TLP **10** (1): 49–112, 2010. © Cambridge University Press 2009
doi:10.1017/S1471068409990160 First published online 24 September 2009

On the interaction between sharing and linearity

GIANLUCA AMATO and FRANCESCA SCOZZARI
Dipartimento di Science, Università "G. d'Annunzio" di Chieti–Pescara, Pescara, Italy
(*e-mail:* {amato,scozzari}@sci.unich.it)

submitted 26 March 2008; revised 3 July 2008, 15 July 2009; accepted 27 July 2009

Abstract

In the analysis of logic programs, abstract domains for detecting sharing and linearity information are widely used. Devising abstract unification algorithms for such domains has proved to be rather hard. At the moment, the available algorithms are correct but not optimal; i.e., they cannot fully exploit the information conveyed by the abstract domains. In this paper, we define a new (infinite) domain $\mathtt{ShLin}^{\omega}$ which can be thought of as a general framework from which other domains can be easily derived by abstraction. $\mathtt{ShLin}^{\omega}$ makes the interaction between sharing and linearity explicit. We provide a constructive characterization of the optimal abstract unification operator on $\mathtt{ShLin}^{\omega}$, and we lift it to two well-known abstractions of $\mathtt{ShLin}^{\omega}$, namely, to the classical $\mathtt{Sharing} \times \mathtt{Lin}$ abstract domain and to the more precise $\mathtt{ShLin}^2$ abstract domain by Andy King. In the case of single-binding substitutions, we obtain optimal abstract unification algorithms for such domains.

KEYWORDS: static analysis, abstract interpretation, sharing, linearity, unification

1 Introduction

In the analysis of logic programs, the theory of abstract interpretation (Cousot and Cousot 1979, 1992a) has been widely used to design new analyses and to improve existing ones. Given a concrete semantics working over a concrete domain, an abstract interpretation formalizes an analysis by providing an abstract domain and an abstract semantics (working on the abstract domain) and relating them to their concrete counterparts. An abstract domain is a collection of abstract objects which encode the property to analyze. The concrete and abstract domains are related by means of abstraction and concretization maps, which allow each concrete object to be abstracted into an abstract object which describes it. The abstract semantics, in most cases, is given by a set of abstract operators on the abstract domain, which are the counterparts of the concrete ones. For example, in the case of logic programs, one can individuate in the concrete semantics the main operations (unification, projection, union), and an abstract semantics can be specified by giving the abstract unification, abstract projection, and abstract union operations. The theory of abstract interpretation assures us that for any concrete operator, there exists a best abstract operator, called the optimal operator. It computes the most precise result among all possible correct operators, on a given abstract domain. Designing the optimal abstract counterpart of each concrete operator is often a very difficult task. In fact, even if the definition of the optimal operator for any abstract domain is known from the theory of abstract interpretation (as a composition of the concrete operator and the abstraction map), the hard task is to provide an explicit definition of the abstract operators and to devise algorithms on the abstract domain which compute them.

1.1 The context

The property of sharing has been the subject of many papers (Hans and Winkler 1992; Jacobs and Langen 1992; Muthukumar and Hermenegildo 1992; Codish *et al.* 1999; Bagnara *et al.* 2002), from both the theoretical and the practical point of view. Typical applications of sharing analysis are in the fields of optimization of unification (Søndergaard 1986) and parallelization of logic programs (Hermenegildo and Rossi 1995). The goal of (set) sharing analysis is to detect sets of variables which share a common variable in the answer substitutions. For instance, consider the substitution $\{x/f(u,v), y/g(u,u,u), z/v\}$. We say that x and y share the variable u, while x and z share the variable v, and no single variable is shared by x, y, and z. Many domains concerning sharing properties also consider linearity in order to improve the precision of the analysis. We say that a term is linear if it does not contain multiple occurrences of the same variable. For instance, the term $f(x, f(y,z))$ is linear, while $f(x, f(y,x))$ is not, since x occurs twice.

1.2 The problem

It is now widely recognized that the original domain proposed for sharing analysis, namely, `Sharing` (Langen 1990; Jacobs and Langen 1992), is not very precise, so that it is often combined with other domains for handling freeness, linearity, groundness, or structural information (see Bagnara *et al.* 2005 for a comparative evaluation). In particular, adding some kind of linearity information seems to be very profitable, both for the gain in precision and speed which can be obtained and for the fact that it can be easily and elegantly embedded inside the sharing groups (see King 1994). In the literature, many authors have proposed abstract unification operators (e.g., Codish *et al.* 1991; Hans and Winkler 1992; Muthukumar and Hermenegildo 1992; King 1994) for domains of sharing properties, encoding different amounts of linearity information. However, optimal operators for combined analysis of sharing and linearity have never been devised, either for the domain `ShLin`^{2} (King 1994) or for the more broadly adopted `Sharing`$\times$`Lin` (Hans and Winkler 1992; Muthukumar and Hermenegildo 1992).

With the lack of optimal operators, the analysis loses precision and might even be slower. The latter is typical of sharing analysis, where abstract domains are usually defined in such a way that the less information we have, the more complex the abstract objects are. This is not the case for other kinds of analyses, such as groundness analysis, where the complexity of abstract objects may grow according to the amount of groundness information they encode.

The lack of optimal operators is due to the fact that the role played by linearity in the unification process has never been fully clarified. The traditional domains which combine sharing and linearity information are too abstract to capture in a clean way the effect of repeated occurrences of a variable in a term and most of the effects of (non)linearity are obscured by the abstraction process.

1.3 The solution

We propose an abstract domain $\mathtt{ShLin}^{\omega}$ which is able to encode the *amount* of nonlinearity, i.e., which keeps track of the exact number of occurrences of the same variable in a term. Consider again the substitution $\theta = \{x/f(u,v), y/g(u,u,u), z/v\}$. Intuitively, to each variable w in the range of the substitution, we associate the multiset of domain variables which are bound to a term where w occurs, and we call it an ω-sharing group. For instance, we associate, to the variable u, the ω-sharing group $\{x,y,y,y\}$, to denote that u appears once in $\theta(x)$ and three times in $\theta(y)$. To the variable v, we associate the ω-sharing group $\{x,z\}$, to denote that v appears once in $\theta(x)$ and once in $\theta(z)$. Then we consider the collection of all the multisets so obtained $\{\{x,y,y,y\},\{x,z\}\}$, which describes both the sharing property and the exact amount of nonlinearity in the given substitution. The domain we obtain is conceptually simple but cannot be directly used for static analysis, without a widening operator (Cousot and Cousot 1992c), since it contains infinite ascending chains. However, in this domain the role played by (non)linearity is manifest, and we can provide a constructive characterization of the optimal abstract unification operator. The cornerstone of the abstract unification is the concept of *sharing graph* which plays the same role as alternating paths (Søndergaard 1986; King 2000) for pair-sharing analysis. We use sharing graphs to combine different ω-sharing groups during unification. The use of sharing graphs offers a new perspective for looking at variables in the process of unification and simplifies the proofs of correctness and optimality of the abstract operators.

We prove that sharing graphs yield an optimal abstract unification operator for single-binding substitutions. We also provide a purely algebraic characterization of the unification process, which should help in implementing the domain through widening operators and in devising abstract operators for further abstractions of $\mathtt{ShLin}^{\omega}$.

1.4 The applications

We consider two well-known domains for sharing properties, namely, the reduced product (Cousot and Cousot 1979) $\mathtt{Sharing} \times \mathtt{Lin}$ and the more precise domain $\mathtt{ShLin}^2$ by Andy King, and show that they can be immediately obtained as abstractions of $\mathtt{ShLin}^{\omega}$. By exploiting the unification operator on $\mathtt{ShLin}^{\omega}$, we provide the optimal abstract unification operators, in the case of single-binding substitutions, for both domains. We show that we gain in precision w.r.t. any previous attempt to design an abstract unification operator on these domains. This is the first time that abstract unification has been provided optimal for a domain including sharing and linearity information.

Surprisingly, the optimal abstract operators are able to improve not only aliasing and linearity information but also groundness. We show that in certain cases, we improve over Pos (Armstrong *et al.* 1994). This is mainly due to the fact that our operators exploit the occur-check condition. As far as we know, there is no abstract unification operator in the literature, for a domain dealing with sharing, freeness, and linearity, which is more precise than Def for groundness.

Unification for multibinding substitutions is usually computed by considering one binding at a time. For instance, the unification of a substitution θ with $\{x_1/t_1, x_2/t_2, \ldots, x_n/t_n\}$ is performed by first computing the unification of θ with $\{x_1/t_1\}$ and then unifying the result with $\{x_2/t_2, \ldots, x_n/t_n\}$. Actually, computing abstract unification one binding at a time is optimal in $\mathtt{ShLin}^{\omega}$ (Amato and Scozzari 2005). We show that this is not the case for $\mathtt{ShLin}^2$ and $\mathtt{Sharing} \times \mathtt{Lin}$. This means that the classical schema of computing unification iteratively on the number of bindings cannot be used when looking for optimality with multibinding substitutions, at least with these two domains.

1.5 Structure of the paper

In Section 2 we recall some basic notions and the notations about substitutions, multisets, and abstract interpretation. In Section 3 we briefly recall the domain of existential substitutions and its operators, which will be used throughout the paper. In Section 4 we define the domain $\mathtt{ShLin}^{\omega}$, together with the unification operator, and we show the optimality result and give an alternative algebraic characterization of the unification operator. In Section 5 we exploit our results to devise the optimal unification operators for $\mathtt{ShLin}^2$ and $\mathtt{Sharing} \times \mathtt{Lin}$, in the case of single-binding substitutions. Section 6 gives some evidence that there are practical advantages in using the optimal unification operators for $\mathtt{ShLin}^2$ and $\mathtt{Sharing} \times \mathtt{Lin}$. In Section 7 we compare our domains and operators with those known in the literature. We conclude with some open questions for future work. The proofs of the main results of the paper are in Appendix A, and the proofs of the results in Section 5 are in Appendix B.

The paper is a substantial expansion of Amato and Scozzari (2003), which introduces preliminary results of optimality for domains involving sharing and linearity properties.

2 Notation

Given a set A, let $\wp(A)$ be the powerset of A and $\wp_f(A)$ be the set of finite subsets of A. Given two posets $(A, \leqslant_A)$ and $(B, \leqslant_B)$, we denote by $A \rightarrow B$ the poset of monotonic functions from A to B ordered pointwise. We use $\leqslant_{A \rightarrow B}$ to denote the order relation over $A \rightarrow B$. When an order for A or B is not specified, we assume the least informative order ($x \leqslant y \iff x = y$). We also use $A \uplus B$ to denote disjoint union and $|A|$ for the cardinality of the set A.

2.1 Terms and substitutions

In the following, we fix a first-order signature and a denumerable set of variables $\mathscr{V}$. Given a term or other syntactic object o, we denote by $vars(o)$ the set of variables occurring in o and by $occ(v, o)$ the number of occurrences of v in o. When it does not cause ambiguities, we abuse the notation and prefer to use o itself in the place of $vars(o)$. For example, if t is a term and $x \in \mathscr{V}$, then $x \in t$ should be read as $x \in vars(t)$.

We denote by ϵ the empty substitution, by $\{x_1/t_1, \ldots, x_n/t_n\}$ a substitution θ with $\theta(x_i) = t_i \neq x_i$, and by $\mathrm{dom}(\theta) = \{x \in \mathscr{V} \mid \theta(x) \neq x\}$ and $\mathrm{rng}(\theta) = \cup_{x \in \mathrm{dom}(\theta)} vars(\theta(x))$ the domain and range of θ respectively. Let $vars(\theta)$ be the set $\mathrm{dom}(\theta) \cup \mathrm{rng}(\theta)$, and given $U \in \wp_f(\mathscr{V})$, let $\theta|_U$ be the projection of θ over U, i.e., the unique substitution such that $\theta|_U(x) = \theta(x)$ if $x \in U$ and $\theta|_U(x) = x$ otherwise. Given θ_1 and θ_2, two substitutions with disjoint domains, we denote by $\theta_1 \uplus \theta_2$ the substitution θ such that $\mathrm{dom}(\theta) = \mathrm{dom}(\theta_1) \cup \mathrm{dom}(\theta_2)$ and $\theta(x) = \theta_i(x)$ if $x \in \mathrm{dom}(\theta_i)$, for each $i \in \{1, 2\}$. The application of a substitution θ to a term t is written as $t\theta$ or $\theta(t)$. Given two substitutions θ and δ, their composition, denoted by $\theta \circ \delta$, is given by $(\theta \circ \delta)(x) = \theta(\delta(x))$. A substitution θ is idempotent when $\theta \circ \theta = \theta$ or, equivalently, when $\mathrm{dom}(\theta) \cap \mathrm{rng}(\theta) = \emptyset$. A substitution ρ is called renaming if it is a bijection from $\mathscr{V}$ to $\mathscr{V}$. (This is equivalent to saying that there exists a substitution ρ^{-1} such that $\rho \circ \rho^{-1} = \rho^{-1} \circ \rho = \epsilon$.) Instantiation induces a preorder on substitutions: θ is more general than δ, denoted by $\delta \leqslant \theta$, if there exists σ such that $\sigma \circ \theta = \delta$. If $\approx$ is the equivalence relation induced by $\leqslant$, we say that σ and θ are equal up to renaming when $\sigma \approx \theta$. The sets of substitutions, idempotent substitutions, and renamings are denoted by *Subst*, *ISubst*, and *Ren* respectively. Given a set of equations E, we write $\sigma = \mathrm{mgu}(E)$ to denote that σ is a most general unifier of E. Any idempotent substitution σ is a most general unifier of the corresponding set of equations $\mathrm{Eq}(\sigma) = \{x = \sigma(x) \mid x \in \mathrm{dom}(\sigma)\}$. In the following, we will abuse the notation and denote by $\mathrm{mgu}(\sigma_1, \ldots, \sigma_n)$ the substitution $\mathrm{mgu}(\mathrm{Eq}(\sigma_1) \cup \ldots \cup \mathrm{Eq}(\sigma_n))$, when it exists. In spite of a single-binding substitution $\{x/t\}$ we often use just the *binding* x/t. In the rest of the paper we assume that a binding x/t is idempotent, namely, that $x \notin vars(t)$.

A *position* is a sequence of positive natural numbers. We denote with Ξ the set of all positions and with $\mathbb{N}^+$ the set of all positive natural numbers. Given a term t and a position ξ, we define $t(\xi)$ inductively as follows:

$$t(\epsilon) = t \qquad \text{(where } \epsilon \text{ denotes the empty sequence)}$$

$$t(i \cdot \xi') = \begin{cases} t_i(\xi') & \text{if } t \equiv f(t_1, \ldots, t_n) \text{ and } i \leqslant n; \\ \text{undefined} & \text{otherwise.} \end{cases}$$

For any variable x, an *occurrence* of x in t is a position ξ such that $t(\xi) = x$.

In the rest of the paper, we use U, V, W to denote finite sets of variables; h, k, u, v, w, x, y, z for variables; t for terms; f, r, s for term symbols; a, b for constants; $\eta, \theta, \sigma, \delta$ for substitutions; ρ for renamings.

2.2 Multisets

A *multiset* is a set in which repetitions are allowed. We denote by $\{\!\{x_1, \dots, x_m\}\!\}$ a multiset, where $x_1, \dots, x_m$ is a sequence with (possible) repetitions. We denote by $\{\!\{\}\!\}$ the empty multiset. We will often use the polynomial notation $v_1^{i_1} \dots v_n^{i_n}$, where $v_1, \dots, v_n$ is a sequence without repetitions, to denote a multiset A whose element v_j appears i_j times. The set $\{v_j \mid i_j > 0\}$ is called the *support* of A and is denoted by $\lfloor\!\lfloor A \rfloor\!\rfloor$. We also use the functional notation $A : \{v_1, \dots, v_n\} \to \mathbb{N}$, where $A(v_j) = i_j$.

In this paper, we only consider multisets whose support is *finite*. We denote with $\wp_m(X)$ the set of all the multisets whose support is *any finite subset* of X. For example, both a^2c^4 and $a^1b^2c^3$ are elements of $\wp_m(\{a, b, c\})$. The cardinality of a multiset is $|A| = \sum_{v \in \lfloor\!\lfloor A \rfloor\!\rfloor} A(v)$.

The new fundamental operation for multisets is the *sum*, defined as

$$A \uplus B = \lambda v \in \lfloor\!\lfloor A \rfloor\!\rfloor \cup \lfloor\!\lfloor B \rfloor\!\rfloor . A(v) + B(v).$$

Multiset sum is associative and commutative and $\{\!\{\}\!\}$ is the neutral element. Note that we also use $\uplus$ to denote disjoint union for standard sets. The context will allow us to identify the proper semantics of $\uplus$.

Given a multiset A and $X \subseteq \lfloor\!\lfloor A \rfloor\!\rfloor$, the *restriction* of A over X, denoted by $A|_X$, is the only multiset B such that $\lfloor\!\lfloor B \rfloor\!\rfloor = X$ and $B(v) = A(v)$ for each $v \in X$. Finally, if $A \in \wp_m(X)$, $E[x]$ is an integer expression and $x \in X$, we define

$$\sum_{x \in A} E[x] = \sum_{x \in \lfloor\!\lfloor A \rfloor\!\rfloor} A(x) \cdot E[x].$$

For example, given a multiset $A = \{\!\{5, 5, 6, 8, 8, 8\}\!\}$, $\sum_{x \in A} x^2 = 2 \times 5^2 + 6^2 + 3 \times 8^2 = 278$.

2.3 Abstract interpretation

Given two sets C and A of concrete and abstract objects respectively, an *abstract interpretation* (Cousot and Cousot 1992b) is given by an approximation relation $\rhd \subseteq A \times C$. When $a \rhd c$ holds, this means that a is a correct abstraction of c. In particular, we are interested in the case in which $(A, \leqslant_A)$ is a poset and $a \leqslant_A a'$ means that a is more precise than a'. In this case we require that if $a \rhd c$ and $a \leqslant_A a'$, then $a' \rhd c$, too. In more detail, we require what Cousot and Cousot (1992b) call the *existence of a best abstract approximation assumption*, i.e., the existence of a map $\alpha : C \to A$ such that for all $a \in A, c \in C$, it holds that $a \rhd c \iff \alpha(c) \leqslant_A a$. The map α is called the *abstraction function* and maps each c to its best approximation in A.

Given a (possibly partial) function $f : C \to C$, we say that $\tilde{f} : A \to A$ is a correct abstraction of f and write $\tilde{f} \rhd f$, whenever

$$a \rhd c \Rightarrow \tilde{f}(a) \rhd f(c),$$

assuming that $\tilde{f}(a) \rhd f(c)$ is true whenever $f(c)$ is not defined. We say that $\tilde{f} : A \to A$ is the *optimal* abstraction of f when it is the best correct approximation of f, i.e., when $\tilde{f} \rhd f$ and

$$\forall f' : A \to A.\ f' \rhd f \Rightarrow \tilde{f} \leqslant_{A \to A} f'.$$

In some cases, we prefer to deal with a stronger framework, in which the domain C is also endowed with a partial order $\leqslant_C$ and $\alpha : C \to A$ is a left adjoint to $\gamma : A \to C$, i.e.,

$$\forall c \in C. \forall a \in A. \alpha(c) \leqslant_A a \iff c \leqslant_C \gamma(a).$$

The pair $\langle \alpha, \gamma \rangle$ is called a *Galois connection*. In particular, we will only consider the case of *Galois insertions*, which are Galois connections such that $\alpha \circ \gamma$ is the identity map. If $\langle \alpha, \gamma \rangle$ is a Galois insertion and $f : C \to C$ is a monotone map, the optimal abstraction $\tilde{f}$ always exists, and it is definable as $\tilde{f} = \alpha \circ f \circ \gamma$.

3 The domain of existential substitutions

The choice of the concrete domain depends on the observable properties we want to analyze. Most of the semantics suited for the analysis of logic programs are based on computed answer substitutions, and most of the domains are expressed as abstractions of sets of substitutions. In general, we are not really interested in the substitutions, but in their quotient set w.r.t. an appropriate equivalence relation. Let us consider a one-clause program $\mathrm{p(x, x)}$, the goal $p(x, y)$, and the following answer substitutions: $\theta_1 = \{y/x\}$, $\theta_2 = \{x/y\}$, $\theta_3 = \{x/u, y/u\}$, and $\theta_4 = \{x/v, y/v\}$. Although θ_1 and θ_2 are equal up to renaming, the same does not hold for θ_3 and θ_4. Nonetheless, they essentially represent the same answer, since u and v are just two different variables we chose when renaming apart the clause $\mathrm{p(x, x)}$ from the goal $p(x, y)$, and therefore are not relevant to the user. On the other hand, if θ_3 and θ_4 are answer substitutions for the goal $q(x, y, u)$, then they correspond to computed answers $q(u, u, u)$ and $q(v, v, u)$ and therefore are fundamentally different. As a consequence, the equivalence relation we need to consider must be coarser then renaming and must take into account the set of variables of interest, i.e., the set of variables which appear in the goal. For these reasons, we think that the best solution is to use a domain of equivalence classes of substitutions. Among the various domains proposed in the literature (e.g., Jacobs and Langen 1992; Marriott *et al.* 1994; Levi and Spoto 2003), we adopt the domain of existential substitutions (Amato and Scozzari 2009), since it is explicitly defined as a quotient of a set of substitutions, w.r.t. a suitable equivalence relation. Moreover, the domain is equipped with all the necessary operators for defining a denotational semantics, namely, projection, renaming, and unification. We briefly recall the basic definitions of the domain and the unification operator.

