A Correct, Precise and Efficient Integration of Set-Sharing, Freeness and Linearity for the Analysis of Finite and Rational Tree Languages∗

PATRICIA M. HILL
School of Computing, University of Leeds, Leeds, U.K.
(*e-mail:* hill@comp.leeds.ac.uk)

ENEA ZAFFANELLA, ROBERTO BAGNARA
Department of Mathematics, University of Parma, Italy
(*e-mail:* {zaffanella,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, Hill and Zaffanella 2002, Zaffanella, Hill and Bagnara 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 have 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, Dams and Yardeni 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,

∗ The present work has been partly funded by MURST project "Certificazione automatica di programmi mediante interpretazione astratta" and by the University of Parma's FIL scientific research project (ex 60%) "Matematica pura e matematica applicata". The work of P. M. Hill and E. Zaffanella has been partly funded by EPSRC under grant M05645.

but for the set-sharing domain. He 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) 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, Dams, Filé and Bruynooghe 1993a, Codish, Dams, Filé and Bruynooghe 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, Codish and Mulkers 1994a, Bruynooghe, Codish and Mulkers 1994b). From a more pragmatic point of view, Codish et al. (Codish, Mulkers, Bruynooghe, Garcìa de la Banda and Hermenegildo 1993b, Codish, Mulkers, Bruynooghe, Garcìa de la Banda and Hermenegildo 1995) integrate the information captured by the domains of (Søndergaard 1986) and (Muthukumar and Hermenegildo 1991) by performing the analysis with both the 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 no explicit structural information is considered, 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, Zaffanella and Hill 2000b). 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).

By extending the results of (Hill, Bagnara and Zaffanella 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 the domain *SFL* can be simplified by removing some redundant information. In Section 5, we provide an experimental evaluation using the China analyzer (Bagnara 1997). In Section 6 we discuss some related work. Section 7 concludes with some final remarks. The proofs of the results stated in Sections 2, 3.1, 3.2 and 4 are given in Appendices A, B, C and D, 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, one having rank 0 (so that there exist finite terms having no variables) and one having rank greater than 0 (so that there exist infinite terms).

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,

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

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

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

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.

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 $\left\{ x = t \;\middle|\; (x \mapsto t) \in \sigma \right\}$. We say that a set of equations e is in *rational solved form* if $\left\{ s \mapsto t \;\middle|\; (s = t) \in e \right\} \in \mathit{RSubst}$. 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, Lassez and Maher 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 set of equations in rational solved form σ is *satisfiable* in an equality theory T if

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

Observe that, when the occurs-check axioms do not hold for σ (for instance, when $\sigma = \{x = f(x)\}$), then σ is not satisfiable in the equality theory $\mathcal{FT}$.

Given a satisfiable set of equations $e \in \wp_{\mathrm{f}}(\mathit{Eqs})$ in an equality theory T, then a substitution $\sigma \in \mathit{RSubst}$ is called a *solution for e in T* if σ is satisfiable in T and $T \vdash \forall(\sigma \rightarrow e)$. 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 $\mathrm{mgs}(e)$.

We have the following useful result regarding 'rt' and satisfiable substitutions that are equivalent with respect to any given syntactic equality theory.

Proposition 1
Let $\sigma, \tau \in \mathit{RSubst}$ be satisfiable in the syntactic equality theory T and suppose that $T \vdash \forall(\sigma \leftrightarrow \tau)$. Then

$$\mathrm{rt}(y, \sigma) \in \mathit{Vars} \quad \Longleftrightarrow \quad \mathrm{rt}(y, \tau) \in \mathit{Vars}, \tag{9}$$

$$\mathrm{rt}(y,\sigma) \in \mathit{GTerms} \iff \mathrm{rt}(y,\tau) \in \mathit{GTerms}, \tag{10}$$

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

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. Given a Galois connection, the function $\rho \stackrel{\text{def}}{=} \gamma \circ \alpha$ 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, 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 2
(The *set-sharing* lattice.) Let $\mathit{SG} \stackrel{\text{def}}{=} \wp(\mathit{VI}) \setminus \{\varnothing\}$ be the set of *sharing groups*. The set-sharing lattice is defined as $\mathit{SH} \stackrel{\text{def}}{=} \wp(\mathit{SG})$, ordered by subset inclusion.

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

Definition 3
(Auxiliary operators on SH.) For each $\mathit{sh}, \mathit{sh}_1, \mathit{sh}_2 \in \mathit{SH}$ and each $V \subseteq \mathit{VI}$, we define the following functions:
the *star-union* function $(\cdot)^\star\colon \mathit{SH} \to \mathit{SH}$, is defined as

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

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

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

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

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

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

the *binary union* function $\mathrm{bin}\colon SH \times SH \to SH$ is defined as

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

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

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

the *abstract existential quantification* function $\mathrm{aexists}\colon SH \times \wp(VI) \to SH$ is defined as

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

In (Bagnara, Hill and Zaffanella 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, Hill and Bagnara 1999b, Zaffanella et al. 2002).

Definition 4
(The observables of SH.) The *groundness* and *pair-sharing* observables (on SH) $\rho_{Con}, \rho_{PS} \in \mathrm{uco}(SH)$ are defined, for each $sh \in SH$, by

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

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

Definition 5
(The *pair-sharing dependency* lattice PSD.) The operator $\rho_{PSD} \in \mathrm{uco}(SH)$ is defined, for each $sh \in SH$, by

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

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

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.

Definition 6
(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 $(\varnothing, VI, VI)$ 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.[2] 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 rational tree (which is usually denoted by f^ω).

Ideally, a strong normal form for the set of rational trees described by a substitution $\sigma \in \mathit{RSubst}$ 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, Bagnara and Zaffanella 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

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

the fact that the number of iterations needed to reach the fixpoint is bounded by the number of bindings in the substitution.

Definition 7
(Occurrence operator.) For each $\sigma \in \mathit{RSubst}$ and $v \in \mathit{Vars}$, the *occurrence operator* $\mathrm{occ}\colon \mathit{RSubst} \times \mathit{Vars} \to \wp_{\mathrm{f}}(\mathit{Vars})$ is defined as

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

For each $\sigma \in \mathit{RSubst}$, the operator $\mathrm{ssets}\colon \mathit{RSubst} \to \mathit{SH}$ is defined as

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

The operator 'ssets' is introduced for notational convenience only; its additive extension corresponds to the abstraction function mapping concrete elements into elements of the set-sharing domain SH.

Example 8
Let

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

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 $\mathrm{occ}(\sigma, x_4) = \mathrm{occ}(\tau, x_4) = \{x_1, x_2, x_3, x_4\}$.

In a similar way, it is possible to define suitable operators for freeness, groundness 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 previously defined abstraction for set-sharing; 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 9
(Groundness operator.) The *groundness operator* $\mathrm{gvars}\colon \mathit{RSubst} \to \wp_{\mathrm{f}}(\mathit{Vars})$ is defined, for each $\sigma \in \mathit{RSubst}$, by

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

Example 10
Consider $\sigma \in \mathit{RSubst}$, where

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

Then $\mathrm{gvars}(\sigma) = \{x_1, x_2, x_3, x_4\}$. Observe that $x_1 \in \mathrm{gvars}(\sigma)$ although $x_1\sigma \in \mathit{Vars}$. 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 11
(Freeness operator.) The *freeness operator* $\mathrm{fvars}\colon \mathit{RSubst} \to \wp(\mathit{Vars})$ is defined, for each $\sigma \in \mathit{RSubst}$, by

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

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

Example 12
Consider $\sigma \in \mathit{RSubst}$, 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, $\mathrm{fvars}(\sigma) = \{x_3, x_4, x_5\}$. Thus, $x_1 \notin \mathrm{fvars}(\sigma)$ although $x_1\sigma \in \mathit{Vars}$. 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 13
(Linearity operator.) The *linearity operator* $\mathrm{lvars}\colon \mathit{RSubst} \to \wp(\mathit{Vars})$ is defined, for each $\sigma \in \mathit{RSubst}$, by

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

In the next example we consider the extraction of linearity from two substitutions. The substitution σ shows that, in contrast with respect to 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 14
Let $\mathit{VI} = \{x_1, x_2, x_3, x_4\}$ and consider $\sigma \in \mathit{RSubst}$, where

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

Then, $\mathrm{lvars}(\sigma) = \{x_4\}$. Observe that $x_1 \notin \mathrm{lvars}(\sigma)$ since $x_4 \notin \mathrm{dom}(\sigma)$, $x_4 \in x_1\sigma^3 = f(x_1, x_4)$ 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 $i = 3, 4, 5$.

Let now $\tau \in \mathit{RSubst}$, where

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

Then $\mathrm{lvars}(\tau) = \mathit{VI}$. 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 properties of sharing, groundness, freeness and linearity are invariant with respect to substitutions that are equivalent in the given syntactic equality theory.

Proposition 15
Let $\sigma, \tau \in \mathit{RSubst}$ be satisfiable in the syntactic equality theory T and suppose that $T \vdash \forall(\sigma \leftrightarrow \tau)$. Then

$$\mathrm{ssets}(\sigma) = \mathrm{ssets}(\tau), \tag{12}$$

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

Moreover, the occurrence, groundness, freeness and linearity operators precisely capture the intended properties over the domain of rational trees.

Proposition 16
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 17
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 18
(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

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

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 $\Sigma \in \wp(\mathit{RSubst})$,

$$\alpha_S(\Sigma) \stackrel{\text{def}}{=} \mathrm{alub}_S\big\{\, \alpha_S(\sigma) \;\big|\; \sigma \in \Sigma \,\big\}.$$

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

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

With Definition 18 and Proposition 15, one of our objectives is fulfilled: substitutions in *RSubst* that are equivalent have the same abstraction.

Corollary 19

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, suppose that $\langle sh, f, l\rangle \in \mathit{SFL}$. If (as is the case for $\bot_S$) $f \not\subseteq \mathrm{vars}(sh)$, then it follows from Corollary 17 that $\gamma_S\big(\langle sh, f, l\rangle\big) = \gamma_S(\bot_S) = \varnothing$. Thus all such elements represent the semantics of those program fragments that have no successful computations. Also, if $V = \big(\mathit{VI} \setminus \mathrm{vars}(sh)\big) \cup f$, then, for any l', such that $V \cup l = V \cup l'$, we have, by Corollary 17, $\gamma_S\big(\langle sh, f, l'\rangle\big) = \gamma_S\big(\langle sh, f, l\rangle\big)$.

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[3]

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

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. 2000b, 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 20

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

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

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

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

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

t is *linear in* d if and only if $\mathrm{lin}_d\colon \mathit{HTerms} \to \mathit{Bool}$ 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 21
Let $d \in \mathit{SFL}$, $\sigma \in \gamma_S(d)$, $y \in \mathit{VI}$ and $s,t \in \mathit{HTerms}$ be such that $\mathrm{vars}(s) \cup \mathrm{vars}(t) \subseteq \mathit{VI}$. Then

$$\mathrm{ind}_d(s,t) \implies \mathrm{vars}\big(\mathrm{rt}(s,\sigma)\big) \cap \mathrm{vars}\big(\mathrm{rt}(t,\sigma)\big) = \varnothing; \tag{20}$$
$$\mathrm{ind}_d(y,t) \iff y \notin \mathrm{share\_with}_d(t); \tag{21}$$
$$\mathrm{free}_d(t) \implies \mathrm{rt}(t,\sigma) \in \mathit{Vars}; \tag{22}$$
$$\mathrm{ground}_d(t) \implies \mathrm{rt}(t,\sigma) \in \mathit{GTerms}; \tag{23}$$
$$\mathrm{lin}_d(t) \implies \mathrm{rt}(t,\sigma) \in \mathit{LTerms}. \tag{24}$$

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 22
(amgu_S.) The function $\mathrm{amgu}_S\colon \mathit{SFL} \times \mathit{Bind} \to \mathit{SFL}$ captures the effects of a binding on an element of *SFL*. Let $d = \langle \mathit{sh}, f, l\rangle \in \mathit{SFL}$ and $(x \mapsto t) \in \mathit{Bind}$, where $\{x\} \cup \mathrm{vars}(t) \subseteq \mathit{VI}$. Let also

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

where

$$\mathit{sh}_x \stackrel{\mathrm{def}}{=} \mathrm{rel}\big(\{x\}, \mathit{sh}\big), \qquad \mathit{sh}_t \stackrel{\mathrm{def}}{=} \mathrm{rel}\big(\mathrm{vars}(t), \mathit{sh}\big),$$

$$sh_{xt} \stackrel{\text{def}}{=} sh_x \cap sh_t, \qquad sh_{\_} \stackrel{\text{def}}{=} \overline{\text{rel}}\big(\{x\} \cup \text{vars}(t), sh\big),$$

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

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

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

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

where

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

Then

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

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

Theorem 23
Let $d \in SFL$ and $(x \mapsto t) \in Bind$, where $\{x\} \cup \text{vars}(t) \subseteq VI$. Then, for all $\sigma \in \gamma_{\mathcal{S}}(d)$ and $\tau \in \text{mgs}\big(\sigma \cup \{x = t\}\big)$ in the syntactic equality theory T, we have $\tau \in \gamma_{\mathcal{S}}\big(\text{amgu}_{\mathcal{S}}(d, x \mapsto t)\big)$.

