mathrm{dom}(\nu) \cap \mathrm{vars}(\mu_1) = \{s_1\} \cap \mathrm{vars}(\mu_1) = \varnothing$. Hence, since $\mu_1 \in \mathit{VSubst}$, by Lemma 53 we obtain $\mu = \mu_1 \circ \nu \in \mathit{VSubst}$. Also note that, by Lemma 64, $\mu \in \mathrm{mgs}(\bar{s} = \bar{t})$.

Suppose that $z \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s})$. Then, as $\mathrm{vars}(\bar{s}) = \mathrm{vars}(\bar{s}_1) \cup \{s_1\}$, $z \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$. Thus, the inductive properties (C3), (C4) and (C5) using μ_1 and $\bar{s}_1$ can be applied to $z\mu_1$. Knowing this, we now show that the same properties using μ and $\bar{s}$ can be applied to $z\mu$. Since $\mathrm{dom}(\mu) = \mathrm{dom}(\mu_1) \cup \{s_1\}$ and $z \neq s_1$, we have $z\mu_1 = z\mu$ and $s_1 \notin \mathrm{vars}(z\mu)$. Each property is proved separately.

1. By the inductive property (C3), we have $z\mu_1$ is V_1-linear. As $z\mu = z\mu_1$ and $V = V_1$, $z\mu$ is V-linear.
2. By inductive property (C4), we have $\mathrm{vars}(z\mu_1) \cap \mathrm{dom}(\mu_1) \subseteq G_1$. Since $z\mu = z\mu_1$, $G = G_1$, $\mathrm{dom}(\mu) = \mathrm{dom}(\mu_1) \cup \{s_1\}$ and $s_1 \notin \mathrm{vars}(z\mu)$, we obtain $\mathrm{vars}(z\mu) \cap \mathrm{dom}(\mu) \subseteq G$.
3. Let $z' \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s})$ be such that $z \neq z'$. Since $z' \notin \mathrm{vars}(\bar{s})$, we have $z' \notin \mathrm{vars}(\bar{s}_1)$ and $z'\mu = z'\mu_1$. By applying inductive property (C5), $\mathrm{vars}(z\mu_1) \cap \mathrm{vars}(z'\mu_1) \subseteq G_1$. As $z\mu = z\mu_1$, $z'\mu = z'\mu_1$ and $G = G_1$, we obtain $\mathrm{vars}(z\mu) \cap \mathrm{vars}(z'\mu) \subseteq G$.

Case c. Assume that $t_1 \in V \setminus \mathrm{vars}(s_1)$ and $s_1 \notin \mathit{Vars}$. Let

$$\bar{s}_1 \stackrel{\text{def}}{=} \big(s_2\{t_1 \mapsto s_1\}, \ldots, s_n\{t_1 \mapsto s_1\}\big),$$
$$\bar{t}_1 \stackrel{\text{def}}{=} \big(t_2\{t_1 \mapsto s_1\}, \ldots, t_n\{t_1 \mapsto s_1\}\big).$$

As $\bar{s}$ is V-linear and $t_1 \in V$, there is at most one occurrence of t_1 in $(s_2, \ldots, s_n)$; also s_1 is V-linear and, for $i = 2, \ldots, n$, $\mathrm{vars}(s_1) \cap \mathrm{vars}(s_i) \cap V = \varnothing$ so that $\bar{s}_1$ is V-linear. Also, by the assumption for this case $t_1 \notin \mathrm{vars}(s_1)$ and all occurrences of t_1 in $(s_2, \ldots, s_n)$ and $(t_2, \ldots, t_n)$ are replaced in $\bar{s}_1$ and $\bar{t}_1$ by s_1. Thus

$$t_1 \notin W_1 \stackrel{\text{def}}{=} \mathrm{vars}(\bar{s}_1) \cup \mathrm{vars}(\bar{t}_1), \tag{C7}$$

so that $W_1 \subset W$. Let $G_1 \stackrel{\text{def}}{=} \mathrm{gvars}(\bar{s}_1 = \bar{t}_1)$ and $V_1 \stackrel{\text{def}}{=} \mathit{Vars} \setminus G_1$. Note that $G \setminus \{t_1\} \subseteq G_1 \subseteq G$ and, by the assumption for this case, $t_1 \in V$, so that $t_1 \notin G$. As a consequence, $G_1 = G$, $V_1 = V$ and $\bar{s}_1$ is V_1-linear, so that the inductive hypothesis applies to $\bar{s}_1$ and $\bar{t}_1$. Thus, there exists $\mu_1 \in \mathrm{mgs}(\bar{s}_1 = \bar{t}_1) \in \mathit{VSubst}$ such that, for each $z, z' \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$, the three inductive properties hold.

Let $\mu \subseteq \mathit{Eqs}$ be defined as

$$\mu \stackrel{\text{def}}{=} \{t_1 = s_1\mu_1\} \cup \mu_1. \tag{C8}$$

We show that $\mu \in \mathit{VSubst}$. Note that $\nu \stackrel{\text{def}}{=} \{t_1 \mapsto s_1\} \in \mathit{VSubst}$ and, by (C7), $\mathrm{dom}(\nu) \cap \mathrm{vars}(\mu_1) = \{t_1\} \cap \mathrm{vars}(\mu_1) = \varnothing$. Thus, since $\mu_1 \in \mathit{VSubst}$, by Lemma 53 we obtain $\mu = \mu_1 \circ \nu \in \mathit{VSubst}$. Also note that, by Lemma 64, $\mu \in \mathrm{mgs}(\bar{s} = \bar{t})$.

Observe that, as $\bar{s}$ is V-linear, if $t_1 \notin \mathrm{vars}(\bar{s})$ then $\mathrm{vars}(s_1) \cap \mathrm{vars}(\bar{s}_1) \subseteq G_1$. Suppose that $z \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s})$. Then either $z \neq t_1$ so that $z\mu = z\mu_1$ or $z = t_1$ so that $t_1 \notin \mathrm{vars}(\bar{s})$, $\mathrm{vars}(s_1) \cap \mathrm{vars}(\bar{s}_1) \subseteq G_1$ and $z\mu = s_1\mu_1$. We show in each case that $z\mu$ satisfies the three required properties.

1. Suppose $z \neq t_1$. By inductive property (C3), $z\mu_1$ is V_1-linear. As $z\mu = z\mu_1$ and $V = V_1$, $z\mu$ is V-linear.
   Otherwise, let $z = t_1$ so that $\mathrm{vars}(s_1) \cap \mathrm{vars}(\bar{s}_1) \subseteq G_1$ and $z\mu = s_1\mu_1$. Consider an arbitrary variable $u \in \mathrm{vars}(s_1)$. If $u \in G_1$ then $\mathrm{vars}(u\mu_1) \subseteq G_1$. If $u \notin G_1$, then $u \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$ and the inductive properties using μ_1 and $\bar{s}_1$ can be applied to $u\mu_1$. Therefore, by property (C3), $u\mu_1$ is V_1-linear. Moreover, for any $u' \in \mathrm{vars}(s_1)$, $u' \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$ so that, by property (C5), we have $u \neq u' \implies \mathrm{vars}(u\mu_1) \cap \mathrm{vars}(u'\mu_1) \subseteq G_1$. As a consequence, $z\mu = s_1\mu_1$ is V_1-linear. As $V = V_1$, $z\mu$ is V-linear.
2. Suppose $z \neq t_1$. By property (C4), we have $\mathrm{vars}(z\mu_1) \cap \mathrm{dom}(\mu_1) \subseteq G_1$. Since $z\mu = z\mu_1$, $G = G_1$, $\mathrm{dom}(\mu) = \mathrm{dom}(\mu_1) \cup \{t_1\}$ and $t_1 \notin \mathrm{vars}(z\mu)$, we obtain $\mathrm{vars}(z\mu) \cap \mathrm{dom}(\mu) \subseteq G$.
   Otherwise, let $z = t_1$ so that $\mathrm{vars}(s_1) \cap \mathrm{vars}(\bar{s}_1) \subseteq G_1$ and $z\mu = s_1\mu_1$. Consider an arbitrary variable $u \in \mathrm{vars}(s_1)$. If $u \in G_1$ then $\mathrm{vars}(u\mu_1) \subseteq G_1$. If $u \notin G_1$, then $u \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$ and the inductive properties using μ_1 and $\bar{s}_1$ can be applied to $u\mu_1$. By property (C4), $\mathrm{vars}(u\mu_1) \cap \mathrm{dom}(\mu_1) \subseteq G_1$. As this holds for all $u \in \mathrm{vars}(s_1)$, we have $\mathrm{vars}(s_1\mu_1) \cap \mathrm{dom}(\mu_1) \subseteq G_1$. As $z\mu = s_1\mu_1$, $G = G_1$, $\mathrm{dom}(\mu) = \mathrm{dom}(\mu_1) \cup \{t_1\}$ and $t_1 \notin \mathrm{vars}(z\mu)$, we obtain $\mathrm{vars}(z\mu) \cap \mathrm{dom}(\mu) \subseteq G$.
3. Suppose $z \neq t_1$ and let $z' \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s})$ be such that $z \neq z'$. Then, by inductive property (C5), we have $\mathrm{vars}(z\mu_1) \cap \mathrm{vars}(z'\mu_1) \subseteq G_1$. Since $z\mu = z\mu_1$ and $G = G_1$, if also $z' \neq t_1$ (so that $z'\mu = z'\mu_1$) we obtain $\mathrm{vars}(z\mu) \cap \mathrm{vars}(z'\mu) \subseteq G$. Otherwise, let $z' = t_1$ so that $t_1 \notin \mathrm{vars}(\bar{s})$, $\mathrm{vars}(s_1) \cap \mathrm{vars}(\bar{s}_1) \subseteq G_1$ and $z'\mu = s_1\mu_1$. We will show that, for any $u \in \mathrm{vars}(s_1)$,

   $$\mathrm{vars}(z\mu_1) \cap \mathrm{vars}(u\mu_1) \subseteq G_1. \tag{C9}$$

   Note that $u \neq z$ since $z \notin \mathrm{vars}(\bar{s})$. If $u \in G_1$ then $\mathrm{vars}(u\mu_1) \subseteq G_1$ so that (C9) holds. On the other hand, if $u \notin G_1$, then $u \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$ so that (C9) holds by the inductive property (C5) (taking $z' = u$). As (C9) holds for all $u \in \mathrm{vars}(s_1)$, we have $\mathrm{vars}(z\mu_1) \cap \mathrm{vars}(s_1\mu_1) \subseteq G_1$. Thus, by observing that $z\mu = z\mu_1$, $z'\mu = s_1\mu_1$ and $G = G_1$, we can conclude $\mathrm{vars}(z\mu) \cap \mathrm{vars}(z'\mu) \subseteq G$.
   Otherwise, let $z = t_1$ so that $\mathrm{vars}(s_1) \cap \mathrm{vars}(\bar{s}_1) \subseteq G$ and $z\mu = s_1\mu_1$. Let $z' \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s})$ be such that $z \neq z'$ (note that this implies $z' \neq t_1$, so that $z'\mu = z'\mu_1$). We will prove that, for any $u \in \mathrm{vars}(s_1)$,

   $$\mathrm{vars}(u\mu_1) \cap \mathrm{vars}(z'\mu_1) \subseteq G_1. \tag{C10}$$

   Note that $u \neq z'$ since $z' \notin \mathrm{vars}(\bar{s})$. If $u \in G_1$ then $\mathrm{vars}(u\mu_1) \subseteq G_1$ and (C10) holds. Suppose now $u \notin G_1$. Then $u \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$ and the inductive property (C5) can be applied (taking $z = u$) to obtain (C10). As (C10) holds for all $u \in \mathrm{vars}(s_1)$, we have $\mathrm{vars}(s_1\mu_1) \cap \mathrm{vars}(z'\mu_1) \subseteq G_1$. By observing that $z\mu = s_1\mu_1$, $z'\mu = z'\mu_1$ and $G = G_1$ we obtain $\mathrm{vars}(z\mu) \cap \mathrm{vars}(z'\mu) \subseteq G$.

□

For all the proofs of correctness in the next four subsections, as far as possible, we will use the same notation and assume a common set of initial hypotheses. Namely,

we will assume that:

$$\begin{array}{l} d = \langle sh, f, l\rangle \in \mathit{SFL} \text{ and } \sigma \in \gamma_s(d) \cap \mathit{VSubst};\\ \forall y \in \mathrm{dom}(\sigma) \cap \mathrm{range}(\sigma) : y \in \mathrm{vars}(y\sigma);\\ (x \mapsto t) \in \mathit{Bind}, \text{ where } \{x\} \cup \mathrm{vars}(t) \subseteq \mathit{VI};\\ r, r' \in \mathit{HTerms} \text{ are such that } \{r, r'\} = \{x, t\};\\ \sigma \cup \{x = t\} \text{ is satisfiable in the syntactic equality theory } T. \end{array} \tag{C11}$$

Let

$$\begin{array}{rl} \{u_1, \ldots, u_k\} & \stackrel{\mathrm{def}}{=} \mathrm{dom}(\sigma) \cap \big(\mathrm{vars}(x\sigma) \cup \mathrm{vars}(t\sigma)\big),\\ \sigma' & \stackrel{\mathrm{def}}{=} \{u_i = u_i\sigma \mid 1 \leq i \leq k\},\\ \mu & \in \mathrm{mgs}\big(\sigma' \cup \{x\sigma = t\sigma\}\big) \cap \mathit{VSubst},\\ \nu & \stackrel{\mathrm{def}}{=} \sigma \setminus \sigma'. \end{array} \tag{C12}$$

Intuitively, $\sigma' \subseteq \sigma$ (resp., $\nu \subseteq \sigma$) is the set of the bindings of σ that will be affected (resp., will not be affected) by the addition of the equation $x\sigma = t\sigma$. Note that, by Lemma 52, $\nu \in \mathit{VSubst}$. By Lemma 48, $T \vdash \forall\big(\sigma \rightarrow \{x = x\sigma, t = t\sigma\}\big)$ so that, as $\sigma \cup \{x = t\}$ is satisfiable, also $\sigma \cup \{x\sigma = t\sigma\}$ is satisfiable and, a fortiori, $\mathrm{mgs}\big(\sigma' \cup \{x\sigma = t\sigma\}\big) \neq \varnothing$. Therefore, by Proposition 54, μ is well defined. Note also that, since $\sigma \in \mathit{VSubst}$, we have $\mathrm{vars}(u_i\sigma) \subseteq \mathrm{vars}(x\sigma) \cup \mathrm{vars}(t\sigma)$ for each $i = 1$, $\ldots$, k. Therefore, we have

$$\mathrm{vars}(\mu) \subseteq \mathrm{vars}(x\sigma) \cup \mathrm{vars}(t\sigma) \tag{C13}$$

and hence, as $\mathrm{dom}(\nu) = \mathrm{dom}(\sigma) \setminus \big(\mathrm{vars}(x\sigma) \cup \mathrm{vars}(t\sigma)\big)$, we have

$$\mathrm{dom}(\nu) \cap \mathrm{vars}(\mu) = \varnothing. \tag{C14}$$

Thus we can apply Lemma 53 to obtain $\mu \circ \nu \in \mathit{VSubst}$. By applying Lemma 64, we have

$$\mu \circ \nu \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big) \cap \mathit{VSubst}. \tag{C15}$$

The following result is used in Subsections C.4 and C.6.

Lemma 69
Let $\mathrm{rt}(r, \sigma) \in \mathit{LTerms}$. Then $(u_1, \ldots, u_k, r\sigma)$ is $\big(\mathit{Vars} \setminus \mathrm{gvars}(\mu)\big)$-linear.