Given $\theta_1, \theta_2 \in \mathit{Subst}$ and $U \in \wp_f(\mathcal{V})$, the preorder $\preceq_U$ is defined as follows:

$$\theta_1 \preceq_U \theta_2 \iff \exists \delta \in \mathit{Subst}. \forall x \in U.\ \theta_1(x) = \delta(\theta_2(x)).$$

The notation $\theta_1 \preceq_U \theta_2$ states that θ_1 is an instance of θ_2 w.r.t. the variables in U. The equivalence relation induced by the preorder $\preceq_U$ is given by

$$\theta_1 \sim_U \theta_2 \iff \exists \rho \in \mathit{Ren}. \forall x \in U.\ \theta_1(x) = \rho(\theta_2(x)).$$

This relation precisely captures the extended notion of renaming which is needed to work with computed answer substitutions.

Example 3.1

It is easy to check that $\{x/w, y/u\} \sim_{\{x,y\}} \epsilon$ by choosing the renaming $\rho = \{x/w, w/x, y/u, u/y\}$. Note that $\sim_U$ is coarser than the standard equivalence relation $\approx$: there is no renaming ρ such that $\epsilon = \rho \circ \{x/w, y/u\}$. As it happens for $\preceq$, if we enlarge the set of variables of interest, not all equivalences between substitutions are preserved: for instance, $\{x/w, y/u\} \not\sim_{\{w,x,y\}} \epsilon$. $\square$

Let $\mathit{ISubst}_{\sim_U}$ be the quotient set of ISubst w.r.t. $\sim_U$. The domain $\mathit{ISubst}_{\sim}$ of *existential substitutions* is defined as the disjoint union of all the $\mathit{ISubst}_{\sim_U}$ for $U \in \wp_f(\mathscr{V})$, namely,

$$\mathit{ISubst}_{\sim} = \biguplus_{U \in \wp_f(\mathscr{V})} \mathit{ISubst}_{\sim_U}.$$

In the following we write $[\theta]_U$ for the equivalence class of θ w.r.t. $\sim_U$. The partial order $\preceq$ over $\mathit{ISubst}_{\sim}$ is given by

$$[\theta]_U \preceq [\theta']_V \iff U \supseteq V \land \theta \preceq_V \theta'.$$

Intuitively, $[\theta]_U \preceq [\theta']_V$ means that θ is an instance of θ' w.r.t. the variables in V, provided that they are all variables of interest of θ.

To ease notation, we often omit braces from the sets of variables of interest when they are given extensionally. So we write $[\theta]_{x,y}$ instead of $[\theta]_{\{x,y\}}$ and $\sim_{x,y,z}$ instead of $\sim_{\{x,y,z\}}$. When the set of variables of interest is clear from the context or when it is not relevant, it will be omitted. Finally, we omit the braces which enclose the bindings of a substitution when the latter occurs inside an equivalence class; i.e., we write $[x/y]_U$ instead of $[\{x/y\}]_U$.

3.1 Unification

Given $U, V \in \wp_f(\mathscr{V})$, $[\theta_1]_U, [\theta_2]_V \in \mathit{ISubst}_{\sim}$, the most general unifier between these two classes is defined as the mgu of suitably chosen representatives, where variables not of interest are renamed apart. In formulas

$$\mathrm{mgu}([\theta_1]_U, [\theta_2]_V) = [\mathrm{mgu}(\theta'_1, \theta'_2)]_{U \cup V}, \tag{1}$$

where $\theta_1 \sim_U \theta'_1 \in \mathit{ISubst}$, $\theta_2 \sim_V \theta'_2 \in \mathit{ISubst}$, and $(U \cup \mathit{vars}(\theta'_1)) \cap (V \cup \mathit{vars}(\theta'_2)) \subseteq U \cap V$. The last condition is needed to avoid variable clashes between the chosen representatives θ'_1 and θ'_2. Moreover, mgu is the greatest lower bound of $\mathit{ISubst}_{\sim}$ ordered by $\preceq$.

Example 3.2
Let $\theta_1 = \{x/a, y/r(v_1, v_1, v_2)\}$ and $\theta_2 = \{y/r(a, v_2, v_1), z/b\}$. Then

$$\text{mgu}([\theta_1]_{x,y}, [\theta_2]_{y,z}) = [x/a, y/r(a, a, v), z/b]_{x,y,z},$$

by choosing $\theta_1' = \theta_1$ and $\theta_2' = \{y/r(a, w, v), z/b\}$. In this case we have

$$\{x/a, y/r(a, a, v), z/b\} \sim_{x,y,z} \text{mgu}(\theta_1', \theta_2') = \{x/a, y/r(a, a, v), z/b, v_1/a, w/a, v_2/v\}. \quad \square$$

A different version of unification is obtained when one of the two arguments is an existential substitution and the other one is a standard substitution. In this case, the latter argument may be viewed as an existential substitution where all the variables are of interest:

$$\text{mgu}([\theta]_U, \delta) = \text{mgu}([\theta]_U, [\delta]_{vars(\delta)}). \tag{2}$$

Note that deriving the general unification in (1) from the special case in (2) is not possible. This is because there are elements in $\mathit{ISubst}_\sim$ which cannot be obtained as $[\delta]_{vars(\delta)}$ for any $\delta \in \mathit{ISubst}$ (see Example 4.10).

This is the form of unification which is better suited for analysis of logic programs, where existential substitutions are the denotations of programs while standard substitutions are the result of unification between goals and heads of clauses. Therefore, the rest of the paper will be concerned with the problem of devising optimal abstract operators corresponding to (2), for three different abstract domains. Of course, unification is not the only operator needed to give semantics to logic programs: we also need projection, renaming, and union. However, providing optimal abstract counterparts for these operators is generally a trivial task and will not be considered here.

We want to conclude the section with a small remark about our choice of the concrete domain. By adopting existential substitutions and the corresponding notion of unification, we greatly simplify all the semantic definitions which are heavily based on renaming variables apart. This is because all the details concerning renamings are shifted toward the inner level of the semantic domain, where they are more easily managed (Jacobs and Langen 1992; Amato and Scozzari 2009).

4 The abstract domain $\mathtt{ShLin}^\omega$

The domain $\mathtt{Sharing} \times \mathtt{Lin}$ is one of the best known domains in the literature that combine sharing and linearity information. The domain $\mathtt{Sharing}$ records the information of variable aliasing, by abstracting the substitution $\theta = \{x/f(u, v), y/g(u, u, u), z/v\}$ into the set $\{uxy, vxz\}$. The object uxy, called a *sharing group*, states that $\theta(u), \theta(x)$, and $\theta(y)$ do share some variable (the variable u in this case). Analogously, the sharing group vxz states that $\theta(v), \theta(x)$, and $\theta(z)$ do share (in this case the variable v). One of the simplest way of adding linearity information is to record, in a separate object, the set of variables w such that $\theta(w)$ is a linear term. In our example, only $\theta(y)$ is not linear. Thus the substitution is abstracted into the pair $(\{uxy, vxz\}, \{u, v, x, z\})$. Another known domain in the literature is ASub whose main difference w.r.t. Sharing × Lin is that it only records sharing information between pairs of variables. Thus, in ASub, each sharing group has at most two elements. Developing optimal unification operators for such abstract domains is a difficult task. In our opinion, this is because the gap between the substitutions and Sharing × Lin (or ASub) is too wide and the combined effect of aliasing and linearity is difficult to grasp.

We solve this problem by defining a new abstract domain $\mathtt{ShLin}^{\omega}$ which can be used to approximate $ISubst_{\sim}$. Since $\mathtt{ShLin}^{\omega}$ has infinite ascending chains, in most cases it cannot be directly used for the analysis. It should be thought of as a general framework from which other domains can be easily derived by abstraction. In this sense, $\mathtt{ShLin}^{\omega}$ closes the gap between the concrete domain of substitutions and the abstractions like Sharing × Lin or ASub. The structure of $\mathtt{ShLin}^{\omega}$ has made it possible to develop clean and optimal abstract unification operators. From these, optimal operators for the simpler domains are easy to obtain, at least for single-binding substitutions.

The idea underlying $\mathtt{ShLin}^{\omega}$ is to count the exact number of occurrences of the same variable in a term. It extends the standard domain Sharing by recording, for each $v \in \mathcal{V}$ and $\theta \in ISubst$, not only the set $\{w \in \mathcal{V} \mid v \in \theta(w)\}$ but also the multiset $\lambda w \in \mathcal{V}. occ(v, \theta(w))$.

Definition 4.1 (*ω-sharing group*)
An *ω-sharing group* is a multiset of variables, i.e., an element of $\wp_m(\mathcal{V})$.

Example 4.2
Given variables $u, v, w, x, y \in \mathcal{V}$, examples of ω-sharing groups are $u^2v^3x^{19}$, xyz, and $u^{23}vwx^2y^3$. □

Definition 4.3
Given a substitution θ and a variable $v \in \mathcal{V}$, we define

$$\theta^{-1}(v) = \lambda w.\ occ(v, \theta(w)).$$

Intuitively, $\theta^{-1}(v)$ is an ω-sharing group which maps each variable w to the number of occurrences of v in $\theta(w)$.

Example 4.4
Given $\theta = \{x/f(u,u,u), y/g(u,v), z/f(u,v,v)\}$, we have that $\theta^{-1}(u) = ux^3yz$, $\theta^{-1}(v) = vyz^2$, $\theta^{-1}(w) = w$, and $\theta^{-1}(x) = \{\!\!\{\}\!\!\}$. □

Definition 4.5 (*Correct approximation*)
Given a set of variables U and a set of ω-sharing groups S (i.e., $S \subseteq \wp_m(U)$), we say that the pair (S, U) correctly approximates a substitution $[\theta]_W$ if $U = W$ and for each $v \in \mathcal{V}$, $\theta^{-1}(v)|_W \in S$. In the following we denote by $[S]_U$ the pair (S, U) and write $[S]_U \rhd [\theta]_W$ to mean that $[S]_U$ correctly approximates $[\theta]_W$.

Therefore, $[S]_U$ correctly approximates $[\theta]_U$ when S contains at least all the ω-sharing groups which may arise in θ, restricted to the variables U. Note that $[\theta]_U$ is an equivalence class of substitutions, as defined in Section 3, while $[S]_U$ is just a symbol to denote the pair of objects (S, U). We prefer this notation for the sake of uniformity with substitutions.

Theorem 4.6
The relation $\rhd$ is well defined.

We can now define the domain $\mathtt{ShLin}^{\omega}$ of ω-sharing groups.

Definition 4.7 ($\mathtt{ShLin}^{\omega}$)
The domain $\mathtt{ShLin}^{\omega}$ is defined as

$$\mathtt{ShLin}^{\omega} = \{[S]_U \mid U \in \wp_f(\mathcal{V}), S \subseteq \wp_m(U), S \neq \emptyset \Rightarrow \{\!\!\{\}\!\!\} \in S\}$$

and ordered by $[S_1]_{U_1} \leqslant_{\omega} [S_2]_{U_2}$ iff $U_1 = U_2$ and $S_1 \subseteq S_2$.

The order relation corresponds to the approximation ordering, since bigger (w.r.t $\leqslant_{\omega}$) elements correctly approximate a larger number of substitutions than smaller elements. The existence of the empty multiset, when S is not empty, is required in order to obtain a Galois insertion, instead of a Galois connection. In order to simplify the notation, in the following we write an object $[\{\{\!\!\{\}\!\!\}, B_1, \ldots, B_n\}]_U \in \mathtt{ShLin}^{\omega}$ as $[B_1, \ldots, B_n]_U$ by omitting the braces and the empty multiset. Moreover, if $X \in \mathtt{ShLin}^{\omega}$, we write $B \in X$ in place of $X = [S]_U \wedge B \in S$.

Definition 4.8 (*Abstraction for* $\mathtt{ShLin}^{\omega}$)
We define the abstraction for a substitution $[\theta]_U$ as

$$\alpha_{\omega}([\theta]_U) = [\{\theta^{-1}(v)|_U \mid v \in \mathcal{V}\}]_U.$$

This is the least element of $\mathtt{ShLin}^{\omega}$ which correctly approximates $[\theta]_U$. Note that by the proof of Theorem 4.6 it immediately follows that α_{ω} is well defined; i.e., it does not depend from the choice of the representative for $[\theta]_U$.

Example 4.9
Given $\theta = \{x/r(y,u,u), z/y, v/u\}$ and $U = \{w, x, y, z\}$, we have $\theta^{-1}(u) = x^2vu$, $\theta^{-1}(y) = xyz$, $\theta^{-1}(z) = \theta^{-1}(v) = \theta^{-1}(x) = \{\!\!\{\}\!\!\}$, and $\theta^{-1}(s) = s$ for all the other variables (included w). Projecting over U we obtain $\alpha_{\omega}([\theta]_U) = [x^2, xyz, w]_U$. □

Example 4.10
As we have said in Section 3, we show an element of $\mathit{ISubst}_{\sim}$, namely, the existential substitution $[x/r(v,v)]_x$, which cannot be obtained as $[\delta]_{vars(\delta)}$ for any substitution δ. In fact, consider any ω-sharing group $B = \delta^{-1}(u)|_{vars(\delta)} \in \alpha_{\omega}([\delta]_{vars(\delta)})$. Then either $u \notin \mathrm{rng}(\delta)$ and $B = \{\!\!\{\}\!\!\}$ or $u \in \mathrm{rng}(\delta)$ and $B(u) = 1$. However, $\alpha([x/r(v,v)]_x) = [x^2]_x$ and x^2 does not contain any variable with multiplicity one. □

4.1 Multigraphs

In order to define an abstract unification operator over $\mathtt{ShLin}^{\omega}$, we need to introduce the concept of multigraph. We call (directed) *multigraph* a graph in which multiple distinguished edges are allowed between nodes. We use the definition of multigraph which is customary in category theory (Mac Lane 1988).

Definition 4.11 (*Multigraph*)
A *multigraph* G is a tuple $\langle N_G, E_G, \mathrm{src}_G, \mathrm{tgt}_G\rangle$, where $N_G \neq \emptyset$ and E_G are the sets of *nodes* and *edges* respectively; $\mathrm{src}_G : E_G \to N_G$ is the *source* function which maps each edge to its start node; and $\mathrm{tgt}_G : E_G \to N_G$ is the *target* function which maps each edge to its end node.

A *labeled* multigraph G is a multigraph equipped with a *labeling* function $l_G : N_G \to L_G$ which maps each node to its *label* in the given set L_G.

We write $e : n_1 \to n_2 \in G$ to denote the edge $e \in E_G$ such that $\mathrm{src}_G(e) = n_1$ and $\mathrm{tgt}_G(e) = n_2$. We also write $n_1 \to n_2 \in G$ to denote any edge $e \in E_G$ such that $\mathrm{src}_G(e) = n_1$ and $\mathrm{tgt}_G(e) = n_2$. Moreover, with $|n_1 \to n_2 \in G|$ we denote the cardinality of the set $\{e \in E_G \mid \mathrm{src}_G(e) = n_1 \wedge \mathrm{tgt}_G(e) = n_2\}$. In the notation above, we omit "$\in G$" whenever the multigraph G is clear from the context.

We call *in-degree* (respectively *out-degree*) of a node n the cardinality of the set $\{e \in E_G \mid \mathrm{tgt}(e) = n\}$ (respectively $\{e \in E_G \mid \mathrm{src}(e) = n\}$).

Given a multigraph G, a *path* π is a *nonempty* sequence of nodes $n_1, \ldots, n_k$ such that for each $i \in \{1, \ldots, k-1\}$, there is either an edge $n_i \to n_{i+1} \in G$ or an edge $n_{i+1} \to n_i \in G$. Nodes n_1 and n_k are the *endpoints* of π, and we say that π *connects* n_1 and n_k. A multigraph is *connected* when all pairs of nodes are connected by at least one path.

4.2 Abstract unification

We need to find the abstract counterpart of mgu over $\mathtt{ShLin}^{\omega}$, i.e., an operation mgu_{ω} such that if $[S]_U \rhd [\theta]_U$, then

$$\mathrm{mgu}_{\omega}([S]_U, \delta) \rhd \mathrm{mgu}([\theta]_U, \delta) \tag{3}$$

for each $\delta \in \mathit{ISubst}$. Note that we are looking for an abstract counterpart to the mixed unification in (2), where one of the two arguments is a plain substitution. In particular, we would like to find an operator which is the minimum element that satisfies the condition in (3), i.e., the *optimal* abstract counterpart of mgu. Observe that for a fixed U, the set of all the elements $[S]_U \in \mathtt{ShLin}^{\omega}$ is a complete lattice w.r.t. $\leqslant_{\omega}$ with the top element given by $[\wp_m(U)]_U$ and the meet operator given by

$$\textstyle\bigwedge_{\omega}\{[S_i]_U \mid i \in I\} = \left[\bigcap_{i\in I} S_i\right]_U,$$

for any family $\{[S_i]_U \mid i \in I\}$ of elements of $\mathtt{ShLin}^{\omega}$. Moreover, the relation $\rhd$ is meet preserving on the left, since if $[S_i]_U \rhd [\theta]_U$ for each $i \in I$, then $\bigwedge_{\omega}\{[S_i]_U \mid i \in I\} \rhd [\theta]_U$. Therefore, we may define the abstract mgu as follows:

$$\mathrm{mgu}_{\omega}([S]_U, \delta) = \bigwedge\nolimits_{\omega}\{[S']_{U'} \mid \forall\theta.[S]_U \rhd [\theta]_U \Rightarrow [S']_{U'} \rhd \mathrm{mgu}([\theta]_U, \delta)\},$$

where the definitions of $\rhd$ and mgu force U' to be $U \cup \mathit{vars}(\delta)$. Note that this is just a translation of the general definition of optimal operator in Cousot and Cousot (1992b), and it satisfies (3).

This definition is completely nonconstructive. The rest of this section is devoted to providing a constructive characterization for $\mathrm{mgu}_{\omega}([S]_U, \delta)$. We begin to characterize the operation of abstract unification by means of graph theoretic notions.

Definition 4.12 (*Multiplicity of ω-sharing groups*)
The *multiplicity* of an ω-sharing group B in a term t is defined as

$$\chi(B,t) = \sum_{v \in B} occ(v,t) = \sum_{v \in \llbracket B \rrbracket} B(v) \cdot occ(v,t).$$

For instance, $\chi(x^3yz^4, r(x,y,f(x,y,z))) = 3 \times 2 + 1 \times 2 + 4 \times 1 = 12$. The meaning of the map χ is made clear by the following proposition.

Proposition 4.13
Given a substitution θ, a variable v, and a term t, we have that $\chi(\theta^{-1}(v),t) = occ(v,\theta(t))$. Moreover, given a set of variables U, when $vars(t) \subseteq U$, it holds that $\chi(\theta^{-1}(v)|_U,t) = occ(v,\theta(t))$.

Example 4.14
Let $B = xy^2z^3$ and $\theta = \{y/r(x,x), z/r(x,x,x)\}$, so that $\theta^{-1}(x) = \{xy^2z^3\}$. Given $t \equiv s(x,z)$ we have

$$occ(x,\theta(t)) = occ(x, s(x, r(x,x,x))) = 4,$$

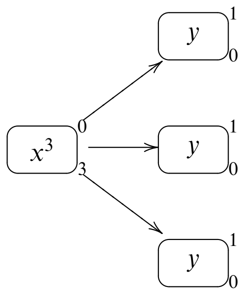
and

$$\chi(B,t) = B(x) \cdot occ(x,t) + B(z) \cdot occ(z,t) = 1 \times 1 + 3 \times 1 = 4. \quad \square$$

If $[S]_U \rhd [\theta]_U$ and we unify $[\theta]_U$ with δ, some of the ω-sharing groups in S may be glued together to obtain a bigger resultant group. It happens that the gluing of the sharing groups during the unification of $[\theta]_U$ with a single-binding substitution $\{x/t\}$ may be represented by special labeled multigraphs which we call *sharing graphs*.

Example 4.15
Let $S = \{x^3, y\}$ and $U = \{x,y\}$. We look for a representation of the unification process between any substitution θ approximated by S and the binding $x/r(y)$. We show that multigraphs can be easily used for this purpose. For instance, the substitution $\theta = \{x/r(g(u,u,u))\}$ is approximated by S. By unifying θ with $\{x/r(y)\}$ we obtain $\delta = \{x/r(g(u,u,u)), y/g(u,u,u)\}$. Note that any approximation of δ on the variables $\{x,y\}$ must include the sharing group x^3y^3 generated by the variable u. Thus, any correct approximation of the unification must also contain x^3y^3.