We now highlight the similarities and differences of the operator $\text{amgu}_{\mathcal{S}}$ with respect to the corresponding ones defined in the "classical" proposals for an integration of set-sharing with freeness and linearity, such as (Bruynooghe et al. 1994a, 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 24
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 18, 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 abstractly evaluating the binding, both $\mathrm{lin}_d(x)$ and $\mathrm{lin}_d(y)$ hold so that we apply the second case of the definition of sh''. By using the notation of Definition 22, we have

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

Since we compute the star-closure of sh_{xt} only, we obtain the set-sharing component

$$sh' = \{xx_1yy_1, xx_1yy_2, xx_1yz, xx_2yy_1, xx_2yy_2, xx_2yz, xyy_1z, xyy_2z, xyz\}.$$

Thus, we precisely capture 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 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. Example 24 has another interesting, unexpected consequence. By repeating the above abstract computation on the domain ASub (e.g., using the abstract semantics operators specified in (King 2000)), we discover that even this simpler domain precisely captures the independence of pairs P_x and P_y. Therefore, the example provides a formal proof that all the classical approaches based on set-sharing are not *uniformly* more precise than the pair-sharing domain ASub. Such a property is enjoyed by our combination *SFL* with the improved abstract unification operator. The next example shows the precision improvements arising from the use of the 'cyclic^t_x' operator.

Example 25
Let $VI = \{x, x_1, x_2, y\}$ and $\sigma \stackrel{\text{def}}{=} \{x \mapsto f(x_1, x_2)\}$. By Definition 18, we have $d \stackrel{\text{def}}{=} \alpha_s(\{\sigma\}) = \langle sh, f, l\rangle$, 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) $\tau \in \mathrm{mgs}(\sigma \cup \{x = t\})$, where

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

Note that if we further instantiate τ by grounding y, then variables x, x_1 and x_2 would become ground too. Formally, $\alpha_s(\{\tau\}) = \langle sh_\tau, f_\tau, l_\tau\rangle$, where $sh_\tau = \{xyx_1x_2\}$. Thus, as observed above, y covers x, x_1 and x_2.

When abstractly evaluating the binding, we compute

$$sh_x = \{xx_1, xx_2\}, \qquad sh_t = \{xx_1, xx_2, y\},$$
$$sh_{xt} = sh_x, \qquad sh_- = \varnothing,$$

so that

$$sh_- \cup sh'' = \{xx_1, xx_1x_2, xx_1x_2y, xx_1y, xx_2, xx_2y\},$$
$$sh' = \mathrm{cyclic}^t_x(sh_- \cup sh'')$$
$$= \{xx_1x_2y, xx_1y, xx_2y\}.$$

Note that, in the element $sh_- \cup sh''$ (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 26
(**aunify_S.**) The operator $\text{aunify}_S \colon \mathit{SFL} \times \mathit{Bind}^* \to \mathit{SFL}$ is defined, for each $d \in \mathit{SFL}$ and each sequence of bindings $bs \in \mathit{Bind}^*$, by

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

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*.

Definition 27
(**aexists_S.**) The function $\text{aexists}_S \colon \mathit{SFL} \times \wp_{\text{f}}(\mathit{VI}) \to \mathit{SFL}$ 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 $V \subseteq \mathit{VI}$,

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

Note that the correctness of the aexists_S operator does not pose any problem.

4 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 amgu_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, pair-sharing, freeness and linearity. In order to prove such a claim, we now formalize the new observable properties.

Definition 28
(The observables of *SFL*.) The (overloaded) *groundness* and *pair-sharing* observables $\rho_{Con}, \rho_{PS} \in \mathrm{uco}(SFL)$ are defined, for each $\langle sh, f, l\rangle \in SFL$, by

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

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 29
(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 30
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, Filé and Winsborough 1998, Giacobazzi, Ranzato and Scozzari 1998) of SFL with respect to the reduced product

$$(\rho_{Con} \sqcap \rho_{PS} \sqcap \rho_F \sqcap \rho_L).$$

Theorem 31
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 22).

Theorem 32
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

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

Then it holds

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

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 $sh^\diamond$.

5 Experimental Evaluation

Example 24 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, Zaffanella and Hill n.d.), where the non-redundant version SFL_2 of the domain SFL is further combined, as described in (Bagnara et al. n.d., Section 4), with the definite groundness

information computed by *Pos* and compare the results using the (classical) abstract unification operator of (Bagnara et al. n.d., Definition 4) with the (new) operator amgu_S given in Definition 22. 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, Bagnara and Hill 1999a);
3. the abstract domain may or may not be upgraded with structural information using the Pattern($\cdot$) operator (see (Bagnara, Hill and Zaffanella 2000a) and (Bagnara et al. n.d., 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

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.*

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.

6 Related work

Sharing information has been shown to be important for finite-tree analysis (Bagnara, Gori, Hill and Zaffanella 2001a, Bagnara, Zaffanella, Gori and Hill 2001b). 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 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 (SIC 1995), and Oz (Smolka and Treinen 1994). The analysis specified in (Bagnara et al. 2001a) 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. 2001a), 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, Lagoon and Bueno 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 variable 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. 2000b, 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 technical report (Howe and King 2001), 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 2001) is not uniformly more precise than the classical proposals. The following example illustrates this point.

Example 33

Let $VI = \{x, y, z_1, z_2, z_3\}$, $(x \mapsto y) \in Bind$ 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 $d_1 = \langle sh_1, \varnothing, \{x, y\}\rangle$, 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 2001, Definition 3.2), results

[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.

in the less precise abstract element $d_2 = \langle sh_2, \varnothing, \{x, y\}\rangle$, 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 2001) 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 2001) exploits the combination of the domain *SFL* with the groundness domain *Pos*.

7 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* 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 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.

References

Bagnara, R. (1997). *Data-Flow Analysis for Constraint Logic-Based Languages*, PhD 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. (2001a). Finite-tree analysis for constraint logic-based languages, *in* P. Cousot (ed.), *Static Analysis: 8th International Symposium, SAS 2001*, Vol. 2126 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Paris, France, pp. 165–184.

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

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

Bagnara, R., Hill, P. M. and Zaffanella, E. (2002). Set-sharing is redundant for pair-sharing, *Theoretical Computer Science*. To appear.

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

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

Bagnara, R., Zaffanella, E., Gori, R. and Hill, P. M. (2001b). Boolean functions for finite-tree dependencies, *in* R. Nieuwenhuis and A. Voronkov (eds), *Proceedings of the 8th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR 2001)*, Vol. 2250 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Berlin, Havana, Cuba, pp. 575–589.

Bruynooghe, M. and Codish, M. (1993). Freeness, sharing, linearity and correctness — All at once, *in* P. Cousot, M. Falaschi, G. Filé and A. Rauzy (eds), *Static Analysis, Proceedings of the Third International Workshop*, Vol. 724 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Padova, Italy, pp. 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* F. S. de Boer and M. Gabbrielli (eds), *Verification and Analysis of Logic Languages, Proceedings of the W2 Post-Conference Workshop, International Conference on Logic Programming*, Santa Margherita Ligure, Italy, pp. 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.

Clark, K. L. (1978). Negation as failure, *in* H. Gallaire and J. Minker (eds), *Logic and Databases*, Plenum Press, Toulouse, France, pp. 293–322.

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

Codish, M., Dams, D., Filé, G. and Bruynooghe, M. (1993a). Freeness analysis for logic programs — and correctness?, *in* D. S. Warren (ed.), *Logic Programming: Proceedings of the Tenth International Conference on Logic Programming*, MIT Press Series in Logic Programming, The MIT Press, Budapest, Hungary, pp. 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., 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. (1993b). Improving abstract interpretations by combining domains, *Proceedings of the ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation*, ACM Press, Copenhagen, Denmark, pp. 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 Hermenegildo, M. (1995). Improving abstract interpretations by combining domains, *ACM Transactions on Programming Languages and Systems* **17**(1): 28–44.

Colmerauer, A. (1982). Prolog and infinite trees, *in* K. L. Clark and S. Å. Tärnlund (eds), *Logic Programming, APIC Studies in Data Processing*, Vol. 16, Academic Press, New York, pp. 231–251.

Colmerauer, A. (1984). Equations and inequations on finite and infinite trees, *Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'84)*, ICOT, Tokyo, Japan, pp. 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, *Proceedings of the Fourth Annual ACM Symposium on Principles of Programming Languages*, pp. 238–252.

Filé, G. (1994). Share × Free: Simple and correct, *Technical Report 15*, Dipartimento di Matematica, Università di Padova.

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* J. Gruska and J. Zlatuska (eds), *Proceedings of 23rd International Symposium on Mathematical Foundations of Computer Science (MFCS'98)*, Vol. 1450 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 366–377.

Hans, W. and Winkler, S. (1992). Aliasing and groundness analysis of logic programs through abstract interpretation and its safety, *Technical Report 92–27*, Technical University of Aachen (RWTH Aachen).

Hill, P. M., Bagnara, R. and Zaffanella, E. (1998). The correctness of set-sharing, *in* G. Levi (ed.), *Static Analysis: Proceedings of the 5th International Symposium*, Vol. 1503 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Pisa, Italy, pp. 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. To appear. Available at `http://arXiv.org/abs/cs.PL/0102030`.

Howe, J. and King, A. (2001). Three optimisations for sharing, *Technical Report 11-01*, Computing Laboratory, University of Kent at Canterbury.

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* E. L. Lusk and R. A. Overbeek (eds), *Logic Programming: Proceedings of the North American Conference*, MIT Press Series in Logic Programming, The MIT Press, Cleveland, Ohio, USA, pp. 154–165.

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

Keisu, T. (1994). *Tree Constraints*, PhD 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* D. Sannella (ed.), *Proceedings of the Fifth European Symposium on Programming*, Vol.

788 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Edinburgh, UK, pp. 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* P. Van Hentenryck (ed.), *Logic Programming: Proceedings of the Eleventh International Conference on Logic Programming*, MIT Press Series in Logic Programming, The MIT Press, Santa Margherita Ligure, Italy, pp. 553–568.

Langen, A. (1990). *Advanced Techniques for Approximating Variable Aliasing in Logic Programs*, PhD 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, *Proceedings, Third Annual Symposium on Logic in Computer Science*, IEEE Computer Society, Edinburgh, Scotland, pp. 348–357.

Muthukumar, K. and Hermenegildo, M. (1991). Combined determination of sharing and freeness of program variables through abstract interpretation, *in* Furukawa (1991), pp. 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.

SIC (1995). *SICStus Prolog User's Manual*, release 3 #0 edn.

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* B. Robinet and R. Wilhelm (eds), *Proceedings of the 1986 European Symposium on Programming*, Vol. 213 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 327–338.

Zaffanella, E. (2001). *Correctness, Precision and Efficiency in the Sharing Analysis of Real Logic Languages*, PhD 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. (1999a). Widening Sharing, *in* G. Nadathur (ed.), *Principles and Practice of Declarative Programming*, Vol. 1702 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Paris, France, pp. 414–431.

Zaffanella, E., Hill, P. M. and Bagnara, R. (1999b). Decomposing non-redundant sharing by complementation, *in* A. Cortesi and G. Filé (eds), *Static Analysis: Proceedings of the 6th International Symposium*, Vol. 1694 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Venice, Italy, pp. 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. To appear. Available at `http://arXiv.org/abs/cs.PL/0101025`.

A Proofs of the Results of Section 2

The proof of Proposition 1 requires the concept of a *path* $p \in \left(\mathbb{N} \setminus \{0\}\right)^*$ 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, proven in (Hill et al. 2002), state some basic properties of equality theories.

Lemma 34
((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 35
((Hill et al. 2002, Lemma 2).) Let T be an equality theory, $\sigma \in \mathit{RSubst}$ and $t \in \mathit{HTerms}$. Then

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

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

Lemma 36
Let $\sigma \in \mathit{RSubst}$ be satisfiable in the syntactic equality theory T. Suppose $s, t \in \mathit{HTerms}$ are such that $T \vdash \forall\big(\sigma \to (s = t)\big)$. 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 35, we have $T \vdash \forall\big(\sigma \to (s\sigma^i = t\sigma^i)\big)$; from this, by the identity axioms, we obtain that

$$T \vdash \forall\Big(\sigma \to \big(s\sigma^i[p] = t\sigma^i[p]\big)\Big). \tag{A 1}$$

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 34, 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 (A 1), 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 34, $\sigma' = \sigma \cup \{x \mapsto r'\} \in \mathit{RSubst}$ 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 (A 1), $T \vdash \forall\big(\sigma \to (x = r)\big)$.