Proof
Let $\bar{s} = (u_1, \ldots, u_k, r\sigma)$. Then, for each $y \in \mathrm{vars}(r)$, $\mathrm{rt}(y, \sigma) \in \mathit{LTerms}$ so that, by Proposition 19, $y \in \mathrm{lvars}(\sigma)$. It follows that $\mathrm{vars}(r) \subseteq \mathrm{lvars}(\sigma)$ and hence, by Proposition 56,

$$\begin{array}{c} \forall v \in \mathrm{vars}(r\sigma) \setminus \mathrm{dom}(\sigma) : \mathrm{occ_lin}(v, r\sigma),\\ \mathrm{vars}(r\sigma) \cap \mathrm{dom}(\sigma) \subseteq \mathrm{gvars}(\sigma). \end{array}$$

Hence $\bar{s}$ is $\big(\mathit{Vars} \setminus \mathrm{gvars}(\sigma)\big)$-linear. As $\mathrm{gvars}(\sigma') \subseteq \mathrm{gvars}(\mu)$ and $\mathrm{gvars}(\sigma) \cap \mathrm{vars}(\bar{s}) = \mathrm{gvars}(\sigma')$, we have $\mathrm{gvars}(\sigma) \cap \mathrm{vars}(\bar{s}) \subseteq \mathrm{gvars}(\mu)$. Therefore $\bar{s}$ is $\big(\mathit{Vars} \setminus \mathrm{gvars}(\mu)\big)$-linear. □

The following result is used in Subsections C.5 and C.6.

Lemma 70
Assume the notation and hypotheses given in (C11) and (C12) and let $y \in \mathrm{lvars}(\sigma) \setminus \mathrm{gvars}(\sigma)$. Then $y \in \mathrm{dom}(\mu)$ implies that $y \notin \mathrm{dom}(\sigma)$.

Proof
Suppose $y \in \mathrm{dom}(\sigma)$. Then we show that $y \notin \mathrm{dom}(\mu)$. Since $\sigma \in \mathit{VSubst}$ and $y \in \mathrm{lvars}(\sigma)$, by Proposition 56 we have

$$\mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma) \subseteq \mathrm{gvars}(\sigma).$$

As $y \notin \mathrm{gvars}(\sigma)$, it follows that $y \notin \mathrm{vars}(y\sigma)$. Hence, by the definition of σ, $y \notin \mathrm{dom}(\sigma) \cap \mathrm{range}(\sigma)$ so that $y \in \mathrm{dom}(\sigma) \setminus \mathrm{range}(\sigma)$. Therefore $y \notin \mathrm{vars}(x\sigma) \cup \mathrm{vars}(t\sigma)$ so that, by (C13), $y \notin \mathrm{dom}(\mu)$. □

In Subsections C.4 and C.7, we use the following subsets of sh.

$$\begin{aligned} sh_x &= \mathrm{rel}\big(\{x\}, sh\big), & sh_t &= \mathrm{rel}\big(\mathrm{vars}(t), sh\big), \\ sh_r &= \mathrm{rel}\big(\mathrm{vars}(r), sh\big), & sh_{r'} &= \mathrm{rel}\big(\mathrm{vars}(r'), sh\big), \\ sh_- &= \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), sh\big), & sh_{xt} &= sh_x \cap sh_t. \end{aligned} \tag{C16}$$

We also use the corresponding subsets of $\mathrm{ssets}(\sigma)$.

$$\begin{aligned} R_x &= \mathrm{rel}\big(\{x\}, \mathrm{ssets}(\sigma)\big), & R_t &= \mathrm{rel}\big(\mathrm{vars}(t), \mathrm{ssets}(\sigma)\big), \\ R_r &= \mathrm{rel}\big(\mathrm{vars}(r), \mathrm{ssets}(\sigma)\big), & R_{r'} &= \mathrm{rel}\big(\mathrm{vars}(r'), \mathrm{ssets}(\sigma)\big), \\ R_- &= \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), \mathrm{ssets}(\sigma)\big), & R_{xt} &= R_x \cap R_t. \end{aligned} \tag{C17}$$

C.4 The Correctness for Set-Sharing

Lemma 71
Assume the notation and hypotheses in (C11), (C12) and (C17) and suppose that $\mathrm{rt}(r, \sigma) \in \mathit{LTerms}$. Let also $\tau = \mu \circ \nu$, $S \in \mathrm{ssets}(\tau)$ and $S_{r'} \in R^\star_{r'}$, where $S_{r'} \subseteq S$. Then $S_{r'} \in R_{r'} \cup \mathrm{bin}(R_{r'}, R^\star_{xt})$.

Proof
As $S_{r'} \in R^\star_{r'}$, $S_{r'} = S_1 \cup S_2$ where $S_1 \in (R_{r'} \setminus R_{xt})^\star \cup \{\varnothing\}$ and $S_2 \in R^\star_{xt} \cup \{\varnothing\}$. Note that as $S_{r'} \neq \varnothing$, we cannot have $S_1 = S_2 = \varnothing$. Suppose first that $S_1 = \varnothing$ so that $S_{r'} = S_2 \neq \varnothing$. Then $S_{r'} \in R^\star_{xt}$. However, since $R_{xt} \subseteq R_{r'}$, $R^\star_{xt} \subseteq \mathrm{bin}(R_{r'}, R^\star_{xt})$. Thus $S_{r'} \in \mathrm{bin}(R_{r'}, R^\star_{xt})$. Suppose next that $S_1 \neq \varnothing$. As $R_{r'} \setminus R_{xt} = R_{r'} \setminus R_r$, we have $S_1 \subseteq \bigcup \big\{\, \mathrm{occ}(\sigma, w) \;\big|\; w \in \big(\mathrm{vars}(r'\sigma) \setminus \big(\mathrm{dom}(\sigma) \cup \mathrm{vars}(r\sigma)\big)\big) \,\big\}$ and hence, $S_1 = \bigcup sh_1$ where

$$sh_1 = \big\{\, \mathrm{occ}(\sigma, w) \;\big|\; w \in \big(S_1 \cap \mathrm{vars}(r'\sigma)\big) \setminus \big(\mathrm{dom}(\sigma) \cup \mathrm{vars}(r\sigma)\big) \,\big\}.$$

Let $w_1, w_2 \in S_1 \setminus \big(\mathrm{dom}(\sigma) \cup \mathrm{vars}(r\sigma)\big)$. Then we show that $w_1 = w_2$. By (C15), $\tau \in \mathit{VSubst}$ so that, as $S_1 \subseteq S$ and $S \in \mathrm{ssets}(\tau)$, by case (B1) of Proposition 56, $\big(\mathrm{vars}(w_1\tau) \cap \mathrm{vars}(w_2\tau)\big) \setminus \mathrm{dom}(\tau) \neq \varnothing$. Note that $\mathrm{dom}(\mu) \subseteq \mathrm{dom}(\tau)$ and, as $w_i \notin \mathrm{dom}(\sigma)$, we have $w_i\tau = w_i\mu$, for $i \in \{1, 2\}$. Thus, $\big(\mathrm{vars}(w_1\mu) \cap \mathrm{vars}(w_2\mu)\big) \setminus \mathrm{dom}(\mu) \neq \varnothing$. By definition, $\mu \in \mathit{VSubst}$ so that, by case (B1) of Proposition 56,

$$\mathrm{rel}\big(\{w_1\}, \mathrm{ssets}(\mu)\big) \cap \mathrm{rel}\big(\{w_2\}, \mathrm{ssets}(\mu)\big) \neq \varnothing.$$

By definition, $u_1, \ldots, u_k \in \mathrm{dom}(\sigma)$ so that $w_1, w_2 \notin \{u_1, \ldots, u_k\} \cup \mathrm{vars}(r\sigma)$. By hypothesis, $\mathrm{rt}(r, \sigma) \in \mathit{LTerms}$. Therefore Lemma 69 can be applied to obtain $(u_1, \ldots, u_k, r\sigma)$ is $\big(\mathit{Vars} \setminus \mathrm{gvars}(\mu)\big)$-linear. Thus, we can apply property (C2) of Proposition 68. to conclude that $w_1 = w_2$.

As a consequence $\#sh_1 \leq 1$. Thus, as $S_1 \neq \varnothing$, $\#sh_1 = 1$ and $S_1 = \mathrm{occ}(\sigma, w)$ for some $w \in \mathrm{vars}(r'\sigma) \setminus \big(\mathrm{dom}(\sigma) \cup \mathrm{vars}(r\sigma)\big)$. It follows that $S_1 \in R_{r'}$ as required. □

Proposition 72
Assume the notation and hypotheses given in (C11) and (C16) and suppose that $\mathrm{free}_d(r)$ holds. Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$,

$$sh_- \cup \mathrm{bin}(sh_r, sh_{r'}) \supseteq \mathrm{ssets}(\tau). \tag{C18}$$

Proof
Assuming the definitions in (C12), we show that $\tau \stackrel{\mathrm{def}}{=} \mu \circ \nu$ satisfies (C18).

Since $\sigma \in \gamma_S(d)$, we have $sh \supseteq \mathrm{ssets}(\sigma)$ so that, using the monotonicity of $\overline{\mathrm{rel}}$, rel, $(\cdot)^\star$ and bin and assuming the notation given in (C17),

$$sh_- \cup \mathrm{bin}(sh_r, sh_{r'}) \supseteq R_- \cup \mathrm{bin}(R_r, R_{r'}).$$

Let S be an arbitrary sharing set in $\mathrm{ssets}(\tau)$. Then, in order to prove (C18), it is sufficient to show that

$$S \in R_- \cup \mathrm{bin}(R_r, R_{r'}). \tag{C19}$$

By hypothesis, $\mathrm{free}_d(r)$ holds so that, by Theorem 24, $\mathrm{rt}(r, \sigma) \in \mathit{Vars}$.

By Definition 62 and Theorem 63, $S \in R_- \cup \mathrm{bin}(R^\star_r, R^\star_{r'})$. If $S \in R_-$, then (C19) holds trivially. Therefore suppose, for some $S_r \in R^\star_r$ and $S_{r'} \in R^\star_{r'}$, $S = S_r \cup S_{r'}$. As $\mathrm{rt}(r, \sigma) \in \mathit{LTerms}$, we can apply Lemma 71 to obtain $S_{r'} \in R_{r'} \cup \mathrm{bin}(R_{r'}, R^\star_{xt})$. However, as $R_{xt} \subseteq R_r$, $\mathrm{bin}\big(R^\star_r, R_{r'} \cup \mathrm{bin}(R_{r'}, R^\star_{xt})\big) = \mathrm{bin}(R^\star_r, R_{r'})$. Thus $S \in \mathrm{bin}(R^\star_r, R_{r'})$.

As $\mathrm{rt}(r, \sigma) \in \mathit{Vars}$, by Propositions 19 and 56, $r\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. Therefore, by Definition 10, $R_r = \big\{\mathrm{occ}(\sigma, r\sigma)\big\}$ so that $R^\star_r = R_r$. Thus $S \in \mathrm{bin}(R_r, R_{r'})$ and (C19) holds. □

Proposition 73
Assume the notation and hypotheses given in (C11) and (C16) and suppose that both $\mathrm{lin}_d(x)$ and $\mathrm{lin}_d(t)$ hold. Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$,

$$sh_- \cup \mathrm{bin}\big(sh_x \cup \mathrm{bin}(sh_x, sh^\star_{xt}), sh_t \cup \mathrm{bin}(sh_t, sh^\star_{xt})\big) \supseteq \mathrm{ssets}(\tau). \tag{C20}$$

Proof
Assuming the definitions in (C12), we show that $\tau \stackrel{\mathrm{def}}{=} \mu \circ \nu$ satisfies (C20).

Since $\sigma \in \gamma_S(d)$, we have $sh \supseteq \mathrm{ssets}(\sigma)$ so that, using the monotonicity of $\overline{\mathrm{rel}}$, rel, $(\cdot)^\star$ and bin and assuming the notation given in (C17), we obtain

$$sh_- \cup \mathrm{bin}\big(sh_x \cup \mathrm{bin}(sh_x, sh^\star_{xt}), sh_t \cup \mathrm{bin}(sh_t, sh^\star_{xt})\big) \supseteq R_- \cup \mathrm{bin}\big(R_x \cup \mathrm{bin}(R_x, R^\star_{xt}), R_t \cup \mathrm{bin}(R_t, R^\star_{xt})\big).$$

Let S be an arbitrary sharing set in $\mathrm{ssets}(\tau)$. Then, in order to prove (C20), it is sufficient to show

$$S \in R_- \cup \mathrm{bin}\big(R_x \cup \mathrm{bin}(R_x, R^\star_{xt}), R_t \cup \mathrm{bin}(R_t, R^\star_{xt})\big) \tag{C21}$$

By Definition 62 and Theorem 63, we have

$$S \in R_- \cup \mathrm{bin}(R^\star_x, R^\star_t).$$

If $S \in R_-$, then (C21) holds trivially. Therefore suppose $S \in \mathrm{bin}(R^\star_x, R^\star_t)$. That is, there are $S_x \in R^\star_x$ and $S_t \in R^\star_t$ such that $S = S_x \cup S_t$. By hypothesis, $\mathrm{lin}_d(x)$ and $\mathrm{lin}_d(t)$ hold. Hence, by Theorem 24, $\mathrm{rt}(x, \sigma)$ and $\mathrm{rt}(t, \sigma) \in \mathit{LTerms}$ so that we can apply Lemma 71 twice, once with $r = x$ and $r' = t$ and once with $r = t$ and $r' = x$ to obtain $S_x \in R_x \cup \mathrm{bin}(R_x, R^\star_{xt})$ and $S_t \in R_t \cup \mathrm{bin}(R_t, R^\star_{xt})$. Thus (C21) holds. □

Proposition 74
Assume the notation and hypotheses given in (C11) and (C16) and suppose $\mathrm{lin}_d(r)$ holds. Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$,

$$sh_- \cup \mathrm{bin}(sh^\star_r, sh_{r'}) \supseteq \mathrm{ssets}(\tau). \tag{C22}$$

Proof
Assuming the definitions in (C12), we show that $\tau \stackrel{\mathrm{def}}{=} \mu \circ \nu$ satisfies (C22).

Since $\sigma \in \gamma_S(d)$, we have $sh \supseteq \mathrm{ssets}(\sigma)$ so that, using the monotonicity of $\overline{\mathrm{rel}}$, rel, $(\cdot)^\star$ and bin and assuming the notation given in (C17), we obtain

$$sh_- \cup \mathrm{bin}(sh^\star_r, sh_{r'}) \supseteq R_- \cup \mathrm{bin}(R^\star_r, R_{r'}).$$

Let S be an arbitrary sharing set in $\mathrm{ssets}(\tau)$. Then, in order to prove (C22) it is sufficient to show that

$$S \in R_- \cup \mathrm{bin}(R^\star_r, R_{r'}). \tag{C23}$$

By Definition 62 and Theorem 63, we have

$$S \in R_- \cup \mathrm{bin}(R^\star_r, R^\star_{r'}).$$

If $S \in R_-$, then (C23) holds trivially. Therefore suppose $S \in \mathrm{bin}(R^\star_r, R^\star_{r'})$. That is, there are $S_r \in R^\star_r$ and $S_{r'} \in R^\star_{r'}$ such that $S = S_r \cup S_{r'}$. By hypothesis, $\mathrm{lin}_d(r)$ holds. Hence, by Theorem 24, $\mathrm{rt}(r, \sigma) \in \mathit{LTerms}$ so that we can apply Lemma 71, to obtain $S_{r'} \in R_{r'} \cup \mathrm{bin}(R_{r'}, R^\star_{xt})$. However, as $R_{xt} \subseteq R_r$, $\mathrm{bin}\big(R^\star_r, R_{r'} \cup \mathrm{bin}(R_{r'}, R^\star_{xt})\big) = \mathrm{bin}(R^\star_r, R_{r'})$. Thus $S \in \mathrm{bin}(R^\star_r, R_{r'})$ and (C23) holds. □

To prove the main correctness result for set-sharing in the case that the binding is cyclic, we need the following lemma.