We want to associate to any ω-sharing group B in δ a special multigraph which represents the way the ω-sharing groups in S have been merged in order to obtain B. The nodes of this multigraph are the ω-sharing groups in S (possibly repeated any number of times). The following is a sharing graph for $x/r(y)$ and S:



where pedices and apices on a sharing group B are respectively the values of $\chi(B,x)$ and $\chi(B,r(y))$. For instance, since $\chi(x^3,x) = 3$, we put the pedice 3 on the node x^3 to mean that x is bound to a term containing three occurrences of the same variable. Symmetrically, since $\chi(x^3,r(y)) = 0$, we put the apice 0 on the node x^3. The in-degree and the out-degree of the nodes reflect the values of apices and pedices. In this case, we have three outgoing edges from x^3 and no ingoing edges. Moreover, the multigraph must be connected, in order to guarantee that we can use a single variable to form the sharing group x^3y^3.

By summing the labels of all the nodes, namely, $x^3 \uplus y \uplus y \uplus y$, we obtain the ω-sharing group x^3y^3 which must appear in any correct approximation of the unification. □

Given any labeled multigraph G, in the rest of the paper we assume that the codomain of the labeling function l_G is $\wp_m(\mathcal{V})$, the set of ω-sharing groups.

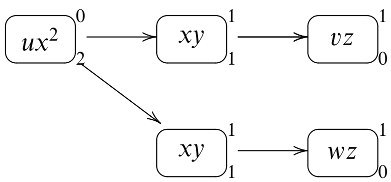
Definition 4.16 (*Sharing graph*)
A *sharing graph* for the binding x/t and a set of ω-sharing groups S is a labeled multigraph G such that

(1) G is connected;
(2) for each node $n \in N_G$, $l_G(n) \in S$;
(3) for each node $n \in N_G$, the out-degree of n is equal to $\chi(l_G(n),x)$ and the in-degree of n is equal to $\chi(l_G(n),t)$.

The *resultant ω-sharing group* of G is

$$res(G) = \biguplus_{n \in N_G} l_G(n).$$

Example 4.17
Let $S = \{ux^2, xy, vz, wz, xyz\}$. The following is a sharing graph for $x/r(y,z)$ and S:



where pedices and apices on a sharing group B are respectively the value of $\chi(B,x)$ and $\chi(B,r(y,z))$. Therefore the resultant sharing group is $uvwx^4y^2z^2$. □

It is worth noting that given any set of ω-sharing groups S and binding x/t, there exist many different sharing graphs for x/t and S. Each sharing graph yields a resultant sharing group which must be included in the result of the abstract unification operator. Of course, different sharing graphs may give the same resultant sharing group. The abstract unification operator is defined by collecting all the resultant sharing groups.

Definition 4.18 (*Single-binding unification*)
Let $U \in \wp_f(\mathcal{V})$, S be a set of ω-sharing groups with $[S]_U \in \mathtt{ShLin}^{\omega}$, x/t be a binding, and $vars(x/t) \subseteq U$. The set of resultant ω-sharing groups for x/t and S is

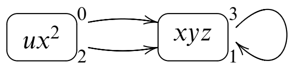
$$\mathrm{mgu}_{\omega}(S, x/t) = \{res(G) \mid G \text{ is a sharing graph for } S \text{ and } x/t\}.$$

We lift mgu_{ω} to an operation over $\mathtt{ShLin}^{\omega}$:

$$\mathrm{mgu}_{\omega}([S]_U, x/t) = [\mathrm{mgu}(S, x/t)]_U.$$

This is a particular case of the abstract unification operator, for single-binding substitutions and $vars(x/t) \subseteq U$.

Example 4.19
Let S be as in Example 4.17. The following is a sharing graph for $x/r(y, y, z)$ and S:



where pedices and apices on a sharing group B are respectively the value of $\chi(B, x)$ and $\chi(B, r(y, y, z))$. Therefore $ux^3yz \in \mathrm{mgu}_{\omega}(S, x/r(y, y, z))$. Note that this sharing group can actually be generated by the substitution $\theta = \{x/r(v_1, v_1, v_2), y/v_2, z/v_2, u/v_1, v/a, w/a\}$, where a is a ground term. Let $U = \{u, v, w, x, y, z\}$. It is the case that $[S]_U \rhd [\theta]_U$ and $\mathrm{mgu}([\theta]_U, \{x/r(y, y, z)\})$ performs exactly the variable aliasings depicted by the sharing graph. Actually $\mathrm{mgu}([\theta]_U, \{x/r(y, y, z)\}) = [x/r(v_1, v_1, v_1), y/v_1, u/v_1, v/a, w/a]_U = [\eta]_U$, and $\eta^{-1}(v_1)|_U = ux^3yz$. $\square$

We give here an intuition of the way sharing graphs work. Assume given a set of ω-sharing groups $[S]_U$ and a binding x/t with $vars(x/t) \subseteq U$. We want to compute $[\mathrm{mgu}_{\omega}(S, x/t)]_U$. To this aim, for any substitution θ approximated by $[S]_U$, that is, $[S]_U \rhd [\theta]_U$, we compute $\alpha_{\omega}(\mathrm{mgu}([\theta]_U, \{x/t\}))$.

For any $B_1, B_2 \in S$, assume that there exist $v_1, v_2 \in \mathcal{V}$ such that $B_1 = \theta^{-1}(v_1)|_U$ and $B_2 = \theta^{-1}(v_2)|_U$. When unifying θ with the binding x/t, we use the fact that $\mathrm{mgu}(\mathrm{Eq}(\theta) \cup \{x = t\}) = \mathrm{mgu}(\{\theta(x) = \theta(t)\}) \circ \theta$. By Proposition 4.13, $\theta(x)$ contains $\chi(B_1, x)$ instances of v_1 and $\chi(B_2, x)$ instances of v_2. Symmetrically, $\theta(t)$ contains $\chi(B_1, t)$ instances of v_1 and $\chi(B_2, t)$ instances of v_2.

Assume that $\theta(x)$ and $\theta(t)$ only differ for the variables occurring in them (and not for the structure of terms). Then, an arrow from the sharing group B_1 to B_2 represents the fact that in $\mathrm{mgu}(\{\theta(x) = \theta(t)\})$, one of the copies of v_1 is aliased to one of the copies of v_2, i.e., that there are corresponding positions in $\theta(x)$ and $\theta(t)$ where the two terms contain the variables v_1 and v_2 respectively. The third condition for sharing graphs implies that each occurrence of v_1 and v_2 is aliased to some other variable. The first condition (the sharing graph must be connected) ensures that all the variables corresponding to the ω-sharing groups involved in the sharing graph are aliased to each other. In other words, given any two such variables, they are aliased. Although here we are only considering the case in which $\theta(x)$ and $\theta(t)$ differ for the variables occurring in them, we will show that it is enough to reach correctness and optimality. The next example applies this intuition to a concrete case.

Example 4.20
Consider Example 4.19, where $\theta = \{x/r(v_1, v_1, v_2), y/v_2, z/v_2, u/v_1, v/a, w/a\}$ and $U = \{u, v, w, x, y, z\}$. Let $B_1 = ux^2$ and $B_2 = xyz$; thus $B_1 = \theta^{-1}(v_1)|_U$ and $B_2 = \theta^{-1}(v_2)|_U$. When unifying θ with the binding $x/r(y, y, z)$ we have that $\theta(x) = r(v_1, v_1, v_2)$ and $\theta(r(y, y, z)) = r(v_2, v_2, v_2)$.

Note that $\theta(x)$ contains $\chi(ux^2, x) = 2$ instances of v_1 and $\chi(xyz, x) = 1$ instance of v_2. Symmetrically, $\theta(r(y, y, z))$ contains $\chi(ux^2, r(y, y, z)) = 0$ instances of v_1 and $\chi(xyz, r(y, y, z)) = 3$ instances of v_2. Moreover, $\theta(x)$ and $\theta(r(y, y, z))$ only differ for the variables occurring in them. Thus, the three edges in the sharing graph of Example 4.19 correspond to the following aliasings:

$$\begin{array}{ll} \theta(r(y, y, z)) = & r(v_2, v_2, v_2) \\ & \;\;\;\uparrow\;\;\; \uparrow\;\;\; \uparrow \\ \theta(x) \qquad = & r(v_1, v_1, v_2) \end{array}$$

In particular, the last arrow from v_2 to itself corresponds to the self-loop in the sharing graph. □

The unification operator $\mathrm{mgu}_\omega([S]_U, x/t)$ can be extended to the case $vars(x/t) \not\subseteq U$. The idea is to enlarge S by including all the singletons in $vars(x/t) \setminus U$.

Definition 4.21 (*Single-binding unification with extension*)
Let $U \in \wp_f(\mathcal{V})$, S be a set of ω-sharing groups with $[S]_U \in \mathtt{ShLin}^\omega$ and x/t be a binding:

$$\mathrm{mgu}_\omega([S]_U, x/t) = \mathrm{mgu}_\omega([S \cup \{\{\!\{v\}\!\} \mid v \in vars(x/t) \setminus U\}]_{U \cup vars(x/t)}, x/t).$$

Note that for a generic abstract domain, the method of extending the abstract object to include all the variables in the concrete substitution δ may result in a nonoptimal abstract unification. For example, this is what happens in the case of the domain `Sharing`, as shown in Amato and Scozzari (to appear). However, we will prove that in the case of $\mathtt{ShLin}^\omega$, the abstract mgu in Definition 4.21 is optimal.

This operator can be extended to multibinding substitutions in the obvious way, namely, by iterating the single-binding operator.

Definition 4.22 (*Multibinding unification*)
We define $\mathrm{mgu}_\omega([S]_U, \delta)$ with $\delta \in ISubst$ and $[S]_U \in \mathtt{ShLin}^\omega$ by induction on the number of bindings:

$$\begin{aligned} \mathrm{mgu}_\omega([S]_U, \epsilon) &= [S]_U, \\ \mathrm{mgu}_\omega([S]_U, \{x/t\} \uplus \delta) &= \mathrm{mgu}_\omega(\mathrm{mgu}_\omega([S]_U, x/t), \delta). \end{aligned}$$

It is possible to prove that $\mathrm{mgu}_\omega([S]_U, \delta)$ is optimal for multibinding substitutions (Amato and Scozzari 2005). Since optimality of iterative multibinding unification is not inherited by the abstractions of $\mathtt{ShLin}^\omega$ (as we show in Section 5.3), we will focus on single-binding unification. In the rest of the paper, we only consider bindings x/t which are idempotent, namely, such that $x \notin vars(t)$.

4.3 Correctness of abstract unification

We now show that $\text{mgu}_\omega([S]_U, \delta)$ is correct w.r.t. concrete unification. We show correctness for multibinding substitutions, since it is a trivial extension of the single-binding case. In fact, composition of correct operators is still correct.

First of all, we extend the definition of θ^{-1} to the case in which it is applied to a sharing group B.

Definition 4.23
Given $\theta \in \mathit{ISubst}$ and B an ω-sharing group, we define

$$\theta^{-1}(B) = \lambda v \in \mathcal{V}.\chi(B, \theta(v)).$$

In order to prove the correctness of abstract unification, we need the following auxiliary property.

Proposition 4.24
Given substitutions θ, $\eta \in \mathit{ISubst}$ and an ω-sharing group B, we have

$$(\eta \circ \theta)^{-1}(B) = \theta^{-1}(\eta^{-1}(B)).$$

Theorem 4.25 (*Correctness of* mgu_ω)
The operation mgu_ω is correct w.r.t. mgu, i.e.,

$$\forall [S]_U \in \mathtt{ShLin}^\omega, \delta \in \mathit{ISubst}.\ [S]_U \rhd [\theta]_U \implies \text{mgu}_\omega([S]_U, \delta) \rhd \text{mgu}([\theta]_U, \delta).$$

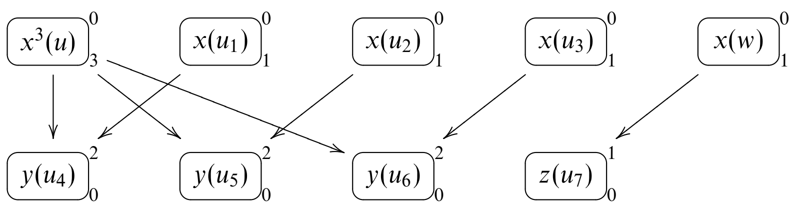
Example 4.26
Let $\theta = \{x/r(s(u,u,u), v, w), y/v', z/w'\}$, $\delta = \{x/r(y,y,z)\}$, and $U = \{x,y,z\}$. Therefore $\alpha_\omega([\theta]_U) = [x^3, x, y, z]_U$. If we proceed with the concrete unification of $[\theta]_U$ with δ, we have $\text{mgu}([\theta]_U, \delta) = [\theta']_U$ with $\theta' = \text{mgu}(\theta, \delta) = \eta \circ \theta$ and $\eta = \text{mgu}(\theta(x) = \theta(r(y,y,z)))$. This gives the following results:

$$\eta = \{v'/s(u,u,u), v/s(u,u,u), w'/w\},$$
$$\theta' = \{x/r(s(u,u,u), s(u,u,u), w), y/s(u,u,u), z/w, v'/s(u,u,u), w'/w\},$$

with $[\theta']_U = [\theta]_U$. Now, let η' be obtained from η by replacing each occurrence of a variable in $\text{rng}(\eta)$ with a different fresh variable, $\beta = \eta' \circ \theta$, and ρ be a substitution mapping variables to variables such that $\rho(\beta(x)) = \theta'(x)$ for each $x \in U$. Note that ρ is not a renaming, since it is not bijective. We have

$$\eta = \{v/s(u_1,u_2,u_3), v'/s(u_4,u_5,u_6), w'/u_7\},$$
$$\beta = \{x/r(s(u,u,u), s(u_1,u_2,u_3), w), y/s(u_4,u_5,u_6), z/u_7, v'/s(u_4,u_5,u_6), w'/u_7\},$$
$$\rho = \{u_1/u, u_2/u, u_3/u, u_4/u, u_5/u, u_6/u, u_7/w\}.$$

Following the proof, we build a multigraph G as follows:

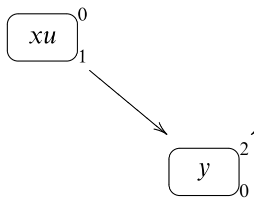


Note that we have chosen to annotate every sharing group with the corresponding variable in $vars(\beta(U))$. This is not a sharing graph, since it is not connected, but if we take $Y = \llbracket \rho^{-1}(u) \rrbracket = \{u, u_1, u_2, u_3, u_4, u_5, u_6\}$, the restriction of G to the nodes annotated with a variable in Y is a sharing graph whose resultant ω-sharing group is $x^6 y^3$. □

4.4 Optimality of abstract unification

We now prove that mgu_ω is not only correct but also optimal for a single-binding substitution; i.e., it is the least correct abstraction. This means proving that given a set of ω-sharing groups $[S]_U \in \mathtt{ShLin}^\omega$, a binding x/t, and an ω-sharing group $B \in \mathrm{mgu}_\omega([S]_U, x/t)$, there exists a substitution $[\delta]_U$ such that $[S]_U \rhd [\delta]_U$ and $B \in \alpha_\omega(\mathrm{mgu}([\delta]_U, \{x/t\}))$. First of all, we prove optimality of $\mathrm{mgu}_\omega([S]_U, x/t)$ in the special case of $vars(x/t) \subseteq U$. Next, we extend this result to the general case.

Example 4.27
Consider $S = \{xu, xv, y\}$ and the binding $x/s(y, y)$. The following is a sharing graph for $x/s(y, y)$ and S whose resultant ω-sharing group is x^2uvy:



We show how to find a substitution $[\delta]_U$ such that the ω-sharing group $x^2uvy \in \alpha_\omega(\mathrm{mgu}([\delta]_U, \{x/s(y, y)\}))$. Let $U = \{u, v, x, y\}$. For each node n of the sharing graph, we consider a different fresh variable w_n. Assume that the node labeled with xu in the upper-left corner is node 1, and proceed clockwise to number the other nodes.

For each variable $z \in U \setminus \{x\}$, we associate to $\delta(z)$ a term containing all the variables w_i such that the label of the ith node contains the variable z. Thus, we define $\delta(u) = r(w_1)$, where w_1 correspond to the node containing u. Analogously, we define $\delta(v) = r(w_2)$ and $\delta(y) = r(w_3)$.

We now define $\delta(x)$ in a different way, namely, by replacing in $s(y, y)$ each occurrence of the variable y with a term similar to $\delta(y)$, with the difference that w_3 is replaced with the variables w_1 and w_2. The choice of w_1 and w_2 is obvious by looking at the sharing graph, since the first and second nodes are the sources of the two edges targeted at the node three. Therefore we obtain $\delta(x) = s(r(w_1), r(w_2))$.

Summing up, we have

$$\delta = \{u/r(w_1), v/r(w_2), x/s(r(w_1), r(w_2)), y/r(w_3)\}.$$

It is easy to check that $[S]_U \rhd [\delta]_U$ and

$$\mathrm{mgu}(\delta, \{x/s(y, y)\}) = \{u/r(w_1), v/r(w_1), x/s(r(w_1), r(w_1)), y/r(w_1), w_2/w_1, w_3/w_1\};$$

hence $\alpha_\omega([\mathrm{mgu}(\delta, \{x/s(y, y)\})]_U) = [x^2uvy]_U$. □

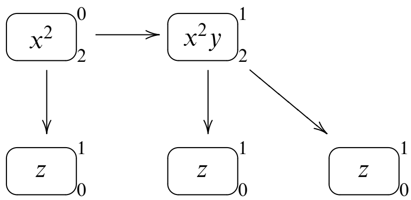
In the above example we have shown how to find a special substitution such that its fresh variables are unified according to the arrows in a sharing graph. The same idea is exploited in the next theorem for proving the optimality of the abstract unification operator mgu_ω. For any ω-sharing group $X \in \mathrm{mgu}_\omega([S]_U, x/t)$, we provide a substitution δ obtained as in Example 4.27, such that $[S]_U$ approximates $[\delta]_U$ and $X \in \alpha_\omega(\mathrm{mgu}([\delta]_U, \{x/t\}))$.

Theorem 4.28 (*Optimality of* mgu_ω)
The single-binding unification $\mathrm{mgu}_\omega([S]_U, x/t)$ is optimal w.r.t. mgu, under the assumption that $vars(x/t) \subseteq U$, i.e.,

$$\forall B \in \mathrm{mgu}_\omega([S]_U, x/t)\ \exists \delta \in \mathit{ISubst}.\ [S]_U \rhd [\delta]_U \text{ and } B \in \alpha_\omega(\mathrm{mgu}([\delta]_U, \{x/t\})).$$

The previous proof requires that $vars(x/t) \subseteq U$. However, the same construction also works when this condition does not hold.

Example 4.29
Given $U = \{x, y\}$ and $S = \{x^2, x^2y\}$, we want to compute $\mathrm{mgu}_\omega([S]_U, x/s(y,z))$. By extending the domain of the variables of interests, we obtain $[S']_V = [x^2, x^2y, z]_{x,y,z}$. One of the sharing graphs for $x/s(y,z)$ and $[S']_V$ is



Following the proof of the previous theorem, we obtain the substitution

$$\delta' = \{x/s(r(w_1), r(w_1, w_2, w_2)), y/r(w_2), z/r(w_3, w_4, w_5)\},$$

where $[S']_V \rhd [\delta']_V$ and $x^4yz^3 \in \alpha_\omega(\mathrm{mgu}([\delta']_V, \{x/s(y,z)\}))$. However we are looking for a substitution δ such that $[S]_U \rhd [\delta]_U$ and $x^4yz^3 \in \alpha_\omega(\mathrm{mgu}([\delta]_U, \{x/s(y,z)\}))$. Nonetheless, we may choose $\delta = \delta'$ (or, if we prefer, $\delta = \delta'|_{x,y}$) to get the required substitution. □

This is not a fortuitous coincidence. We may show that it consistently happens every time we apply Theorem 4.28 to an abstract unification where $vars(x/t) \not\subseteq U$. Therefore, we can prove the main result of the paper.

Theorem 4.30 (*Optimality of* mgu_ω *with extension*)
The single-binding unification mgu_ω with extension is optimal w.r.t. mgu.

4.5 A characterization for resultant sharing groups

The domain $\mathtt{ShLin}^\omega$ has not been designed to be directly implemented, but some of its abstractions could. Providing a simpler definition for the set of resultant ω-sharing groups could help in developing the abstract operators for its abstractions.

We show that given a set S of ω-sharing groups and a binding x/t, the set of resultant ω-sharing groups has an elegant algebraic characterization.