Finally, consider case 3. In this case $T \vdash \forall(r_1 \neq r_2)$. This immediately leads to a contradiction, since, by (A 1), $T \vdash \forall\big(\sigma \to (r_1 = r_2)\big)$. □

Proof of Proposition 1 on page 6 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 35 we have $T \vdash \forall\big(\tau \to (y = y\sigma^i)\big)$. By Lemma 36, 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 35, we have $T \vdash \forall\big(\sigma \to (y\sigma^i = y\tau^i)\big)$; from this, by the identity axioms, we obtain that

$$T \vdash \forall\Big(\sigma \to \big(y\sigma^i[p] = y\tau^i[p]\big)\Big). \tag{A 2}$$

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 34, $\tau' = \tau \cup \{x \mapsto r'\} \in \mathit{RSubst}$ is satisfiable; we also have $T \vdash \forall(\tau' \to \tau)$ and $T \vdash \forall\big(\tau' \to (x = r')\big)$. This is a contradiction, since, by (A 2), $T \vdash \forall\big(\tau \to (x = r)\big)$.

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

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 $T \vdash \forall(\sigma \to \tau)$, by Lemma 35 we have $T \vdash \forall\big(\sigma \to (y = y\tau^i\sigma^j)\big)$. By Lemma 36, we obtain $\mathrm{rt}(y, \sigma) = \mathrm{rt}(y\tau^i\sigma^j, \sigma)$, so that $\mathrm{rt}(y, \sigma) \notin \mathit{LTerms}$.

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 37
(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 any idempotent substitution is also variable-idempotent, so that $\mathit{ISubst} \subset \mathit{VSubst} \subset \mathit{RSubst}$. This definition of variable-idempotence, 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), where we disregard 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 37.

The following result provides an alternative characterization of variable-idempotence. The proof is by induction on the size of a term where, for each $t \in \mathit{HTerms}$, the function size: $\mathit{HTerms} \to \mathbb{N}$ is defined by

$$\mathrm{size}(t) \stackrel{\mathrm{def}}{=} \begin{cases} 1, & \text{if } t \in \mathit{Vars}; \\ 1 + \sum_{i=1}^{n} \mathrm{size}(t_i), & \text{if } t = f(t_1, \ldots, t_n). \end{cases}$$

Lemma 38
Let $\sigma \in \mathit{RSubst}$. Then

$$\sigma \in \mathit{VSubst} \iff \forall(x \mapsto r) \in \sigma : \mathrm{vars}(r\sigma) = \mathrm{vars}(r).$$

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

$$\mathrm{vars}(x\sigma\sigma) = \mathrm{vars}(x\sigma)$$

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

Next, suppose that for all $(x \mapsto r) \in \sigma$, $\mathrm{vars}(r\sigma) = \mathrm{vars}(r)$ and consider $t \in \mathit{HTerms}$. We will show that $\mathrm{vars}(t\sigma\sigma) = \mathrm{vars}(t\sigma)$ by induction on the size of t. If t is a constant or $t \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$, then the result follows from the fact that $t\sigma = t$.

If $t \in \mathrm{dom}(\sigma)$, then there exists $(y \mapsto s) \in \sigma$ such that $t = y$, so that $t\sigma = s$. Thus, we have

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

Finally, if $t = f(t_1, \ldots, t_n)$, then by the inductive hypothesis $\mathrm{vars}(t_i\sigma\sigma) = \mathrm{vars}(t_i\sigma)$ for $i = 1, \ldots, n$. Therefore we have

$$\mathrm{vars}(t\sigma\sigma) = \bigcup_{i=1}^{n} \mathrm{vars}(t_i\sigma\sigma) = \bigcup_{i=1}^{n} \mathrm{vars}(t_i\sigma) = \mathrm{vars}(t\sigma).$$

Thus, by Definition 37, as $\sigma \in \mathit{RSubst}$, $\sigma \in \mathit{VSubst}$. □

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

Lemma 39
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
Let $(x \mapsto t) \in \sigma' \subseteq \sigma$. We will prove that $\mathrm{vars}(t\sigma') = \mathrm{vars}(t)$, so that the thesis will follow from Lemma 38.

To prove the first implication, let $y \in \mathrm{vars}(t\sigma')$, so that $y \in \mathrm{range}(\sigma)$. If it also holds $y \in \mathrm{dom}(\sigma)$, then by the hypothesis $y \in \mathrm{vars}(y\sigma)$, so that $y \in \mathrm{vars}(t\sigma)$. Otherwise, if $y \notin \mathrm{dom}(\sigma)$, then again $y \in \mathrm{vars}(t\sigma)$. Thus, in both cases, since $\sigma \in \mathit{VSubst}$, by Lemma 38 we obtain $y \in \mathrm{vars}(t)$.

To prove the other implication, let $y \in \mathrm{vars}(t)$, so that $y \in \mathrm{range}(\sigma)$. If $y \notin \mathrm{dom}(\sigma')$ then $y \in \mathrm{vars}(t\sigma')$. Otherwise, if $y \in \mathrm{dom}(\sigma')$, then we have $y \in \mathrm{dom}(\sigma) \cap \mathrm{range}(\sigma)$. Thus, by hypothesis, $y \in \mathrm{vars}(y\sigma)$. Since $y\sigma = y\sigma'$, we have $y \in \mathrm{vars}(y\sigma')$, so that $y \in \mathrm{vars}(t\sigma')$. □

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 40
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 $z \notin \mathrm{dom}(\sigma) \cup \mathrm{dom}(\tau)$, 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, $z \in \mathrm{vars}(t\sigma\sigma)$, so that there exists $v \in \mathrm{vars}(t\sigma) \cap \mathrm{dom}(\sigma)$ such that $z \in \mathrm{vars}(v\sigma)$. Thus $v \notin \mathrm{vars}(\tau)$, so that $v \in \mathrm{vars}(t\sigma\tau)$. 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 $z \in \mathrm{dom}(\tau)$, so that $z \notin \mathrm{dom}(\sigma)$. There exists $v \in \mathrm{vars}(t\sigma) \cap \mathrm{dom}(\tau)$

such that $z \in \mathrm{vars}(v\tau)$. Since τ is variable-idempotent, $z \in \mathrm{vars}(v\tau\tau)$ so that there exists $w \in \mathrm{vars}(v\tau) \cap \mathrm{dom}(\tau)$ such that $z \in \mathrm{vars}(w\tau)$. Since $w \notin \mathrm{dom}(\sigma)$ then $w \in \mathrm{vars}(t\sigma\tau\sigma)$. 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, $w \in \mathrm{vars}(t\sigma)$, $v \in \mathrm{vars}(t\sigma\sigma)$ and $z \in \mathrm{vars}(t\sigma\sigma\tau)$. Thus, as $\sigma \in \mathit{VSubst}$, we obtain $z \in \mathrm{vars}(t\sigma\tau)$.

Otherwise, if $v = w$, there exists $x \in \mathrm{vars}(t\sigma)$ such that $z \in \mathrm{vars}(x\tau\tau)$. Thus, $z \in \mathrm{vars}(t\sigma\tau\tau)$ and, since $\tau \in \mathit{VSubst}$, $z \in \mathrm{vars}(t\sigma\tau)$. □

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

Proposition 41
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. □

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

Lemma 42
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 43
For each $\sigma \in \mathit{VSubst}$ and $v \in \mathit{Vars}$, we have

$$\mathrm{occ}(\sigma, v) = \big\{\, y \in \mathit{Vars} \;\big|\; v \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) \,\big\}, \tag{B 1}$$

$$\mathrm{gvars}(\sigma) = \big\{\, y \in \mathit{Vars} \;\big|\; \mathrm{vars}(y\sigma) \subseteq \mathrm{dom}(\sigma) \,\big\}, \tag{B 2}$$

$$\mathrm{fvars}(\sigma) = \big\{\, y \in \mathit{Vars} \;\big|\; y\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma) \,\big\}, \tag{B 3}$$

$$\mathrm{lvars}(\sigma) = \left\{ y \in \mathit{Vars} \;\middle|\; \begin{array}{l} \forall z \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) : \mathrm{occ\_lin}(z, y\sigma), \\ \forall z \in \mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma) : z \in \mathrm{gvars}(\sigma) \end{array} \right\}. \tag{B 4}$$

Proof
We prove each equation separately.

(B 1). This has been proved in (Hill et al. 2002, Lemma 13).

(B 2). By Definition 9, $y \in \mathrm{gvars}(\sigma)$ if and only if, for all $v \in \mathit{Vars}$, we have

$y \notin \mathrm{occ}(\sigma, v)$. By equation (B 1) proven 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 $n = \#\sigma$.

(B 3). By Definition 11, $y \in \mathrm{fvars}(\sigma)$ if and only if $y\sigma^n \in \mathit{Vars}$. First note that, if $y\sigma^n \notin \mathit{Vars}$, then $n \geq 1$ and $y\sigma \notin \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. Conversely, if $y\sigma^n \in \mathit{Vars}$, then $y\sigma \in \mathit{Vars}$. As $\sigma \in \mathit{VSubst}$, $\{y\sigma\} = \{y\sigma^n\}$, so that, by Lemma 42, $y\sigma = y\sigma^n \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$.

(B 4). First, suppose that, for some $z \in \mathrm{vars}(y\sigma)$, we have either $z \notin \mathrm{dom}(\sigma)$ and $\mathrm{occ\_lin}(z, y\sigma)$ does not hold or $z \in \mathrm{dom}(\sigma)$ and $z \notin \mathrm{gvars}(\sigma)$. In both cases, we have $n \geq 1$ and we show that $y \notin \mathrm{lvars}(\sigma)$. In the first case it follows that $z \in \mathrm{vars}(y\sigma^n)$ and $\mathrm{occ\_lin}(z, y\sigma^{2n})$ does not hold. For the second case, by equation (B 2) proved above, there exists $w \in \mathrm{vars}(z\sigma) \setminus \mathrm{dom}(\sigma)$. As $\sigma \in \mathit{VSubst}$, we have $w \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma)$. Since $z \in \mathrm{vars}(y\sigma)$, we obtain that $\mathrm{occ\_lin}(w, y\sigma^2)$ does not hold and, as $n \geq 1$, $\mathrm{occ\_lin}(w, y\sigma^{2n})$ does not hold. Therefore, by Definition 13, $y \notin \mathrm{lvars}(\sigma)$.

Secondly, suppose that $y \notin \mathrm{lvars}(\sigma)$ so that $n \geq 1$. Then, by Definition 13, 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 (B 2), $z \notin \mathrm{gvars}(\sigma)$. □

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 44
Let $\sigma \in \mathit{VSubst}$, $y \in \mathit{VI}$ and $v \in \mathit{Vars}$. Then:

$$y \in \mathrm{occ}(\sigma, v) \iff v \in \mathrm{vars}\big(\mathrm{rt}(y, \sigma)\big), \tag{B 5}$$

$$y \in \mathrm{gvars}(\sigma) \iff \mathrm{rt}(y, \sigma) \in \mathit{GTerms}, \tag{B 6}$$

$$y \in \mathrm{fvars}(\sigma) \iff \mathrm{rt}(y, \sigma) \in \mathit{Vars}, \tag{B 7}$$

$$y \in \mathrm{lvars}(\sigma) \iff \mathrm{rt}(y, \sigma) \in \mathit{LTerms}. \tag{B 8}$$

Proof
We prove each item separately.

(B 5). By Proposition 43, $y \in \mathrm{occ}(\sigma, v)$ 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 $v \in \mathrm{vars}(y\sigma)$.

(B 6). By Definition 9, we have $y \in \mathrm{gvars}(\sigma)$ if and only if $y \notin \mathrm{occ}(\sigma, v)$, for all $v \in \mathit{Vars}$. By item (B 5) 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 $\mathrm{rt}(y, \sigma) \in \mathit{GTerms}$.

(B 7). By Proposition 43, $y \in \mathrm{fvars}(\sigma)$ 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, $\mathrm{rt}(y,\sigma) \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$ and, more generally, $\mathrm{rt}(y,\sigma) \in \mathit{Vars}$. 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, $\mathrm{rt}(y,\sigma) \notin \mathit{Vars}$.

(B 8). In order to prove the first implication ($\Rightarrow$), assume $y \in \mathrm{lvars}(\sigma)$ so that, by Proposition 43, we have

$$\forall z \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) : \mathrm{occ\_lin}(z, y\sigma), \tag{B 9}$$

$$\forall z \in \mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma) : \mathrm{vars}(z\sigma) \subseteq \mathrm{dom}(\sigma). \tag{B 10}$$

We need to show that $\mathrm{rt}(y,\sigma) \in \mathit{LTerms}$ and we proceed by contradiction, negating the conclusion. Thus assume that there exists $v \in \mathrm{vars}\big(\mathrm{rt}(y,\sigma)\big)$ such that $\mathrm{occ\_lin}\big(v, \mathrm{rt}(y,\sigma)\big)$ does not hold. Note that $v \notin \mathrm{dom}(\sigma)$; also, since $\sigma \in \mathit{VSubst}$, $\mathrm{vars}(y\sigma) = \mathrm{vars}(y\sigma^i)$, for all $i > 0$, so that $v \in \mathrm{vars}(y\sigma)$. If $\mathrm{occ\_lin}(v, y\sigma)$ does not hold, then we obtain the negation of (B 9), hence a contradiction. So, assume that $\mathrm{occ\_lin}(v, y\sigma)$ hold. As a consequence, there exists an index $j > 1$ such that $\mathrm{occ\_lin}(v, y\sigma^{j-1})$ holds and $\mathrm{occ\_lin}(v, y\sigma^j)$ does not hold. Thus, there exists $w \in \mathrm{vars}(y\sigma^{j-1}) \cap \mathrm{dom}(\sigma)$ such that $v \in \mathrm{vars}(w\sigma) \setminus \mathrm{dom}(\sigma)$. Since $\sigma \in \mathit{VSubst}$, $w \in \mathrm{vars}(y\sigma^{j-1})$ if and only if $w \in \mathrm{vars}(y\sigma)$. Hence $j = 2$, $w \in \mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma)$ and $\mathrm{vars}(w\sigma) \not\subseteq \mathrm{dom}(\sigma)$, therefore contradicting (B 10).