Lemma 75
Let $sh \in \mathit{SH}$ and $V, W \subseteq \mathit{VI}$ where $\mathrm{rel}(V, sh) \subseteq \mathrm{rel}(W, sh)$. Let $x \mapsto t \in \mathit{Bind}$ where $\{x\} \cup \mathrm{vars}(t) \subseteq \mathit{VI}$ and $sh' \stackrel{\mathrm{def}}{=} \mathrm{amgu}(sh, x \mapsto t)$. Then $\mathrm{rel}(V, sh') \subseteq \mathrm{rel}(W, sh')$.

Proof
Suppose that $S \in \mathrm{rel}(V, sh')$. By Definition 62, we have two cases.

1. Suppose first that $S \in \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), sh\big)$. Then, $S \in sh$ and, in particular, $S \in \mathrm{rel}(V, sh)$. Thus, by hypothesis, $S \in \mathrm{rel}(W, sh)$ and we can conclude $S \in \mathrm{rel}(W, sh')$.
2. Otherwise, let $S \in \mathrm{bin}\Big(\mathrm{rel}\big(\{x\}, sh\big)^\star, \mathrm{rel}\big(\mathrm{vars}(t), sh\big)^\star\Big)$. Then, it holds $S = S_0 \cup \cdots \cup S_n$ where $n \in \mathbb{N}$ and $S_i \in sh$, for each $0 \leq i \leq n$. Moreover, since $S \in \mathrm{rel}(V, sh')$, there exists an index $j \in \{0, \ldots, n\}$ such that $S_j \in \mathrm{rel}(V, sh)$. Hence, by the hypothesis, $S_j \in \mathrm{rel}(W, sh)$ and, since $S_j \subseteq S$, we can conclude $S \in \mathrm{rel}(W, sh')$.

□

Proposition 76
Let $sh \in SH$ and $\tau \in RSubst$ be satisfiable in T and suppose that $T \vdash \forall(\tau \to x = t)$ and $\mathrm{ssets}(\tau) \subseteq sh$. Then $\mathrm{ssets}(\tau) \subseteq \mathrm{cyclic}^t_x(sh)$.

Proof
Take $V_x = \{x\}$ and $W_t = \mathrm{vars}(t) \setminus \{x\}$. Let $\tau = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$, where $n = \#\tau$. Define

$$\tau_0 \stackrel{\mathrm{def}}{=} \{x \mapsto t\}, \qquad sh_0 \stackrel{\mathrm{def}}{=} \mathrm{ssets}\big(\{x \mapsto t\}\big),$$

and, for each $i = 1, \ldots, n$,

$$\tau_i \in \mathrm{mgs}\big(\{x_1 = t_1, \ldots, x_i = t_i\} \cup \{x = t\}\big), \qquad sh_i \stackrel{\mathrm{def}}{=} \mathrm{amgu}(sh_{i-1}, x_i \mapsto t_i).$$

We show by induction on $i = 0, \ldots, n$ that $\mathrm{ssets}(\tau_i) \subseteq sh_i$ and

$$\mathrm{rel}(V_x, sh_i) \subseteq \mathrm{rel}(W_t, sh_i). \tag{C24}$$

The base case, when $i = 0$, follows directly from Definition 10; note that (C24) holds because $x \in \mathrm{dom}(\tau_0)$, so that $\mathrm{occ}(\tau_0, x) = \varnothing$.

Consider the inductive case, when $0 < i \leq n$. By the inductive hypothesis, $\mathrm{ssets}(\tau_{i-1}) \subseteq sh_{i-1}$ so that, by Theorem 63, $\mathrm{ssets}(\tau_i) \subseteq sh_i$. By the inductive hypothesis, (C24) holds for sh_{i-1}. Thus, by Lemma 75 (taking $V = V_x$ and $W = W_t$), we obtain that (C24) also holds for sh_i.

By taking $i = n$, we obtain $\mathrm{rel}(V_x, sh_n) \subseteq \mathrm{rel}(W_t, sh_n)$. Note that, by hypothesis, we have $\tau \in \mathrm{mgs}\big(\tau \cup \{x = t\}\big) = \mathrm{mgs}(\tau_n)$, so that $T \vdash \forall(\tau \leftrightarrow \tau_n)$. By Lemma 67, we have $\mathrm{ssets}(\tau) = \mathrm{ssets}(\tau_n)$, so that $\mathrm{ssets}(\tau) \subseteq sh_n$. As a consequence, $\mathrm{ssets}(\tau) \subseteq sh \cap sh_n$. Thus, by Definition 23, we obtain $\mathrm{ssets}(\tau) \subseteq \mathrm{cyclic}^t_x(sh)$. □

C.5 The Correctness for Freeness

Lemma 77
Assume the notation and definitions given in (C11) and (C12) and suppose $\tau = \mu \circ \nu$ and $y \in \mathrm{fvars}(\sigma)$ be such that $y\sigma \in \mathrm{fvars}(\mu)$. Then $y \in \mathrm{fvars}(\tau)$.

Proof
If $y \notin \mathrm{dom}(\tau)$, then $y \in \mathrm{fvars}(\tau)$. Suppose now that $y \in \mathrm{dom}(\tau)$. By hypothesis, $y \in \mathrm{fvars}(\sigma)$ so that $y \in \mathrm{lvars}(\sigma) \setminus \mathrm{gvars}(\sigma)$ and Lemma 70 applies. Thus, if $y \in \mathrm{dom}(\mu)$, $y \notin \mathrm{dom}(\sigma)$ and hence $y\tau = y\sigma\mu$. In addition, as $\nu \subseteq \sigma$, if $y \in \mathrm{dom}(\nu)$, then again $y\tau = y\sigma\mu$.

As $y\sigma \in \mathrm{fvars}(\mu)$ and $\mu \in \mathit{VSubst}$, by case B3 of Proposition 56, $y\sigma\mu \in \mathit{Vars} \setminus \mathrm{dom}(\mu)$. If $y\sigma \in \mathrm{dom}(\mu)$, then $y\sigma\mu \in \mathrm{vars}(\mu)$, so that, by (C14), $y\sigma\mu \notin \mathrm{dom}(\nu)$. On the other hand, if $y\sigma \notin \mathrm{dom}(\mu)$, then $y\sigma = y\sigma\mu$. Since $y \in \mathrm{fvars}(\sigma)$ and $\sigma \in \mathit{VSubst}$, by case B3 of Proposition 56, $y\sigma \notin \mathrm{dom}(\sigma)$. Thus, in this case, $y\sigma\mu \notin \mathrm{dom}(\sigma)$ and hence, as $\nu \subseteq \sigma$, $y\sigma\mu \notin \mathrm{dom}(\nu)$. Therefore, for any y satisfying the hypothesis, $y\sigma\mu \in \mathit{Vars} \setminus \big(\mathrm{dom}(\nu) \cup \mathrm{dom}(\mu)\big)$ and hence, as $\mathrm{dom}(\tau) \subseteq \mathrm{dom}(\nu) \cup \mathrm{dom}(\mu)$, $y\sigma\mu \in \mathit{Vars} \setminus \mathrm{dom}(\tau)$. Since, by (C15), $\tau \in \mathit{VSubst}$, we can apply Proposition 56 to obtain $y \in \mathrm{fvars}(\tau)$ as required. □

Proposition 78
Assume the notation and hypotheses given in (C11) and suppose $\{x, t\} \subseteq \mathrm{fvars}(\sigma)$. Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$, we have

$$\mathrm{fvars}(\sigma) \subseteq \mathrm{fvars}(\tau). \tag{C25}$$

Proof
Assuming the definitions in (C12), we show that $\tau \stackrel{\mathrm{def}}{=} \mu \circ \nu$ satisfies (C25). Suppose $y \in \mathrm{fvars}(\sigma)$. As $\{x, t\} \subseteq \mathrm{fvars}(\sigma)$ we have, using Proposition 56, $\{x\sigma, t\sigma\} \subseteq \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. Thus $\mu \in \mathrm{mgs}\big(\{x\sigma = t\sigma\}\big) \cap \mathit{VSubst}$. As a single binding is in VSubst, we can assume that either $x\sigma \neq t\sigma$ and $\mu = \{x\sigma \mapsto t\sigma\}$ or $\mu = \varnothing$. In both cases, $\mathrm{fvars}(\mu) = \mathit{Vars}$ so that $y\sigma \in \mathrm{fvars}(\mu)$. Therefore Lemma 77 can be applied to give $y \in \mathrm{fvars}(\tau)$. □

The following simple consequence of Proposition 56 will be used to complete the proofs of correctness for the freeness component.

Lemma 79
Suppose $y \in \mathit{VI}$ and $s \in \mathit{HTerms}$ are such that $\mathrm{vars}(s) \subseteq \mathit{VI}$. Suppose also that $y \notin \mathrm{share_with}_d(s)$. Then $\mathrm{vars}(y\sigma) \cap \mathrm{vars}(s\sigma) \subseteq \mathrm{dom}(\sigma)$.

Proof
Let $V = \mathrm{vars}(s)$ so that, by Definition 23, $y \notin \mathrm{vars}\big(\mathrm{rel}(V, \mathit{sh})\big)$. Thus

$$\forall w \in V, S \in \mathit{sh} : \{y, w\} \not\subseteq S.$$

By Definition 21, since $\sigma \in \gamma_s(d)$, this implies

$$\forall v \in \mathit{Vars}, w \in V : \{y, w\} \not\subseteq \mathrm{occ}(\sigma, v).$$

Thus, since $\sigma \in \mathit{VSubst}$, by Proposition 56 we obtain

$$\forall w \in V : \mathrm{vars}(y\sigma) \cap \mathrm{vars}(w\sigma) \subseteq \mathrm{dom}(\sigma),$$

which is equivalent to the thesis. □

Proposition 80
Assume the notation and hypotheses given in (C11) and suppose that $x \in \text{fvars}(\sigma)$. Then, for some $\tau \in \text{mgs}\big(\sigma \cup \{x = t\}\big)$, we have

$$\text{fvars}(\sigma) \setminus \text{share_with}_d(x) \subseteq \text{fvars}(\tau). \tag{C26}$$

Proof
Assuming the definitions in (C12), we show that $\tau \stackrel{\text{def}}{=} \mu \circ \nu$ satisfies (C26). If $x\sigma = t\sigma$, then $t \in \text{fvars}(\sigma)$ and the result follows from Proposition 78. Suppose next that $x\sigma \neq t\sigma$ and that $y \in \text{fvars}(\sigma) \setminus \text{share_with}_d(x)$. We prove that $y \in \text{fvars}(\tau)$. First we show that $y\sigma \in \text{fvars}(\mu)$.

As $x \in \text{fvars}(\sigma)$ and $\sigma \in \mathit{VSubst}$, by Proposition 56, $x\sigma \in \mathit{Vars} \setminus \text{dom}(\sigma)$. Thus $\mu' \stackrel{\text{def}}{=} \sigma' \cup \{x\sigma = t\sigma\}$ has neither identities nor circular subsets so that $\mu' \in \mathit{RSubst}$. We show that $y\sigma \in \text{fvars}(\mu')$. Moreover, $\sigma \in \mathit{VSubst}$, so that, by Proposition 56, $y\sigma \notin \text{dom}(\sigma)$. However, since $y \notin \text{share_with}_d(x)$ and $\sigma \in \mathit{VSubst}$, we can apply Lemma 79 to give $\text{vars}(y\sigma) \cap \text{vars}(x\sigma) \subseteq \text{dom}(\sigma)$. Therefore $y\sigma \neq x\sigma$ and hence $y\sigma \notin \text{dom}(\sigma) \cup \{x\sigma\}$. Thus $y\sigma \notin \text{dom}(\mu')$ so that $y\sigma \in \text{fvars}(\mu')$. By Lemma 67, we have $y\sigma \in \text{fvars}(\mu)$.

Therefore Lemma 77 can be applied to give $y \in \text{fvars}(\tau)$. □

Proposition 81
Assume the notation and hypotheses given in (C11). Therefore we have, for some $\tau \in \text{mgs}\big(\sigma \cup \{x = t\}\big)$,

$$\text{fvars}(\sigma) \setminus \big(\text{share_with}_d(x) \cup \text{share_with}_d(t)\big) \subseteq \text{fvars}(\tau). \tag{C27}$$

Proof
Assuming the definitions in (C12), we show that $\tau \stackrel{\text{def}}{=} \mu \circ \nu$ satisfies (C27). Suppose $y \in \text{fvars}(\sigma) \setminus \big(\text{share_with}_d(x) \cup \text{share_with}_d(t)\big)$. Then we need to show that $y \in \text{fvars}(\tau)$.

As $y \in \text{fvars}(\sigma)$, by (B3) of Proposition 56, $y\sigma \in \mathit{Vars} \setminus \text{dom}(\sigma)$. Thus, as

$$y \notin \text{share_with}_d(x) \cup \text{share_with}_d(t),$$

we can apply Lemma 79 to obtain

$$y\sigma \notin \text{vars}(x\sigma) \cup \text{vars}(t\sigma)$$

so that, by (C13), $y\sigma \notin \text{vars}(\mu)$. It follows that $y\sigma \in \text{fvars}(\mu)$. Therefore Lemma 77 can be applied to give $y \in \text{fvars}(\tau)$ as required. □

C.6 The Correctness for Linearity

Lemma 82
Assume the notation and definitions given in (C11) and (C12). Suppose also that $\tau = \mu \circ \nu$, $W \subseteq \text{lvars}(\mu)$ and $y \in \text{lvars}(\sigma) \cap W$ where $\text{vars}(y\sigma) \setminus W \subseteq \text{gvars}(\sigma)$ and, for all $w, w' \in W$ such that $w \neq w'$, we have $\text{vars}(w\mu) \cap \text{vars}(w'\mu) \subseteq \text{gvars}(\mu)$. Then $y \in \text{lvars}(\tau)$.

Proof
By Lemma 66, we have $\mathrm{gvars}(\sigma) \subseteq \mathrm{gvars}(\tau)$ and $\mathrm{gvars}(\mu) \subseteq \mathrm{gvars}(\tau)$.

If $y \notin \mathrm{dom}(\tau)$ then $y \in \mathrm{fvars}(\tau)$. Also, if $y \in \mathrm{gvars}(\sigma)$ then $y \in \mathrm{gvars}(\tau)$. Thus, in both cases, by Corollary 20, $y \in \mathrm{lvars}(\tau)$. Therefore, for the rest of the proof, we assume $y \in \mathrm{dom}(\tau) \setminus \mathrm{gvars}(\sigma)$.