By definition of sharing graph, a set of nodes N labeled with ω-sharing groups of S can be turned into a sharing graph for S and x/t iff the condition on the out-degree and in-degree is satisfied and the obtained graph is connected. The condition on the degrees says that for each node s labeled with the sharing group B_s, the out-degree of s must be equal to $\chi(B_s, x)$. Symmetrically, the in-degree must be equal to $\chi(B_s, t)$. As a consequence, the sum of the out-degrees of all the nodes $\sum_{s\in N} \chi(B_s, x)$ must be equal to the sum of the in-degrees of all the nodes $\sum_{s\in N} \chi(B_s, t)$. This is because each edge has a source and a target node. Moreover, in order to be connected, any graph needs at least $|N| - 1$ edges. Since the number of edges is equal to the sum of in-degrees of all the nodes, it turns out that such a sum must be equal to or greater than $|N| - 1$. Surprisingly, this is enough to construct a sharing graph from N.

Theorem 4.31
Let S be a set of ω-sharing groups and x/t be a binding. Then $B \in \mathrm{mgu}_\omega(S, x/t)$ iff there exist $n \in \mathbb{N}^+$, $B_1, \ldots, B_n \in S$ which satisfy the following conditions:

(1) $B = \uplus_{1\leqslant i\leqslant n} B_i$;
(2) $\sum_{1\leqslant i\leqslant n} \chi(B_i, x) = \sum_{1\leqslant i\leqslant n} \chi(B_i, t) \geqslant n - 1$;
(3) either $n = 1$ or $\forall 1 \leqslant i \leqslant n.\ \chi(B_i, x) + \chi(B_i, t) > 0$.

Following the above theorem, we can give an algebraic characterization of the abstract unification operator as follows.

Corollary 4.32 (*Algebraic characterization of* mgu_ω)
Given a set of ω-sharing groups S and a binding x/t, we have that

$$\begin{aligned}\mathrm{mgu}_\omega(S, x/t) &= (S \setminus rel(S, x, t)) \cup \\ &\quad \left\{ \uplus\mathscr{R} \mid \mathscr{R} \in \wp_m(rel(S, x, t)), \sum_{B\in\mathscr{R}} \chi(B, x) = \sum_{B\in\mathscr{R}} \chi(B, t) \geqslant |\mathscr{R}| - 1 \right\},\end{aligned}$$

where

$$\begin{aligned} rel(S, x, t) &= \{B \in S.\ \chi(B, x) + \chi(B, t) > 0\} \\ &= \{B \in S.\ \llbracket B \rrbracket \cap vars(x/t) \neq \emptyset\}. \end{aligned}$$

Example 4.33
Consider $S = \{xa, xb, z^2, zc\}$ and the equation $x = z$. Then if we choose $\mathscr{R} = \{\!\{xa, xb, z^2\}\!\}$, we have $\sum_{B\in\mathscr{R}} \chi(B, x) = 2 = \sum_{B\in\mathscr{R}} \chi(B, z) \geqslant |\mathscr{R}| - 1$. Therefore $x^2z^2ab \in \mathrm{mgu}_\omega(S, x/z)$. If we take $\mathscr{R} = \{\!\{xa, xb, zc, zc\}\!\}$, although $\sum_{B\in\mathscr{R}} \chi(B, x) = 2 = \sum_{B\in\mathscr{R}} \chi(B, z)$, we have $|\mathscr{R}| - 1 = 3$. This only proves that $z^2c^2x^2ab$ cannot be obtained by the multiset $\mathscr{R}$. If we check for every possible multiset over S, we have that $z^2c^2x^2ab \notin \mathrm{mgu}_\omega(S, x/z)$. □

This characterization of the abstract mgu will be the key point for devising the optimal abstract unification operators on the abstractions of $\mathtt{ShLin}^\omega$. Let α be the abstraction function from $ISubst_\sim$ to an abstract domain A. If we are able to factor

α through a Galois connection $\langle \alpha' : \mathtt{ShLin}^{\omega} \to A, \gamma' : A \to \mathtt{ShLin}^{\omega} \rangle$ as $\alpha = \alpha' \circ \alpha_{\omega}$, then the optimal abstract unification for α is exactly $\alpha'(\mathrm{mgu}_{\omega}(\gamma'(\cdot), \cdot))$. However, this expression is helpful when it may be simplified in order to use only objects in A. Our algebraic characterization makes the simplification feasible, as we show in the following section.

5 Practical domains for program analysis

We consider two domains for sharing analysis with linearity information, namely, the domain proposed in King (1994) and the classical reduced product $\mathtt{Sharing} \times \mathtt{Lin}$. They are defined as abstractions of $\mathtt{ShLin}^{\omega}$ through Galois insertions. This allows us to design optimal abstract operators for both of them, by exploiting the results introduced so far. By composing each Galois insertion with α_{ω}, we get the corresponding abstraction function for substitutions (Cousot and Cousot 1992a, Section 4.2.3.1).

5.1 King's domain for linearity and aliasing

We first consider the domain for combined analysis of sharing and linearity in King (1994)(King 1994). The idea is to enhance the domain Sharing by annotating each sharing group with linearity information on each variable. For instance, the object $xy^{\infty}z$ represents the sharing group xyz and the information that y may be nonlinear (while x and z are definitely linear). The objects in this domain can be easily viewed as abstraction of ω-sharing groups. Intuitively, in order to abstract an ω-sharing groups, one simply needs to replace each exponent equal to or greater than two with ∞. Let us now formalize the domain as an abstraction of $\mathtt{ShLin}^{\omega}$.

An ω-sharing group (which is a multiset $\mathcal{V} \to \mathbb{N}$ whose support is finite) is abstracted into a map $o : \mathcal{V} \to \{0, 1, \infty\}$ such that its support $\llbracket o \rrbracket = \{v \in \mathcal{V} \mid o(v) \neq 0\}$ is finite. We call such a map the *2-sharing group*. We use a polynomial notation for 2-sharing groups as for ω-sharing groups. For instance, $o = xy^{\infty}z$ denotes the 2-sharing group whose support is $\llbracket o \rrbracket = \{x, y, z\}$, such that $o(x) = o(z) = 1$ and $o(y) = \infty$. We denote with $\emptyset$ the 2-sharing group with empty support. Note that in King (1994)(King 1994) the number 2 is used as an exponent instead of ∞, but we prefer this notation to be coherent with ω-sharing groups.

We denote $\min\{o(x), 2\}$ by $o_m(x)$, where the ordering on $\mathbb{N}$ is extended in the obvious way; i.e., for all $n \in \mathbb{N}$ we have that $n < \infty$. A 2-sharing group o represents the sets $\gamma_2(o)$ of ω-sharing group given by

$$\gamma_2(o) = \{B \in \wp_m(\mathcal{V}) \mid \llbracket o \rrbracket = \llbracket B \rrbracket \land \forall x \in \llbracket o \rrbracket . o_m(x) \leqslant B(x) \leqslant o(x)\}.$$

For instance, the 2-sharing group $xy^{\infty}z$ represents the set of ω-sharing groups $\{xy^2z, xy^3z, xy^4z, xy^5z, \ldots\}$. The idea is to use 2-sharing groups to keep track of linearity: if $o(x) = \infty$, it means that the variable x is not linear in the 2-sharing group o. In the rest of this section, we use the term "sharing group" as a short form of 2-sharing group, when this does not cause ambiguity.

An ω-sharing group B may be abstracted into the 2-sharing group $\alpha_2(B)$ given by

$$\alpha_2(B) = \lambda v \in \llbracket B \rrbracket. \begin{cases} 1 & \text{if } B(x) = 1, \\ \infty & \text{otherwise.} \end{cases}$$

The next proposition shows two useful properties of the maps α_2 and γ_2.

Proposition 5.1

The following properties hold:

(1) $\alpha_2(\biguplus \mathscr{R}) = \biguplus \alpha_2(\mathscr{R})$.
(2) $rel(\gamma_2(S), x, t)) = \gamma_2(rel(S, x, t))$.

Since we do not want to represent definite nonlinearity, we introduce an order relation over sharing groups,

$$o \leqslant o' \iff \llbracket o \rrbracket = \llbracket o' \rrbracket \wedge \forall x \in \llbracket o \rrbracket.\ o(x) \leqslant o'(x),$$

and we restrict our attention to downward closed sets of sharing groups. We denote by $Sg^2(V)$ the set of 2-sharing groups whose support is a subset of V. The domain we are interested in is the following:

$$\mathtt{ShLin}^2 = \{[S]_U \mid S \in \wp_{\downarrow}(Sg^2(U)), U \in \wp_f(\mathscr{V}), S \neq \emptyset \Rightarrow \emptyset \in S\},$$

where $\wp_{\downarrow}(Sg^2(U))$ is the powerset of downward closed subsets of $Sg^2(U)$ according to $\leqslant$ and $[S_1]_{U_1} \leqslant_2 [S_2]_{U_2}$ iff $U_1 = U_2$ and $S_1 \subseteq S_2$. For instance, the set $\{xy^{\infty}z\}$ is not downward closed, while $\{xyz, xy^{\infty}z\}$ is downward closed. There is a Galois insertion of $\mathtt{ShLin}^2$ into $\mathtt{ShLin}^{\omega}$ given by the pair of adjoint maps $\gamma_2 : \mathtt{ShLin}^2 \to \mathtt{ShLin}^{\omega}$ and $\alpha_2 : \mathtt{ShLin}^{\omega} \to \mathtt{ShLin}^2$:

$$\gamma_2([S]_U) = \left[\bigcup\{\gamma_2(o) \mid o \in S\}\right]_U,$$

$$\alpha_2([S]_U) = \left[\downarrow\{\alpha_2(B) \mid B \in S\}\right]_U.$$

With an abuse of notation, we also apply γ_2 and α_2 to subsets of ω-sharing groups and 2-sharing groups respectively, by ignoring the set of variables of interest. For instance, $\gamma_2(\{xyz, xy^{\infty}z\}) = \{xyz, xy^2z, xy^3z, xy^4z, xy^5z, \ldots\}$.

Theorem 5.2

The pair $\langle \alpha_2, \gamma_2 \rangle$ is a Galois insertion.

Now we may define the optimal mgu for $\mathtt{ShLin}^2$ and single-binding substitutions as follows:

Definition 5.3 (*Unification for* $\mathtt{ShLin}^2$)

Given $[S]_U \in \mathtt{ShLin}^2$ and the binding x/t, we define

$$\mathrm{mgu}_2([S]_U, x/t) = \alpha_2(\mathrm{mgu}_{\omega}(\gamma_2([S]_U), x/t)).$$

By construction, mgu_2 is the optimal abstraction of mgu_ω and hence also of mgu. In the case in which $vars(x/t) \subseteq U$, by using additivity of α_2 we get

$$\begin{aligned}\mathrm{mgu}_2([S]_U, x/t) \;=\; \Big[&\alpha_2(\gamma_2(S) \setminus rel(\gamma_2(S), x, t)) \cup \\ &\alpha_2\Big(\{\uplus\mathscr{R} \mid \mathscr{R} \in \wp_m(rel(\gamma_2(S), x, t)), \\ &\sum_{B\in\mathscr{R}} \chi(B, x) = \sum_{B\in\mathscr{R}} \chi(B, t) \geqslant |\mathscr{R}| - 1\}\Big)\Big]_U. \end{aligned} \tag{4}$$

Now we want to simplify equation (4). In particular we would like to get rid of the abstraction and concretization maps and to express the result using only objects and operators in $\mathtt{ShLin}^2$. Therefore, we need to define operations in $\mathtt{ShLin}^2$ which correspond to $\uplus$ and χ in $\mathtt{ShLin}^\omega$.

The operation on 2-sharing groups which corresponds to multiset union on ω-sharing groups is given by

$$o \uplus o' = \lambda v \in \mathscr{V}.o(v) \oplus o'(v),$$

where $0 \oplus x = x \oplus 0 = x$ and $\infty \oplus x = x \oplus \infty = 1 \oplus 1 = \infty$. We will use $\biguplus\{\!\{o_1, \dots, o_n\}\!\}$ for $o_1 \uplus \dots \uplus o_n$. Given a sharing group o, we also define the *delinearization* operator:

$$o^2 = o \uplus o. \tag{5}$$

Note that $o^2 = \lambda x \in \llbracket o \rrbracket.\infty$. The operator is extended pointwise to sets and multisets.

A fundamental role is played by the notion of multiplicity of a sharing group in a term. While the multiplicity of an ω-sharing group in a term is a single natural number, every object in $\mathtt{ShLin}^2$ represents a set of ω-sharing groups; hence its multiplicity should be a set of natural numbers. Actually, it is enough to consider intervals. We define the minimum χ_m and maximum χ_M multiplicity of o in t as follows:

$$\chi_m(o,t) = \sum_{v\in\llbracket o\rrbracket} o_m(v) \cdot occ(v,t) \qquad \chi_M(o,t) = \sum_{v\in\llbracket o\rrbracket} o(v) \cdot occ(v,t).$$

Sum and product on integers are lifted in the obvious way; namely, the sum is ∞ iff at least one of the addenda is ∞ and $n \cdot \infty = \infty \cdot n = \infty$ for any $n \in \mathbb{N}^+$, while $0 \cdot \infty = \infty \cdot 0 = 0$. The maximum multiplicity $\chi_M(o,t)$ either is equal to the minimum multiplicity $\chi_m(o,t)$ or is infinite. Note that if B is an ω-sharing group represented by o, i.e., $B \in \gamma_2(o)$, then $\chi_m(o,t) \leqslant \chi(B,t) \leqslant \chi_M(o,t)$. Actually, not all the values between $\chi_m(o,t)$ and $\chi_M(o,t)$ may be assumed by $\chi(B,t)$.

Example 5.4
Let $o = x^\infty$ and $t = f(x,x)$. According, to our definition, $\chi(o,t) = [4,\infty)$. However, it is obvious that if $B \in \gamma_2(o)$, then $\chi(B,t)$ is an even number. □

According to the above definitions, we define the multiplicity of a multiset of sharing groups as

$$\chi(Y,t) = \Big\{n \in \mathbb{N} \mid \sum_{o\in Y} \chi_m(o,t) \leqslant n \leqslant \sum_{o\in Y} \chi_M(o,t)\Big\}.$$

Even if this is a superset of all the possible values which can be obtained by combining the multiplicities of all the sharing groups in Y, this definition is sufficiently accurate to allow us to design the optimal abstract unification.

We extend in the obvious way the definition of *rel* (see Corollary 4.32) from ω-sharing groups to 2-sharing groups, i.e.,

$$rel(S,x,t) = \{o \in S \mid \llbracket o \rrbracket \cap vars(x/t) \neq \emptyset\},$$

and we prove the following.

Theorem 5.5 (*Characterization of abstract unification for* ShLin^2)
Given $[S]_U \in \texttt{ShLin}^\omega$ and the binding x/t with $vars(x/t) \subseteq U$, we have that

$$\begin{aligned}\mathrm{mgu}_2([S]_U, x/t) = [&(S \setminus S') \cup \\ &\downarrow\{\biguplus Y \mid Y \in \wp_m(S'), n \in \chi(Y,x) \cap \chi(Y,t).\ n \geqslant |Y|-1\}]_U,\end{aligned}$$

where $S' = rel(S,x,t)$.

Example 5.6
Let $S = \downarrow\{\emptyset, ux^\infty, vx^\infty, x^\infty y, z^\infty\}$ and $Y = \{\!\{ux^\infty, vx^\infty, xy, z^\infty\}\!\}$. We have that $\chi(Y,x) = \{n \mid n \geqslant 5\}$ and $\chi(Y, f(z,z)) = \{n \mid n \geqslant 4\}$. Since $f(z,z)$ contains two occurrences of z, the "actual" multiplicity of the sharing group z^∞ in $f(z,z)$ should be a multiple of two. But we do not need to check this condition and can safely approximate this set with $\{n \mid n \geqslant 4\}$. This works because we can always choose a number which is contained in both $\chi(Y,x)$ and $\chi(Y,t)$ and which is an "actual" multiplicity. For instance, we can take $n = 6 \in \chi(Y,x) \cap \chi(Y, f(z,z))$, and since we have $6 \geqslant 3 = |Y| - 1$, we get that the sharing group $\biguplus Y = uvx^\infty yz^\infty$ belongs to $\mathrm{mgu}_2([S]_U, x/f(z,z))$. This sharing group can be generated by the substitution $\{x/f(f(u,u,y), f(v,v,y)), z/f(w,w,w)\}$ when the variables of interest are $\{u,v,x,y,z\}$. $\square$

Theorem 5.5 gives a characterization of the abstract unification over ShLin^2. However, it cannot be directly implemented, since one needs to check a certain condition for each element of $\wp_m(rel(S,x,t))$, which is an infinite set. Nonetheless, this is an important starting point to prove correctness and completeness of the abstract unification algorithm which we are going to introduce.

The characterization in Theorem 5.5 may be used even when $vars(x/t) \nsubseteq U$, if we first enlarge the set of variables of interest in order to include all $vars(x/t)$.

Theorem 5.7 (*Characterization of abstract unification with extension for* ShLin^2)
Given $[S]_U$ in ShLin^2 and the binding x/t, let $V = \{v_1, \ldots, v_n\}$ be $vars(x/t) \setminus U$. Then,

$$\mathrm{mgu}_2([S]_U, x/t) = \mathrm{mgu}_2([S \cup \{v_1, \ldots, v_n\}]_{U \cup V}, x/t).$$

The previous theorem states that enlarging the set of variables of interest preserves optimality.

5.2 *An algorithm for abstract unification in* ShLin^2

In order to obtain an algorithm from the characterization in Theorem 5.5 we need to avoid the use of $\wp_m(rel(S,x,t))$ and to develop a procedure able to compute the resultant sharing groups by inspecting subsets (not multisets) of $rel(S,x,t)$. In general, any $X \subseteq rel(S,x,t)$ yields more than one sharing group, since every element in X may be considered more than once. However, since $\mathtt{ShLin}^2$ is downward closed, it is enough to compute the maximal resultant sharing groups.

Given $X \subseteq rel(S,x,t)$ and the binding x/t, assume that we are only interested in those sharing groups whose support is $\llbracket \biguplus X \rrbracket$. By joining (multiple copies of) the sharing groups in X, any resultant sharing group o is between $\biguplus X$ and $\biguplus X^2$, i.e., $\biguplus X \leqslant o \leqslant \biguplus X^2$, where X^2 is the pointwise extension of the delinearization operator (see equation (5)). Note that if X is badly chosen, it is possible that we are not able to generate any sharing group with this support. In this computation, the notion of multiplicity of a sharing group in a term plays a major role.

For example, given the binding x/t, if $\chi_M(o,x) \leqslant 1$ for each $o \in X$, then $\biguplus X$ is a resultant sharing group only if there is a unique sharing group $o \in X$ such that $vars(t) \cap \llbracket o \rrbracket \neq \emptyset$. If there are $o_1, o_2 \in X$ such that $\chi_M(o_1,x) > 1$ and $\chi_M(o_2,t) > 1$, then $\biguplus X$ is a resultant sharing group. Moreover, we may join two copies of each sharing group in X, and therefore also $\biguplus X^2$ is a resultant sharing group.

Now we can define the notions of linearity and nonlinearity on the abstract domain. In addition, we also introduce a new notion of strong nonlinearity. Given $X \subseteq rel(S,x,t)$, we partition X in three subsets $X_x = \{o \in X \mid \chi_M(o,t) = 0\}$, $X_t = \{o \in X \mid \chi_M(o,x) = 0\}$, and $X_{xt} = X \setminus (X_x \cup X_t)$.

Definition 5.8
Given a set S of sharing groups and $X \subseteq rel(S,x,t)$, we say that X is

- *linear* for the term t if for all $o \in X$ it holds that $\chi_M(o,t) \leqslant 1$;
- *nonlinear* for the term t if there exists $o \in X$ such that $\chi_M(o,t) > 1$;
- *strongly nonlinear* for the term t if there exists $o \in X$ such that $\chi_M(o,t) = \infty$ or there exists $o \in X_{xt}$ such that $\chi_M(o,t) > 1$.

Analogously, we define linearity and nonlinearity of X for the variable x.

Note that if t is a variable, the nonlinear and strongly nonlinear cases coincide. We now present the algorithm for computing the abstract unification in $\mathtt{ShLin}^2$.

Theorem 5.9 (*Abstract unification algorithm for* $\mathtt{ShLin}^2$)
Given $[S]_U \in \mathtt{ShLin}^2$ and the binding x/t with $vars(x/t) \subseteq U$, we have

$$\mathrm{mgu}_2([S]_U, x/t) = [(S \setminus S') \cup \downarrow \bigcup_{X \subseteq S'} res(X,x,t)]_U,$$

where $S' = rel(S,x,t)$ and $res(X,x,t)$ is defined as follows:

(1) if X is nonlinear for x and t, then $res(X,x,t) = \{\biguplus X^2\}$;
(2) if X is nonlinear for x and linear for t, $|X_x| \leqslant 1$ and $|X_t| \geqslant 1$, then we have $res(X,x,t) = \{(\biguplus X_x) \uplus (\biguplus X_{xt}^2) \uplus (\biguplus X_t^2)\}$;
(3) if X is linear for x and strongly nonlinear for t, $|X_x| \geqslant 1$ and $|X_t| \leqslant 1$, then we have $res(X,x,t) = \{(\biguplus X_x^2) \uplus (\biguplus X_{xt}^2) \uplus (\biguplus X_t)\}$;
(4) if X is linear for x and not strongly nonlinear for t, $|X_t| \leqslant 1$, then we have

$$\begin{aligned} res(X,x,t) = \{(\biguplus Z) \uplus (\biguplus X_{xt}^2) \uplus (\biguplus X_t) \mid\ & Z \in \wp_m(X_x), \\ & |Z| = \chi_M(X_t,t) = \chi_m(X_t,t), \\ & \llbracket Z \rrbracket = X_x\}; \end{aligned}$$

(5) otherwise $res(X,x,t) = \emptyset$.