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 (B 9) and (B 10) hold. We proceed by contradiction, negating the conclusion. There are two cases.

1. Assume that (B 9) 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 (B 10) 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 15, we first prove some preliminary results.

Lemma 45

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 42, $s\sigma^m \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$ so that $\mathrm{rt}(s\sigma^m, \sigma) = s\sigma^m$. Moreover, by Lemma 35, $T \vdash \forall\big(\sigma \rightarrow (s\sigma^m = t\tau^n)\big)$ so that, by Lemma 36, $\mathrm{rt}(t\tau^n, \sigma) = s\sigma^m \in \mathit{Vars}$. Therefore $t\tau^n \in \mathit{Vars}$. □

Lemma 46
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 size of term s.

If $\mathrm{size}(s) = 1$ and $z \in \mathrm{vars}(s)$, then we have $s = z$. Then, for all $i \geq 1$, $s\sigma^i \in \mathit{Vars}$. Thus, by Lemma 45, $t\tau^n \in \mathit{Vars}$ and hence, by Lemma 42, $t\tau^n \in \mathit{Vars} \setminus \mathrm{dom}(\tau)$. By Lemma 35, $T \vdash \forall\big(\tau \rightarrow (z = t\tau^n)\big)$. 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 45, 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 47
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 size of term s.

If $\mathrm{size}(s) = 1$ and $z \in \mathrm{vars}(s)$, then we have $s = z$. In this case there is nothing to prove, since $\mathrm{occ\_lin}(z, s)$ 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 45, $t\tau^n \notin \mathit{Vars}$. 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 46 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 36, 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 15 on page 11. 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 9, 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 11, $y\sigma^m \in \mathit{Vars}$. so that, by Lemma 45, $y\tau^n \in \mathit{Vars}$. Hence, by Definition 11, $y \in \mathrm{fvars}(\tau)$.

Consider (15) and let $y \notin \mathrm{lvars}(\sigma)$. Then, by Definition 13, 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 35, $T \vdash \forall\big(\sigma \to (y\sigma^{2m} = y)\big)$ so that, by Lemma 47, 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 13, $y \notin \mathrm{lvars}(\tau)$.

Lemma 48
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 $v \in \mathrm{vars}\big(\mathrm{rt}(y\sigma^i, \tau)\big)$. Since, by Lemma 36, $\mathrm{rt}(y, \tau) = \mathrm{rt}(y\sigma^i, \tau)$, it follows that $v \in \mathrm{vars}\big(\mathrm{rt}(y, \tau)\big)$. □

Proof of Proposition 16 on page 12. By Proposition 41, there exists $\tau \in \mathit{VSubst}$ such that $\mathrm{dom}(\sigma) = \mathrm{dom}(\tau)$ and $T \vdash \forall(\sigma \leftrightarrow \tau)$. By Proposition 15, 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 44, we obtain

$$\begin{aligned} y \in \mathrm{occ}(\sigma, v) \quad &\iff \quad v \in \mathrm{vars}\big(\mathrm{rt}(y, \tau)\big), \\ y \in \mathrm{gvars}(\sigma) \quad &\iff \quad \mathrm{rt}(y, \tau) \in \mathit{GTerms}, \end{aligned}$$

$$y \in \mathrm{fvars}(\sigma) \iff \mathrm{rt}(y,\tau) \in \mathit{Vars},$$
$$y \in \mathrm{lvars}(\sigma) \iff \mathrm{rt}(y,\tau) \in \mathit{LTerms}.$$

Thus (16) follows from Lemma 48 while (17), (18) and (19) follow from Proposition 1.

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 20. 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 23.

C.1 Abstract unification for SH

Definition 49
(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 $sh_x \stackrel{\text{def}}{=} \mathrm{rel}\big(\{x\}, sh\big)$, $sh_t \stackrel{\text{def}}{=} \mathrm{rel}\big(\mathrm{vars}(t), sh\big)$ and $sh_- \stackrel{\text{def}}{=} \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), sh\big)$. Then

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

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

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

Theorem 50
((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 21 on page 14. Let $d = \langle sh, f, l\rangle$, $V_s = \mathrm{vars}(s)$ and $V_t = \mathrm{vars}(t)$. By Definition 18, 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 20, the hypothesis and Proposition 16, we have

$$\begin{aligned}
\mathrm{ind}_d(s,t) &\iff \mathrm{rel}(V_s, sh) \cap \mathrm{rel}(V_t, sh) = \varnothing \\
&\iff \forall S \in sh, w_1 \in V_s, w_2 \in V_t : \{w_1, w_2\} \nsubseteq S \\
&\implies \forall v \in \mathit{Vars}, w_1 \in V_s, w_2 \in V_t : \{w_1, w_2\} \nsubseteq \mathrm{occ}(\sigma, v) \\
&\iff \forall w_1 \in V_s, w_2 \in V_t : \mathrm{vars}\big(\mathrm{rt}(w_1, \sigma)\big) \cap \mathrm{vars}\big(\mathrm{rt}(w_1, \sigma)\big) = \varnothing \\
&\iff \mathrm{vars}\big(\mathrm{rt}(s, \sigma)\big) \cap \mathrm{vars}\big(\mathrm{rt}(t, \sigma)\big) = \varnothing.
\end{aligned}$$

Consider the equivalence (21). By Definition 20, 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 20, the hypothesis and Proposition 16, 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 20, the hypothesis and Proposition 16, we have

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

Finally consider the implication (24). By Definition 20, the hypothesis, the above results and Proposition 16, we have

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

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

Lemma 51
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 35, 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 52
((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 53
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 $x \notin \mathrm{gvars}(\tau)$ implies $x \notin \mathrm{gvars}(\sigma)$.

By Proposition 41, 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 15, we have $\mathrm{gvars}(\sigma) = \mathrm{gvars}(\sigma')$ and $\mathrm{gvars}(\tau) = \mathrm{gvars}(\tau')$. Therefore, it is sufficient to prove that $x \notin \mathrm{gvars}(\tau')$ implies $x \notin \mathrm{gvars}(\sigma')$.

Assume $x \notin \mathrm{gvars}(\tau')$. By Definition 9, there exists $v \in \mathit{Vars}$ such that $x \in \mathrm{occ}(\tau', v)$. By Proposition 43, $v \in \mathrm{vars}(x\tau') \setminus \mathrm{dom}(\tau')$. Also, by Lemma 35, $T \vdash \forall(\tau' \rightarrow x\tau' = x)$. Therefore, by Lemma 52 (taking $s = x\tau'$ and $t = x$) there

exists $z \in \mathrm{vars}(x\sigma') \setminus \mathrm{dom}(\sigma')$ such that $v \in \mathrm{vars}(z\tau')$. By Definition 7, we have $x \in \mathrm{occ}(\sigma', z)$ so that, by Definition 9, $x \notin \mathrm{gvars}(\sigma')$. □

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

Lemma 54
Let $e \subseteq \mathit{Eqs}$ be satisfiable in the syntactic equality theory T. If $\sigma, \tau \in \mathrm{mgs}(e)$, then $\mathrm{ssets}(\sigma) = \mathrm{ssets}(\tau)$, $\mathrm{gvars}(\sigma) = \mathrm{gvars}(\tau)$, $\mathrm{fvars}(\sigma) = \mathrm{fvars}(\tau)$ and $\mathrm{lvars}(\sigma) = \mathrm{lvars}(\tau)$.

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 15. □

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 $\mathrm{occ\_lin}(v, t)$ holds for all variables $v \in \mathrm{vars}(t) \cap V$. 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 $\mathrm{mgs}(\bar{s} = \bar{t}) \neq \varnothing$, then we write $\mathrm{gvars}(\bar{s} = \bar{t})$ to denote the set $\mathrm{gvars}(\mu)$, where $\mu \in \mathrm{mgs}(\bar{s} = \bar{t})$. Note that, by Lemma 54, this notation is not ambiguous.

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

Proposition 55
Suppose $\bar{s}, \bar{t} \in \mathit{HTerms}^n$ and $\mathrm{mgs}(\bar{s} = \bar{t}) \neq \varnothing$. Let $G \stackrel{\mathrm{def}}{=} \mathrm{gvars}(\bar{s} = \bar{t})$ and suppose also that $\bar{s}$ is $(\mathit{Vars} \setminus G)$-linear. Then there exists $\mu \in \mathrm{mgs}(\bar{s} = \bar{t}) \cap \mathit{VSubst}$ such that

$$\mathit{Vars} \setminus \mathrm{vars}(\bar{s}) \subseteq \mathrm{lvars}(\mu), \tag{C 1}$$

$$\begin{aligned} &\forall z, z' \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}) : \\ &\qquad z \neq z' \implies \mathrm{rel}\big(\{z\}, \mathrm{ssets}(\mu)\big) \cap \mathrm{rel}\big(\{z'\}, \mathrm{ssets}(\mu)\big) = \varnothing. \end{aligned} \tag{C 2}$$

Proof
By Proposition 15 and 41, we just have to show that *there exists* $\mu \in \mathit{VSubst}$ such that (C 1) and (C 2) hold.

By Proposition 43, if $z \in \mathit{Vars}$, then $z \in \mathrm{lvars}(\mu)$ if and only if

$$\begin{aligned} &\forall y \in \mathrm{vars}(z\mu) \setminus \mathrm{dom}(\mu) : \mathrm{occ\_lin}(y, z\mu), \\ &\mathrm{vars}(z\mu) \cap \mathrm{dom}(\mu) \subseteq G. \end{aligned}$$

As $G \subseteq \mathrm{dom}(\mu)$, it follows from Proposition 43, that (C 1) and (C 2) will hold if and only if, for each $z \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s})$:

$$z\mu \text{ is } (\mathit{Vars} \setminus G)\text{-linear}; \tag{C 3}$$

$$\mathrm{vars}(z\mu) \cap \mathrm{dom}(\mu) \subseteq G; \tag{C 4}$$

$$\forall z' \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}) : z \neq z' \implies \mathrm{vars}(z\mu) \cap \mathrm{vars}(z'\mu) \subseteq G. \qquad \text{(C 5)}$$

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 (C 3), (C 4) and (C 5) 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 $n > 0$. Note that, in the case that $\mathrm{vars}(\bar{t}) \subseteq \mathrm{vars}(\bar{s})$, then all three properties hold trivially. This is because for all $\mu \in \mathrm{mgs}(\bar{s} = \bar{t})$ and for all $z \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s})$, we have $z \notin \mathrm{vars}(\mu)$. Similarly, the three properties hold trivially whenever $\mathrm{vars}(\bar{t}) \subseteq G$. This is because, if $z \in \mathrm{dom}(\mu)$, then $\mathrm{vars}(z\mu) \subseteq G$. We therefore assume for the rest of the proof that for some $i = 1, \ldots, n$, $\mathrm{vars}(t_i) \setminus \big(\mathrm{vars}(\bar{s}) \cup G\big) \neq \varnothing$. As the order of equations is irrelevant, without loss of generality we assume that this property holds when $i = 1$, so that $\mathrm{vars}(t_1) \setminus \big(\mathrm{vars}(\bar{s}) \cup G\big) \neq \varnothing$. This can be re-written as

$$\mathrm{vars}(t_1) \cap \big(V \setminus \mathrm{vars}(\bar{s})\big) \neq \varnothing. \qquad \text{(C 6)}$$

Note that this implies that $t_1 \neq s_1$. By Proposition 16, another consequence of the above assumption is that, for all $\mu \in \mathrm{mgs}(\bar{s} = \bar{t})$, we have $\mathrm{rt}(t_1, \mu) \notin \mathit{GTerms}$. Since $T \vdash \forall\big(\mu \rightarrow (s_1 = t_1)\big)$, by Lemma 36, we obtain $\mathrm{rt}(s_1, \mu) \notin \mathit{GTerms}$, so that again by Proposition 16, $\mathrm{vars}(s_1) \setminus G \neq \varnothing$. This in turn can be re-written as

$$\mathrm{vars}(s_1) \cap V \neq \varnothing. \qquad \text{(C 7)}$$

By exploiting (C 6) and (C 7), 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. $t_1 \in V \setminus \mathrm{vars}(\bar{s})$ 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 41, 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 $s_1 \in V \setminus \mathrm{vars}(t_1)$ 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,

$$s_1 \notin W_1 \stackrel{\text{def}}{=} \mathrm{vars}(\bar{s}_1) \cup \mathrm{vars}(\bar{t}_1), \tag{C 8}$$

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 $G_1 \subseteq G$ 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 $z \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$, 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 (C 8), 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 40 we obtain $\mu = \mu_1 \circ \nu \in \mathit{VSubst}$. Also note that, by Lemma 51, $\mu \in \mathrm{mgs}(\bar{s} = \bar{t})$.