By Lemma 70, if $y \in \mathrm{dom}(\mu)$, then $y \notin \mathrm{dom}(\sigma)$ and hence $y\tau = y\sigma\mu$. Alternatively, if $y \in \mathrm{dom}(\nu)$, then, since $\nu \subseteq \sigma$, again $y\tau = y\sigma\mu$. Thus, as $\mathrm{dom}(\tau) \subseteq \mathrm{dom}(\nu) \cup \mathrm{dom}(\mu)$,

$$y\tau = y\sigma\mu. \tag{C28}$$

By definition $\mu \in \mathit{VSubst}$. Thus, by Proposition 56, for all $w \in \mathrm{lvars}(\mu)$ (and hence, for all $w \in W$),

$$\forall v \in \mathrm{vars}(w\mu) \setminus \mathrm{dom}(\mu) : \mathrm{occ_lin}(v, w\mu), \tag{C29}$$

$$\mathrm{vars}(w\mu) \cap \mathrm{dom}(\mu) \subseteq \mathrm{gvars}(\mu). \tag{C30}$$

By definition, $\sigma \in \mathit{VSubst}$. Thus, as $y \in \mathrm{lvars}(\sigma)$, by Proposition 56,

$$\forall v \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) : \mathrm{occ_lin}(v, y\sigma), \tag{C31}$$

$$\mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma) \subseteq \mathrm{gvars}(\sigma). \tag{C32}$$

By (C15), $\tau \in \mathit{VSubst}$. Thus, by Proposition 56, to prove $y \in \mathrm{lvars}(\tau)$, we just need to show that

$$\forall v \in \mathrm{vars}(y\tau) \setminus \mathrm{dom}(\tau) : \mathrm{occ_lin}(v, y\tau), \tag{C33}$$

$$\mathrm{vars}(y\tau) \cap \mathrm{dom}(\tau) \subseteq \mathrm{gvars}(\tau). \tag{C34}$$

As $y \notin \mathrm{gvars}(\sigma)$, $\mathrm{vars}(y\sigma) \neq \varnothing$. Let w be any variable in $\mathrm{vars}(y\sigma)$. Since (C32) holds and $y \in \mathrm{lvars}(\sigma) \setminus \mathrm{gvars}(\sigma)$, $w \notin \mathrm{dom}(\sigma)$. Thus, by (C31), $\mathrm{occ_lin}(w, y\sigma)$. Then, it follows from (C28) that to prove (C33), we need to prove:

$$\forall v \in \mathrm{vars}(w\mu) \setminus \mathrm{dom}(\tau) : \mathrm{occ_lin}(v, w\mu), \tag{C35}$$

$$\forall w' \in \mathrm{vars}(y\sigma) \setminus \{w\} : \mathrm{vars}(w\mu) \cap \mathrm{vars}(w'\mu) \subseteq \mathrm{dom}(\tau), \tag{C36}$$

and, to prove (C34), we need to prove:

$$\mathrm{vars}(w\mu) \cap \mathrm{dom}(\tau) \subseteq \mathrm{gvars}(\tau). \tag{C37}$$

We first prove (C37) and (C35) hold. Suppose $w \in \mathrm{vars}(y\sigma) \setminus W$. Then, by hypothesis, $w \in \mathrm{gvars}(\sigma)$ and hence, $w \in \mathrm{gvars}(\tau)$. As $T \vdash \forall(\tau \to \mu)$, by Lemma 48, $T \vdash \forall\big(\tau \to (w = w\mu)\big)$; hence, by Lemma 49, $\mathrm{rt}(w, \tau) = \mathrm{rt}(w\mu, \tau)$; and hence, by Proposition 19, $\mathrm{vars}(w\mu) \subseteq \mathrm{gvars}(\tau)$. Thus, as $\mathrm{gvars}(\tau) \subseteq \mathrm{dom}(\tau)$, (C37) and (C35) hold. Suppose now that $w \in \mathrm{vars}(y\sigma) \cap W$ so that (C29) and (C30) hold. Thus, as $\mathrm{dom}(\mu) \subseteq \mathrm{dom}(\tau)$ and $\mathrm{gvars}(\mu) \subseteq \mathrm{gvars}(\tau)$, both (C37) and (C35) hold.

To prove (C36), let $w' \in \mathrm{vars}(y\sigma) \setminus \{w\}$. By (C34) and as $\mathrm{gvars}(\tau) \subseteq \mathrm{dom}(\tau)$, we just need to show that $\mathrm{vars}(w\mu) \cap \mathrm{vars}(w'\mu) \subseteq \mathrm{gvars}(\tau)$. If $\{w, w'\} \subseteq W$ then, by the hypothesis and as $\mathrm{gvars}(\mu) \subseteq \mathrm{gvars}(\tau)$, $\mathrm{vars}(w\mu) \cap \mathrm{vars}(w'\mu) \subseteq \mathrm{gvars}(\tau)$. If, however, $w \notin W$, then as $w \in \mathrm{vars}(y\sigma)$, by the hypothesis, $w \in \mathrm{gvars}(\sigma)$ and hence, by the same argument as in the previous paragraph, $\mathrm{vars}(w\mu) \subseteq \mathrm{gvars}(\tau)$. Similarly, if $w' \notin W$, $\mathrm{vars}(w'\mu) \subseteq \mathrm{gvars}(\tau)$. Thus, in all cases, (C36) holds. □

Proposition 83

Assume the notation and hypotheses given in (C11) and suppose $\mathrm{lin}_d(r)$ holds. Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$,

$$\mathrm{lvars}(\sigma) \setminus \mathrm{share_with}_d(r) \subseteq \mathrm{lvars}(\tau). \tag{C38}$$

Proof

Assuming the definitions in (C12), we show that $\tau \stackrel{\mathrm{def}}{=} \mu \circ \nu$ satisfies (C38). To prove this, we assume that $y \in \mathrm{lvars}(\sigma) \setminus \mathrm{share_with}_d(r)$ and show that $y \in \mathrm{lvars}(\tau)$. Suppose $y \in \mathrm{gvars}(\sigma)$. Then, by Lemma 66, $y \in \mathrm{gvars}(\tau)$ so that $y \in \mathrm{lvars}(\tau)$. For the rest of the proof, we assume that $y \notin \mathrm{gvars}(\sigma)$. Let $\bar{s} \stackrel{\mathrm{def}}{=} (u_1, \ldots, u_k, r\sigma)$. We will show that the hypothesis of Lemma 82 holds where $W = \mathrm{vars}(\mu) \setminus \mathrm{vars}(\bar{s})$.

As $y \in \mathrm{lvars}(\sigma)$, by Proposition 56, $\mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma) \subseteq \mathrm{gvars}(\sigma)$ so that, as $y \notin \mathrm{gvars}(\sigma)$, $y \notin \mathrm{vars}(y\sigma)$. By (C11), $y \notin \mathrm{dom}(\sigma) \cap \mathrm{range}(\sigma)$ so that $y \notin \{u_1, \ldots, u_k\}$. As $y \notin \mathrm{share_with}_d(r)$, by Definition 23 and Theorem 24, we have $y \notin \mathrm{vars}(r\sigma)$. Thus $y \notin \mathrm{vars}(\bar{s})$ and hence, $y \in W$. Therefore we have $y \in \mathrm{lvars}(\sigma) \cap W$.

By (C12), $\mathrm{vars}\big(\{u_1, \ldots, u_k\}\big) \subseteq \mathrm{dom}(\sigma)$ so that, as we have already shown that $\mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma) \subseteq \mathrm{gvars}(\sigma)$, we have $\mathrm{vars}(y\sigma) \cap \mathrm{vars}\big(\{u_1, \ldots, u_k\}\big) \subseteq \mathrm{gvars}(\sigma)$. As $\sigma \in \mathit{VSubst}$ and $y \notin \mathrm{share_with}_d(r)$, by Theorem 24, $\mathrm{vars}(y\sigma) \cap \mathrm{vars}(r\sigma) \subseteq \mathrm{gvars}(\sigma)$. Thus we have $\mathrm{vars}(y\sigma) \cap \mathrm{vars}(\bar{s}) \subseteq \mathrm{gvars}(\sigma)$ and hence, $\mathrm{vars}(y\sigma) \setminus W \subseteq \mathrm{gvars}(\sigma)$.

Since $\mathrm{lin}_d(r)$ holds, by Theorem 24, $\mathrm{rt}(r, \sigma) \in \mathit{LTerms}$ so that, by Lemma 69, $\bar{s}$ is $\big(\mathit{Vars} \setminus \mathrm{gvars}(\mu)\big)$-linear. We can thus apply Proposition 68: by (C1), we obtain $W \subseteq \mathrm{lvars}(\mu)$; and, by (C2), for all $w, w' \in W$ where $w \neq w'$, $\mathrm{rel}\big(\{w\}, \mathrm{ssets}(\mu)\big) \cap \mathrm{rel}\big(\{w'\}, \mathrm{ssets}(\mu)\big) = \varnothing$. By Proposition 56, as $\mu \in \mathit{VSubst}$, for all $w, w' \in W$ where $w \neq w'$, we have $\mathrm{vars}(w\mu) \cap \mathrm{vars}(w'\mu) \subseteq \mathrm{dom}(\mu)$, $\mathrm{vars}(w\mu) \cap \mathrm{dom}(\mu) \subseteq \mathrm{gvars}(\mu)$ and $\mathrm{vars}(w'\mu) \cap \mathrm{dom}(\mu) \subseteq \mathrm{gvars}(\mu)$. Hence $\mathrm{vars}(w\mu) \cap \mathrm{vars}(w'\mu) \subseteq \mathrm{gvars}(\mu)$.

Thus all the conditions needed in order to apply Lemma 82 hold. Therefore we can conclude that $y \in \mathrm{lvars}(\tau)$. □

Proposition 84

Assume the notation and hypotheses given in (C11) and suppose $\mathrm{lin}_d(x)$ and $\mathrm{lin}_d(t)$ hold. Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$,

$$\mathrm{lvars}(\sigma) \setminus \big(\mathrm{share_with}_d(x) \cap \mathrm{share_with}_d(t)\big) \subseteq \mathrm{lvars}(\tau).$$

Proof

By applying Proposition 83 twice, the first time taking $r = x$ and the second time taking $r = t$, we can conclude that

$$\begin{aligned}&\mathrm{lvars}(\sigma) \setminus \big(\mathrm{share_with}_d(x) \cap \mathrm{share_with}_d(t)\big)\\ &\qquad \subseteq \big(\mathrm{lvars}(\sigma) \setminus \mathrm{share_with}_d(x)\big) \cup \big(\mathrm{lvars}(\sigma) \setminus \mathrm{share_with}_d(t)\big) \subseteq \mathrm{lvars}(\tau).\end{aligned}$$

□

Proposition 85
Assume the notation and hypotheses given in (C11). Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$, we have

$$\mathrm{lvars}(\sigma) \setminus \big(\mathrm{share_with}_d(x) \cup \mathrm{share_with}_d(t)\big) \subseteq \mathrm{lvars}(\tau). \tag{C39}$$

Proof
Assuming the definitions in (C12), we show that $\tau \stackrel{\mathrm{def}}{=} \mu \circ \nu$ satisfies (C39). To prove this, we assume that $y \in \mathrm{lvars}(\sigma) \setminus \big(\mathrm{share_with}_d(x) \cup \mathrm{share_with}_d(t)\big)$ and show that $y \in \mathrm{lvars}(\tau)$.

We will show that the hypothesis of Lemma 82 holds where

$$W = \mathit{Vars} \setminus \big(\mathrm{vars}(x\sigma) \cup \mathrm{vars}(t\sigma)\big).$$

As $y \notin \mathrm{share_with}_d(x) \cup \mathrm{share_with}_d(t)$, by Definition 23 and Theorem 24, $y \notin \mathrm{vars}(x\sigma) \cup \mathrm{vars}(t\sigma)$ and also $\mathrm{vars}(y\sigma) \cap \big(\mathrm{vars}(x\sigma) \cup \mathrm{vars}(t\sigma)\big) \subseteq \mathrm{gvars}(\sigma)$. Thus we can apply Lemma 82, to obtain $y \in \mathrm{lvars}(\tau)$. □

C.7 Putting Results Together

By exploiting the correctness results regarding each of the three components of the domain *SFL*, we now prove the correctness of the amgu_s operator. We start by proving a restricted result that only applies to variable-idempotent substitutions.

Lemma 86
Assume the notation and hypotheses given in (C11). Suppose $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$. Then $\tau \in \gamma_s\big(\mathrm{amgu}_s(d, x \mapsto t)\big)$.

Proof
Let $d' = \langle sh', f', l' \rangle \stackrel{\mathrm{def}}{=} \mathrm{amgu}_s(d, x \mapsto t)$. It follows from Definition 21 that to prove $\tau \in \gamma_s(d')$, we have to show that

$$\mathrm{ssets}(\tau) \subseteq sh', \tag{C40}$$
$$\mathrm{fvars}(\tau) \supseteq f', \tag{C41}$$
$$\mathrm{lvars}(\tau) \supseteq l'. \tag{C42}$$

We prove, for each component of d' separately, that there exists $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$ such that (C40), (C41) and (C42) hold. By Lemma 67 the same inclusions hold for any $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$ so that we have the thesis.

(C40). By Definition 26, we have $sh' = \mathrm{cyclic}_x^t(sh_- \cup sh'')$ where

$$sh'' \stackrel{\mathrm{def}}{=} \begin{cases} \mathrm{bin}(sh_x, sh_t), & \text{if } \mathrm{free}_d(x) \vee \mathrm{free}_d(t); \\ \mathrm{bin}\big(sh_x \cup \mathrm{bin}(sh_x, sh^\star_{xt}), & \\ \qquad sh_t \cup \mathrm{bin}(sh_t, sh^\star_{xt})\big), & \text{if } \mathrm{lin}_d(x) \wedge \mathrm{lin}_d(t); \\ \mathrm{bin}(sh^\star_x, sh_t), & \text{if } \mathrm{lin}_d(x); \\ \mathrm{bin}(sh_x, sh^\star_t), & \text{if } \mathrm{lin}_d(t); \\ \mathrm{bin}(sh^\star_x, sh^\star_t), & \text{otherwise.} \end{cases}$$

It follows from Definition 21 and the hypothesis $\sigma \in \gamma_s(d)$ that $\mathrm{ssets}(\sigma) \subseteq sh$. Therefore we apply Proposition 72 (when $\mathrm{free}_d(x) \vee \mathrm{free}_d(t)$ holds), Proposition 73 (when $\mathrm{lin}_d(x) \wedge \mathrm{lin}_d(t)$ holds), Proposition 74 (when $\mathrm{lin}_d(x)$ or $\mathrm{lin}_d(t)$ holds) or Theorem 63 (when nothing is known about x or t) to conclude that $\mathrm{ssets}(\tau) \subseteq sh_- \cup sh''$. The thesis now follows from Proposition 76.

(C41). It follows from Definition 21 and the hypothesis $\sigma \in \gamma_s(d)$ that $f \subseteq \mathrm{fvars}(\sigma)$. Therefore we can apply Proposition 78 (when $\mathrm{free}_d(x) \wedge \mathrm{free}_d(t)$ holds), Proposition 80 (when $\mathrm{free}_d(x)$ or $\mathrm{free}_d(t)$ holds) or Proposition 81 (when nothing is known about x or t) to conclude that $\mathrm{fvars}(\tau) \supseteq f'$.

(C42). It follows from Definition 21 and the hypothesis $\sigma \in \gamma_s(d)$ that $l \subseteq \mathrm{lvars}(\sigma)$. Therefore we can apply Proposition 84 (when $\mathrm{lin}_d(x) \wedge \mathrm{lin}_d(t)$ holds), Proposition 83 (when $\mathrm{lin}_d(x)$ or $\mathrm{lin}_d(t)$ holds) or Proposition 85 (when nothing is known about x or t) to conclude that

$$\mathrm{lvars}(\tau) \supseteq l''.$$

By (C40) proved above, $\mathrm{ssets}(\tau) \subseteq sh'$. By Definition 12, $VI \setminus \mathrm{vars}(sh') \subseteq \mathrm{gvars}(\tau)$. Moreover, by (C41) proved above, we have $\mathrm{fvars}(\tau) \supseteq f'$. Thus, by applying Corollary 20 we obtain the thesis:

$$\begin{aligned}\mathrm{lvars}(\tau) &\supseteq \mathrm{gvars}(\tau) \cup \mathrm{fvars}(\tau) \cup \mathrm{lvars}(\tau)\\ &\supseteq \big(VI \setminus \mathrm{vars}(sh')\big) \cup f' \cup l''\\ &= l'.\end{aligned}$$

□

Finally, by exploiting the results of Subsection 3.1, we drop the assumption about variable-idempotent substitutions.