Example 5.10
Let $U = \{u, v, x, y\}$ and consider the set of 2-sharing groups $S = \{\emptyset, xu, x^\infty, xy, yv\}$ and consider the binding $x/r(y, y)$. Note that $S' = \{xu, x^\infty, xy, yv\}$. Let us compute $res(X, x, r(y, y))$ for some X's, subsets of S'.

- $X = \{x^\infty, yv\}$. In this case, $\chi_M(x^\infty, x) = \infty$ and $\chi_M(yv, r(y, y) = 2$; hence X is nonlinear for x and $r(y, y)$. From the first case of Theorem 5.9, we have that $res(X, x, r(y, y)) = \{\biguplus X^2\} = \{\biguplus\{x^\infty, y^\infty v^\infty\} = \{x^\infty y^\infty v^\infty\}$.
- $X = \{xu, xy, yv\}$. Then X is linear for x and strongly nonlinear for $r(y, y)$, since $xy \in X_{xt}$ and $\chi_M(xy, r(y, y)) = 2$. From the third case, it follows that $res(X, x, t) = \{(\biguplus\{xu\}^2) \uplus (\biguplus\{xy\}^2) \uplus (\biguplus\{yv\})\} = \{x^\infty y^\infty u^\infty v\}$.
- $X = \{xu, yv\}$. Then X is linear for x and not strongly nonlinear for $r(y, y)$. (Note that $\chi_M(yv, r(y, y)) = 2 > 1$ and $yv \in X_t$; hence X is nonlinear for $r(y, y)$, but it is not strongly nonlinear.) Since $\chi_M(X_t, r(y, y)) = 2$, we only need to consider those $Z \in \wp_m(X_x)$ such that $|Z| = 2$. There is only one such set, which is $Z = \{\!\{xu, xu\}\!\}$. Therefore $res(X, x, r(y, y)) = \{(\biguplus\{\!\{xu, xu\}\!\}) \uplus (\biguplus\{\}^2) \uplus (\biguplus\{yv\})\} = \{x^\infty y u^\infty v\}$. □

Note that given $X \subseteq S'$, if x does not appear in any sharing group of S, then $res(X, x, t) \subseteq \{\emptyset\}$. In fact, we can only apply the fourth or the fifth case. In the fourth case, we have that $X_x = X_{xt} = \emptyset$, and thus the only $Z \in \wp_m(X_x)$ is the empty multiset. Thus, $|Z| = 0$, which implies that $X_t = \emptyset$, and $res(X, x, t) = \{\emptyset\}$. In the fifth case, the result is trivially the emptyset. Symmetrically, when none of the variables of t appears in S, again we can apply only the fourth or the fifth case and $res(X, x, t) \subseteq \{\emptyset\}$.

Example 5.11
Consider S and U as in Example 5.10. We compute $\text{mgu}_2([S]_U, x/r(y, y))$. We show the value of $res(X, x, r(y, y))$ for every $X \subseteq S' = rel(S, x, r(y, y))$ which contains both the variables x and y:

X	$res(\mathbf{S}, \mathbf{x}, \mathbf{r}(\mathbf{y}, \mathbf{y}))$	Case in Theorem 5.9
x^∞, xy	$x^\infty y^\infty$	1
x^∞, yv	$x^\infty y^\infty v^\infty$	1
x^∞, xy, yv	$x^\infty y^\infty v^\infty$	1
x^∞, xu, xy	$x^\infty y^\infty u^\infty$	1
x^∞, xu, yv	$x^\infty y^\infty u^\infty v^\infty$	1
x^∞, xu, xy, yv	$x^\infty y^\infty u^\infty v^\infty$	1
xu, xy	$x^\infty y^\infty u^\infty$	3
xu, yv	$x^\infty y u^\infty v$	4
xu, xy, yv	$x^\infty y^\infty u^\infty v$	3

Hence

$$\text{mgu}_2([S]_U, x/r(y, y)) = \downarrow\{\emptyset, x^\infty y^\infty, x^\infty y^\infty v^\infty, x^\infty y^\infty u^\infty, x^\infty y^\infty u^\infty v^\infty, x^\infty y u^\infty v, x^\infty y^\infty u^\infty v\}.$$ □

The main difference between the algorithm in Theorem 5.9 and the characterization in Theorem 5.5 is that in the former X is a subset of S' while in Theorem 5.5 Y is a multiset over S'. Since the number of subsets of S' is finite, the characterization in Theorem 5.9 is an algorithm.

Obviously, a direct implementation of mgu_2 would be very slow, so that appropriate data structures and procedures should be developed for a real implementation. Although this is mostly out of the scope of this paper, we show here that the definition of $\mathrm{mgu}_2([S]_U, x/t)$ may be modified to consider only *maximal* subsets of $rel(S, x, t)$. This should help in reducing the computational complexity of the abstract operator.

Given $[A]_U \in \mathtt{ShLin}^2$, let $\max A$ be the set of maximal elements of A, i.e., $\max A = \{a \in A \mid \nexists b \in A. b >_2 a\}$. Given a sharing group o, we define the *linearized* version of o, denoted by $l(o)$, as

$$l(o)(v) = \begin{cases} 1 & \text{if } v \in \llbracket o \rrbracket, \\ 0 & \text{otherwise.} \end{cases}$$

The linearization operator l is extended pointwise to sets of sharing groups. We show that instead of choosing X as a subset of S' in the definition of mgu_2, we may only consider those X's which are subsets of $\max S'$.

Theorem 5.12
Given $[S]_U \in \mathtt{ShLin}^2$ and the binding x/t with $vars(x/t) \subseteq U$, we have

$$\mathrm{mgu}_2([S]_U, x/t) = [(S \setminus S') \cup \downarrow \bigcup_{X \sqsubseteq \max S'} (res(X, x, t) \cup res'(X, x, t))]_U,$$

where $S' = rel(S, x, t)$ and

$$res'(X, x, t) = \begin{cases} \{\biguplus X^2\} & \text{if } X = X_{xt} \text{ and } l(X) \text{ is linear for } t, \\ \emptyset & \text{otherwise.} \end{cases}$$

The next examples compare our optimal abstract unification operator to the original one and show the increase in precision.

Example 5.13
Let $U = \{u, v, w, x, y\}$. Consider the set of 2-sharing groups $S = \{\emptyset, xu, xv, xw, y\}$. We compute $\mathrm{mgu}_2([S]_U, x/r(y, y))$. Since $rel(S, x, r(y, y)) = S$, we need to consider any $X \sqsubseteq S$. If $y \notin X$, then clearly $res(X, x, r(y, y)) = \emptyset$. If $y \in X$, since $\chi_M(y, r(y, y)) = 2$, it follows that X is linear for x and not strongly nonlinear for $r(y, y)$. Thus

$$\mathrm{mgu}_2([S]_U, x/r(y, y)) = [\downarrow\{\emptyset, x^\infty u^\infty y, x^\infty uvy, x^2uwy, x^\infty v^\infty y, x^\infty vwy, x^\infty w^\infty y\}]_U$$

On the other hand, computing with the unification algorithm given in King (1994), the result is

$$\downarrow\{\emptyset, x^\infty u^\infty y, x^\infty u^\infty v^\infty y, x^\infty u^\infty w^\infty y, x^\infty v^\infty y, x^\infty v^\infty w^\infty y, x^\infty w^\infty y, x^\infty u^\infty v^\infty w^\infty y\}.$$

The old algorithm is not able to infer the linearity which arises when combining two distinct sharing groups from $\{xu, xv, xw\}$ with $\{y\}$. Moreover, it does not assert that the variables u, v, w cannot share a common variable. □

Example 5.14
Let $U = \{u, x, y, z\}$ and $S = \{\emptyset, xu, xy, yz\}$. By computing $\mathrm{mgu}_2([S]_U, x/r(y))$ we obtain $\downarrow\{\emptyset, x^\infty y^\infty, x^\infty u y^\infty z\}$, which shows that u and z are linear after the unification. This is not the case when computing with the unification algorithm in King (1994), since we obtain $\downarrow\{\emptyset, x^\infty y^\infty, x^\infty u^\infty y^\infty z^\infty, x^\infty u^\infty y^\infty, x^\infty y^\infty z^\infty\}$. Note that we also improve the groundness information. In fact, in our result, groundness of u implies groundness of z. □

Both examples show the increased precision w.r.t. King's algorithm. In the first example, we obtain optimality thanks to the introduction of the notion of (non-)strong nonlinearity. In the second example, we improve the result, since we do not need to consider independence between x and t, in order to exploit linearity information.

5.3 Unification for multibinding substitutions

The unification operator on $\mathtt{ShLin}^2$ has been defined for single-binding substitutions. It is possible to extend this operator to multibinding substitutions in the obvious way, namely, by iterating the single-binding operators:

$$\mathrm{mgu}_2([S]_U, \{x/t\} \uplus \theta) = \mathrm{mgu}_2(\mathrm{mgu}_2([S]_U, x/t), \theta).$$

However, defined in such a way, mgu_2 is not optimal. Consider, for example, $S = \{\emptyset, xz, yw\}$, $U = \{x, y, z, w\}$, and the substitution $\theta = \{x/r(y, y), z/w\}$. We have that $\mathrm{mgu}_2([S]_U, x/r(y, y)) = [\downarrow \{\emptyset, x^\infty z^\infty yw\}]_U$. Since $x^\infty zyw \leqslant_2 x^\infty z^\infty yw$, by applying the third case of mgu_2 to $Y = \{x^\infty zyw\}$ we get

$$\mathrm{mgu}_2([\downarrow \{\emptyset, z^\infty x^\infty yw\}]_U, z/w) = [\downarrow \{\emptyset, x^\infty y^\infty z^\infty w^\infty\}]_U.$$

However,

$$\begin{aligned}
&\alpha_2(\mathrm{mgu}_\omega(\gamma_2([\{\emptyset, xz, yw\}]_U, \theta)))\\
&\quad= \alpha_2(\mathrm{mgu}_\omega([\{xz, yw\}]_U, \theta))\\
&\quad= \alpha_2(\mathrm{mgu}_\omega([\{wx^2yz^2\}]_U, \{z/w\}))\\
&\quad= \alpha_2([\{\!\!\{\}\!\!\}]_U) = [\emptyset]_U,
\end{aligned}$$

which shows that mgu_2 is not optimal. Note that we do not use optimality of mgu_ω to prove this result, since correctness is enough.

The problem is that to be able to conclude that the unification of S with θ is ground, we need to keep track of the fact that after the first binding w is linear and z is definitively nonlinear. Since $\mathtt{ShLin}^2$ is downward closed, we are not able to state this property. Note that in the case we have presented here, by changing the order of the bindings we get an optimal result in $\mathtt{ShLin}^2$, but this happens just by accident.

Now, consider the substitution $\theta = \{x/r(y, \ldots, y), z/s(y, \ldots, y), u/v\}$ with $S = \{\emptyset, xu, zv, y\}$ and $U = \{u, v, x, y, z\}$. Assume that $r(y, \ldots, y)$ is an n-ary term and that

$s(y,\ldots,y)$ is an m-ary term with $n \neq m$ and $n, m \geqslant 2$. We have that

$$\begin{aligned}
&\mathrm{mgu}_2([S]_U, x/r(y,\ldots,y)) = [\downarrow\{\emptyset, x^\infty u^\infty y, zv\}]_U,\\
&\mathrm{mgu}_2([\downarrow\{\emptyset, x^\infty u^\infty y, zv\}]_U, z/s(y,\ldots,y)) = [\downarrow\{\emptyset, x^\infty u^\infty z^\infty v^\infty y\}]_U,\\
&\mathrm{mgu}_2([\downarrow\{\emptyset, x^\infty u^\infty z^\infty v^\infty y\}]_U, u/v) = [\downarrow\{\emptyset, x^\infty u^\infty z^\infty v^\infty y\}]_U.
\end{aligned}$$

On the other hand, we have that

$$\begin{aligned}
&\alpha_2(\mathrm{mgu}_\omega(\gamma_2([\{\emptyset, xu, zv, y\}]_U, \theta)))\\
&\quad = \alpha_2(\mathrm{mgu}_\omega([\{xu, zv, y\}]_U, \theta))\\
&\quad = \alpha_2(\mathrm{mgu}_\omega([\{x^n u^n y, zv\}]_U, \{z/s(y,\ldots,y), u/v\}))\\
&\quad = \alpha_2(\mathrm{mgu}_\omega([\{x^n u^n y z^m v^m\}]_U, \{u/v\}))\\
&\quad = \alpha_2([\{\!\!\{\}\!\!\}]_U) = [\emptyset]_U.
\end{aligned}$$

However, if $n = m$, we have

$$\begin{aligned}
&\alpha_2(\mathrm{mgu}_\omega(\gamma_2([\{\emptyset, xu, zv, y\}]_U, \theta)))\\
&\quad = \alpha_2([\{\{\!\!\{\}\!\!\}\} \cup \{x^{kn} u^{kn} y^k z^{kn} v^{kn} \mid k \in \mathbb{N}\}]_U)\\
&\quad = [\downarrow\{\emptyset, x^\infty u^\infty z^\infty v^\infty y\}]_U.
\end{aligned}$$

In this case, keeping track of the variables which are definitively nonlinear does not help. It seems that in order to compute abstract unification one binding at a time, we need to work in a domain which is able to keep track of the exact multiplicity of variables in a sharing group. Actually, this is how $\mathtt{ShLin}^\omega$ works. Obviously, we could try to develop a different algorithm for unification in $\mathtt{ShLin}^2$ which directly works with multibinding substitutions. However, since the algorithm for single-binding substitutions is already quite complex, we think this is not worth the effort.

5.4 The domain Sharing × Lin

The reduced product ShLin = Sharing × Lin has been used for a long time in the analysis of aliasing properties, since it was recognized that the precision of these analyses could be greatly improved by keeping track of the linear variables. Among the papers which consider the domain ShLin, we refer to Hans and Winkler (1992) and Hill *et al.* (2004). Actually, these papers also deal with freeness properties, which we do not consider here, to further improve precision. Although the domain ShLin has been used for many years, the optimal unification operator is as yet unknown, even for a single-binding substitution. We provide here a new abstract operator for ShLin, designed from the abstract unification for $\mathtt{ShLin}^2$, and we prove that it is optimal for single-binding substitutions.

The domain ShLin keeps track of linearity by recording, for each object of Sharing, the set of linear variables. Each element is now a triple: the first component is an object of Sharing; the second component is an object of Lin, that is, the set of variables which are linear in all the sharing groups of the first component; and the third component is the set of variables of interest. It is immediate that ShLin is an abstraction of $\mathtt{ShLin}^2$ (and thus of $\mathtt{ShLin}^\omega$). In the following, we briefly recall the definition of the abstract domain and provide the abstraction function from $\mathtt{ShLin}^2$,

$$\mathtt{ShLin} = \{[S, L, U] \mid S \subseteq \wp(U), (S \neq \emptyset \Rightarrow \emptyset \in S), L \supseteq U \setminus vars(S), U \in \wp_f(\mathscr{V})\},$$

with the approximation relation $\leqslant_{sl}$ defined as $[S, L, U] \leqslant_{sl} [S', L', U']$ iff $U = U'$, $S \subseteq S'$, $L \supseteq L'$. There is a Galois insertion of $\mathtt{ShLin}$ into $\mathtt{ShLin}^2$ given by the pair of maps,

$$\alpha_{sl}([S]_U) = [\{\llbracket o \rrbracket \mid o \in S\}, \{x \in U \mid \forall o \in S.\ o(x) \leqslant 1\}, U],$$

$$\gamma_{sl}([S, L, U]) = [\{B_L \mid B \in S\}]_U,$$

where B_L is the 2-sharing group which has the same support of B, with linear variables dictated by the set L. In formula

$$B_L = \lambda v \in \mathscr{V}. \begin{cases} \infty & \text{if } B \in U \setminus L, \\ 1 & \text{if } B \in L, \\ 0 & \text{otherwise.} \end{cases}$$

The functional composition of α_ω, α_2, and α_{sl} gives the standard abstraction map from substitutions to $\mathtt{ShLin}$. We still use the polynomial notation to represent sharing groups, but now all the exponents are fixed to one. Note that the last component U in $[S, L, U]$ is redundant, since it can be retrieved as $L \cup vars(S)$. This is because the set L contains all the ground variables.

5.5 *Abstract unification for* Sharing × Lin

In order to obtain a correct and optimal abstract unification over $\mathtt{ShLin}$, the trivial way is to directly compute $\alpha_{sl}(\mathrm{mgu}_2(\gamma_{sl}([S, L, U]), x/t))$. However, we prefer to give an unification operator similar to the other operators for $\mathtt{ShLin}$ in the literature (Howe and King 2003; Hill *et al.* 2004; Bagnara *et al.* 2005). As for the domain $\mathtt{ShLin}^2$, we now provide the notions of multiplicity and linearity over $\mathtt{ShLin}$.

Given a set L of linear variables, we define the maximum multiplicity of a sharing group o in a term t as follows:

$$\chi_M^L(o, t) = \begin{cases} \sum_{v \in o} occ(v, t) & \text{if } o \cap vars(t) \subseteq L, \\ \infty & \text{otherwise.} \end{cases}$$

According to the similar definition for 2-sharing groups, given $[S, L, U] \in \mathtt{ShLin}$, we say that (S, L) is *linear* for a term t when for all $o \in S$ it holds that $\chi_M^L(o, t) \leqslant 1$. Note that when t is a variable, the definition boils down to check whether $t \in L$.

Given $X \subseteq rel(S, x, t)$, we fix the set L of linear variables and partition X into the three subsets $X_x = \{o \in X \mid \chi_M^L(o, t) = 0\}$, $X_t = \{o \in X \mid \chi_M^L(o, x) = 0\}$, and $X_{xt} = X \setminus (X_x \cup X_t)$. Moreover, we need to define the following subsets of X:

$$X_t^{=\infty} = \{B \in X_t \mid \chi_M^L(B, t) = \infty\}, \qquad X_t^{\in\mathbb{N}} = \{B \in X_t \mid \chi_M^L(B, t) \in \mathbb{N}\},$$
$$X_t^{=1} = \{B \in X_t \mid \chi_M^L(B, t) = 1\}, \qquad X_t^{>1} = \{B \in X_t \mid \chi_M^L(B, t) > 1\},$$
$$X_{xt}^{=1} = \{B \in X_{xt} \mid \chi_M^L(B, t) = 1\}, \qquad X_{xt}^{>1} = \{X_{xt} \mid \chi_M^L(B, t) > 1\}.$$

Since we do not deal with definite linearity, we need to take into account the sharing groups which can be obtained by linearizing variables. This may be accomplished by using the set U instead of L when computing the multiplicity. We denote by X_{xt}^U the set

$$X_{xt}^U = \{B \in X_{xt} \mid \chi_M^U(B,t) = 1\},$$

which corresponds to the *linearizable* sharing groups.

Moreover, given sets $A_1, \ldots, A_n$ with $n \geqslant 2$ we denote by $\mathrm{bin}(A_1, \ldots, A_n)$ the set $\{\bigcup\{a_1, \ldots, a_n\} \mid a_1 \in A_1, \ldots, a_n \in A_n\}$, by A^* the set $\{\bigcup B \mid B \subseteq A\}$, and by A^+ the set $\{\bigcup B \mid B \subseteq A, B \neq \emptyset\}$. This notation slightly deviates from most of other literature on `Sharing`, where A^* does not include the empty set. We prefer to adopt a double notation, namely, A^* and A^+, which is more standard in the rest of the research community.