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 C 3, C 4 and C 5 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 C 3, 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 C 4, 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 C 5, $\mathrm{vars}(z\mu_1) \cap \mathrm{vars}(z'\mu_1) \subseteq G_1$. 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 $t_1 \in V \setminus \mathrm{vars}(\bar{s})$ and $s_1 \notin \mathit{Vars}$. Let

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

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

$$t_1 \notin W_1 \stackrel{\mathrm{def}}{=} \mathrm{vars}(\bar{s}_1) \cup \mathrm{vars}(\bar{t}_1), \tag{C 9}$$

so that $W_1 \subset W$. Let $G_1 \stackrel{\mathrm{def}}{=} \mathrm{gvars}(\bar{s}_1 = \bar{t}_1)$ and $V_1 \stackrel{\mathrm{def}}{=} \mathit{Vars} \setminus G_1$. Note that $G_1 \subseteq G$ 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 $z \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$, the three inductive properties hold.

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

$$\mu \stackrel{\mathrm{def}}{=} \{t_1 = s_1\mu_1\} \cup \mu_1. \tag{C 10}$$

We show that $\mu \in \mathit{VSubst}$. Note that $\nu \stackrel{\mathrm{def}}{=} \{t_1 \mapsto s_1\} \in \mathit{VSubst}$ and, by (C 9), $\mathrm{dom}(\nu) \cap \mathrm{vars}(\mu_1) = \{t_1\} \cap \mathrm{vars}(\mu_1) = \varnothing$. Thus, since $\mu_1 \in \mathit{RSubst}$, by Lemma 40 we obtain $\mu = \mu_1 \circ \nu \in \mathit{VSubst}$. Also note that, by Lemma 51, $\mu \in \mathrm{mgs}(\bar{s} = \bar{t})$.

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 $z\mu = s_1\mu_1$. We show in each case that $z\mu$ satisfies the three required properties.

1. Suppose $z \neq t_1$. By inductive property C 3, $z\mu_1$ 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 $z\mu = s_1\mu_1$. Consider an arbitrary variable $u \in \mathrm{vars}(s_1)$. 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 C 3, $u\mu_1$ is V_1-linear.

Moreover, by property C 5, we have

$$\forall u' \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1) : u \neq u' \implies \mathrm{vars}(u\mu_1) \cap \mathrm{vars}(u'\mu_1) \subseteq G_1.$$

In particular, this holds for all $u' \in \mathrm{vars}(s_1)$ such that $u \neq u'$. 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 C 4, 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 $z\mu = s_1\mu_1$. Consider $u \in \mathrm{vars}(s_1)$. Then $u \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$, so that the inductive properties using μ_1 and $\bar{s}_1$ can be applied to $u\mu_1$. By property C 4, $\mathrm{vars}(u\mu_1) \cap \mathrm{dom}(\mu_1) \subseteq G_1$. 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 C 5, 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 $z'\mu = s_1\mu_1$). We will show that

$$\forall u \in \mathrm{vars}(s_1) : \mathrm{vars}(z\mu_1) \cap \mathrm{vars}(u\mu_1) \subseteq G_1. \tag{C 11}$$

In fact, in the case that $u \in G_1$ then $\mathrm{vars}(u\mu_1) \subseteq G_1$. On the other hand, if $u \in \mathrm{vars}(s_1) \setminus G_1$, then we have $u \neq z$. As $\bar{s}$ is V-linear, $u \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$ so that the property holds by inductive property C 5 (taking $z' = u$). As (C 11) holds, 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 $z\mu = s_1\mu_1$. 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 all $u \in \mathrm{vars}(s_1)$,

$$\mathrm{vars}(u\mu_1) \cap \mathrm{vars}(z'\mu_1) \subseteq G_1. \tag{C 12}$$

In fact, if $u \in G_1$ then $\mathrm{vars}(u\mu_1) \subseteq G_1$. Suppose now $u \in \mathrm{vars}(s_1) \setminus G_1$. As $\bar{s}$ is V-linear, $u \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$ so that the inductive property C 5 can be applied to $u\mu_1$. Thus, for all $u' \in \mathit{Vars} \setminus \mathrm{vars}(\bar{s}_1)$, if $u \neq u'$ we have $\mathrm{vars}(u\mu_1) \cap \mathrm{vars}(u'\mu_1) \subseteq G_1$. In particular, since $u \in \mathrm{vars}(s_1)$ and $z' \notin \mathrm{vars}(\bar{s})$, we have $u \neq z'$ so that, by taking $u' = z'$, we obtain (C 12). As the choice of $u \in \mathrm{vars}(s_1)$ is arbitrary, 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:

$$\begin{array}{l} d = \langle sh, f, l\rangle \in \mathit{SFL} \text{ and } \sigma \in \gamma_s(d) \cap \mathit{VSubst};\\ y \in \mathrm{dom}(\sigma) \cap \mathrm{range}(\sigma) \implies y \in \mathrm{vars}(y\sigma);\\ (x \mapsto t) \in \mathit{Bind}, \text{ where } \{x\} \cup \mathrm{vars}(t) \subseteq \mathit{VI};\\ r, r' \in \mathit{HTerms} \text{ are such that } \{r, r'\} = \{x, t\};\\ \sigma \cup \{x = t\} \text{ is satisfiable in the syntactic equality theory } T. \end{array} \tag{C 13}$$

Let

$$\begin{array}{rl} \{u_1, \ldots, u_k\} & \stackrel{\mathrm{def}}{=} \mathrm{dom}(\sigma) \cap \big(\mathrm{vars}(x\sigma) \cup \mathrm{vars}(t\sigma)\big),\\ \sigma' & \stackrel{\mathrm{def}}{=} \{u_i = u_i\sigma \mid 1 \le i \le k\},\\ \mu & \in \mathrm{mgs}\big(\sigma' \cup \{r\sigma = r'\sigma\}\big) \cap \mathit{VSubst},\\ \nu & \stackrel{\mathrm{def}}{=} \sigma \setminus \sigma'. \end{array} \tag{C 14}$$

As $\sigma \cup \{x = t\}$ is satisfiable, by Lemma 35, $\mathrm{mgs}\big(\sigma' \cup \{x\sigma = t\sigma\}\big) \neq \varnothing$. Therefore, by Proposition 41, μ is well defined and, by Lemma 39, $\nu \in \mathit{VSubst}$. 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

$$\mathrm{vars}(\mu) \subseteq \mathrm{vars}(x\sigma) \cup \mathrm{vars}(t\sigma) \tag{C 15}$$

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

$$\mathrm{dom}(\nu) \cap \mathrm{vars}(\mu) = \varnothing. \tag{C 16}$$

Thus we can apply Lemma 40 to obtain $\nu \circ \mu \in \mathit{VSubst}$. By applying Lemma 51, we have

$$\nu \circ \mu \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big) \cap \mathit{VSubst}. \tag{C 17}$$

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

Lemma 56
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 16, $y \in \mathrm{lvars}(\sigma)$. It follows that $\mathrm{vars}(r) \subseteq \mathrm{lvars}(\sigma)$ and hence, by Proposition 43,

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

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

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

Lemma 57
Assume the notation and hypotheses given in (C 13) and (C 14) 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 43 we have

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

Therefore, as $y \notin \mathrm{gvars}(\sigma)$, $y \notin \mathrm{vars}(y\sigma)$. and hence, by the definition of σ, $y \notin \mathrm{dom}(\sigma) \cap \mathrm{range}(\sigma)$. Thus, by (C 15), $y \notin \mathrm{dom}(\mu)$. □

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

$$\begin{aligned} sh_x &= \mathrm{rel}\big(\mathrm{vars}(x), sh\big), & sh_t &= \mathrm{rel}\big(\mathrm{vars}(t), sh\big), \\ sh_r &= \mathrm{rel}\big(\mathrm{vars}(r), sh\big), & sh_{r'} &= \mathrm{rel}\big(\mathrm{vars}(r'), sh\big), \\ sh_- &= \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), sh\big), & sh_{xt} &= sh_x \cap sh_t. \end{aligned} \tag{C 18}$$

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

$$\begin{aligned} R_x &= \mathrm{rel}\big(\mathrm{vars}(x), \mathrm{ssets}(\sigma)\big), & R_t &= \mathrm{rel}\big(\mathrm{vars}(t), \mathrm{ssets}(\sigma)\big), \\ R_r &= \mathrm{rel}\big(\mathrm{vars}(r), \mathrm{ssets}(\sigma)\big), & R_{r'} &= \mathrm{rel}\big(\mathrm{vars}(r'), \mathrm{ssets}(\sigma)\big), \\ R_- &= \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), \mathrm{ssets}(\sigma)\big), & R_{xt} &= R_x \cap R_t. \end{aligned} \tag{C 19}$$

C.4 The Correctness for Set-Sharing

Lemma 58
Assume the notation and hypotheses in (C 13), (C 14) and (C 19). Suppose also $\mathrm{rt}(r, \sigma) \in \mathit{LTerms}$, $\tau = \nu \circ \mu$, $S \in \mathrm{ssets}(\tau)$, $S_{r'} \subseteq S$ and $S_{r'} \in R^\star_{r'}$. 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 $S_1 = \bigcup\big\{\, \mathrm{occ}(\sigma, w) \;\big|\; w \in S_1 \setminus \big(\mathrm{dom}(\sigma) \cup \mathrm{vars}(r'\sigma)\big) \,\big\}$. Let $w_1, w_2 \in S_{r'} \setminus \big(\mathrm{dom}(\sigma) \cup \mathrm{vars}(r\sigma)\big)$. Then, it remains to show that $w_1 = w_2$.

As $S_{r'} \subseteq S$ and $S \in \mathrm{ssets}(\tau)$, by (C 17) and Proposition 43, $\big(\mathrm{vars}(w_1\tau) \cap \mathrm{vars}(w_2\tau)\big) \setminus \mathrm{dom}(\tau) \neq \varnothing$. 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 Proposition 43,

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

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 56 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 (C 2) of Proposition 55. to conclude that $w_1 = w_2$. □

Proposition 59

Assume the notation and hypotheses given in (C 13) and (C 18) and suppose that $\mathrm{free}_d(r)$ holds. Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$,

$$sh_- \cup \mathrm{bin}(sh_r, sh_{r'}) \supseteq \mathrm{ssets}(\tau). \tag{C 20}$$

Proof

Assuming the definitions in (C 14), we show that $\tau \stackrel{\mathrm{def}}{=} \nu \circ \mu$ satisfies (C 20).

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 (C 19),

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

Let S be an arbitrary sharing set in $\mathrm{ssets}(\tau)$. Then, in order to prove (C 20), it is sufficient to show that

$$S \in R_- \cup \mathrm{bin}(R_r, R_{r'}). \tag{C 21}$$

By hypothesis, $\mathrm{free}_d(r)$ holds so that, by Theorem 21, $\mathrm{rt}(r, \sigma) \in \mathit{Vars}$.

By Definition 49 and Theorem 50, $S \in R_- \cup \mathrm{bin}(R^\star_r, R^\star_{r'})$. If $S \in R_-$, then (C 21) 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 58 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 16 and 43, $r\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. Therefore, by Definition 7, $R_r = \big\{\mathrm{occ}(\sigma, r\sigma)\big\}$ so that $R^\star_r = R_r$. Thus $S \in \mathrm{bin}(R_r, R_{r'})$ and (C 21) holds. □

Proposition 60

Assume the notation and hypotheses given in (C 13) and (C 18) and suppose that both $\mathrm{lin}_d(x)$ and $\mathrm{lin}_d(t)$ hold. Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$,

$$sh_- \cup \mathrm{bin}\big(sh_x \cup \mathrm{bin}(sh_x, sh^\star_{xt}), sh_t \cup \mathrm{bin}(sh_t, sh^\star_{xt})\big) \supseteq \mathrm{ssets}(\tau). \tag{C 22}$$

Proof

Assuming the definitions in (C 14), we show that $\tau \stackrel{\mathrm{def}}{=} \nu \circ \mu$ satisfies (C 22).

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 (C 19), we obtain

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

Let S be an arbitrary sharing set in $\mathrm{ssets}(\tau)$. Then, in order to prove (C 22), it is sufficient to show

$$S \in R_- \cup \mathrm{bin}\big(R_x \cup \mathrm{bin}(R_x, R^\star_{xt}), R_t \cup \mathrm{bin}(R_t, R^\star_{xt})\big) \tag{C 23}$$

By Definition 49 and Theorem 50, we have

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

If $S \in R_-$, then (C 23) 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 21, $\mathrm{rt}(x,\sigma)$ and $\mathrm{rt}(t,\sigma) \in \mathit{LTerms}$ so that we can apply Lemma 58 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 (C 23) holds. □

Proposition 61
Assume the notation and hypotheses given in (C 13) and (C 18) and suppose $\mathrm{lin}_d(r)$ holds. Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$,

$$sh_- \cup \mathrm{bin}(sh^\star_r, sh_{r'}) \supseteq \mathrm{ssets}(\tau). \tag{C 24}$$

Proof
Assuming the definitions in (C 14), we show that $\tau \stackrel{\text{def}}{=} \nu \circ \mu$ satisfies (C 24).