Proof of Theorem 27 on page 17. Let $d' = \mathrm{amgu}_s(d, x \mapsto t)$.

If $d = \bot_s$ then we have $d' = \bot_s$ and the result holds trivially, since $\gamma_s(d) = \varnothing$. Similarly, if $T = \mathcal{FT}$ is the theory of finite trees and $x \in \mathrm{vars}(t)$, then $d' = \bot_s$. Again, the result holds trivially, since the equation $\{x = t\}$ is not satisfiable in $\mathcal{FT}$, so that $\mathrm{mgs}\big(\sigma \cup \{x = t\}\big) = \varnothing$.

Therefore suppose there exists $\sigma \in \gamma_s(d)$ and $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$. By Proposition 54, there exists $\sigma' \in \mathit{VSubst}$ such that $T \vdash \forall(\sigma \leftrightarrow \sigma')$ and $y \in \mathrm{dom}(\sigma') \cap \mathrm{range}(\sigma')$ implies $y \in \mathrm{vars}(y\sigma')$. By Corollary 22 and Definition 21, we have $\sigma \in \gamma_s(d)$ if and only if $\sigma' \in \gamma_s(d)$. Therefore, the result follows by application of Lemma 86.

Appendix D Proofs of the Results of Section 4

In this section we prove that the abstract unification operator defined on the domain *SFL* is uniformly more precise than the corresponding operator defined on the domain ASub.

Lemma 87
Let $d = \langle sh, f, l\rangle \in \mathit{SFL}$, where $d \neq \bot_S$; let also $\kappa = \alpha_{\mathsf{ASub}}(d) = \langle G, R\rangle \in \mathsf{ASub}$. Then, for all $s, t \in \mathit{HTerms}$ such that $\mathrm{vars}(s) \cup \mathrm{vars}(t) \subseteq \mathit{VI}$, we have

$$\chi_\kappa(t) = 0 \implies \mathrm{rel}\big(\mathrm{vars}(t), sh\big) = \varnothing; \tag{D1}$$

$$\chi_\kappa(t) = 1 \implies \mathrm{lin}_d(t); \tag{D2}$$

$$\{u, v\} \subseteq S \in sh \implies u \overset{\kappa}{\Longleftrightarrow} v; \tag{D3}$$

$$u \in \mathrm{vars}(sh) \wedge u \notin l \implies u \overset{\kappa}{\longleftrightarrow} u. \tag{D4}$$

Proof
To prove case (D1), suppose that $\chi_\kappa(t) = 0$. By Definition 35, $\mathrm{vars}(t) \subseteq G$ so that by Definition 34, $\mathrm{vars}(t) \cap \mathrm{vars}(sh) = \varnothing$. Hence, we obtain $\mathrm{rel}\big(\mathrm{vars}(t), sh\big) = \varnothing$.

To prove case (D2), we first prove that, for all variables $y \in \mathrm{vars}(t)$,

$$\mathrm{occ_lin}_\kappa(y, t) \implies \mathrm{occ_lin}_d(y, t).$$

Suppose that $\mathrm{occ_lin}_\kappa(y, t)$. By Definition 35, we have $y \in G$ or $\mathrm{occ_lin}(y, t)$ and $\forall z \in \mathrm{vars}(t) : \{y, z\} \notin R$. Note that, by taking $z = y$, this also implies that $\{y\} \notin R$. By Definition 34, the condition $y \in G$ implies $y \notin \mathrm{vars}(sh)$, i.e., $\mathrm{ground}_d(y)$; when $y \in \mathrm{vars}(sh)$, the condition $\{y\} \notin R$ implies $y \in l$; finally, we also have

$$\big(\forall z \in \mathrm{vars}(t) : \{y, z\} \notin R\big) \implies \big(\forall z \in \mathrm{vars}(t) : y \neq z \implies \mathrm{ind}_d(y, z)\big).$$

Therefore, by Definition 23, we obtain $\mathrm{occ_lin}_d(y, t)$. Suppose now that $\chi_\kappa(t) = 1$ so that, by Definition 35, we have $\mathrm{occ_lin}_\kappa(y, t)$ for all $y \in \mathrm{vars}(t)$. As shown above, this implies $\mathrm{occ_lin}_d(y, t)$ for all $y \in \mathrm{vars}(t)$. By Definition 23, we conclude $\mathrm{lin}_d(t)$.

To prove case (D3), suppose that there exists $S \in sh$ such that $\{u, v\} \subseteq S$. If $u = v$, then $u \overset{\kappa}{\Longleftrightarrow} v$. On the other hand, if $u \neq v$, by Definition 34, we obtain $\{u, v\} \in R$, which implies $u \overset{\kappa}{\Longleftrightarrow} v$.

To prove case (D4), suppose that $u \in \mathrm{vars}(sh)$ is such that $u \notin l$. Then, by Definition 34, we obtain $\{u\} \in R$, which implies $u \overset{\kappa}{\longleftrightarrow} u$. □

Lemma 88
Let $d \in \mathit{SFL}$ and $(x \mapsto t) \in \mathit{Bind}$ be such that $d \neq \bot_S$ and $\{x\} \cup \mathrm{vars}(t) \subseteq \mathit{VI}$. Then

$$\alpha_{\mathsf{ASub}}\big(\mathrm{amgu}_S(d, x \mapsto t)\big) \preceq_{\mathsf{ASub}} \mathrm{amgu}_{\mathsf{ASub}}\big(\alpha_{\mathsf{ASub}}(d), x \mapsto t\big).$$

Proof
Let $d = \langle sh, f, l\rangle$. By Definition 34, since $d \neq \bot_S$, we have $\alpha_{\mathsf{ASub}}(d) \neq \bot_{\mathsf{ASub}}$. Let

$$d_1 = \langle sh_1, f_1, l_1\rangle = \mathrm{amgu}_S(d, x \mapsto t), \qquad \kappa = \langle G, R\rangle = \alpha_{\mathsf{ASub}}(d),$$
$$\kappa_1 = \langle G_1, R_1\rangle = \alpha_{\mathsf{ASub}}(d_1), \qquad \kappa' = \langle G', R'\rangle = \mathrm{soln}(\kappa, x \mapsto t),$$
$$\kappa_2 = \langle G_2, R_2\rangle = \kappa \circ \kappa'.$$

By Definition 38, $\kappa_2 = \mathrm{amgu}_{\mathsf{ASub}}\big(\alpha_{\mathsf{ASub}}(d), x \mapsto t\big)$. Therefore we have to show that $\kappa_1 \preceq_{\mathsf{ASub}} \kappa_2$, i.e., $G_1 \supseteq G_2$ and $R_1 \subseteq R_2$. Let sh_-, sh_x, sh_t, and sh_{xt} be as in Definition 26.

To prove that $G_1 \supseteq G_2$, we consider an arbitrary variable $u \in G_2$ and prove that $u \in G_1$. By Definition 37, $G_2 = G \cup G'$. Suppose $u \in G$. Then, by Definition 34, $u \notin \mathrm{vars}(sh)$ so that, by Definition 26, $u \notin \mathrm{vars}(sh_1)$. Thus, again by Definition 34, we obtain $u \in G_1$ as required. On the other hand, if $u \in G'$ then, by Definition 36, $u \in \{x\} \cup \mathrm{vars}(t)$ and $\chi_\kappa(x \mapsto t) = 0$. Thus, by Definition 35, $\chi_\kappa(x) = 0$ or $\chi_\kappa(t) = 0$. By case (D1) of Lemma 87, we obtain $sh_x = \varnothing$ or $sh_t = \varnothing$, so that, by Definition 26, $sh_1 = sh_-$. Since $u \in \{x\} \cup \mathrm{vars}(t)$, we obtain $u \notin \mathrm{vars}(sh_1)$ and hence, by Definition 34, $u \in G_1$ as required.

To prove that $R_1 \subseteq R_2$, we consider an arbitrary sharing group $\{u, v\} \in R_1$ and prove that $\{u, v\} \in R_2$. First note that, by Definition 33, $\{u, v\} \cap G_1 = \varnothing$; thus, by Definition 34, $\{u, v\} \subseteq \mathrm{vars}(sh_1)$. Hence, by Definition 26, $\{u, v\} \subseteq \mathrm{vars}(sh)$. The cases when $u = v$ and $u \neq v$ are considered separately.

Case a. Assume $u = v$ so that $\{u\} \in R_1$. Then, by Definition 34, $u \in \mathrm{vars}(sh_1)$ and $u \notin l_1$. By Definition 26, as $u \in \mathrm{vars}(sh_1)$, $u \in \mathrm{vars}(sh)$. Suppose first that $u \notin l$. Then, by case (D4) of Lemma 87, $u \overset{\kappa}{\longleftrightarrow} u$. Since $\kappa_2 = \kappa \circ \kappa'$, by Definition 37, $\{u\} \in R_2$. Now suppose that $u \in l \setminus l_1$. We note that in this case

$$\chi_\kappa(x \mapsto t) \neq 0. \tag{D5}$$

To see this, suppose, by contraposition, that $\chi_\kappa(x \mapsto t) = 0$. Then, by Definition 35 and case (D1) of Lemma 87, we obtain $sh_x = \varnothing$ or $sh_t = \varnothing$ so that, by Definition 26, $l \subseteq l_1$. Thus, since we are assuming that $u \in l \setminus l_1$, we have a contradiction. Therefore (D5) holds. Consider the set l'' as specified in Definition 26, so that, as $u \in l \setminus l_1$, we have $u \notin l''$. For each of the four cases in the specification of l'' we will now prove that $\{u\} \in R_2$. In all four subproofs, we will use the notation $S_x = \mathrm{share_with}_d(x)$ and $S_t = \mathrm{share_with}_d(t)$.

1. Suppose $\mathrm{lin}_d(x) \wedge \mathrm{lin}_d(t)$ holds. By Definition 26, as $u \in l \setminus l_1$, $u \in S_x \cap S_t$. Thus, by Definition 23, there exist $S_0 \in sh_x, S_1 \in sh_t$ such that $\{u, x\} \subseteq S_0$ and $\{y, u\} \subseteq S_1$, where $y \in \mathrm{vars}(t)$. By case (D3) of Lemma 87, $u \overset{\kappa}{\Longleftrightarrow} x$ and $y \overset{\kappa}{\Longleftrightarrow} u$. By Definition 36 and (D5), we also have $\{x, y\} \in R'$, so that $x \overset{\kappa'}{\longleftrightarrow} y$. It follows that $u \overset{\kappa}{\Longleftrightarrow} x \overset{\kappa'}{\longleftrightarrow} y \overset{\kappa}{\Longleftrightarrow} u$ and, by Definition 37, $\{u\} \in R_2$.
2. Suppose $\mathrm{lin}_d(x)$ holds but $\mathrm{lin}_d(t)$ does not hold. By case (D2) of Lemma 87, $\chi_\kappa(t) = 2$; as a consequence, by Definition 36 and (D5), we have $\{x\} \in R'$, so that $x \overset{\kappa'}{\longleftrightarrow} x$. By Definition 26, as $u \in l \setminus l_1$, $u \in S_x$. By Definition 23, there exist $S \in sh_x$ such that $\{u, x\} \subseteq S$ so that, by case (D3) of Lemma 87, $u \overset{\kappa}{\Longleftrightarrow} x$. Thus, we obtain $u \overset{\kappa}{\Longleftrightarrow} x \overset{\kappa'}{\longleftrightarrow} x \overset{\kappa}{\Longleftrightarrow} u$ and, by Definition 37, $\{u\} \in R_2$.
3. Suppose $\mathrm{lin}_d(t)$ holds but $\mathrm{lin}_d(x)$ does not hold. By Definition 26, as $u \in l \setminus l_1$, $u \in S_t$. By Definition 23, there exist $S \in sh_t$ such that $\{u, y\} \subseteq S$, where $y \in \mathrm{vars}(t)$. Thus, by case (D3) of Lemma 87, $u \overset{\kappa}{\Longleftrightarrow} y$. Since $\mathrm{lin}_d(x)$ does not hold, by case (D2) of Lemma 87, $\chi_\kappa(x) = 2$. As a consequence, by Definition 36 and (D5), for all $\{w_1, w_2\} \subseteq \mathrm{vars}(t)$, we have $\{w_1, w_2\} \in R'$;

in particular, by taking $w_1 = w_2 = y$, we obtain $\{y\} \in R'$, so that $y \overset{\kappa'}{\longleftrightarrow} y$. Thus, $u \overset{\kappa}{\Longleftrightarrow} y \overset{\kappa'}{\longleftrightarrow} y \overset{\kappa}{\Longleftrightarrow} u$ and, by Definition 37, $\{u\} \in R_2$.

4. Suppose neither $\mathrm{lin}_d(t)$ nor $\mathrm{lin}_d(x)$ hold. By Definition 26, as $u \in l \setminus l_1$, $u \in S_x \cup S_t$. Thus, by Definition 23, there exist $S \in sh_x \cup sh_t$ such that $\{u, y\} \in S$, where $y \in \{x\} \cup \mathrm{vars}(t)$. Thus, by case (D3) of Lemma 87, $u \overset{\kappa}{\Longleftrightarrow} y$. Since neither $\mathrm{lin}_d(t)$ nor $\mathrm{lin}_d(x)$ hold, by case (D2) of Lemma 87, $\chi_\kappa(x) = \chi_\kappa(t) = 2$ and hence, by Definition 36, we obtain $\{z\} \in R'$ for all $z \in \{x\} \cup \mathrm{vars}(t)$; in particular, $\{y\} \in R'$, so that $y \overset{\kappa'}{\longleftrightarrow} y$. It follows that $u \overset{\kappa}{\Longleftrightarrow} y \overset{\kappa'}{\longleftrightarrow} y \overset{\kappa}{\Longleftrightarrow} u$ and, by Definition 37, $\{u\} \in R_2$.

Case b. Now consider the case when $u \neq v$. By Definition 34, there exists $S \in sh_1$ such that $\{u, v\} \subseteq S$. First, suppose that there exists $S' \in sh$ such that $\{u, v\} \subseteq S'$. Then, by Definition 34, $\{u, v\} \in R$. Thus, $u \overset{\kappa}{\longleftrightarrow} v$ and, by Definition 37, we obtain $\{u, v\} \in R_2$ as required. Therefore, from now on, we assume that

$$\forall S' \in sh : \{u, v\} \nsubseteq S'. \tag{D6}$$

We note that in this case

$$\chi_\kappa(x \mapsto t) \neq 0. \tag{D7}$$

To see this, suppose, by contraposition, that $\chi_\kappa(x \mapsto t) = 0$. Then, by Definition 35 and case (D1) of Lemma 87, we obtain $sh_x = \varnothing$ or $sh_t = \varnothing$ so that, by Definition 26, $sh_1 \subseteq sh$. Thus, since we are assuming (D6) holds, we have a contradiction. Therefore (D7) holds. Note also that since by (D6), $S \notin sh$, $S \notin sh_-$. Thus we can assume that $S \in sh''$. Therefore, by Definition 26, $S \in \mathrm{bin}(sh_x^*, sh_t^*)$. Thus there exists $S_u, S_v \in sh_x \cup sh_t$ such that $S_u \cup S_v \subseteq S$, $u \in S_u$ and $v \in S_v$. Then by (D6), $S_u \neq S_v$. The five subcases in the computation of sh'' as specified in Definition 26 are proved separately.