Definition 5.15 (*Abstract unification algorithm for* `ShLin`)
Given $[S, L, U] \in$ `ShLin` and the binding x/t such that $vars(x/t) \subseteq U$, we define

$$\mathrm{mgu}_{sl}([S, L, U], x/t) = [(S \setminus X) \cup K, U' \cup L', U],$$

where $X = rel(S, x, t) = \{B \in S \mid B \cap vars(x/t) \neq \emptyset\}$ and $U' = U \setminus vars((S \setminus X) \cup K)$. Here, K is the set of new sharing groups created by the unification process and U' is the set of variables which do not appear in any sharing group of the result, i.e., the set of ground variables. K is defined as follows:

- If $x \in L$,

$$\begin{aligned} K = {} & \mathrm{bin}(X_t^{=\infty}, X_x^+, X_{xt}^*) \cup \\ & \mathrm{bin}(X_t \cup \{\emptyset\}, X_{xt}^{>1}, X_x^+, X_{xt}^*) \cup \\ & \mathrm{bin}(\{\{o\} \cup (\cup Z) \mid o \in X_t^{\in\mathbb{N}}, Z \subseteq X_x, 1 \leqslant |Z| \leqslant \chi_M^L(o,t)\}, (X_{xt}^{=1})^*) \cup \\ & (X_{xt}^U)^+. \end{aligned} \tag{6}$$

- If $x \notin L$,

$$\begin{aligned} K = {} & \mathrm{bin}(X_t^{>1} \cup X_{xt}^{>1}, X_x \cup X_{xt}, X^*) \cup \\ & \mathrm{bin}((X_t^{=1})^+, X_x \cup X_{xt}^{=1}, (X_{xt}^{=1})^*) \cup \\ & (X_{xt}^{=1})^+. \end{aligned} \tag{7}$$

Finally, the set L' of linear variables which are not ground is

$$L' = \begin{cases} L \setminus (vars(X_x \cup X_{xt}) \cap vars(X_t \cup X_{xt})) & \text{if } (S, L) \text{ is linear for } x \text{ and } t, \\ L \setminus vars(X_x \cup X_{xt}) & \text{otherwise, if } (S, L) \text{ is linear for } x, \\ L \setminus vars(X_t \cup X_{xt}) & \text{otherwise, if } (S, L) \text{ is linear for } t, \\ L \setminus vars(X) & \text{otherwise.} \end{cases} \tag{8}$$

Theorem 5.16 (*Optimality of* mgu_{sl})
The operator mgu_{sl} in Definition 5.15 is correct and optimal w.r.t. mgu, when $vars(x/t) \subseteq U$.

Example 5.17
Let $S = \{\emptyset, xv, xy, zw\}, L = \{v, w, x, y\}$, and $U = \{v, w, x, y, z\}$ and consider the binding $x/f(y, z)$. It is easy to check that (S, L) is linear for x but not for t. Applying our operator, we obtain $\mathrm{mgu}_{sl}([S, L, U], x/f(y, z)) = [S', L', U]$ with $S' = \{\emptyset, xy, vwxyz, vwxz\}$ and $L' = \{w\}$. This is more precise than the operators for $\mathtt{Sharing} \times \mathtt{Lin}$ in Hans and Winkler (1992). Actually, even using the optimizations proposed in Howe and King (2003) and Hill *et al.* (2004), one obtains as result the object

$$[\{vxy, vwxz, xy, wxyz, vwxyz\}, \{w\}, U].$$

The optimization proposed in Bagnara *et al.* (2005) is not applicable as it is, since it requires $vars(rel(S, x))$ and $vars(rel(S, f(y, z)))$ to be disjoint. Even assuming that this test for independence may be removed as unnecessary, the final result would be the same as above. In both cases, our operator is able to prove that vxy and $wxyz$ are not possible sharing groups.

Note that in a domain for rational trees, the sharing group vxy is needed for correctness, since the unification of $\{x/f(f(v, y), c), z/w\}$ with the binding $x/f(y, z)$ succeeds with $\{x/f(f(v, y), c), z/c, w/c, y/f(v, y)\}$. This means that we are able to exploit the occur-check of the unification in finite trees. As a consequence, our abstract unification operator is not correct w.r.t. a concrete domain of rational substitutions (King 2000). □

An alternative would be to compute the abstract unification following Theorem 5.9 with χ_M and $\uplus$ replaced by χ_M^L and $\cup$ respectively. (We can obviously ignore the delinearization operator $(\_)^2$, since $B \cup B = B$.) However, we do not pursue further this approach.

In the case $vars(x/t) \nsubseteq U$, we may proceed as for $\mathtt{ShLin}^\omega$ and $\mathtt{ShLin}^2$: enlarge the set of variables of interest in order to include all $vars(x/t)$, and compute unification with mgu_{sl}.

Definition 5.18 (*Abstract unification algorithm with extension in* ShLin)
Given $[S, L, U] \in \mathtt{ShLin}$ and the binding x/t, let $V = \{v_1, \ldots, v_n\}$ be $vars(x/t) \setminus U$. We define

$$\mathrm{mgu}_{sl}([S, L, U], x/t) = \mathrm{mgu}_{sl}([S \cup \{v_1, \ldots, v_n\}, L \cup V, U \cup V], x/t).$$

Theorem 5.19 (*Optimality of* mgu_{sl} *with extension*)
The operator mgu_{sl} in Definition 5.18 is the optimal abstraction of mgu.

Although the abstract operator mgu_{sl} is optimal for the unification with a single binding, the optimal operator for a multibinding substitution cannot be obtained by considering one binding at a time. This is a consequence of the fact that the corresponding operator for single-binding unification on $\mathtt{ShLin}^2$ cannot be extended to an optimal multibinding operator by simply considering one binding at a time.

In fact, all the counterexamples in Section 5.3 are also counterexamples for mgu_{sl}, since it is the case that $[S]_U = \gamma_{sl}(\alpha_{sl}([S]_U))$.

6 Optimal unification in practice

In this section, we give some evidence that there are practical advantages in using the optimal unification operators for `ShLin`. It is far beyond the scope of this paper to provide an experimental evaluation of the new algorithms, but the results in Bagnara *et al.* (2005) give some hints on its possible outcome. Bagnara *et al.* (2005) introduced an improvement for `Sharing` × `Lin` × `Free`, exploiting some ideas from King's unification operator for the domain $\mathtt{ShLin}^2$. In this way, they improved precision in a few cases and showed that efficiency of the analysis is more likely to be increased than decreased. In fact, even if the final result of the analysis does not change, a more precise operator may reduce the number of sharing groups in the intermediate steps, which helps performance. Hence, we expect the optimal unification for `ShLin` to further improve the analysis, in both efficiency and precision. This is more evident if we consider that Bagnara *et al.* (2005) measured precision in terms of the number of independent pairs (as well as definitively ground, free, and linear variables) and did not consider set-sharing. However, Bueno and García de la Banda (2004) showed that set-sharing information may be useful in several application of the analysis, such as parallelization of logic programs. Hence, a greater improvement in precision is to be expected if we consider the full set-sharing property.

We now provide a concrete example of a simple program in which our abstract operators give better results than the operators known in the literature.

6.1 An example: Difference lists

We work with *difference lists*, an alternative data structure to lists for representing a sequence of elements. A difference list is a term of the kind $A \setminus B$, where A and B are lists, which represents the list obtained by removing B from the tail of A. For example, using `PROLOG` notation for lists, $[1,2,3,4] \setminus [3,4]$ represents the list $[1,2]$, while $[1,2,3|x] \setminus x$ and $[1,2,3] \setminus []$ represent the list $[1,2,3]$. The difference lists whose tails are variables (such as $[1,2,3|x] \setminus x$) are mostly useful, since they can be concatenated in constant time. An overview of difference lists may be found in Sterling and Shapiro (1994).

We define the predicate *difflist*/3, which translates lists to difference lists and vice versa. The goal $\leftarrow \mathit{difflist}(l,h,t)$ succeeds when the difference list $h \setminus t$ represents the standard list l. For example, $\mathit{difflist}([],x,x)$ and $\mathit{difflist}([1,2,3],[1,2,3|x],x)$ succeed without any further instantiation of variables. In order to improve the precision of the analysis, we keep head and tail of difference lists in separate predicate arguments. The code for *difflist*/3, in head normal form, is the following:

$$\mathit{difflist}(l,h,t) \leftarrow l = [], h = t.$$
$$\mathit{difflist}(l,h,t) \leftarrow l = [x|l'], h = [x|h'], \mathit{difflist}(l',h',t).$$

where l, l' (list), h, h' (head), t (tail), and x are variables. We informally compute the goal-independent analysis of *difflist* on the domain ShLin, which gives

$$[\![difflist]\!]^0 = [\{\emptyset\}, \{l, h, t\}, \{l, h, t\}],$$
$$[\![difflist]\!]^1 = [\{\emptyset, ht, hl\}, \{l, h, t\}, \{l, h, t\}],$$
$$[\![difflist]\!]^2 = [\![difflist]\!]^1.$$

The result of the analysis is not affected by our improved unification operator: the standard mgu for ShLin, as given in Hans and Winkler (1992), yields exactly the same result. Now, suppose we want to analyze the goal $\leftarrow difflist(l, h, h)$. This corresponds to the goal $\leftarrow difflist(l, h, t), h = t$ in head normal form. Its semantics may be computed, using our operators, as

$$\text{mgu}_{sl}([\{\emptyset, ht, hl\}, \{l, h, t\}, \{l, h, t\}], h/t) = [\{\emptyset, ht\}, \{l\}, \{l, h, t\}].$$

By projecting over l and h, we get $[\{\emptyset, h\}, \{l\}, \{l, h\}]$. Hence, the analysis is able to infer that upon exiting the goal $\leftarrow difflist(l, h, h)$, the variable l is ground.

By using the standard mgu for ShLin in Hans and Winkler (1992), we get

$$[\{\emptyset, ht, htl\}, \{l\}, \{l, h, t\}]; \tag{9}$$

hence l is detected to be linear but not ground. The optimizations introduced in Howe and King (2003), Hill *et al.* (2004) and Bagnara *et al.* (2005) do not improve this result. This is a consequence of the fact that these optimizations have been developed to be correct also for rational trees. In this case, you cannot infer that l is ground after $\leftarrow difflist(l, h, h)$, since the substitution in rational solved form $\{l/[v], h/[v|h]\}$ is a correct answer for the same goal.

If we perform the analysis in $\texttt{ShLin}^2$, using our operators we have $[\![difflist]\!] = [\{\emptyset, hl, ht\}]_{lht}$ and the result for the goal $\leftarrow difflist(l, h, h)$ is $[\{\emptyset, h\}]_{lh}$. However, by using the original operator in King (1994), the semantics of *difflist* does not change, but the result for the goal $\leftarrow difflist(l, h, h)$ is $[\{\emptyset, h^\infty, h^\infty l^\infty\}]_{lh}$; thus l is not proven to be either ground or linear.

The fact that optimal operators improve groundness information is somehow surprising. Generally, one expects that groundness affects aliasing analysis, but not vice versa. In fact, it is well known that Sharing is a refinement (Cortesi *et al.* 1997) of the domain Def. However, as far as groundness is concerned, the precision of Sharing and Def is the same; i.e., the other objects included in Sharing do not improve groundness analysis (Cortesi *et al.* 1998). As far as we know, there is no abstract unification operator in the literature, for a domain dealing with sharing, freeness, and linearity, which is more precise than Def for groundness. On the contrary, the example above shows that ShLin, endowed with the optimal unification, improves over Def. Amazingly, in this example ShLin is even better than Pos (Armstrong *et al.* 1994). In the latter, the abstract semantics of *difflist* is $h \leftrightarrow (l \wedge t)$, i.e., h is ground iff both l and t are ground. The result of the analysis for the goal $\leftarrow difflist(l, h, h)$ is $\exists_t\ (h \leftrightarrow (l \wedge t) \wedge h \leftrightarrow t)$. This is equivalent to $h \rightarrow l$ which does not imply groundness of l. Actually, $h \rightarrow l$ is the groundness information which may be inferred by (9).

6.2 Another example for ShLin²

As far as we know, there is no implementation or experimental evaluation of the domain $\texttt{ShLin}^2$. We think that it would be worthwhile to give such an implementation and that there is some evidence that $\texttt{ShLin}^2$ improves over ShLin also in practice. For instance, we show a simple program in which King's domain is more precise than ShLin with optimal operators.

We provide a variant of the predicate *difflist*/3, which we call *difflist'*/2, with only two arguments—head and tail of the difference list are encoded in the second argument as the term $head \setminus tail$:

$$\begin{aligned}&\mathit{difflist}'(l,d) \leftarrow l = [], d = h \setminus h.\\&\mathit{difflist}'(l,d) \leftarrow l = [x|l'], d = [x|h] \setminus t, d' = h \setminus t, \mathit{difflist}'(l',d').\end{aligned}$$

We informally compute the goal-independent analysis of *difflist'* on the domain ShLin, which gives

$$\begin{aligned}[\![\mathit{difflist}']\!]^0 &= [\{\emptyset\},\{d,l\},\{d,l\}],\\ [\![\mathit{difflist}']\!]^1 &= [\{\emptyset, dl, d\},\{l\},\{d,l\}],\\ [\![\mathit{difflist}']\!]^2 &= [\![\mathit{difflist}']\!]^1.\end{aligned}$$

The same analysis, computed over $\texttt{ShLin}^2$, gives

$$\begin{aligned}[\![\mathit{difflist}']\!]^0 &= [\{\emptyset\}]_{dl},\\ [\![\mathit{difflist}']\!]^1 &= [\{\emptyset, dl, d, d^\infty\}]_{dl},\\ [\![\mathit{difflist}']\!]^2 &= [\![\mathit{difflist}']\!]^1.\end{aligned}$$

Now, suppose we want to analyze the goal $\leftarrow \mathit{difflist}'(l,d), d = [x_1,x_2|h] \setminus t$, which extracts the first two elements from the difference list d. In ShLin we have the following:

$$\begin{aligned}&\mathrm{mgu}_{sl}([\{\emptyset, dl, d\},\{l\},\{d,l\}], d/[x_1,x_2|h] \setminus t)\\&\qquad = [\{\emptyset\} \cup \mathrm{bin}(\{dl,d\},\{x_1,x_2,h,t\}^*),\{l\},\{d,l,x_1,x_2,h,t\}].\end{aligned}$$

Note that the sharing group dlx_1x_2 is part of the result. If we repeat the analysis in $\texttt{ShLin}^2$, we have

$$\begin{aligned}&\mathrm{mgu}_2([\{\emptyset, dl, d, d^\infty\}]_{dl}, d/[x_1,x_2|h] \setminus t) = \big[\{\emptyset, dlx_1, dlx_2, dlh, dlt\} \cup\\&\qquad \downarrow\{\textstyle\biguplus X \mid X \in \wp(\{d^\infty x_1^\infty, d^\infty x_2^\infty, d^\infty h^\infty, d^\infty t^\infty\})\}\big]_{dlx_1x_2ht}.\end{aligned}$$

This result does not contain the sharing group dlx_1x_2.

Generally speaking, it is easier to analyze the predicate *difflist*/3 than *difflist'*/2. Codish *et al.* (2000) proposed a method named *untupling* which is able to automatically recover *difflist*/3 from *difflist'*/2.

7 Related work

In this paper, we work with a concrete domain of substitutions on finite trees. In the literature, some authors deal with rational trees.

Since any correct operator for rational trees is also correct for finite trees, we can compare the unification operators for rational trees with ours. (Of course, this is not entirely fair as far as the precision is concerned.) The opposite is not true, since an abstract unification operator for finite trees may be able to exploit the occur-check condition. We have shown in Example 5.17 that our optimal operator can exploit the occur-check condition, and thus it is not correct for rational trees.

7.1 Sharing

It is well known that the abstract unification operator of the domain Sharing alone (i.e., without any freeness or linearity information) is optimal. Cortesi and Filé (1999) gave a formal proof of optimality, considering a slightly different unification operator with two abstract objects and a concrete substitution. Since the two abstract objects are renamed apart, it is equivalent to consider a single abstract object. The basic idea underlying the proof is to exhibit, for each sharing group in the result of the unification, a pair of concrete substitutions generating the resulting sharing group. We follow the same constructive schema in the proof of optimality for $\mathtt{ShLin}^{\omega}$ (but we look for a single substitution, due to the different concrete operator). Instead, to prove optimality for ShLin and $\mathtt{ShLin}^2$, we use a direct approach and show that the abstract unification operator corresponds to the best correct abstraction (i.e., $\alpha \circ \mathrm{mgu}_{\omega} \circ \gamma$) of the unification on $\mathtt{ShLin}^{\omega}$ with simple (although tedious) algebraic manipulations.

A different unification operator has been proposed in Amato and Scozzari (2002, to appear) for goal-dependent analysis of Sharing. In this paper, the standard unification operator is split into two different operators for forward and backward unification. Both operators are proved to be optimal, and the overall analysis is strictly more precise than the analysis performed on Sharing equipped with the standard operator.

As far as we know, these are the only optimality results for domains encoding aliasing properties.

7.2 Sharing $\times$ Lin

In most of the work combining sharing and linearity, freeness information is included in the abstract domain. In fact, freeness may improve the precision of the aliasing component, and it is also interesting by itself, for example, in the parallelization of logic programs (Hermenegildo and Rossi 1995). In this comparison, we do not consider the freeness component.

The first work which combines set-sharing with linearity is that of Langren (1990), followed by that of Hans and Winkler (1992). The initial unification algorithm has been improved by Howe and King (2003) and Hill *et al.* (2004) by removing an independence test. This increases the number of cases in which linearity information may be exploited. Bagnara *et al.* (2005) proposed a different improvement, adopting an idea by King (1994) for the domain $\mathtt{ShLin}^2$, which simplifies the unification of a linear term with a nonlinear one. Example 5.17 shows that even adopting all these improvements, we still obtain a strictly more precise operator. Since our operator is optimal, any further improvement is now impossible.

Bagnara *et al.* (2002) showed that if we are only interested in pair-sharing information, Sharing is redundant. They proposed a new domain SS^{ρ} which is obtained by discharging redundant sharing groups. A sharing group B in a set S is redundant if $|B| > 2$ and $\forall x, y \in B.\ \exists C \in S.\ \{x, y\} \subseteq C \subset B$. Analyses performed with SS^{ρ} are shown to be as precise as those performed with Sharing, if only pair-sharing information is required. Hill *et al.* (2004) introduced the domain $SS^{\rho} \times$ Lin $\times$ Free. Example 5.17 shows that our operator is still more precise (of course, without considering the freeness component) because of the sharing group vxy which does not appear in S' and is not redundant for SS^{ρ}. In any case, Bueno and García de la Banda (2004) have shown that classical applications of sharing analyses, such as parallelization of logic programs, are able to exploit information which is encoded in Sharing $\times$ Free but not in $SS^{\rho} \times$ Free.

An alternative presentation of Sharing $\times$ Lin, based on *set logic programs*, has been introduced by Codish *et al.* (2000). However, the proposed operators are not optimal, as shown in Hill *et al.* 2004.

The domain $\mathtt{ShLin}^2$ was introduced by King (1994), which provides correct operators for abstract unification. However, these operators are not optimal, as Examples 5.13 and 5.14 show.

7.3 ASub

An alternative approach to aliasing analysis is to only record sharing between pairs of variables (and possibly linearity and groundness information). The best known domain of this category is ASub, introduced by Søndergaard (1986) and formalized by Codish *et al.* (1991). The domain ASub is the reduced product of pair-sharing, Lin and Con (Jones and Søndergaard 1987), which is the simplest domain for definite groundness. Recently, King (2000) reformulated the proofs in order to work with rational trees. Moreover, King's algorithms are parametric w.r.t. the groundness domain, allowing the replacement of Con with more precise domains such as Def and Pos.

The domain Sharing $\times$ Lin is strictly more precise than ASub, since it embeds more groundness information (equivalent to Def) and set-sharing information. Since our operator for Sharing $\times$ Lin is optimal, we are sure that the analyses performed in Sharing $\times$ Lin are strictly more precise than those in ASub.

The following is a counterexample to the optimality of the abstract unification in King (2000), in the case of finite trees, when pair-sharing is equipped with Def or Pos.

Example 7.1

Consider the object $\kappa = (x \leftrightarrow y, \{xy\})$, where the first component is a formula of Def and Pos and $\{xy\}$ is the set of pairs of variables which may possibly share. In this domain, linearity information is embedded in the second component in the following way: If v is not linear, then vv must be included in the second component. Thus, both x, y, and z are linear in $(x \leftrightarrow y, \{xy\})$. We want to unify κ with $x/f(y,z)$. By using the algorithm (King 2000), we obtain $(y \leftrightarrow x \wedge x \rightarrow z, \{xy, xz, yz, xx, yy\})$. However, in Sharing $\times$ Lin we may represent κ with $[S, L, U] = [\{xy, z\}, \{x, y, z\}, \{x, y, z\}]$ and $\text{mgu}_{sl}([S, L, U], x/f(y,z)) = [\{xy\}, \{z\}, \{x, y, z\}]$ which proves that z is ground. □

Actually, King (2000) did not state explicitly how to compute the groundness component of the result, although he said that it must be computed before the linearity and pair-sharing components, in order to improve precision. However, it seems safe to assume that the author's intention was to compute the groundness component using the abstract operators already known and therefore independently from the pair-sharing component. This is what makes our operator more precise, since linearity information may help in tracking ground variables when working over finite trees.

7.3.1 Alternating paths

The domain ASub and its derivatives (King 2000) use the concept of *alternating path*. Alternating paths may seem the counterparts, for pair-sharing, of sharing graphs. We now investigate this idea, and show to what extent this correspondence is faithful.

We call *carrier graph* a special graph defined by a set of equations E. Each distinct occurrence of a variable in E is a node. Edges in the carrier graphs can be of two types:

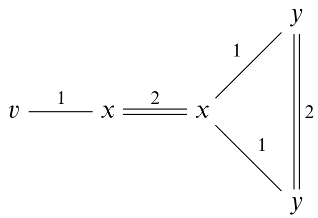
- edges of *type one* between two variable occurrences if the occurrences are on opposite sides of a single equation in E and
- edges of *type two* linking two (distinct) occurrences of the same variable.