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 (C 19), 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 (C 24) it is sufficient to show that

$$S \in R_- \cup \mathrm{bin}(R^\star_r, R_{r'}). \tag{C 25}$$

By Definition 49 and Theorem 50, we have

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

If $S \in R_-$, then (C 25) 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 21, $\mathrm{rt}(r,\sigma) \in \mathit{LTerms}$ so that we can apply Lemma 58, 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 (C 25) holds. □

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

Lemma 62
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 $sh' \stackrel{\text{def}}{=} \mathrm{amgu}(sh, x \mapsto t)$. Then $\mathrm{rel}(V, sh') \subseteq \mathrm{rel}(W, sh')$.

Proof
Suppose that $S \in \mathrm{rel}(V, sh')$. By Definition 49, 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 $0 \le i \le n$. 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 63
Let $sh \in SH$ and $\tau \in RSubst$ be satisfiable in T and suppose that $T \vdash \forall(\tau \rightarrow x = t)$ and $\mathrm{ssets}(\tau) \subseteq sh$. Then $\mathrm{ssets}(\tau) \subseteq \mathrm{cyclic}_x^t(sh)$.

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

$$\mathrm{rel}(V_x, sh_i) \subseteq \mathrm{rel}(W_t, sh_i). \tag{C 26}$$

The base case, when $i = 0$, follows directly from Definition 7; note that (C 26) 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 50, $\mathrm{ssets}(\tau_i) \subseteq sh_i$. By the inductive hypothesis, (C 26) holds for sh_{i-1}. Thus, by Lemma 62 (taking $V = V_x$ and $W = W_t$), we obtain that (C 26) 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 54, 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 20, we obtain $\mathrm{ssets}(\tau) \subseteq \mathrm{cyclic}_x^t(sh)$. □

C.5 The Correctness for Freeness

Lemma 64
Assume the notation and definitions given in (C 13) and (C 14) and suppose $\tau = \nu \circ \mu$ 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 57 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 $\mu \in VSubst$, by Proposition 43, $y\sigma\mu \in Vars \setminus \mathrm{dom}(\mu)$. If $y\sigma \in \mathrm{dom}(\mu)$, then $y\sigma\mu \in \mathrm{vars}(\mu)$, so that, by (C 16), $y\sigma\mu \notin \mathrm{dom}(\nu)$. On the other hand, if $y\sigma \notin \mathrm{dom}(\mu)$, then $y\sigma\mu \notin \mathrm{dom}(\mu)$. Thus, in both cases, $y\sigma\mu \in Vars \setminus \big(\mathrm{dom}(\sigma) \cup \mathrm{dom}(\mu)\big)$ and hence, as $\mathrm{dom}(\tau) \subseteq \mathrm{dom}(\sigma) \cup \mathrm{dom}(\mu)$, $y\sigma\mu \in Vars \setminus \mathrm{dom}(\tau)$. Since, by (C 17), $\tau \in VSubst$, we can apply Proposition 43 to obtain $y \in \mathrm{fvars}(\tau)$ as required. □

Proposition 65
Assume the notation and hypotheses given in (C 13) and suppose $\{x, t\} \subseteq \mathrm{fvars}(\sigma)$. Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$, we have

$$\mathrm{fvars}(\sigma) \subseteq \mathrm{fvars}(\tau). \tag{C 27}$$

Proof
Assuming the definitions in (C 14), we show that $\tau \stackrel{\mathrm{def}}{=} \nu \circ \mu$ satisfies (C 27). Suppose $y \in \mathrm{fvars}(\sigma)$. As $\{x, t\} \subseteq \mathrm{fvars}(\sigma)$ we have, using Proposition 43, $\{x\sigma, t\sigma\} \subseteq \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. 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 64 can be applied to give $y \in \mathrm{fvars}(\tau)$. □

The following simple consequence of Proposition 43 will be used to complete the proofs of correctness for the freeness component.

Lemma 66
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 20, $y \notin \mathrm{vars}\big(\mathrm{rel}(V, \mathit{sh})\big)$. Thus

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

By Definition 18, 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 43 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 67
Assume the notation and hypotheses given in (C 13) and suppose that $x \in \mathrm{fvars}(\sigma)$. Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$, we have

$$\mathrm{fvars}(\sigma) \setminus \mathrm{share\_with}_d(x) \subseteq \mathrm{fvars}(\tau). \tag{C 28}$$

Proof
Assuming the definitions in (C 14), we show that $\tau \stackrel{\mathrm{def}}{=} \nu \circ \mu$ satisfies (C 28). If $x\sigma = t\sigma$, then $t \in \mathrm{fvars}(\sigma)$ and the result follows from Proposition 65. Suppose next that $x\sigma \neq t\sigma$ and that $y \in \mathrm{fvars}(\sigma) \setminus \mathrm{share\_with}_d(x)$. We prove that $y \in \mathrm{fvars}(\tau)$. First we show that $y\sigma \in \mathrm{fvars}(\mu)$.

As $x \in \mathrm{fvars}(\sigma)$ and $\sigma \in \mathit{VSubst}$, by Proposition 43, $x\sigma \in \mathit{Vars} \setminus \mathrm{dom}(\sigma)$. Thus $\mu' \stackrel{\mathrm{def}}{=} \sigma' \cup \{x\sigma = t\sigma\}$ has no identities or circular subsets so that $\mu' \in \mathit{RSubst}$. We show that $y\sigma \in \mathrm{fvars}(\mu')$. Moreover, $\sigma \in \mathit{VSubst}$, so that, by Proposition 43, $y\sigma \notin \mathrm{dom}(\sigma)$. However, since $y \notin \mathrm{share\_with}_d(x)$ and $\sigma \in \mathit{VSubst}$, we can apply Lemma 66 to give $\mathrm{vars}(y\sigma) \cap \mathrm{vars}(x\sigma) \subseteq \mathrm{dom}(\sigma)$. Therefore $y\sigma \neq x\sigma$ and hence

$y\sigma \notin \mathrm{dom}(\sigma) \cup \{x\sigma\}$. Thus $y\sigma \notin \mathrm{dom}(\mu')$ so that $y\sigma \in \mathrm{fvars}(\mu')$. By Proposition 54, we have $y\sigma \in \mathrm{fvars}(\mu)$.

Therefore Lemma 64 can be applied to give $y \in \mathrm{fvars}(\tau)$. □

Proposition 68
Assume the notation and hypotheses given in (C 13). Therefore we have, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$,

$$\mathrm{fvars}(\sigma) \setminus \big(\mathrm{share\_with}_d(x) \cup \mathrm{share\_with}_d(t)\big) \subseteq \mathrm{fvars}(\tau). \tag{C 29}$$

Proof
Assuming the definitions in (C 14), we show that $\tau \stackrel{\mathrm{def}}{=} \nu \circ \mu$ satisfies (C 29). Suppose $y \in \mathrm{fvars}(\sigma) \setminus \big(\mathrm{share\_with}_d(x) \cup \mathrm{share\_with}_d(t)\big)$. Then we need to show that $y \in \mathrm{fvars}(\tau)$.

As $y \in \mathrm{fvars}(\sigma)$, we have $y \in \mathit{Vars}$ and $y\sigma \in \mathit{Vars}$. Moreover, as

$$y \notin \mathrm{share\_with}_d(x) \cup \mathrm{share\_with}_d(t),$$

we can apply Lemma 66 to obtain

$$\mathrm{vars}(y\sigma) \cap \big(\mathrm{vars}(x\sigma) \cup \mathrm{vars}(t\sigma)\big) \subseteq \mathrm{dom}(\sigma)$$

so that, by (C 15), $y\sigma \notin \mathrm{vars}(\mu)$. It follows that $y\sigma \in \mathrm{fvars}(\mu)$. Therefore Lemma 64 can be applied to give $y \in \mathrm{fvars}(\tau)$ as required. □

C.6 The Correctness for Linearity

Lemma 69
Assume the notation and definitions given in (C 13) and (C 14). Suppose also that $\tau = \nu \circ \mu$, $W \subseteq \mathrm{lvars}(\mu)$ and $y \in \mathrm{lvars}(\sigma) \cap W$ where $\mathrm{vars}(y\sigma) \setminus W \subseteq \mathrm{gvars}(\sigma)$ and, for all $w, w' \in W$ such that $w \neq w'$, we have $\mathrm{vars}(w\mu) \cap \mathrm{vars}(w'\mu) \subseteq \mathrm{gvars}(\mu)$. Then $y \in \mathrm{lvars}(\tau)$.

Proof
By Lemma 53, 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 17, $y \in \mathrm{lvars}(\tau)$. Therefore, for the rest of the proof, we assume $y \in \mathrm{dom}(\tau) \setminus \mathrm{gvars}(\sigma)$.

By Lemma 57, if $y \in \mathrm{dom}(\mu)$ then $y \notin \mathrm{dom}(\sigma)$. Thus, as $\nu \subseteq \sigma$, we have

$$y\tau = y\sigma\mu. \tag{C 30}$$

By definition $\mu \in \mathit{VSubst}$. Thus, by Proposition 43, for all $w \in \mathrm{lvars}(\mu)$ (and hence, for all $w \in W$),

$$\forall v \in \mathrm{vars}(w\mu) \setminus \mathrm{dom}(\mu) : \mathrm{occ\_lin}(v, w\mu), \tag{C 31}$$

$$\mathrm{vars}(w\mu) \cap \mathrm{dom}(\mu) \subseteq \mathrm{gvars}(\mu). \tag{C 32}$$

By definition, $\sigma \in \mathit{VSubst}$. Thus, as $y \in \mathrm{lvars}(\sigma)$, by Proposition 43,

$$\forall v \in \mathrm{vars}(y\sigma) \setminus \mathrm{dom}(\sigma) : \mathrm{occ\_lin}(v, y\sigma), \tag{C 33}$$

$$\mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma) \subseteq \mathrm{gvars}(\sigma). \tag{C 34}$$

By (C 17), $\tau \in \mathit{VSubst}$. Thus, by Proposition 43, to prove $y \in \mathrm{lvars}(\tau)$, we just need to show that

$$\forall v \in \mathrm{vars}(y\tau) \setminus \mathrm{dom}(\tau) : \mathrm{occ\_lin}(v, y\tau), \tag{C 35}$$

$$\mathrm{vars}(y\tau) \cap \mathrm{dom}(\tau) \subseteq \mathrm{gvars}(\tau). \tag{C 36}$$

Let $w \in \mathrm{vars}(y\sigma)$. Then, it follows from (C 30) that to prove (C 35) and (C 36), we need to prove:

$$\mathrm{vars}(w\mu) \cap \mathrm{dom}(\tau) \subseteq \mathrm{gvars}(\tau), \tag{C 37}$$

$$\forall v \in \mathrm{vars}(w\mu) \setminus \mathrm{gvars}(\tau) : \mathrm{occ\_lin}(v, w\mu), \tag{C 38}$$

$$\forall w' \in \mathrm{vars}(y\sigma) \setminus \{w\} : \mathrm{vars}(w\mu) \cap \mathrm{vars}(w'\mu) \subseteq \mathrm{gvars}(\tau). \tag{C 39}$$

We first prove (C 37) and (C 38) hold. Suppose $w \in \mathrm{vars}(y\sigma) \setminus W$. Then, by hypothesis, $w \in \mathrm{gvars}(\sigma)$ and hence, $w \in \mathrm{gvars}(\tau)$. Thus, as $T \vdash \forall(\tau \to \mu)$, by Lemma 35, Lemma 36 and Proposition 16, $\mathrm{vars}(w\mu) \subseteq \mathrm{gvars}(\tau)$. Thus (C 37) and (C 38) hold. Suppose now that $w \in \mathrm{vars}(y\sigma) \cap W$ so that (C 31) and (C 32) hold. Thus, as $\mathrm{dom}(\mu) \subseteq \mathrm{dom}(\tau)$ and $\mathrm{gvars}(\mu) \subseteq \mathrm{gvars}(\tau)$, both (C 37) and (C 38) hold.

We now prove (C 39) holds. Suppose $w' \in \mathrm{vars}(y\sigma)$. If $w, w' \notin \mathrm{dom}(\tau)$. Then $w, w' \notin \mathrm{dom}(\mu)$ so that $\mathrm{vars}(w\mu) \cap \mathrm{vars}(w'\mu) = \{w\} \cap \{w'\} = \varnothing$. If $w \in \mathrm{dom}(\tau)$, then, by (C 37), $\mathrm{vars}(w\mu) \subseteq \mathrm{gvars}(\tau)$. Similarly, if $w' \in \mathrm{dom}(\tau)$, then, by (C 37) where w is replaced by w', $\mathrm{vars}(w'\mu) \subseteq \mathrm{gvars}(\tau)$. Thus, in all cases (C 39) holds. □

Proposition 70

Assume the notation and hypotheses given in (C 13) and suppose $\mathrm{lin}_d(r)$ holds. Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$,

$$\mathrm{lvars}(\sigma) \setminus \mathrm{share\_with}_d(r) \subseteq \mathrm{lvars}(\tau). \tag{C 40}$$

Proof

Assuming the definitions in (C 14), we show that $\tau \stackrel{\text{def}}{=} \nu \circ \mu$ satisfies (C 40). To prove this, we assume that $y \in \mathrm{lvars}(\sigma) \setminus \mathrm{share\_with}_d(r)$ and show that $y \in \mathrm{lvars}(\tau)$.