1. Consider the first case for sh'' in Definition 26, so that $S \in \mathrm{bin}(sh_x, sh_t)$. Then $S_u \in sh_x$ and $S_v \in sh_t$ or vice versa. Without loss of generality, suppose $S_u \in sh_x$ and $S_v \in sh_t$; then $x \in S_u$ and there exists $y \in \mathrm{vars}(t) \cap S_v$. Thus, by By case (D3) of Lemma 87, $u \overset{\kappa}{\Longleftrightarrow} x$ and $y \overset{\kappa}{\Longleftrightarrow} v$. Moreover, by Definition 36 and (D7), $x \overset{\kappa'}{\longleftrightarrow} y$ and hence, by Definition 37, $\{u, v\} \in R_2$.
2. Consider the second case for sh'' in Definition 26, so that

$$S \in \mathrm{bin}\big(sh_x \cup \mathrm{bin}(sh_x, sh_{xt}^\star), sh_t \cup \mathrm{bin}(sh_t, sh_{xt}^\star)\big).$$

   It follows from this that, if $\{S_u, S_v\} \subseteq sh_x$ then one of S_u and S_v must be in sh_{xt} and hence one of S_u or S_v is in sh_t. Similarly, if $\{S_u, S_v\} \subseteq sh_t$, one of S_u or S_v is in sh_x. Thus, as $\{S_u, S_v\} \subseteq sh_x \cup sh_t$, we have $S_u \in sh_x$ and $S_v \in sh_t$, or vice versa; this is the same situation as the previous case, so that $\{u, v\} \in R_2$.
3. Consider the third case for sh'' in Definition 26, so that $\mathrm{lin}_d(t)$ does not hold and $S \in \mathrm{bin}(sh_x^\star, sh_t)$. Then $S_u \in sh_x$ and $S_v \in sh_x \cup sh_t$, or vice versa. If $S_u \in sh_x$ and $S_v \in sh_t$, or vice versa, then we are in the same situation as the first case, so that $\{u, v\} \in R_2$.
   Suppose now that $\{S_u, S_v\} \subseteq sh_x$; then $x \in S_u$ and $x \in S_v$. Thus, by case (D3)

of Lemma 87, $u \overset{\kappa}{\Longleftrightarrow} x$ and $x \overset{\kappa}{\Longleftrightarrow} v$. As $\mathrm{lin}_d(t)$ does not hold then, by case (D2) of Lemma 87 and the assumption (D7), $\chi_\kappa(t) = 2$. As a consequence, by Definition 36 and (D7), we obtain $x \overset{\kappa'}{\longleftrightarrow} x$ and thus, by Definition 37, $\{u, v\} \in R_2$.

4. Consider the fourth case for sh'' in Definition 26, so that $\mathrm{lin}_d(x)$ does not hold and $S \in \mathrm{bin}(sh_x, sh_t^\star)$. Then $S_u \in sh_x \cup sh_t$ and $S_v \in sh_t$, or vice versa. If $S_u \in sh_x$ and $S_v \in sh_t$, or vice versa, then we are in the same situation as the first case, so that $\{u, v\} \in R_2$.
   Suppose now that $\{S_u, S_v\} \subseteq sh_t$; then there exist $\{y_u, y_v\} \subseteq \mathrm{vars}(t)$ such that $y_u \in S_u$ and $y_v \in S_v$. Thus, by case (D3) of Lemma 87, $u \overset{\kappa}{\Longleftrightarrow} y_u$ and $y_v \overset{\kappa}{\Longleftrightarrow} v$. As $\mathrm{lin}_d(x)$ does not hold then, by case (D2) of Lemma 87 and the assumption (D7), $\chi_\kappa(x) = 2$. As a consequence, by Definition 36 and (D7), we obtain $y_u \overset{\kappa'}{\longleftrightarrow} y_v$ and thus, by Definition 37, $\{u, v\} \in R_2$.
5. Consider the last case for sh'' in Definition 26, so that neither $\mathrm{lin}_d(x)$ nor $\mathrm{lin}_d(t)$ do hold and $S \in \mathrm{bin}(sh_x^\star, sh_t^\star)$. Then $S_u \in sh_x \cup sh_t$ and $S_v \in sh_x \cup sh_t$. If $S_u \in sh_x$ and $S_v \in sh_t$ or vice versa, then we are in the same situation as the first case. If $\{S_u, S_v\} \subseteq sh_t$, then, as $\mathrm{lin}_d(t)$ does not hold, we are in the same situation as the third case. If $\{S_u, S_v\} \subseteq sh_x$, then, as $\mathrm{lin}_d(x)$ does not hold, we are in the same situation as the fourth case. Thus, in all cases, $\{u, v\} \in R_2$.

□

Proof of Theorem 40 on page 24. First note that, by Definitions 26 and 34, if $d = \bot_S$ we have

$$\alpha_{\mathsf{ASub}}\big(\mathrm{amgu}_S(\bot_S, x \mapsto t)\big) = \alpha_{\mathsf{ASub}}(\bot_S) = \bot_{\mathsf{ASub}}$$

so that the result holds for all choices of $\kappa \in \mathsf{ASub}_\bot$.

Thus, let $d \neq \bot_S$. By applying Lemma 88 we have

$$\alpha_{\mathsf{ASub}}\big(\mathrm{amgu}_S(d, x \mapsto t)\big) \preceq_{\mathsf{ASub}} \mathrm{amgu}_{\mathsf{ASub}}\big(\alpha_{\mathsf{ASub}}(d), x \mapsto t\big).$$

The thesis follows by the monotonicity of $\mathrm{amgu}_{\mathsf{ASub}}$, since by hypothesis we have $\alpha_{\mathsf{ASub}}(d) \preceq_{\mathsf{ASub}} \kappa$.

Appendix E Proofs of the Results of Section 5

We will use results from (Bagnara et al. 2002; Zaffanella et al. 2002), that show that the domain *PSD* is the weakest abstraction of *SH* achieving the same precision for the computation of groundness and independence.

Theorem 89
(Zaffanella et al. 2002, Theorem 3). Let $sh_1, sh_2 \in SH$ be such that $\rho_{PSD}(sh_1) = \rho_{PSD}(sh_2)$. Then, for each $\sigma \in RSubst$ and $V \subseteq VI$,

$$\rho_{PSD}\big(\mathrm{aunify}(sh_1, \sigma)\big) = \rho_{PSD}\big(\mathrm{aunify}(sh_2, \sigma)\big), \tag{E1}$$

$$\rho_{PSD}\big(\mathrm{aexists}(sh_1, V)\big) = \rho_{PSD}\big(\mathrm{aexists}(sh_2, V)\big). \tag{E2}$$

Theorem 90
(Zaffanella et al. 2002, Theorem 4). Let $sh_1, sh_2 \in SH$ be such that $\rho_{PSD}(sh_1) \neq \rho_{PSD}(sh_2)$. Then there exist $\sigma \in RSubst$ and $\rho \in \{\rho_{Con}, \rho_{PS}\}$ such that

$$\rho\big(\mathrm{aunify}(sh_1, \sigma)\big) \neq \rho\big(\mathrm{aunify}(sh_2, \sigma)\big). \tag{E3}$$

The following result summarises several results in (Bagnara et al. 2002) establishing useful properties of the operator $\rho_{PSD} \in \mathrm{uco}(SH)$.

Lemma 91
Let $sh, sh_1, sh_2 \in SH$. Then

$$\rho_{PSD}(sh^2) = sh^\star; \tag{E4}$$

$$\rho_{PSD}(sh_1 \cup sh_2) = \rho_{PSD}\big(\rho_{PSD}(sh_1) \cup \rho_{PSD}(sh_2)\big); \tag{E5}$$

$$\rho_{PSD}\big(\mathrm{bin}(sh_1, sh_2)\big) = \rho_{PSD}\Big(\mathrm{bin}\big(\rho_{PSD}(sh_1), \rho_{PSD}(sh_2)\big)\Big). \tag{E6}$$

Proof
Case (E4) corresponds to (Bagnara et al. 2002, Theorem 14); case (E5) corresponds to (Bagnara et al. 2002, Lemma 22); and case (E6) corresponds to (Bagnara et al. 2002, Lemma 30). □

The next three lemmas show that the precision of the abstract evaluation of the operators specified in Definition 23 is not affected by ρ_{PSD}.

Lemma 92
Let $V \subseteq VI$ and $sh \in SH$. Then

$$\mathrm{vars}\big(\mathrm{rel}(V, sh)\big) = \mathrm{vars}\Big(\mathrm{rel}\big(V, \rho_{PSD}(sh)\big)\Big).$$

Proof
If $V = \varnothing$, the result is trivial. Thus, assume $V \neq \varnothing$.

The first inclusion ($\subseteq$) follows from the extensivity of ρ_{PSD} and the monotonicity of the operators rel and vars. To prove the other inclusion, let $S \in \mathrm{rel}\big(V, \rho_{PSD}(sh)\big)$. By Definition 7, we have

$$\forall x \in S : S = \bigcup \{\, T \in sh \mid \{x\} \subseteq T \subseteq S \,\}.$$

In particular, for all $x \in S \cap V$, it holds

$$\begin{aligned} S &= \bigcup \{\, T \in sh \mid \{x\} \subseteq T \subseteq S \,\} \\ &= \bigcup \{\, T \in \mathrm{rel}(V, sh) \mid \{x\} \subseteq T \subseteq S \,\} \\ &\subseteq \bigcup \mathrm{rel}(V, sh) \\ &= \mathrm{vars}\big(\mathrm{rel}(V, sh)\big). \end{aligned}$$

Since the choice of S was arbitrary, we obtain the desired inclusion

$$\mathrm{vars}\big(\mathrm{rel}(V, sh)\big) \supseteq \mathrm{vars}\Big(\mathrm{rel}\big(V, \rho_{PSD}(sh)\big)\Big).$$

□

Lemma 93
Let $V, W \subseteq VI$ and $sh \in SH$. Then

$$\big(\mathrm{rel}(V, sh) \cap \mathrm{rel}(W, sh) = \varnothing\big) \iff \Big(\mathrm{rel}\big(V, \rho_{PSD}(sh)\big) \cap \mathrm{rel}\big(W, \rho_{PSD}(sh)\big) = \varnothing\Big).$$

Proof
To prove the implication ($\Rightarrow$), we reason by contraposition and suppose

$$\mathrm{rel}\big(V, \rho_{PSD}(sh)\big) \cap \mathrm{rel}\big(W, \rho_{PSD}(sh)\big) \neq \varnothing.$$

Thus, there exists $S \in \rho_{PSD}(sh)$ such that $S \cap V \neq \varnothing$ and $S \cap W \neq \varnothing$. Consider $x \in S \cap V$ and $y \in S \cap W$, so that we have $\{x, y\} \subseteq S$.

By Definition 7, we have

$$\forall v \in S : S = \bigcup \{\, T \in sh \mid \{v\} \subseteq T \subseteq S \,\}.$$

In particular, by taking $v = x$, there must exist a sharing group $T \in sh$ such that $\{x, y\} \subseteq T$, so that

$$T \in \mathrm{rel}(V, sh) \cap \mathrm{rel}(W, sh) \neq \varnothing.$$

The other implication ($\Leftarrow$) follows by the extensivity of ρ_{PSD} and the monotonicity of rel. □

Lemma 94
For each $i \in \{1, 2\}$, let $d_i = \langle sh_i, f, l\rangle \in SFL$ and suppose that $\rho_{PSD}(sh_1) = \rho_{PSD}(sh_2)$. Then, for all $s, t \in HTerms$ and $y \in VI$,

$$\mathrm{ind}_{d_1}(s,t) \iff \mathrm{ind}_{d_2}(s,t); \tag{E7}$$
$$\mathrm{free}_{d_1}(t) \iff \mathrm{free}_{d_2}(t); \tag{E8}$$
$$\mathrm{ground}_{d_1}(t) \iff \mathrm{ground}_{d_2}(t); \tag{E9}$$
$$\mathrm{occ_lin}_{d_1}(y,t) \iff \mathrm{occ_lin}_{d_2}(y,t); \tag{E10}$$
$$\mathrm{lin}_{d_1}(t) \iff \mathrm{lin}_{d_2}(t); \tag{E11}$$
$$\mathrm{share_with}_{d_1}(t) = \mathrm{share_with}_{d_2}(t). \tag{E12}$$

Proof
(E7). Let $V = \mathrm{vars}(s)$, $W = \mathrm{vars}(t)$. By Definition 23, Lemma 93 and the hypothesis, we obtain

$$\begin{aligned}
\mathrm{ind}_{d_1}(s,t) &\iff \mathrm{rel}(V, sh_1) \cap \mathrm{rel}(W, sh_1) = \varnothing \\
&\iff \mathrm{rel}\big(V, \rho_{PSD}(sh_1)\big) \cap \mathrm{rel}\big(W, \rho_{PSD}(sh_1)\big) = \varnothing \\
&\iff \mathrm{rel}\big(V, \rho_{PSD}(sh_2)\big) \cap \mathrm{rel}\big(W, \rho_{PSD}(sh_2)\big) = \varnothing \\
&\iff \mathrm{rel}(V, sh_2) \cap \mathrm{rel}(W, sh_2) = \varnothing \\
&\iff \mathrm{ind}_{d_2}(s,t).
\end{aligned}$$

(E8). This follows from Definition 23, since the predicate $\mathrm{free}_{d_i}(t)$ does not depend on the sharing component sh_i of d_i.

(E9). By applying Lemma 92 with $V = VI$, we have that, for $i \in \{1, 2\}$,

$\mathrm{vars}(sh_i) = \mathrm{vars}\big(\rho_{PSD}(sh_i)\big)$. Thus, by the hypothesis, $\mathrm{vars}(sh_1) = \mathrm{vars}(sh_2)$. Therefore, by Definition 23, equivalence (E9) holds.

(E10). This follows from Definition 23 and equivalences (E7) and (E9).

(E11). This follows from Definition 23 and equivalence (E10).

(E12). By applying Lemma 92 with $V = \mathrm{vars}(t)$, we have that, for $i \in \{1, 2\}$,

$$\mathrm{vars}\big(\mathrm{rel}(V, sh_i)\big) = \mathrm{vars}\Big(\mathrm{rel}\big(V, \rho_{PSD}(sh_i)\big)\Big).$$

Thus, by the hypothesis, $\mathrm{vars}\big(\mathrm{rel}(V, sh_1)\big) = \mathrm{vars}\big(\mathrm{rel}(V, sh_2)\big)$. Therefore, by Definition 23, equation (E12) holds. □

Since both ρ_{PSD} (by (Hill et al. 2002, Theorem 7)) and $(\cdot)^\star$ are upper closure operators it follows that

$$sh_1 \subseteq \rho_{PSD}(sh_2) \iff \rho_{PSD}(sh_1) \subseteq \rho_{PSD}(sh_2), \tag{E13}$$

$$sh_1 \subseteq sh_2^\star \iff sh_1^\star \subseteq sh_2^\star. \tag{E14}$$

We next establish a number of results needed to show that the precision of the abstract operator amgu_S specified in Definition 26 is not affected by ρ_{PSD} (see Theorem 100).