An alternating path is a sequence of edges of alternating type over the carrier graph.

Alternating paths in ASub (and derivatives) are used to prove correctness of the abstract unification operators. For example, they are used to prove Proposition 3.1 in King (2000). Sharing graphs are used in this paper to prove Theorem 4.31, which is the starting point to prove correctness and optimality of the unification algorithms for ShLin^2 and ShLin. However, sharing graphs are also used to compute the abstract unification in ShLin^{ω}. Even if alternating paths are not used, in the literature, for computing abstract unification, they could. For any object of pair-sharing o, which is a set of pairs of variables, consider any substitution θ in the concretization of o. Then, the object o is an abstraction of the set of alternating paths in θ. More precisely, it represents all the paths which start and end with edges of type one, which we call *admissible paths*. They are abstracted by considering only the start and end variables. In order to unify o with the binding x/t, we build a carrier graph with all the occurrences of variables in o and x/t. For each pair of variables in o, we add an edge of type one. We add edges of types one and two for the binding x/t, as explained above. Finally, we add all the type two edges between any occurrence in x/t and any occurrence of the same variable in o. We consider all the admissible paths over the graph so obtained. It is not difficult to check that the result of the unification algorithm for pair-sharing in King (2000), without any additional groundness domain, is the set of all the start and end variables for all these admissible paths.

Example 7.2

Let $S = \{xv\}$ be the set of pairs of variables which share, and consider the binding $x/r(y, y)$. We obtain the carrier graph



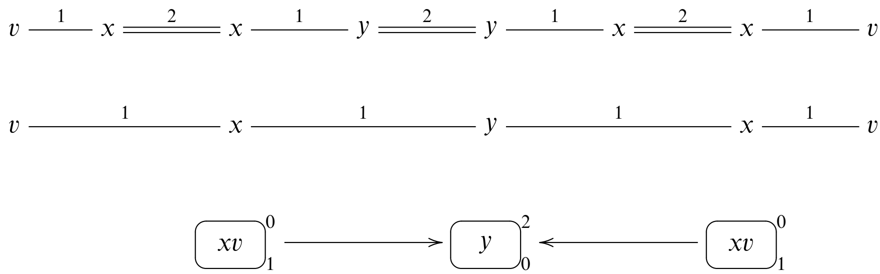
which gives origin to several alternating paths. Among them, there is an admissible path from v to v, which proves that v is not linear after the unification. □

The first difference between alternating paths and sharing graphs is that all the alternating paths are subgraphs of the same carrier graph, while each sharing graph has a different structure, with a different set of nodes. The second difference is that the information coming from the abstract object and the binding is encoded in a different way. For instance, consider the set $S = \{xy\}$ and the binding x/z. We obtain a carrier graph with four nodes x, y, x, z, two edges $x \overset{1}{-\!\!-\!\!-} y$ and $x \overset{1}{-\!\!-\!\!-} z$ of type one, and an edge $x \overset{2}{=\!\!=\!\!=} x$ of type two. Therefore, the sharing information coming from the initial pair-sharing and the binding is treated symmetrically and is entirely encoded on the edges. Performing unification on the carrier graph boils down to devising the alternating paths on the graph. On the contrary, each sharing graph has a set of nodes labeled by xy, x, and z, with suitable multiplicities. The labels of the nodes encode the initial pair-sharing information, while the binding affects the multiplicity of nodes. The process of unification consists of adding the necessary arrows to get a sharing graph.

If we consider a single alternating path in a carrier graph and the sharing graph for the same pair-sharing information and the same binding, they are obviously related, although not in a straightforward manner. Consider an admissible path and delete all type two edges, collapsing in a single node their start and end nodes. (Type two edges are used in the carrier graph to avoid the creation of invalid paths, but in a single alternating path they do not add information.) Then, each type one edge coming from the initial pair-sharing information corresponds to a node in the sharing graph, while a type one edge coming from the binding becomes an arrow in the sharing graph.

Example 7.3

Consider Example 7.2. We depict the (admissible) alternating path from v to v, its collapsed version, and the corresponding sharing graph:



Note that while in the carrier graph, nonlinearity of the variable x is handled by duplicating the variable y which occurs twice, in alternating paths without type two nodes, the duplicated variables are x and v, which are connected to y. The same holds in the sharing graph, where we have only one node labeled by y and two nodes labeled by xv. □ □

In sharing graphs we also require the multiplicities of a node to be equal to its in- and out-degrees. This makes handling groundness at the same level as sharing and linearity, without requiring a separate domain, possible. Remember that a sharing group S with multiplicity n corresponds, in the concrete domain, to a variable u such that $\theta^{-1}(u) = S$. If the degree of the node labeled with this sharing group is not n, this means that one of the occurrences of u is bound to a ground term. This would make ground the entire connected component containing S. Hence, in order to correctly and precisely propagate groundness, we just forbid this kind of sharing graphs. On the contrary, the pair-sharing algorithm in King (2000), which focuses on a single path in the carrier graph, is not able to extract groundness information without the help of an auxiliary domain.

7.4 Lagoon and Stuckey's domain

Lagoon and Stuckey (2002) have recently proposed a different approach to pair-sharing analysis. The authors use multigraphs, called *relation graphs*, to represent sharing and linearity information. The nodes of the multigraph are variables, and two of them may share only if there is a *traversable path* from one variable to the other. Intuitively, each binding generates edges of different types. The definition of traversable paths is very similar to that of alternating paths. A traversable path is a sequence of edges, such that contiguous edges are always of different types.

This domain should be coupled with a groundness domain, and operators are parametric w.r.t. the latter one. The authors show that relation graphs, when coupled with the `Def` groundness domain, are more precise than `Sharing` and `ASub`. However, this is not the case for `Sharing` × `Lin`, at least in the case of finite trees, since the operators in Lagoon and Stuckey (2002) are not able to use linearity to improve the precision of the groundness component.

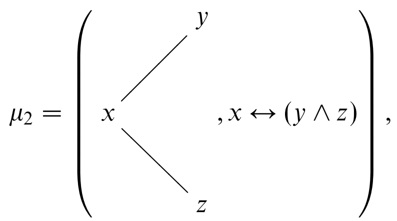
Example 7.4

As shown in Example 7.1, if we unify $[S, L, U] = [\{xy, z\}, \{x, y, z\}, \{x, y, z\}]$ with the binding $x/f(y, z)$, we obtain $\mathrm{mgu}_{sl}([S, L, U], x/f(y, z)) = [\{xy\}, \{z\}, \{x, y, z\}]$, proving that z is ground after the unification. In the domains $\Omega_{\mathtt{Def}}$ and $\Omega_{\mathtt{Pos}}$ of Lagoon and Stuckey (2002), the abstract object corresponding to $[S, L, U]$ is

$$\mu_1 = \big(x \;\text{———}\; y \;, x \leftrightarrow y\big)\,.$$

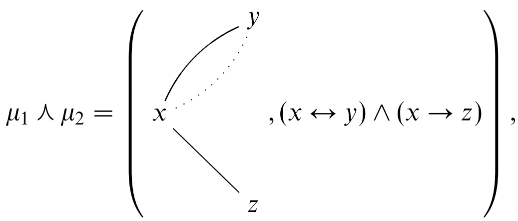
Intuitively, the first element of μ_1 encodes the sharing information, namely, that x and y may share (while z does not share either with x or with y). The second element of μ_1 is an element of `Pos` (and also of `Def`) and denotes the groundness information that x is ground iff y is ground.

The unification of μ_1 with $x/f(y, z)$ in $\Omega_{\mathtt{Pos}}$ is realized by abstracting the substitution and composing the two abstract object. The abstraction of $x/f(y, z)$ is



The first element says that x shares with both y and z, while y and z do not share. The second element says that x is ground iff both y and z are ground.

The abstract conjunction is



where edges drawn in different styles are compatible, namely, that they come from different bindings. From this result, it is not possible to infer that z is ground after the unification. □

In the actual implementation, Lagoon and Stuckey (2002) used another representation for their domain. Each pair of variables is annotated with a formula denoting the groundness models under which the corresponding pair-sharing may occur. For example, a pair uv annotated with the formula $\bar{u} \wedge \bar{v} \wedge \bar{w} \wedge \bar{z}$ means that u and v may share only if none of u, v, w, z is ground. We conjecture that this domain may be embedded in King's `ShLin`[2]. The next example shows how to perform this embedding.

Example 7.5
We consider the example in Figure 4 in Lagoon and Stuckey (2002). The variables of interest are u, v, w, z:

$$
\begin{array}{lllllll}
uw & : & \bar{u} \wedge \bar{w} & \quad & uz & : & \bar{u} \wedge \bar{z} \\
vz & : & \bar{v} \wedge \bar{z} & & uu & : & \bar{u} \wedge \bar{w} \wedge \bar{z} \\
uv & : & \bar{u} \wedge \bar{v} \wedge \bar{w} \wedge \bar{z} & & vv & : & \bar{v} \wedge \bar{w} \wedge \bar{z} \\
wz & : & \bar{w} \wedge \bar{z} & & vw & : & \bar{v} \wedge \bar{w}
\end{array}
$$

For instance, $uv : \bar{u} \wedge \bar{v} \wedge \bar{w} \wedge \bar{z}$ means that u and v may share only if u, v, w, z are not ground, while $uu : \bar{u} \wedge \bar{w} \wedge \bar{z}$ means that u is (possibly) not linear only if u, w, z are not ground. Each of these formulas may be viewed as a condition over 2-sharing groups. For example $uv : \bar{u} \wedge \bar{v} \wedge \bar{w} \wedge \bar{z}$ means that every 2-sharing group which contains u and v should also contain w and z, while $uu : \bar{u} \wedge \bar{w} \wedge \bar{z}$ means that each 2-sharing group in which u is nonlinear should also contain w and z. In order to find the object of $\mathtt{ShLin}^2$ which corresponds to this example, it is enough to collect all the 2-sharing groups which satisfy all the conditions enforced by the formulas. In this case, we get $\downarrow\{u^\infty v^\infty wz, u^\infty wz, v^\infty wz, uw, vz, wz, uz, vw, u, v, w, z\}$. □

7.4.1 Traversable paths

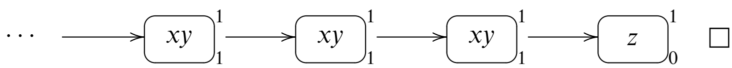
The idea behind traversable paths is very similar to the concept of alternating path, and relation graphs are quite similar to carrier graphs. From a carrier graph, we can obtain a relation graph by removing type two edges and introducing a different type of edge for each binding. This works because the use of nonlinear terms is forbidden: a binding like $x/r(y, y)$ has to be replaced by two bindings $x/r(y, z)$ and y/z. However, the main difference w.r.t. traditional pair-sharing (and also $\mathtt{ShLin}^\omega$) is that Lagoon and Stuckey (2002) did not abstract traversable paths to the set of pairs of variables, but they kept in the abstract object the set of all the edges generated during the unification process. In this way, they are able to record that in order for two variables x and y to share, the only possible path touches another variable z. Hence, if z is ground, x and y cannot share: in this way they recovered pair-sharing dependence information which would be lost otherwise.

We could follow the same approach and use multilayer sharing graphs (namely, sets of sharing graphs over the same set of nodes, where each layer represents the unification with a single binding) as abstract objects, without collapsing them to sharing groups. We do not think this would improve precision of the domain very much, since a sharing group is already a much more concrete abstraction of a graph w.r.t. the set of all the connected pairs of variables. In fact, already `Sharing` can encode the information that grounding a certain variable z, two variables x and y become independent. Moreover, in the Example 7.5 we have shown that relation graphs may be encoded into $\mathtt{ShLin}^2$.

7.5 Rational trees

In the recent years, many authors have studied the behavior of logic programs on *rational trees* (King 2000; Hill *et al.* 2004), which formalize the standard implementations of logic languages. We have shown that our operators, which are optimal for finite trees, are not correct for rational trees, since they exploit the occur-check to reduce the sharing groups generated by the abstract unification (see Example 5.17). It would be interesting to adapt our framework to work with rational trees, in order to obtain optimal operators also in this case. Since a rational tree may contain infinite occurrences of a variable, the notion of ω-sharing group needs to be extended in order to allow infinite exponents. Also, we need to consider infinite sharing graphs (or, at least, a representation of them) and find suitable regularity conditions for them, analogous to the regularity conditions on rational trees.

Example 7.6
Consider the set of ω-sharing groups $S = \{xy, z\}$ and the binding $x/r(z, y)$. On rational trees, unifying $\delta = \{x/y\}$ (such that $[S]_{xyz} \rhd [\delta]_{xyz}$) with $x/r(z, y)$ would get the substitution $\{x/r(z, x), y/r(z, y)\}$ in rational solved form. This, intuitively, corresponds to the sharing group $x^{\omega} y^{\omega} z$ in which the exponent ω denotes an infinite number of occurrences. A possible (infinite) sharing graph generating this sharing group is the following:



Although the structure of abstract objects and operators for adapting $\texttt{ShLin}^{\omega}$ to work with rational trees is more complex, we expect the optimal abstract operators for rational trees on $\texttt{ShLin}^2$ and $\texttt{Sharing} \times \texttt{Lin}$ to be simpler than those presented here for finite trees. This is because we do not need to worry about the occur-check condition (embedded in our unification operator) and infinite multiplicities.

8 Conclusion and future works

We summarize the main results of this paper:

- We define a new domain $\texttt{ShLin}^{\omega}$ as a general framework for investigating sharing and linearity properties and provided the optimal unification operator.
- We show that $\texttt{ShLin}^{\omega}$ is a useful starting point for studying further abstractions. We obtain the optimal operators for single-binding abstract unification in $\texttt{Sharing} \times \texttt{Lin}$ and $\texttt{ShLin}^2$, and we show that these are strictly more precise than all the other operators in the literature for the same domains.
- We show, for the first time, an optimality result for a domain which combines aliasing and linearity information.

Moreover, as a negative result, we prove that the standard schema of the iterative unification algorithm (one binding at a time) does not lead to optimal operators for the domains $\texttt{ShLin}^2$ and $\texttt{Sharing} \times \texttt{Lin}$. As a side result, we show that $\texttt{ShLin}$ and $\texttt{ShLin}^2$ with optimal operators may be more precise than $\texttt{Pos}$ for groundness analysis.

Several things remain to be explored: First of all, we need to study the impact on the precision and performance obtained by adopting the new optimal operators and domains. We plan to implement the operators on $\texttt{ShLin}^2$ and $\texttt{Sharing} \times \texttt{Lin}$ within the CiaoPP static analyzer (Bueno *et al.* 1997). Moreover, we plan to analyze the domain $SS^\rho \times \texttt{Lin}$ (Bagnara *et al.* 2002) in our framework and, possibly, to devise a variant of $\texttt{ShLin}^2$ which enjoys a similar closure property for redundant sharing groups. This could be of great impact on the efficiency of the analysis. Last but not least, we plan to translate our framework to the case of unification over rational trees.

Appendix A

Proofs of Section 4

In this section we give the proofs of the main results of the paper.

Theorem 4.6
The relation $\rhd$ is well defined.

Proof
It is enough to prove that $\{{\theta_1}^{-1}(v)|_U \mid v \in \mathcal{V}\} = \{{\theta_2}^{-1}(v)|_U \mid v \in \mathcal{V}\}$ when $\theta_1 \sim_U \theta_2$. Assume that $\theta_1 \sim_U \theta_2$; then by definition of $\sim_U$ there exists a renaming ρ such that $\rho(\theta_1(u)) = \theta_2(u)$ for each $u \in U$. Given $S = \theta_1^{-1}(v)|_U$, if $w = \rho(v)$ we have $\theta_2^{-1}(w)|_U = \theta_1^{-1}(v)|_U = S$. This concludes the proof. □

Proposition 4.13
Given a substitution θ, a variable v, and a term t, we have that $\chi(\theta^{-1}(v), t) = occ(v, \theta(t))$. Moreover, given a set of variables U, when $vars(t) \subseteq U$, it holds that $\chi(\theta^{-1}(v)|_U, t) = occ(v, \theta(t))$.

Proof
Let $B = \theta^{-1}(v)$. The proof is by induction on the structure of the term t. If $t \equiv a$ is a constant, then $occ(v, \theta(a)) = occ(v, a) = 0$ which is equal to $\chi(B, a)$, since $occ(w, a) = 0$ for each $w \in \mathcal{V}$. If $t \equiv w$ is a variable, then $occ(v, \theta(w)) = \theta^{-1}(v)(w) = B(w)$. At the same time, $\chi(B, t) = B(w)$, since $occ(w, w) = 1$ and $occ(y, w) = 0$ for $y \not\equiv w$. For the inductive case, if $t \equiv f(t_1, \ldots, t_n)$, we have $occ(v, t) = \sum_{i=1}^n occ(v, t_i) = \sum_{i=1}^n \chi(B, t_i)$ by inductive hypothesis. Moreover

$$\chi(B, t) = \sum_{v \in \llbracket B \rrbracket} (B(v) \cdot \sum_{i=1}^n occ(v, t_i)) = \sum_{i=1}^n \sum_{v \in \llbracket B \rrbracket} B(v) \cdot occ(v, t_i) = \sum_{i=1}^n \chi(B, t_i).$$

Let U be a set of variables with $vars(t) \subseteq U$. By definition, $\chi(\theta^{-1}(v)|_U, t) = \sum_{w \in \theta^{-1}(v)|_U} occ(w, t)$. Since $vars(t) \subseteq U$, for any $w \notin U$ it holds that $occ(w, t) = 0$, and thus $\chi(\theta^{-1}(v)|_U, t) = \chi(\theta^{-1}(v), t)$. □

Proposition 4.24
Given substitutions θ, $\eta \in ISubst$ and an ω-sharing group B, we have

$$(\eta \circ \theta)^{-1}(B) = \theta^{-1}(\eta^{-1}(B)).$$

Proof
Using the definitions and simple algebraic manipulations, we have

$$\begin{aligned}
&\theta^{-1}(\eta^{-1}(B))\\
&= \lambda w.\chi\big(\lambda v.\chi(B,\eta(v)),\theta(w)\big)\\
&= \lambda w.\sum_y \chi(B,\eta(y))\cdot occ(y,\theta(w))\\
&= \lambda w.\sum_y \left(\sum_x B(x)\cdot occ(x,\eta(y))\right)\cdot occ(y,\theta(w))\\
&= \lambda w.\sum_x B(x)\cdot \sum_y occ(x,\eta(y))\cdot occ(y,\theta(w))\\
&= \lambda w.\sum_x B(x)\cdot \sum_y \eta^{-1}(x)(y)\cdot occ(y,\theta(w))\\
&= \lambda w.\sum_x B(x)\cdot \chi(\eta^{-1}(x),\theta(w)).
\end{aligned}$$

By Proposition 4.13, we have that $\chi(\eta^{-1}(x),\theta(w)) = occ(x,\eta(\theta(w))$ and therefore

$$\theta^{-1}(\eta^{-1}(B)) = (\eta\circ\theta)^{-1}(B). \quad \square$$

Theorem 4.25 (*Correctness of* mgu_ω)
The operation mgu_ω is correct w.r.t. mgu, i.e.,

$$\forall [S]_U \in \mathtt{ShLin}^\omega, \delta \in \mathit{ISubst}.\ [S]_U \rhd [\theta]_U \implies \mathrm{mgu}_\omega([S]_U,\delta) \rhd \mathrm{mgu}([\theta]_U,\delta).$$

Proof
Given $[S]_U \rhd [\theta]_U$ and $\delta \in \mathit{ISubst}$, we need to prove that $\mathrm{mgu}_\omega([S]_U,\delta) \rhd \mathrm{mgu}([\theta]_U,\delta)$ or the equivalent property $\alpha_\omega(\mathrm{mgu}([\theta]_U,\delta)) \leqslant_\omega \mathrm{mgu}_\omega([S]_U,\delta)$.

Since mgu_ω is defined inductively on the number of bindings in δ, it is enough to prove that $\mathrm{mgu}_\omega([S]_U, x/t) \rhd \mathrm{mgu}([\theta]_U,\{x/t\})$ for a single binding x/t. Since composition of correct operators is still correct, it follows that multibinding unification is correct.