Let $\bar{s} \stackrel{\text{def}}{=} (u_1, \ldots, u_k, r\sigma)$. We will show that the hypothesis of Lemma 69 holds where $W = \mathit{Vars} \setminus \mathrm{vars}(\bar{s})$

As $y \in \mathrm{lvars}(\sigma)$, by Proposition 43, $\mathrm{vars}(y\sigma) \cap \mathrm{dom}(\sigma) \subseteq \mathrm{gvars}(\sigma)$. As $y \notin \mathrm{share\_with}_d(r)$, by Definition 20 and Theorem 21, we have $y \notin \mathrm{vars}(r\sigma)$ and $\mathrm{vars}(y\sigma) \cap \mathrm{vars}(r\sigma) \subseteq \mathrm{gvars}(\sigma)$, so that, we obtain $\mathrm{vars}(y\sigma) \setminus W \subseteq \mathrm{gvars}(\sigma)$. As $y \notin \mathrm{vars}(r\sigma)$ and $\{u_1, \ldots, u_k\} \subseteq \mathrm{dom}(\sigma)$, we also have $y \notin \mathrm{vars}(\bar{s})$ and hence, $y \in W$.

Since $\mathrm{lin}_d(r)$ holds, by Theorem 21, $\mathrm{rt}(r, \sigma) \in \mathit{LTerms}$ so that, by Lemma 56, $\bar{s}$ is $\big(\mathit{Vars} \setminus \mathrm{gvars}(\mu)\big)$-linear. We can thus apply Proposition 55 to obtain $W \subseteq \mathrm{lvars}(\mu)$ and 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 43, as $\mu \in \mathit{VSubst}$, we have, for all $w, w' \in W$ where $w \neq w'$,

$\mathrm{vars}(w\mu) \cap \mathrm{vars}(w'\mu) \subseteq \mathrm{gvars}(\mu)$. Thus we can apply Lemma 69, and obtain $y \in \mathrm{lvars}(\tau)$. □

Proposition 71
Assume the notation and hypotheses given in (C 13) 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 70 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 72
Assume the notation and hypotheses given in (C 13). Then, for some $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$, we have

$$\mathrm{lvars}(\sigma) \setminus \big(\mathrm{share\_with}_d(x) \cup \mathrm{share\_with}_d(t)\big) \subseteq \mathrm{lvars}(\tau). \tag{C 41}$$

Proof
Assuming the definitions in (C 14), we show that $\tau \stackrel{\mathrm{def}}{=} \nu \circ \mu$ satisfies (C 41). 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 69 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 20 and Theorem 21, $y \notin \mathrm{vars}(x\sigma) \cup \mathrm{vars}(t\sigma)$ 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 69, 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 $\mathrm{amgu}_{_S}$ operator. We start by proving a restricted result that only applies to variable-idempotent substitutions.

Lemma 73
Assume the notation and hypotheses given in (C 13). 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 18 that to prove $\tau \in \gamma_{_S}(d')$, we have to show that

$$\mathrm{ssets}(\tau) \subseteq sh', \tag{C 42}$$

$$\mathrm{fvars}(\tau) \supseteq f', \tag{C 43}$$

$$\mathrm{lvars}(\tau) \supseteq l'. \tag{C 44}$$

We prove, for each component of d' separately, that there exists $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$ such that (C 42), (C 43) and (C 44) hold. By Lemma 54 the same inclusions hold for any $\tau \in \mathrm{mgs}\big(\sigma \cup \{x = t\}\big)$ so that we have the thesis.

(C 42). By Definition 22, we have $sh' = \mathrm{cyclic}^t_x(sh_- \cup sh'')$ where

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

It follows from Definition 18 and the hypothesis $\sigma \in \gamma_S(d)$ that $\mathrm{ssets}(\sigma) \subseteq sh$. Therefore we apply Proposition 59 (when $\mathrm{free}_d(x) \vee \mathrm{free}_d(t)$ holds), Proposition 60 (when $\mathrm{lin}_d(x) \wedge \mathrm{lin}_d(t)$ holds), Proposition 61 (when $\mathrm{lin}_d(x)$ or $\mathrm{lin}_d(t)$ holds) or Theorem 50 (when nothing is known about x or t) to conclude that $\mathrm{ssets}(\tau) \subseteq sh_- \cup sh''$. The thesis now follows from Proposition 63.

(C 43). It follows from Definition 18 and the hypothesis $\sigma \in \gamma_S(d)$ that $f \subseteq \mathrm{fvars}(\sigma)$. Therefore we can apply Proposition 65 (when $\mathrm{free}_d(x) \wedge \mathrm{free}_d(t)$ holds), Proposition 67 (when $\mathrm{free}_d(x)$ or $\mathrm{free}_d(t)$ holds) or Proposition 68 (when nothing is known about x or t) to conclude that $\mathrm{fvars}(\tau) \supseteq f'$.

(C 44). It follows from Definition 18 and the hypothesis $\sigma \in \gamma_S(d)$ that $l \subseteq \mathrm{fvars}(\sigma)$. Therefore we can apply Proposition 71 (when $\mathrm{lin}_d(x) \wedge \mathrm{lin}_d(t)$ holds), Proposition 70 (when $\mathrm{lin}_d(x)$ or $\mathrm{lin}_d(t)$ holds) or Proposition 72 (when nothing is known about x or t) to conclude that

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

By (C 42) proved above, $\mathrm{ssets}(\tau) \subseteq sh'$. By Definition 9, $VI \setminus \mathrm{vars}(sh') \subseteq \mathrm{gvars}(\tau)$. Moreover, by (C 43) proved above, we have $\mathrm{fvars}(\tau) \supseteq f'$. Thus, by applying Corollary 17 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 23 on page 15. 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 41, 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 19 and Definition 18, we have $\sigma \in \gamma_S(d)$ if and only if $\sigma' \in \gamma_S(d)$. Therefore, the result follows by application of Lemma 73.

D Proofs of the Results of Section 4

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 pair-sharing.

Theorem 74
(Zaffanella et al. 2002, Theorem 3). Let $sh_1, sh_2 \in \mathit{SH}$ be such that $\rho_{\mathit{PSD}}(sh_1) = \rho_{\mathit{PSD}}(sh_2)$. Then, for each $\sigma \in \mathit{RSubst}$ and $V \subseteq \mathit{VI}$,

$$\rho_{\mathit{PSD}}\big(\mathrm{aunify}(sh_1, \sigma)\big) = \rho_{\mathit{PSD}}\big(\mathrm{aunify}(sh_2, \sigma)\big), \tag{D 1}$$

$$\rho_{\mathit{PSD}}\big(\mathrm{aexists}(sh_1, V)\big) = \rho_{\mathit{PSD}}\big(\mathrm{aexists}(sh_2, V)\big). \tag{D 2}$$

Theorem 75
(Zaffanella et al. 2002, Theorem 4). Let $sh_1, sh_2 \in \mathit{SH}$ be such that $\rho_{\mathit{PSD}}(sh_1) \neq \rho_{\mathit{PSD}}(sh_2)$. Then there exist $\sigma \in \mathit{RSubst}$ and $\rho \in \{\rho_{\mathit{Con}}, \rho_{\mathit{PS}}\}$ such that

$$\rho\big(\mathrm{aunify}(sh_1, \sigma)\big) \neq \rho\big(\mathrm{aunify}(sh_2, \sigma)\big). \tag{D 3}$$

The following result in (Bagnara et al. 2002) shows that, for groundness and pair-sharing information, the exponential star-union operator can be replaced by a quadratic operation without loss of precision.

Theorem 76
(Bagnara et al. 2002, Theorem 15). Let $sh \in \mathit{SH}$ and $(x \mapsto t) \in \mathit{Bind}$. Let also

$$sh_{-} \stackrel{\mathrm{def}}{=} \overline{\mathrm{rel}}\big(\{x\} \cup \mathrm{vars}(t), sh\big), \quad sh_x \stackrel{\mathrm{def}}{=} \mathrm{rel}\big(\{x\}, sh\big), \quad sh_t \stackrel{\mathrm{def}}{=} \mathrm{rel}\big(\mathrm{vars}(t), sh\big).$$

Then

$$\rho_{\mathit{PSD}}\big(\mathrm{amgu}(sh, x \mapsto t)\big) = \rho_{\mathit{PSD}}\big(sh_{-} \cup \mathrm{bin}(sh_x^2, sh_t^2)\big).$$

The next three lemmas show that the precision of the abstract evaluation of the operators specified in Definition 20 is not affected by ρ_{PSD}.

Lemma 77
Let $V \subseteq \mathit{VI}$ and $sh \in \mathit{SH}$. Then

$$\mathrm{vars}\big(\mathrm{rel}(V, sh)\big) = \mathrm{vars}\Big(\mathrm{rel}\big(V, \rho_{\mathit{PSD}}(sh)\big)\Big).$$

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 5, we have

$$\forall x \in S : S = \bigcup \big\{\, T \in sh \;\big|\; \{x\} \subseteq T \subseteq S \,\big\}.$$

In particular, for all $x \in S \cap V$, it holds

$$\begin{aligned} S &= \bigcup \big\{\, T \in sh \;\big|\; \{x\} \subseteq T \subseteq S \,\big\} \\ &= \bigcup \big\{\, T \in \mathrm{rel}(V, sh) \;\big|\; \{x\} \subseteq T \subseteq S \,\big\} \\ &\subseteq \bigcup \mathrm{rel}(V, sh) \\ &= \mathrm{vars}\big(\mathrm{rel}(V, sh)\big). \end{aligned}$$

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 78
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 5, we have

$$\forall v \in S : S = \bigcup \big\{\, T \in sh \;\big|\; \{v\} \subseteq T \subseteq S \,\big\}.$$

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 79
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 $y \in VI$,

$$\mathrm{ind}_{d_1}(s, t) \iff \mathrm{ind}_{d_2}(s, t); \tag{D 4}$$

$$\mathrm{free}_{d_1}(t) \iff \mathrm{free}_{d_2}(t); \tag{D 5}$$

$$\mathrm{ground}_{d_1}(t) \iff \mathrm{ground}_{d_2}(t); \tag{D 6}$$

$$\mathrm{occ\_lin}_{d_1}(y,t) \iff \mathrm{occ\_lin}_{d_2}(y,t); \tag{D 7}$$
$$\mathrm{lin}_{d_1}(t) \iff \mathrm{lin}_{d_2}(t); \tag{D 8}$$
$$\mathrm{share\_with}_{d_1}(t) = \mathrm{share\_with}_{d_2}(t). \tag{D 9}$$

Proof
(D 4). Let $V = \mathrm{vars}(s)$, $W = \mathrm{vars}(t)$. By Definition 20, Lemma 78 and the hypothesis, we obtain

$$\begin{aligned}
\mathrm{ind}_{d_1}(s,t) &\iff \mathrm{rel}(V,sh_1) \cap \mathrm{rel}(W,sh_1) \neq \varnothing \\
&\iff \mathrm{rel}\big(V,\rho_{PSD}(sh_1)\big) \cap \mathrm{rel}\big(W,\rho_{PSD}(sh_1)\big) \neq \varnothing \\
&\iff \mathrm{rel}\big(V,\rho_{PSD}(sh_2)\big) \cap \mathrm{rel}\big(W,\rho_{PSD}(sh_2)\big) \neq \varnothing \\
&\iff \mathrm{rel}(V,sh_2) \cap \mathrm{rel}(W,sh_2) \neq \varnothing \\
&\iff \mathrm{ind}_{d_2}(s,t).
\end{aligned}$$

(D 5). This follows from Definition 20, since the predicate $\mathrm{free}_{d_i}(t)$ does not depend on the sharing component sh_i of d_i.

(D 6). By applying Lemma 77 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 20, equivalence (D 6) holds.

(D 7). This follows from Definition 20 and equivalences (D 4) and (D 6).

(D 8). This follows from Definition 20 and equivalence (D 7).

(D 9). By applying Lemma 77 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 20, equation (D 9) holds. □

Since both ρ_{PSD} (by (Hill et al. 2002, Theorem 7)) and $(\cdot)^\star$ are upper closure operators it follows that

$$sh_1 \subseteq \rho_{PSD}(sh_2) \iff \rho_{PSD}(sh_1) \subseteq \rho_{PSD}(sh_2), \tag{D 10}$$
$$sh_1 \subseteq sh_2^\star \iff sh_1^\star \subseteq sh_2^\star. \tag{D 11}$$

We next establish a number of results needed to show that the precision of the abstract operator amgu_S specified in Definition 22 is not affected by ρ_{PSD} (see Theorem 85).