Lemma 95

(Bagnara et al. 2002, Lemma 18). For each $sh \in SH$ and $V \subseteq VI$,

$$\overline{\mathrm{rel}}\big(V, \rho_{PSD}(sh)\big) = \rho_{PSD}\big(\overline{\mathrm{rel}}(V, sh)\big).$$

Lemma 96

(Bagnara et al. 2002, Lemma 19). For each $sh_1, sh_2 \in SH$ and $V \subseteq VI$,

$$sh_1 \subseteq \rho_{PSD}(sh_2) \implies \mathrm{rel}(V, sh_1)^\star \subseteq \mathrm{rel}(V, sh_2)^\star.$$

Lemma 97

Let $sh_1, sh_2 \in SH$ be such that $sh_1 \subseteq \rho_{PSD}(sh_2)$. Let also $V, W \subseteq VI$ and, for each $i \in \{1, 2\}$,

$$sh_{-,i} = \overline{\mathrm{rel}}(V \cup W, sh_i), \qquad sh_{x,i} = \mathrm{rel}(V, sh_i), \qquad sh_{t,i} = \mathrm{rel}(W, sh_i).$$

Then, we have

$$\mathrm{bin}(sh_{x,1}, sh_{t,1}) \subseteq \rho_{PSD}\big(sh_{-,2} \cup \mathrm{bin}(sh_{x,2}, sh_{t,2})\big); \tag{E15}$$

$$\mathrm{bin}(sh_{x,1}, sh_{t,1}^\star) \subseteq \rho_{PSD}\big(sh_{-,2} \cup \mathrm{bin}(sh_{x,2}, sh_{t,2}^\star)\big). \tag{E16}$$

Proof

(E15). Let $S = S_x \cup S_t \in \mathrm{bin}(sh_{x,1}, sh_{t,1})$ where $S_x \in sh_{x,1}$ and $S_t \in sh_{t,1}$. We show that, for any $y \in S$,

$$S = \bigcup \big\{\, S' \in sh_{-,2} \cup \mathrm{bin}(sh_{x,2}, sh_{t,2}) \;\big|\; \{y\} \subseteq S' \subseteq S \,\big\}. \tag{E17}$$

Suppose first that $y \in S_x$. Let

$$sh_y \stackrel{\mathrm{def}}{=} \{S' \in sh_2 \mid \{y\} \subseteq S' \subseteq S\},$$

$$sh'_t \stackrel{\text{def}}{=} \{\, S' \in sh_2 \mid \exists w \in W \,.\, \{w\} \subseteq S' \subseteq S \,\}.$$

Since $S_x \in sh_{x,1}$ and $S_t \in sh_{t,1}$, by hypothesis, $S_x, S_t \in \rho_{PSD}(sh_2)$. Thus, by Definition 7, $S_x \subseteq \bigcup sh_y \subseteq S$ and $S_t \subseteq \bigcup sh'_t \subseteq S$. As a consequence, $S = \bigcup(sh_y \cup sh'_t)$. By the definition of sh'_t, we have $sh'_t = \{S' \in sh_{t,2} \mid S' \subseteq S\}$, so that $(sh_y \cap sh_{t,2}) \subseteq sh'_t$ and hence,

$$S = \bigcup\big((sh_y \cap sh_{-,2}) \cup (sh_y \cap sh_{x,2}) \cup sh'_t\big).$$

Since $S_x, S_t \neq \varnothing$, it follows that $sh'_t \neq \varnothing$ and $sh_y \cap sh_{x,2} \neq \varnothing$. As a consequence, $\bigcup\big((sh_y \cap sh_{x,2}) \cup sh'_t\big) = \bigcup\big(\mathrm{bin}(sh_y \cap sh_{x,2}, sh'_t)\big)$ so that we have

$$S = \bigcup\big((sh_y \cap sh_{-,2}) \cup \mathrm{bin}(sh_y \cap sh_{x,2}, sh'_t)\big).$$

Moreover, by definition of sh_y, for all $S' \in (sh_y \cap sh_{-,2}) \cup \mathrm{bin}(sh_y \cap sh_{x,2}, sh'_t)$, we have $y \in S'$. Hence (E17) holds for all $y \in S_x$.

By a symmetric argument, (E17) holds for all $y \in S_t$. Thus (E17) holds for all $y \in S$. Therefore, by Definition 7,

$$S \in \rho_{PSD}\big(sh_{-,2} \cup \mathrm{bin}(sh_{x,2}, sh_{t,2})\big).$$

(E16). Let $S = S_x \cup T_t \in \mathrm{bin}(sh_{x,1}, sh^\star_{t,1})$ where $S_x \in sh_{x,1}$ and $T_t \in sh^\star_{t,1}$. Thus, by hypothesis, $S_x \in \rho_{PSD}(sh_2)$ and, by Lemma 96, $T_t \in sh^\star_{t,2}$. We show that, for any $y \in S$,

$$S = \bigcup\{\, S' \in sh_{-,2} \cup \mathrm{bin}(sh_{x,2}, sh^\star_{t,2}) \mid \{y\} \subseteq S' \subseteq S \,\}. \tag{E18}$$

We first consider the case that $y \in S_x$. Let

$$sh_y \stackrel{\text{def}}{=} \{S' \in sh_2 \mid \{y\} \subseteq S' \subseteq S\},$$
$$sh'_t \stackrel{\text{def}}{=} \{S' \in sh^\star_{t,2} \mid S' \subseteq S\}.$$

By Definition 7, as $S_x \in \rho_{PSD}(sh_2)$, $S_x \subseteq \bigcup sh_y \subseteq S$. Also, as $T_t \in sh^\star_{t,2}$, $T_t \subseteq \bigcup sh'_t \subseteq S$. As a consequence, $S = \bigcup(sh_y \cup sh'_t)$. However, we also have $(sh_y \cap sh_{t,2}) \subseteq sh'_t$ so that, as $sh_y \cap sh_2 = sh_y$ and $sh_2 = sh_{-,2} \cup sh_{x,2} \cup sh_{t,2}$, we obtain

$$S = \bigcup\big((sh_y \cap sh_{-,2}) \cup (sh_y \cap sh_{x,2}) \cup sh'_t\big).$$

Since $S_x, T_t \neq \varnothing$, it follows that $sh'_t \neq \varnothing$ and $sh_y \cap sh_{x,2} \neq \varnothing$. As a consequence, $\bigcup\big((sh_y \cap sh_{x,2}) \cup sh'_t\big) = \bigcup\big(\mathrm{bin}(sh_y \cap sh_{x,2}, sh'_t)\big)$. Therefore

$$S = \bigcup\big((sh_y \cap sh_{-,2}) \cup \mathrm{bin}(sh_y \cap sh_{x,2}, sh'_t)\big).$$

Moreover, by definition of sh_y, for all $S' \in (sh_y \cap sh_{-,2}) \cup \mathrm{bin}(sh_y \cap sh_{x,2}, sh'_t)$, we have $y \in S'$. Hence (E18) holds for all $y \in S_x$.

Secondly, consider the case $y \in T_t$. As $T_t \in sh^\star_{t,2}$, there exists $R \subseteq T_t$ such that $y \in R$ and $R \in sh_{t,2}$. Let

$$sh_y \stackrel{\text{def}}{=} \{S' \in sh^\star_{t,2} \mid R \subseteq S' \subseteq S\},$$
$$sh'_x \stackrel{\text{def}}{=} \{\, S' \in sh_2 \mid \exists v \in V \,.\, \{v\} \subseteq S' \subseteq S \,\}.$$

Then $T_t \subseteq \bigcup sh_y \subseteq S$. By Definition 7, as $S_x \in \rho_{PSD}(sh_2)$, $S_x \subseteq \bigcup sh'_x \subseteq S$

and hence $S = \bigcup(sh'_x \cup sh_y)$. Since $S_x, T_t \neq \varnothing$, $\bigcup\big(sh'_x \cup sh_y\big) = \bigcup \mathrm{bin}(sh'_x, sh_y)$. Therefore $S = \bigcup \mathrm{bin}(sh'_x, sh_y)$. Moreover, for all $S' \in \mathrm{bin}(sh'_x, sh_y)$, we have $y \in S'$. Hence, as $sh'_x \subseteq sh_{x,2}$ and $sh_y \subseteq sh^\star_{t,2}$, (E18) holds for all $y \in T_t$.

Therefore, (E18) holds for all $y \in S$. Thus, by Definition 7,

$$S \in \rho_{PSD}\big(sh_{-,2} \cup \mathrm{bin}(sh_{x,2}, sh^\star_{t,2})\big).$$

□

Lemma 98
Let $sh_x, sh_t, sh_{xt}, sh^\diamond \in SH$, where $sh_{xt} = sh_x \cap sh_t$ and

$$sh^\diamond = \mathrm{bin}\big(sh_x \cup \mathrm{bin}(sh_x, sh^\star_{xt}), sh_t \cup \mathrm{bin}(sh_t, sh^\star_{xt})\big).$$

Then $\rho_{PSD}(sh^\diamond) = \rho_{PSD}\big(\mathrm{bin}(sh_x, sh_t)\big)$.

Proof
Observe that

$$\mathrm{bin}\big(\mathrm{bin}(sh_x, sh^\star_{xt}), \mathrm{bin}(sh_t, sh^\star_{xt})\big) = \mathrm{bin}\big(\mathrm{bin}(sh_x, sh_t), sh^\star_{xt}\big).$$

Thus

$$sh^\diamond = \mathrm{bin}(sh_x, sh_t) \cup \mathrm{bin}\big(\mathrm{bin}(sh_x, sh_t), sh^\star_{xt}\big). \tag{E19}$$

Thus, the inclusion $\rho_{PSD}(sh^\diamond) \supseteq \rho_{PSD}\big(\mathrm{bin}(sh_x, sh_t)\big)$ follows by the monotonicity of ρ_{PSD}. We now prove the other inclusion

$$\rho_{PSD}(sh^\diamond) \subseteq \rho_{PSD}\big(\mathrm{bin}(sh_x, sh_t)\big). \tag{E20}$$

Let $S \in sh^\diamond$. Then, by (E19), $S = S_x \cup S_t \cup T_{xt}$, where $S_x \in sh_x$, $S_t \in sh_t$ and $T_{xt} \in sh^\star_{xt} \cup \varnothing$. Thus, for some $k \geq 0$, $T_{xt} = T_1 \cup \cdots \cup T_k$, where $T_i \in sh_{xt}$ for each $i = 1, \ldots, k$. Thus we have

$$S = (S_x \cup S_t) \cup (S_x \cup T_1) \cup \cdots \cup (S_x \cup T_k), \tag{E21}$$
$$S = (S_x \cup S_t) \cup (T_1 \cup S_t) \cup \cdots \cup (T_k \cup S_t), \tag{E22}$$

and, for all $j \in \{1, \ldots, k\}$,

$$S = (S_x \cup T_j) \cup (T_j \cup S_t) \cup (T_1 \cup T_j) \cup \cdots \cup (T_k \cup T_j). \tag{E23}$$

Consider an arbitrary variable $y \in S$. We will show that

$$S = \bigcup\big\{\, S' \in \mathrm{bin}(sh_x, sh_t) \;\big|\; \{y\} \subseteq S' \subseteq S \,\big\}. \tag{E24}$$

If $y \in S_x$, then (E24) follows from (E21). Similarly, if $y \in S_t$, then (E24) follows from (E22). Finally, if $y \notin S_x \cup S_t$, then $k > 0$ and there exists $j \in \{1, \ldots, k\}$ such that $y \in T_j$. Thus, as $T_j \in sh_{xt}$, (E24) follows from (E23).

As (E24) holds for any $y \in S$, by Definition 7, $S \in \rho_{PSD}\big(\mathrm{bin}(sh_x, sh_t)\big)$. Thus, by monotonicity and idempotence of ρ_{PSD}, (E20) holds. □

Lemma 99
Let $sh \in SH$ and $(x \mapsto t) \in Bind$ where $\{x\} \cup \mathrm{vars}(t) \subseteq VI$. Consider $W = \mathrm{vars}(t) \setminus \{x\}$ and let $sh_x, sh_t, sh_W \in SH$ be defined as

$$sh_x = \mathrm{rel}\big(\{x\}, sh\big), \qquad sh_t = \mathrm{rel}\big(\mathrm{vars}(t), sh\big), \qquad sh_W = \mathrm{rel}(W, sh).$$

Then

$$\mathrm{bin}(sh_x, sh_W) = \mathrm{cyclic}_x^t\big(\mathrm{bin}(sh_x, sh_t)\big); \tag{E25}$$
$$\mathrm{bin}(sh_x^\star, sh_W) = \mathrm{cyclic}_x^t\big(\mathrm{bin}(sh_x^\star, sh_t)\big); \tag{E26}$$
$$\mathrm{bin}(sh_x^\star, sh_W^\star) = \mathrm{cyclic}_x^t\big(\mathrm{bin}(sh_x^\star, sh_t^\star)\big). \tag{E27}$$

Proof
We prove equations (E25) and (E26) together, letting $sh'_x \in \{sh_x, sh_x^\star\}$.

To prove the first inclusions ($\subseteq$), let $S \in \mathrm{bin}(sh'_x, sh_W)$. Since $sh_W \subseteq sh_t$, we have $S \in \mathrm{bin}(sh'_x, sh_t)$. Moreover, as $S \in \mathrm{bin}(sh'_x, sh_W)$, $W \cap S \neq \varnothing$. Hence, by Definition 23, $S \in \mathrm{cyclic}_x^t\big(\mathrm{bin}(sh'_x, sh_t)\big)$.

To prove the opposite inclusions ($\supseteq$), let $S \in \mathrm{cyclic}_x^t\big(\mathrm{bin}(sh'_x, sh_t)\big)$. Then, $S \in \mathrm{bin}(sh'_x, sh_t)$, so that $S = S_x \cup S_t$, where $S_x \in sh'_x$ and $S_t \in sh_t$. Thus $x \in S$ so that, by Definition 23, $W \cap S \neq \varnothing$. If $W \cap S_t \neq \varnothing$ then $S_t \in sh_W$ and $S \in \mathrm{bin}(sh'_x, sh_W)$, proving both inclusions. If otherwise $W \cap S_t = \varnothing$, then $W \cap S_x \neq \varnothing$; moreover, as $S_t \in sh_t \setminus sh_W$, we obtain $x \in S_t$ and hence $S_t \in sh_x$. When $sh'_x = sh_x$ the condition $W \cap S_x \neq \varnothing$ is equivalent to $S_x \in sh_W$, so that $S = S_t \cup S_x \in \mathrm{bin}(sh_x, sh_W)$, proving the inclusion for (E25). On the other hand, when $sh'_x = sh_x^\star$, the condition $W \cap S_x \neq \varnothing$ implies that there exists some $S_W \in sh_W$ such that $S_x = S_W \cup S_x$; thus we obtain $S = (S_x \cup S_t) \cup S_W \in \mathrm{bin}(sh_x^\star, sh_W)$, proving the inclusion for (E26).

We now prove equation (E27).

To prove the first inclusion ($\subseteq$), let $S \in \mathrm{bin}(sh_x^\star, sh_W^\star)$. As $sh_W \subseteq sh_t$, we have $S \in \mathrm{bin}(sh_x^\star, sh_t^\star)$. Moreover, as $S \in \mathrm{bin}(sh_x^\star, sh_W^\star)$, $W \cap S \neq \varnothing$. By Definition 23, $S \in \mathrm{cyclic}_x^t\big(\mathrm{bin}(sh_x^\star, sh_t^\star)\big)$.

To prove the opposite inclusion ($\supseteq$), let $S \in \mathrm{cyclic}_x^t\big(\mathrm{bin}(sh_x^\star, sh_t^\star)\big)$. Then $S \in \mathrm{bin}(sh_x^\star, sh_t^\star)$ so that $S = S_x \cup S_t$ where $S_x \in sh_x^\star$ and $S_t \in sh_t^\star$. Thus $x \in S$ and, by Definition 23, $W \cap S \neq \varnothing$. If $W \cap S_t \neq \varnothing$, then $S_t = S_W \cup S_{xt}$, where $S_W \in sh_W^\star$ and $S_{xt} \in sh_x^\star \cup \{\varnothing\}$. Thus $S = (S_x \cup S_{xt}) \cup S_W \in \mathrm{bin}(sh_x^\star, sh_W^\star)$. If $W \cap S_t = \varnothing$, then $S_t \in sh_x^\star$ and $W \cap S_x \neq \varnothing$. In this case, since $S_x = S_W \cup S_x$ for some $S_W \in sh_W$, $S = (S_x \cup S_t) \cup S_W \in \mathrm{bin}(sh_x^\star, sh_W^\star)$. Thus, in both cases, we have the inclusion. □

Theorem 100
Let $d_1, d_2 \in SFL$ be such that $\rho_{PSD}(d_1) = \rho_{PSD}(d_2)$. Then, for all $(x \mapsto t) \in Bind$,

$$\rho_{PSD}\big(\mathrm{amgu}_S(d_1, x \mapsto t)\big) = \rho_{PSD}\big(\mathrm{amgu}_S(d_2, x \mapsto t)\big).$$

Proof
Let $d_1 = \langle sh_1, f, l\rangle$. Then, by Definition 42, $d_2 = \langle sh_2, f, l\rangle$, where $\rho_{PSD}(sh_1) = \rho_{PSD}(sh_2)$. By (E13), it follows that $sh_1 \subseteq \rho_{PSD}(sh_2)$.