Lemma 80
(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 81
(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 82

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

$$\mathrm{bin}(sh_{x,1}, sh_{t,1}) \subseteq \rho_{PSD}\big(sh_{-,2} \cup \mathrm{bin}(sh_{x,2}, sh_{t,2})\big); \tag{D 12}$$

$$\mathrm{bin}(sh_{x,1}, sh^{\star}_{t,1}) \subseteq \rho_{PSD}\big(sh_{-,2} \cup \mathrm{bin}(sh_{x,2}, sh^{\star}_{t,2})\big). \tag{D 13}$$

Proof
(D 12). 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 $y \in S$,

$$S = \bigcup \big\{\, S' \in sh_{-,2} \cup \mathrm{bin}(sh_{x,2}, sh_{t,2}) \;\big|\; \{y\} \subseteq S' \subseteq S \,\big\}. \tag{D 14}$$

Suppose first that $y \in S_x$. Let

$$sh_y \stackrel{\text{def}}{=} \{S' \in sh_2 \mid \{y\} \subseteq S' \subseteq S\},$$
$$sh'_t \stackrel{\text{def}}{=} \big\{\, S' \in sh_2 \;\big|\; \exists w \in W \,.\, \{w\} \subseteq S' \subseteq S \,\big\}.$$

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 5, $S_x \subseteq \bigcup sh_y \subseteq S$ 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 (D 14) holds for all $y \in S_x$.

By a symmetric argument, (D 14) holds for all $y \in S_t$. Thus (D 14) holds for all $y \in S$. Therefore, by Definition 5,

$$S \in \rho_{PSD}\big(sh_{-,2} \cup \mathrm{bin}(sh_{x,2}, sh_{t,2})\big).$$

(D 13). 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 81, $T_t \in sh^{\star}_{t,2}$. We show that, for any $y \in S$,

$$S = \bigcup \big\{\, S' \in sh_{-,2} \cup \mathrm{bin}(sh_{x,2}, sh^{\star}_{t,2}) \;\big|\; \{y\} \subseteq S' \subseteq S \,\big\}. \tag{D 15}$$

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 5, 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

$$S = \bigcup\big((sh_y \cap sh_{-,2}) \cup (sh_y \cap sh_{x,2}) \cup sh'_t\big).$$

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 (D 15) 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

$$sh_y \stackrel{\text{def}}{=} \{S' \in sh^\star_{t,2} \mid R \subseteq S' \subseteq S\},$$
$$sh'_x \stackrel{\text{def}}{=} \big\{\, S' \in sh_2 \;\big|\; \exists v \in V \,.\, \{v\} \subseteq S' \subseteq S \,\big\}.$$

Then $T_t \subseteq \bigcup sh_y \subseteq S$. By Definition 5, 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 $sh_y \subseteq sh^\star_{t,2}$, (D 15) holds for all $y \in T_t$.

Therefore, (D 15) holds for all $y \in S$. Thus, by Definition 5,

$$S \in \rho_{PSD}\big(sh_{-,2} \cup \mathrm{bin}(sh_{x,2}, sh^\star_{t,2})\big).$$

□

Lemma 83
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

$$sh^\diamond = \mathrm{bin}(sh_x, sh_t) \cup \mathrm{bin}\big(\mathrm{bin}(sh_x, sh_t), sh^\star_{xt}\big). \tag{D 16}$$

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

$$\rho_{PSD}(sh^\diamond) \subseteq \rho_{PSD}\big(\mathrm{bin}(sh_x, sh_t)\big). \tag{D 17}$$

Let $S \in sh^\diamond$. Then, by (D 16), $S = S_x \cup S_t \cup T_{xt}$, 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

$$S = (S_x \cup S_t) \cup (T_1 \cup S_x) \cup \cdots \cup (T_k \cup S_x), \tag{D 18}$$

$$S = (S_x \cup S_t) \cup (T_1 \cup S_t) \cup \cdots \cup (T_k \cup S_t), \tag{D 19}$$

and, for all $j \in \{1, \ldots, k\}$,

$$S = (S_x \cup T_j) \cup (S_t \cup T_j) \cup (T_1 \cup T_j) \cup \cdots \cup (T_k \cup T_j). \tag{D 20}$$

Consider an arbitrary variable $y \in S$. We will show that

$$S = \bigcup \{\, S' \in \mathrm{bin}(sh_x, sh_t) \mid \{y\} \subseteq S' \subseteq S \,\}. \tag{D 21}$$

If $y \in S_x$, then (D 21) follows from (D 18). Similarly, if $y \in S_t$, then (D 21) follows from (D 19). 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 $T_j \in sh_{xt}$, (D 21) follows from (D 20).

As (D 21) holds for any $y \in S$, by Definition 5, $S \in \rho_{PSD}\big(\mathrm{bin}(sh_x, sh_t)\big)$. Thus, by monotonicity and idempotence of ρ_{PSD}, (D 17) holds. □

Lemma 84
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

$$\mathrm{bin}(sh_x, sh_W) = \mathrm{cyclic}_x^t\big(\mathrm{bin}(sh_x, sh_t)\big); \tag{D 22}$$

$$\mathrm{bin}(sh_x^\star, sh_W) = \mathrm{cyclic}_x^t\big(\mathrm{bin}(sh_x^\star, sh_t)\big); \tag{D 23}$$

$$\mathrm{bin}(sh_x^\star, sh_W^\star) = \mathrm{cyclic}_x^t\big(\mathrm{bin}(sh_x^\star, sh_t^\star)\big). \tag{D 24}$$

Proof
We prove equations (D 22) and (D 23) 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 20, $S \in \mathrm{cyclic}_x^t\big(\mathrm{bin}(sh'_x, sh_t)\big)$.

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 20, $W \cap S \neq \varnothing$. If $W \cap S_t \neq \varnothing$ then $S_t \in sh_W$ and $S \in \mathrm{bin}(sh'_x, sh_W)$, so that the two inclusions hold. If $W \cap S_t = \varnothing$, then $W \cap S_x \neq \varnothing$ and $x \in S_t$ and hence $S_t \in sh'_x$. Thus, in the case that $sh'_x = sh_x$, $S_t \in sh_x$ so that $S \in \mathrm{bin}(sh_x, sh_W)$, proving the inclusion for (D 22). On the other hand, when $sh'_x = sh_x^\star$, as $S_x = S_W \cup S_x$ for some $S_W \in sh_W$, we have $S = (S_x \cup S_t) \cup S_W \in \mathrm{bin}(sh_x^\star, sh_W)$, proving the inclusion for (D 23).

We now prove equation (D 24).

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 20, $S \in \mathrm{cyclic}_x^t\big(\mathrm{bin}(sh_x^\star, sh_t^\star)\big)$.

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 20, $W \cap S \neq \varnothing$. 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

$S_t \in sh^\star_x$ and $W \cap S_x \neq \varnothing$. In this case, since $S_x = S_W \cup S_x$ for some $S_W \in sh_W$, $S = (S_x \cup S_t) \cup S_W \in \mathrm{bin}(sh^\star_x, sh^\star_W)$. Thus, in both cases, we have the inclusion. □

Theorem 85
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 29, $d_2 = \langle sh_2, f, l\rangle$, where $\rho_{PSD}(sh_1) = \rho_{PSD}(sh_2)$. By (D 10), 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:

$$sh'_1 \subseteq \rho_{PSD}(sh'_2), \tag{D 25}$$
$$f'_1 = f'_2, \tag{D 26}$$
$$l'_1 = l'_2. \tag{D 27}$$

From (D 25) using (D 10), 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 29.

(D 25). 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 $sh_{W,i} = \mathrm{rel}(W, sh_i)$, where $W = \mathrm{vars}(t) \setminus \{x\}$ and $i \in \{1, 2\}$.

By Definitions 20 and 22, 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 80, $sh_{-,1} \subseteq \rho_{PSD}(sh_{-,2})$, from which the required result follows by monotonicity of ρ_{PSD}.

We next show that

$$\mathrm{cyclic}^t_x(sh''_1) \subseteq \rho_{PSD}\big(sh_{-,2} \cup \mathrm{cyclic}^t_x(sh''_2)\big). \tag{D 28}$$

By applying cases (D 5) and (D 8) of Lemma 79 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 (D 22) of Lemma 84, $\mathrm{cyclic}^t_x(sh''_i) = \mathrm{bin}(sh_{x,i}, sh_{W,i})$, for each $i = 1, 2$. Thus, using case (D 12) of Lemma 82, where we take $V = \{x\}$, we obtain (D 28).
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 83, 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 (D 12) of Lemma 82 and the monotonicity of ρ_{PSD}, we obtain (D 28).
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, proven 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 (D 23) of Lemma 84, $\mathrm{cyclic}^t_x(sh''_i) = \mathrm{bin}(sh^\star_{x,i}, sh_{W,i})$. Thus, by using case (D 13) of Lemma 82 (exchanging the roles of $V = \{x\}$ and W), we obtain (D 28).
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 (D 13) of Lemma 82 where $V = \{x\}$, we obtain (D 28).
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 (D 24) of Lemma 84, $\mathrm{cyclic}^t_x(sh''_i) = \mathrm{bin}(sh^\star_{x,i}, sh^\star_{W,i})$. The thesis follows from item (D 1) of Theorem 74, by considering $\sigma = \{x \mapsto t'\}$ such that $t' \in \mathit{HTerms}$ and $\mathrm{vars}(t') = W$.

(D 26). Consider f'_i as specified in Definition 22. By applying case (D 5) of Lemma 79, 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 (D 9) of Lemma 79.

(D 27). Consider l'_i as specified in Definition 22: for each $i \in \{1, 2\}$ we have
$$l'_i = \big(VI \setminus \mathrm{vars}(sh'_i)\big) \cup f'_i \cup l''_i$$
where
$$l''_i \stackrel{\text{def}}{=} \begin{cases} l_i \setminus \big(\mathrm{share\_with}_{d_i}(x) \cap \mathrm{share\_with}_{d_i}(t)\big), & \text{if } \mathrm{lin}_{d_i}(x) \wedge \mathrm{lin}_{d_i}(t); \\ l_i \setminus \mathrm{share\_with}_{d_i}(x), & \text{if } \mathrm{lin}_{d_i}(x); \\ l_i \setminus \mathrm{share\_with}_{d_i}(t), & \text{if } \mathrm{lin}_{d_i}(t); \\ l_i \setminus \big(\mathrm{share\_with}_{d_i}(x) \cup \mathrm{share\_with}_{d_i}(t)\big) & \text{otherwise.} \end{cases}$$
Let $r \in \mathit{HTerms}$ be such that $\mathrm{vars}(r) = VI$. Then, for each $i \in \{1, 2\}$, we have $\mathrm{vars}(sh'_i) = \mathrm{share\_with}_{d'_i}(r)$; also, by equation (D 25), we know that $\rho_{PSD}(sh'_1) =$

$\rho_{PSD}(sh'_2)$; thus, by case (D 9) of Lemma 79, we obtain $\mathrm{vars}(sh'_1) = \mathrm{vars}(sh'_2)$. By equation (D 26), we also know that $f'_1 = f'_2$. By case (D 8) of Lemma 79, 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 (D 9) of Lemma 79, $l''_1 = l''_2$. Therefore equation (D 27) holds. □

Theorem 86
Let $d_1, d_2 \in SFL$ be such that $\rho_{PSD}(d_1) = \rho_{PSD}(d_2)$. Then, for each sequence of bindings $bs \in Bind^*$,

$$\rho_{PSD}\big(\mathrm{aunify}_S(d_1, bs)\big) = \rho_{PSD}\big(\mathrm{aunify}_S(d_2, bs)\big).$$

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 85, we have

$$\rho_{PSD}\big(\mathrm{amgu}_S(d_1, x \mapsto t)\big) = \rho_{PSD}\big(\mathrm{amgu}_S(d_2, x \mapsto t)\big). \tag{D 29}$$

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 (D 29), 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 27 is not affected by ρ_{PSD}.

Theorem 87
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 27 and 29, 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 29, we have $\rho_{PSD}(sh_1) = \rho_{PSD}(sh_2)$, $f_1 = f_2$ and $l_1 = l_2$. We can therefore apply item (D 2) of Theorem 74 to complete the proof. □

Proof of Theorem 30 on page 19. The congruence properties for aunify_S and $\mathrm{aexists}_S$ follow from Theorems 86 and 87, respectively. The congruence property for alub_S holds, as usual, because ρ_{PSD} is an upper closure operator.

Proof of Theorem 31 on page 20. Suppose $\rho_{PSD}(d_1) \neq \rho_{PSD}(d_2)$. By Definition 29, 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 75 (i.e., the proof of Theorem 3 in (Zaffanella et al. 2002)). Then the behavior of $\text{aunify}_{_S}$ 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 (D 3) of Theorem 75.
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 32 on page 20. If $x \notin \text{vars}(t)$, then
$$\text{cyclic}_x^t(sh_- \cup sh^\diamond) = sh_- \cup sh^\diamond.$$
Thus, in this case, the thesis is a corollary of Lemma 83, where $V = \{x\}$ and $W = \text{vars}(t)$. If, on the other hand, $x \in \text{vars}(t)$, then $sh_x = sh_{xt}$, so that $sh^\diamond = \text{bin}(sh_x^\star, sh_t)$. In this case the thesis is a corollary of Theorem 76.