For each $i \in \{1,2\}$, let $\langle sh'_i, f'_i, l'_i\rangle = \mathrm{amgu}_{_S}\big(d_i, x \mapsto t\big)$. We will prove the following results:

$$sh'_1 \subseteq \rho_{_{PSD}}(sh'_2), \tag{E28}$$

$$f'_1 = f'_2, \tag{E29}$$

$$l'_1 = l'_2. \tag{E30}$$

From (E28) using (E13), we obtain $\rho_{_{PSD}}(sh'_1) \subseteq \rho_{_{PSD}}(sh'_2)$ and hence, by symmetry, we have $\rho_{_{PSD}}(sh'_1) = \rho_{_{PSD}}(sh'_2)$ The thesis then follows from Definition 42.

(E28). For each $i \in \{1,2\}$, let

$$sh_{x,i} = \mathrm{rel}\big(\{x\}, sh_i\big), \qquad sh_{t,i} = \mathrm{rel}\big(\mathrm{vars}(t), sh_i\big),$$
$$sh_{xt,i} = sh_{x,i} \cap sh_{t,i}, \qquad sh_{-,i} = \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), sh_i\big).$$

Let also $W = \mathrm{vars}(t) \setminus \{x\}$ and $sh_{W,i} = \mathrm{rel}(W, sh_i)$, for each $i \in \{1,2\}$.

By Definitions 23 and 26, for each $i \in \{1,2\}$ we have

$$sh'_i = \mathrm{cyclic}^t_x(sh_{-,i} \cup sh''_i) = sh_{-,i} \cup \mathrm{cyclic}^t_x(sh''_i)$$

where

$$sh''_i \stackrel{\mathrm{def}}{=} \begin{cases} \mathrm{bin}(sh_{x,i}, sh_{t,i}), & \text{if } \mathrm{free}_{d_i}(x) \vee \mathrm{free}_{d_i}(t); \\ \mathrm{bin}\big(sh_{x,i} \cup \mathrm{bin}(sh_{x,i}, sh^\star_{xt}), & \\ \qquad sh_{t,i} \cup \mathrm{bin}(sh_{t,i}, sh^\star_{xt})\big), & \text{if } \mathrm{lin}_{d_i}(x) \wedge \mathrm{lin}_{d_i}(t); \\ \mathrm{bin}(sh^\star_{x,i}, sh_{t,i}), & \text{if } \mathrm{lin}_{d_i}(x); \\ \mathrm{bin}(sh_{x,i}, sh^\star_{t,i}), & \text{if } \mathrm{lin}_{d_i}(t); \\ \mathrm{bin}(sh^\star_{x,i}, sh^\star_{t,i}), & \text{otherwise.} \end{cases}$$

We first show that $sh_{-,1} \subseteq \rho_{_{PSD}}\big(sh_{-,2} \cup \mathrm{cyclic}^t_x(sh''_2)\big)$. By the definition of $sh_{-,1}$ and the monotonicity of $\overline{\mathrm{rel}}$, we have

$$sh_{-,1} \subseteq \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), \rho_{_{PSD}}(sh_2)\big).$$

Thus, by Lemma 95, $sh_{-,1} \subseteq \rho_{_{PSD}}(sh_{-,2})$, from which the required result follows by monotonicity of $\rho_{_{PSD}}$.

We next show that

$$\mathrm{cyclic}^t_x(sh''_1) \subseteq \rho_{_{PSD}}\big(sh_{-,2} \cup \mathrm{cyclic}^t_x(sh''_2)\big). \tag{E31}$$

By applying cases (E8) and (E11) of Lemma 94 to the conditions in the definition of sh''_1 and sh''_2, it can be seen that they are each computed using the same alternative branch. We have five cases.

1. In the first case, for each $i = 1, 2$, we have $sh''_i = \mathrm{bin}(sh_{x,i}, sh_{t,i})$. By case (E25) of Lemma 99, $\mathrm{cyclic}^t_x(sh''_i) = \mathrm{bin}(sh_{x,i}, sh_{W,i})$, for each $i = 1, 2$. Thus, using case (E15) of Lemma 97, where we take $V = \{x\}$, we obtain (E31).
2. In the second case we have, for each $i = 1, 2$,

   $$sh''_i = \mathrm{bin}\big(sh_{x,i} \cup \mathrm{bin}(sh_{x,i}, sh^\star_{xt,i}), sh_{t,i} \cup \mathrm{bin}(sh_{t,i}, sh^\star_{xt,i})\big).$$

   There are two cases. First assume $x \notin \mathrm{vars}(t)$, so that $\mathrm{cyclic}^t_x(sh''_i) = sh''_i$.

Then, by Lemma 98, we have $sh''_i \subseteq \rho_{PSD}\big(\mathrm{bin}(sh_{x,i}, sh_{t,i})\big)$ for each $i = 1, 2$. Therefore, using case (E15) of Lemma 97 and the monotonicity of ρ_{PSD}, we obtain (E31).

Secondly, suppose that $x \in \mathrm{vars}(t)$. In this case, for each $i = 1, 2$, we have $sh_{xt,i} = sh_{x,i}$, so that $sh''_i = \mathrm{bin}(sh^{\star}_{x,i}, sh_{t,i})$. This case is therefore equivalent to the third case, proved below.

3. In the third case, for each $i = 1, 2$, we have $sh''_i = \mathrm{bin}(sh^{\star}_{x,i}, sh_{t,i})$. By case (E26) of Lemma 99, $\mathrm{cyclic}^t_x(sh''_i) = \mathrm{bin}(sh^{\star}_{x,i}, sh_{W,i})$. Thus, by using case (E16) of Lemma 97 (exchanging the roles of $V = \{x\}$ and W), we obtain (E31).
4. In the fourth case, for each $i = 1, 2$, we have $sh''_i = \mathrm{bin}(sh_{x,i}, sh^{\star}_{t,i})$. Moreover, as $\mathrm{lin}_d(t)$ holds and $\mathrm{lin}_d(x)$ does not hold, we can assume that $x \notin \mathrm{vars}(t)$, so that $\mathrm{cyclic}^t_x(sh''_i) = sh''_i$. Thus, using case (E16) of Lemma 97 where $V = \{x\}$, we obtain (E31).
5. In the fifth case we have, for $i = 1, 2$, $sh''_i = \mathrm{bin}(sh^{\star}_{x,i}, sh^{\star}_{t,i})$. By case (E27) of Lemma 99, $\mathrm{cyclic}^t_x(sh''_i) = \mathrm{bin}(sh^{\star}_{x,i}, sh^{\star}_{W,i})$. The thesis follows from case (E1) of Theorem 89 (using Definition 62) by considering any $\sigma \in \mathit{RSubst}$ for which $\mathrm{dom}(\sigma) = \{x\}$ and $\mathrm{vars}(x\sigma) = W$.

(E29). Consider f'_i as specified in Definition 26. By applying case (E8) of Lemma 94, it can be seen that f'_1 and f'_2 are computed by selecting the same alternative branch. The thesis $f'_1 = f'_2$ thus follows from case (E12) of Lemma 94.

(E30). Consider l'_i as specified in Definition 26: for each $i \in \{1, 2\}$ we have

$$l'_i = \big(\mathit{VI} \setminus \mathrm{vars}(sh'_i)\big) \cup f'_i \cup l''_i$$

where

$$l''_i \stackrel{\mathrm{def}}{=} \begin{cases} l_i \setminus \big(\mathrm{share_with}_{d_i}(x) \cap \mathrm{share_with}_{d_i}(t)\big), & \text{if } \mathrm{lin}_{d_i}(x) \wedge \mathrm{lin}_{d_i}(t); \\ l_i \setminus \mathrm{share_with}_{d_i}(x), & \text{if } \mathrm{lin}_{d_i}(x); \\ l_i \setminus \mathrm{share_with}_{d_i}(t), & \text{if } \mathrm{lin}_{d_i}(t); \\ l_i \setminus \big(\mathrm{share_with}_{d_i}(x) \cup \mathrm{share_with}_{d_i}(t)\big) & \text{otherwise.} \end{cases}$$

Let $r \in \mathit{HTerms}$ be such that $\mathrm{vars}(r) = \mathit{VI}$. Then, for each $i \in \{1, 2\}$, we have $\mathrm{vars}(sh'_i) = \mathrm{share_with}_{d'_i}(r)$; also, by equation (E28), we know that $\rho_{PSD}(sh'_1) = \rho_{PSD}(sh'_2)$; thus, by case (E12) of Lemma 94, we obtain $\mathrm{vars}(sh'_1) = \mathrm{vars}(sh'_2)$. By equation (E29), we also know that $f'_1 = f'_2$. By case (E11) of Lemma 94, it can be seen that the conditions determining which branch will be selected in the definitions of l''_1 and l''_2 are equivalent. Hence, by applying case (E12) of Lemma 94, $l''_1 = l''_2$. Therefore equation (E30) holds. □

Theorem 101

Let $d_1, d_2 \in \mathit{SFL}$ be such that $\rho_{PSD}(d_1) = \rho_{PSD}(d_2)$. Then, for each sequence of bindings $bs \in \mathit{Bind}^*$,

$$\rho_{PSD}\big(\mathrm{aunify}_S(d_1, bs)\big) = \rho_{PSD}\big(\mathrm{aunify}_S(d_2, bs)\big).$$

Proof
The proof is by induction on the length of bs. The base case, when $|bs| = 0$ and thus $bs = \epsilon$, is obvious from the definition of aunify_S. For the inductive case, when $|bs| = m > 0$, let $bs = (x \mapsto t)\,.\,bs'$. By the hypothesis and Theorem 100, we have

$$\rho_{PSD}\big(\mathrm{amgu}_S(d_1, x \mapsto t)\big) = \rho_{PSD}\big(\mathrm{amgu}_S(d_2, x \mapsto t)\big). \tag{E32}$$

Moreover, for each $i \in \{1, 2\}$, by definition of aunify_S we have

$$\mathrm{aunify}_S(d_i, bs) = \mathrm{aunify}_S\big(\mathrm{amgu}_S(d_i, x \mapsto t), bs'\big).$$

Thus, by (E32), we can apply the inductive hypothesis and conclude the proof, since $|bs'| = m - 1 < m$. □

We now prove that the precision of the abstract existential operator $\mathrm{aexists}_S$ specified in Definition 31 is not affected by ρ_{PSD}.

Theorem 102
Let $d_1, d_2 \in SH$ be such that $\rho_{PSD}(d_1) = \rho_{PSD}(d_2)$. Then, for each $V \subseteq VI$,

$$\rho_{PSD}\big(\mathrm{aexists}_S(d_1, V)\big) = \rho_{PSD}\big(\mathrm{aexists}_S(d_2, V)\big).$$

Proof
Let $d_i = \langle sh_i, f_i, l_i\rangle$, for each $i = 1, 2$. By applying Definitions 31 and 42, for each $i = 1, 2$, we have

$$\begin{aligned}\rho_{PSD}\big(\mathrm{aexists}_S(d_i, V)\big) &= \rho_{PSD}\Big(\big\langle \mathrm{aexists}(sh_i), f_i \cup V, l_i \cup V\big\rangle\Big)\\ &= \Big\langle \rho_{PSD}\big(\mathrm{aexists}(sh_i)\big), f_i \cup V, l_i \cup V\Big\rangle.\end{aligned}$$

By the hypothesis and Definition 42, we have $\rho_{PSD}(sh_1) = \rho_{PSD}(sh_2)$, $f_1 = f_2$ and $l_1 = l_2$. We can therefore apply case (E2) of Theorem 89 to complete the proof. □

Proof of Theorem 43 on page 25. The congruence properties for aunify_S and $\mathrm{aexists}_S$ follow from Theorems 101 and 102, respectively. The congruence property for alub_S holds, as usual, because ρ_{PSD} is an upper closure operator.

Proof of Theorem 44 on page 25. Suppose $\rho_{PSD}(d_1) \neq \rho_{PSD}(d_2)$. By Definition 42, we have three cases:

1. Suppose $\rho_{PSD}(sh_1) \neq \rho_{PSD}(sh_2)$. Let $bs \in Bind^*$ be any sequence of the bindings in the substitution σ constructed in the proof of Theorem 90 (i.e., the proof of Theorem 3 in (Zaffanella et al. 2002)). Then the behavior of aunify_S on the sharing component of SFL using bs is exactly the same as the behavior of the aunify operator on the plain set-sharing domain SH using σ. This is because, as σ binds all of its domain variables to terms that are ground and finite, the binary union and star-union operators are never used and the result of the abstract computation is independent of the order in which the bindings in σ are considered. Thus, letting $\rho \in \{\rho_{Con}, \rho_{PS}\}$, the result follows from (E3) of Theorem 90.

2. Suppose $f_1 \neq f_2$. Then, by taking $\rho = \rho_{F}$ and $bs = \epsilon$, we have the result since, for $i = 1, 2$,

$$\rho_{F}\big(\text{aunify}_{S}(d_i, \epsilon)\big) = \rho_{F}(d_i) = \langle SG, f_i, \varnothing\rangle.$$

3. Finally, suppose $l_1 \neq l_2$. Then, by taking $\rho = \rho_{L}$ and $bs = \epsilon$, we have the result since, for $i = 1, 2$,

$$\rho_{L}\big(\text{aunify}_{S}(d_i, \epsilon)\big) = \rho_{F}(d_i) = \langle SG, \varnothing, l_i\rangle.$$

Proof of Theorem 45 on page 26. Suppose first that $x \notin \text{vars}(t)$. Then, by Definition 23,

$$\text{cyclic}_x^t(sh_{-} \cup sh^{\diamond}) = sh_{-} \cup sh^{\diamond}.$$

Thus, by Lemma 98 and case (E5) of Lemma 91, we have

$$\begin{aligned}
\rho_{PSD}(sh_{-} \cup sh^{\diamond}) &= \rho_{PSD}\big(\rho_{PSD}(sh_{-}) \cup \rho_{PSD}(sh^{\diamond})\big) \\
&= \rho_{PSD}\Big(\rho_{PSD}(sh_{-}) \cup \rho_{PSD}\big((\text{bin}(sh_x, sh_t)\big)\Big) \\
&= \rho_{PSD}\big(sh_{-} \cup \text{bin}(sh_x, sh_t)\big).
\end{aligned}$$

If on the other hand, $x \in \text{vars}(t)$, then $sh_x = sh_{xt}$, so that $sh^{\diamond} = \text{bin}(sh_x^{\star}, sh_t)$. Thus, by Definition 23 and case (E26) of Lemma 99,

$$\text{cyclic}_x^t(sh_{-} \cup sh^{\diamond}) = sh_{-} \cup \text{bin}(sh_x^{\star}, sh_W).$$

It follows from Lemma 91 that

$$\rho_{PSD}\big(sh_{-} \cup \text{bin}(sh_x^{\star}, sh_W)\big) = \rho_{PSD}\big(sh_{-} \cup \text{bin}(sh_x^{2}, sh_W)\big)$$

so that we have the thesis.