Moreover, when $vars(x/t) \nsubseteq U$, we exploit the identity $\mathrm{mgu}([\theta]_U,\{x/t\}) = \mathrm{mgu}(\mathrm{mgu}([\theta]_U,[\epsilon]_{vars(x/t)}),\{x/t\})$. When computing $\mathrm{mgu}([\theta]_U,[\epsilon]_{vars(x/t)})$ all the variables in $vars(x/t)\setminus U$ occurring in θ are renamed apart from x/t itself. Therefore each $v \in vars(x/t)\setminus U$ is free (hence linear) in $\mathrm{mgu}([\theta]_U,[\epsilon]_{vars(x/t)})$, i.e.,

$$\alpha_\omega(\mathrm{mgu}([\theta]_U,[\epsilon]_{vars(x/t)})) = \big[S \cup \{\{\!\{v\}\!\} \mid v \in vars(x/t)\setminus U\}\big]_{U\cup vars(x/t)}.$$

Therefore, it is enough to prove that $\mathrm{mgu}_\omega([S]_U, x/t) \rhd \mathrm{mgu}([\theta]_U,\{x/t\})$ when $vars(x/t) \subseteq U$. Let B be a sharing group in $\alpha_\omega(\mathrm{mgu}([\theta]_U,\{x/t\}))$, we prove that $B \in \mathrm{mgu}_\omega([S]_U, x/t)$.

If $B = \{\!\{\}\!\}$, we consider a multigraph G with only one node labeled by $\{\!\{\}\!\}$ and no edges. It is easy to check that G is a sharing graph for S (since $\{\!\{\}\!\} \in S$) and x/t and that $res(G) = \{\!\{\}\!\}$. Therefore, in the following we consider only the case $B \neq \{\!\{\}\!\}$.

The proof is composed of three parts: first, we look for a (special) substitution β obtained by renaming some variables in θ and such that β is still approximated by

S; second, we define a multigraph G exploiting the variables of β; third, we show that we can restrict G to a smaller sharing graph whose resultant ω-sharing group is exactly B.

First part. Without loss of generality, we assume that $\mathrm{dom}(\theta) = U$ (this is always possible since in any class $[\theta]_U$, there exists a substitution whose domain is exactly U). Let $\theta' = \mathrm{mgu}(\theta, \{x/t\}) = \eta \circ \theta$ with $\eta = \mathrm{mgu}(\{\theta(x) = \theta(t)\})$, and we have $[\theta']_U = \mathrm{mgu}([\theta]_U, [x/t]_U)$. Since $\mathrm{dom}(\theta) = U$, we have $vars(\eta) \cap U = \emptyset$. Consider η' obtained from η by replacing each occurrence of a variable in $\mathrm{rng}(\eta)$ with a different fresh variable. This means that there exists $\rho \in Subst$ mapping variables to variables such that $\rho(\eta'(x)) = \eta(x)$ for each $x \in \mathrm{dom}(\eta)$. Namely, we have

$$\rho = \{v_1/v_2 \mid \exists x \in \mathrm{dom}(\eta), \xi \in \Xi \text{ s.t. } \eta'(x)(\xi) = v_1 \wedge \eta(x)(\xi) = v_2\}.$$

Note that ρ is not a renaming, since it is not bijective. We now show that $\beta = \eta' \circ \theta$ has the property that $[S]_U \rhd [\beta]_U$. For any $C \in \alpha([\beta]_U)$, we may distinguish three cases:

- $C = \{\!\!\{\}\!\!\}$. In this case $C \in S$ by definition of $\mathtt{ShLin}^{\omega}$.
- $C = \beta^{-1}(w)|_U$ for $w \in \mathrm{rng}(\theta) \setminus \mathrm{dom}(\eta)$. In this case $occ(w, (\eta' \circ \theta)(v)) = occ(w, \theta(v))$ for each $v \in \mathcal{V}$; therefore $\beta^{-1}(w)|_U = \theta^{-1}(w)|_U \in S$.
- $C = \beta^{-1}(w)|_U$ for $w \in \mathrm{rng}(\eta')$. Hence there exists $v \in \mathrm{rng}(\theta)$ such that $occ(w, \eta'(v)) = 1$ and $occ(w, \eta'(v')) = 0$ for each $v' \notin \{v', w\}$. Hence, for each $u \in U$, $occ(w, \eta'(\theta(u))) = n$ iff $occ(v, \theta(u)) = n$, and this implies $C = \theta^{-1}(v)|_U \in S$.

Moreover $\rho(\beta(u)) = \theta'(u)$ for each $u \in U$; therefore $\theta' \sim_U \rho \circ \beta$.

Second part. Consider the labeled multigraph G such that $N_G = \{v \mid v \in vars(\beta(U))\}$, $l_G(v) = \beta^{-1}(v)|_U \in S$ and $E_G = \{\xi \mid \beta(x)(\xi) \in \mathcal{V}\}$. Note that if $\beta(x)(\xi) \in \mathcal{V}$, then $\beta(t)(\xi) \in \mathcal{V}$, too. Each position ξ in E_G is an arrow such that $\mathrm{src}_G(\xi) = \beta(x)(\xi)$ and $\mathrm{tgt}_G(\xi) = \beta(t)(\xi)$. Observe that the second condition in the definition of sharing graph for S and x/t is satisfied, since $[S]_U \rhd [\beta]_U$.

Let us check the third condition. For each node $v \in N_G$, if $\chi(\beta^{-1}(v)|_U, x) = n$ by Proposition 4.13 we have $occ(v, \beta(x)) = n$; i.e., there are n positions in $\beta(x)$ corresponding to v. Therefore the out-degree of v is n. In the same way, we have that $\chi(\beta^{-1}(v)|_U, t)$ is the in-degree of v.

Third part. Given $B = \theta'^{-1}(u)|_U$, by Proposition 4.24 we have $B = \beta^{-1}(\rho^{-1}(u))|_U$. Since $\theta' \preceq_U \beta \preceq_U \theta$, $[\theta']_U = \mathrm{mgu}([\beta]_U, \{x/t\}) = [\mathrm{mgu}(\beta, \{x/t\})]_U$. Therefore $\rho \circ \beta' \sim_U \theta' = \mathrm{mgu}(\theta, \{x/t\}) \sim_U \mathrm{mgu}(\beta, \{x/t\}) = \mathrm{mgu}(\beta(x) = \beta(t)) \circ \beta$. We call δ the result of $\mathrm{mgu}(\beta(x) = \beta(t))$ and note that $\beta(x) = \beta(t)$ is equivalent to the set of equations $X = \{v_1 = v_2 \mid \text{there is a position } \xi \text{ such that } \beta(x)(\xi) = v_1 \wedge \beta(t)(\xi) = v_2\}$. The relation $\rho \circ \beta \sim_U \delta \circ \beta$ means that if $w_1, w_2 \in \beta(U)$ and $\rho(w_1) = \rho(w_2)$, then $\delta(w_1) = \delta(w_2)$. The latter implies that in X there are equations of the kind $x_1 = x_2$, $x_2 = x_3, \ldots, x_{n-1} = x_n$ with $x_1 = w_1$ and $x_n = w_2$, i.e., w_1 and w_2 are connected in the graph G.

Therefore, let $Y = \{w \mid \rho(w) = u\} = \llbracket \rho^{-1}(u) \rrbracket$. This is not empty, since $B \neq \{\!\!\{\}\!\!\}$. If ξ is an edge such that $\mathrm{src}_G(\xi) \in Y$, then $\mathrm{tgt}_G(\xi) \in Y$, since $\beta(x)(\xi) = \beta(t)(\xi) \in X$. The converse also holds. Hence, if we restrict the graph G to the set of nodes Y, we obtain a sharing graph whose resultant ω-sharing group is $\biguplus_{w \in Y} \beta^{-1}(w)|_U = \beta^{-1}(\rho^{-1}(u))|_U = B$. $\square$

Theorem 4.28 (*Optimality of* mgu_ω)
The single-binding unification $\mathrm{mgu}_\omega([S]_U, x/t)$ is optimal w.r.t. mgu, under the assumption that $vars(x/t) \subseteq U$, i.e.,

$$\forall B \in \mathrm{mgu}_\omega([S]_U, x/t)\ \exists \delta \in \mathit{ISubst}.\ [S]_U \rhd [\delta]_U \text{ and } B \in \alpha_\omega(\mathrm{mgu}([\delta]_U, \{x/t\})).$$

Proof
Let $X \in \mathrm{mgu}_\omega(S, x/t)$. By definition of mgu_ω, there exists a sharing graph $\mathcal{G}$ such that $X \in res(\mathcal{G})$. Let $N_\mathcal{G} = \{n_1, \ldots, n_k\}$. We want to define a substitution δ such that $[S]_U \rhd [\delta]_U$ and $X \in \alpha_\omega(\mathrm{mgu}([\delta]_U, \{x/t\}))$. If $X = \{\!\!\{\}\!\!\}$ this is trivial; hence we assume that $X \neq \{\!\!\{\}\!\!\}$. The structure of the proof is as follows: first, we define a substitution δ which unifies with x/t; second, we show that δ is approximated by $[S]_U$, namely, $[S]_U \rhd [\delta]_U$; third, we show that $X \in \alpha_\omega(\mathrm{mgu}([\delta]_U, \{x/t\}))$.

First part. We now define a substitution δ which unifies with x/t. For each node $n \in N_\mathcal{G}$ we consider a fresh variable w_n, and we denote by W the set of all these new variables.

For any $y \in U \setminus \{x\}$ we define a term t_y of arity $\sum_{n \in N_\mathcal{G}} l_\mathcal{G}(n)(y)$ as follows:

$$t_y = r(\underbrace{w_{n_1}, \ldots, w_{n_1}}_{l_\mathcal{G}(n_1)(y) \text{ times}}, \underbrace{w_{n_2}, \ldots, w_{n_2}}_{l_\mathcal{G}(n_2)(y) \text{ times}}, \ldots, \underbrace{w_{n_k}, \ldots, w_{n_k}}_{l_\mathcal{G}(n_k)(y) \text{ times}}).$$

We know that there exists a map $f : E_G \to \mathcal{V}$ such that for each variable y and node n, the set of edges targeted at n and labeled with y by f is exactly $l_\mathcal{G}(n)(y) \cdot occ(y, t)$. Namely, we require

$$|\{e \in E_G \mid f(e) = y \wedge \mathrm{tgt}_G(e) = n\}| = l_\mathcal{G}(n)(y) \cdot occ(y, t).$$

The idea is that each edge targeted at the node n is actually targeted at one of the specific variables in $l_\mathcal{G}(n)$. In particular, each variable $y \in \lfloor\!\lfloor l_\mathcal{G}(n) \rfloor\!\rfloor$ should have exactly $l_\mathcal{G}(n)(y) \cdot occ(y, t)$ edges targeted at it, so that the total number of edges pointing n is $\sum_{y \in U} l_\mathcal{G}(n)(y) \cdot occ(y, t) = \chi(l_\mathcal{G}(n), t)$, i.e., the in-degree of n. The map f chooses, for each edge targeted at n, a variable in $l_\mathcal{G}(n)$ according to the previous idea.

Now, for each node n and variable $y \in U$, we denote by $M_{n,y}$ the set of edges pointing at y in n, i.e., $M_{n,y} = \{e \in E_G \mid \mathrm{tgt}_G(e) = n \wedge f(e) = y\}$. Thus $M_{n,y}$ may be partitioned in $occ(y, t)$ sets of $l_\mathcal{G}(n)(y)$ elements, denoted by $M_{n,y,\xi}$ such that $\cup\{M_{n,y,\xi} \mid t(\xi) = y\} = M_{n,y}$.

We may define some variations of the terms t_y by replacing the variables occurring in them with those in the set $M_{n,y,l}$. In particular, for $y \in U \setminus \{x\}$ and any occurrence ξ of a variable y in t, we define the term t^y_ξ of arity $\sum_{n \in N_\mathcal{G}} l_\mathcal{G}(n)(y)$ as

$$t^y_\xi = r(w(M_{n_1,y,\xi}), w(M_{n_2,y,\xi}), \ldots, w(M_{n_k,y,\xi})),$$

where if $M = \{e_1, \ldots, e_q\}$, we define $w(M)$ as the sequence $w_{n'_1}, \ldots, w_{n'_q}$ where $n'_j = \mathrm{src}_{E_G}(e_j)$.

Note that t_y and t^y_ξ have, in corresponding positions, variables related to nodes which are connected through edges. We are now ready to define the substitution δ in the following way:

- $\delta(x)$ is the same as t with the difference that each occurrence ξ of a variable $y \in t$ is replaced by the term t^y_ξ;
- for $y \in U \setminus \{x\}$, $\delta(y) = t_y$;
- in all the other cases, i.e., $v \notin U$, $\delta(v) = v$.

Second part. Now we show that $[S]_U \rhd [\delta]_U$. We need to consider all the variables $v \in \mathcal{V}$ and check that $\delta^{-1}(v)|_U \in S$. We distinguish several cases.

- If we choose the variable w_n for some $n \in N$, by construction $occ(w_n, t_y) = l_{\mathcal{G}}(n)(y)$. Moreover, since $\mathcal{G}$ is a sharing graph, there are $l_{\mathcal{G}}(n)(x)$ edges in E departing from n and targeted at nodes m such that $\chi(l_{\mathcal{G}}(m), t) \neq 0$. Thus $\sum_{y \in vars(t), m \in N_{\mathcal{G}}} |\{e \in M_{m,y} | \mathrm{src}_{E_G}(e) = n\}| = l_{\mathcal{G}}(n)(x)$ and $occ(\delta(x), w_n) = l_{\mathcal{G}}(n)(x)$. Since for each $v \in U$ we have that $occ(\delta(v), w_n) = l_{\mathcal{G}}(n)(v)$, we obtain the required result which is $\delta^{-1}(w_n)|_U = l_{\mathcal{G}}(n) \in S$.
- If we choose a variable $v \in U$, then $v \in \mathrm{dom}(\delta)$ and $\delta^{-1}(v) = \{\!\!\{\}\!\!\} \in S$.
- Finally, if $v \notin U \cup W$, then $\delta^{-1}(v) = \{\!\!\{v\}\!\!\}$ and $\delta^{-1}(v)|_U = \{\!\!\{\}\!\!\} \in S$.

Third part. We now show that $X \in \alpha_\omega(\mathrm{mgu}([\delta]_U, \{x/t\}))$. By definition of mgu over $ISubst_\sim$, we have that $\mathrm{mgu}([\delta]_U, \{x/t\}) = [\mathrm{mgu}(\delta, \{x/t\})]_U$. We obtain

$$\begin{aligned}\eta &= \mathrm{mgu}(\delta, \{x/t\})\\ &= \{x/t\} \circ \mathrm{mgu}\big(\{y = t_y \mid y \in U \setminus \{x\}\} \cup \{y = t^y_\xi \mid t(\xi) = y\}\big)\\ &= \{x/t\} \circ \{y/t_y \mid y \in U \setminus \{x\}\} \circ \mathrm{mgu}\{t_y = t^y_\xi \mid t(\xi) = y\}. \qquad \text{(A1)}\end{aligned}$$

Let F be the set of equations $\{t_y = t^y_j \mid t(j) = y\}$. We show that for any edge $n \to m \in E_G$, it follows from F that $w_n = w_m$. Since $n \to m \in E_G$, for some $y \in vars(t)$ it holds that $f(n \to m) = y$. This implies that $n \to m \in M_{m,y}$, and therefore there exists a position ξ such that $n \to m \in M_{m,y,\xi}$. By definition of t^y_ξ, it means that $w_n \in vars(t^y_\xi)$, in the same position at which w_m occurs in t_y; hence $w_n = w_m$ follows from $t_y = t^y_\xi \in F$.

We know that $\mathcal{G}$ is connected; hence for any $n, m \in N_{\mathcal{G}}$, the set of equations in F implies $w_n = w_m$. We choose a particular node $\bar{n} \in N_{\mathcal{G}}$, and for what we said before, we have $\mathrm{mgu}(F) = \{w_n/w_{\bar{n}} \mid n \in N_{\mathcal{G}} \setminus \{\bar{n}\}\}$. We show that $\eta^{-1}(w_{\bar{n}})|_U = X$:

$$\begin{aligned}&\eta^{-1}(w_{\bar{n}})|_U\\ &\quad= \{x/t\}^{-1}(\{y/t_y \mid y \in U \setminus \{x\}\}^{-1}(\{\!\!\{w_{n_1}, \ldots, w_{n_k}\}\!\!\}))|_U\\ &\quad= \{x/t\}^{-1}(\{\!\!\{w_{n_1}, \ldots, w_{n_k}\}\!\!\} \uplus \lambda y \in U \setminus \{x\}. \sum_{n \in N_{\mathcal{G}}} l_{\mathcal{G}}(n)(y))|_U\\ &\quad= \lambda y \in U \setminus \{x\}. \sum_{n \in N_{\mathcal{G}}} l_{\mathcal{G}}(n)(y) \uplus \{\!\!\{x^{\sum_{y \in \mathcal{V}} occ(y,t) \cdot \sum_{n \in N_{\mathcal{G}}} l_{\mathcal{G}}(n)(y)}\}\!\!\}\\ &\quad= \lambda y \in U \setminus \{x\}. \sum_{n \in N_{\mathcal{G}}} l_{\mathcal{G}}(n)(y) \uplus \{\!\!\{x^{\sum_{n \in N_{\mathcal{G}}} \chi(l_{\mathcal{G}}(n), t)}\}\!\!\}.\end{aligned}$$

Since $\mathscr{G}$ is a sharing graph, the total out-degree $\sum_{n\in N_{\mathscr{G}}} \chi(l_{\mathscr{G}}(n), t)$ is equal to the total in-degree $\sum_{n\in N_{\mathscr{G}}} \chi(l_{\mathscr{G}}(n), x)$. Hence

$$\begin{aligned}
&\eta^{-1}(w_{\bar{n}})|_U \\
&\quad = \lambda y \in U \setminus \{x\}. \sum_{n\in N_{\mathscr{G}}} l_{\mathscr{G}}(n)(y) \uplus \{\!\{ x^{\sum_{n\in N_{\mathscr{G}}} \chi(l_{\mathscr{G}}(n),x)} \}\!\} \\
&\quad = \lambda y \in U. \sum_{n\in N_{\mathscr{G}}} l_{\mathscr{G}}(n)(y) \\
&\quad = res(\mathscr{G}).
\end{aligned}$$

This concludes the proof. $\square$

Theorem 4.30 (*Optimality of* mgu_ω *with extension*)
The single-binding unification mgu_ω with extension is optimal w.r.t. mgu.

Proof
Let $S' = S \cup \{\{\!\{v\}\!\} \mid v \in vars(x/t) \setminus U\}$, $V = U \cup vars(x/t)$, and $X \in \mathrm{mgu}_\omega(S', x/t)$. We want to find $[\delta]_U$ such that $[S]_U \rhd [\delta]_U$ and $X \in \alpha_\omega(\mathrm{mgu}([\delta]_U, \{x/t\}))$.

Following the previous theorem, we find δ such that $X \in \alpha_\omega(\mathrm{mgu}([\delta]_V, \{x/t\}))$ and $[S']_V \rhd [\delta]_V$. We want to prove that $[S]_U \rhd [\delta]_U$ and $\alpha_\omega(\mathrm{mgu}([\delta]_V, \{x/t\})) \leqslant_\omega \alpha_\omega(\mathrm{mgu}([\delta]_U, \{x/t\}))$, so that $[\delta]_U$ is the existential substitutions we are looking for.

We first show that $[S]_U \rhd [\delta]_U$. Let $v \in \mathscr{V}$. Since $[S']_V \rhd [\delta]_V$, it follows that $\delta^{-1}(v)|_V \in S'$:

- If $\delta^{-1}(v)|_V \in S$, then $\delta^{-1}(v)|_V = \delta^{-1}(v)|_U$, since $vars(S) \subseteq U$, and thus $\delta^{-1}(v)|_U \in S$.
- If $\delta^{-1}(v)|_V \notin S$, then $\delta^{-1}(v)|_V \in \{\{\!\{v\}\!\} \mid v \in vars(x/t) \setminus U\}$. Then $\delta^{-1}(v)|_U = \{\!\{\}\!\} \in S$.

Now we distinguish two cases: either $x \in U$ or $x \notin U$.

If $x \in U$, with the same considerations which led to (A1), we have

$$\begin{aligned}
\mathrm{mgu}(\{x/t\}, \delta) &= \mathrm{mgu}(\{x = t\} \cup \mathrm{Eq}(\delta|_U) \cup \mathrm{Eq}(\delta|_{V\setminus U})) = \\
&= \mathrm{mgu}(\{x = t\} \cup \mathrm{Eq}(\delta|_U) \cup \{y = t_y \mid y \in vars(t) \setminus U\}.
\end{aligned}$$

For each $y \in vars(t) \setminus U$ there exist a position ξ_y such that $t(\xi_y) = y$ and $\{x/t\} \cup \mathrm{Eq}(\delta|_U) \cup \{y = t_y\}$ is equivalent to $\{x/t\} \cup \mathrm{Eq}(\delta|_U) \cup \{t^y_{\xi_y} = t_y\}$. Note that since $y \notin U$, t_y (which is actually $\delta(y)$) is linear and independent from x/t and the other bindings in δ. Therefore

$$\begin{aligned}
&\mathrm{mgu}(\{x = t\} \cup \mathrm{Eq}(\delta|_U) \cup \{y = t_y \mid y \in vars(t) \setminus U\} \\
&\quad = \mathrm{mgu}(\{x = t\} \cup \mathrm{Eq}(\delta|_U) \cup \{t^y_{\xi_y} = t_y \mid y \in vars(t) \setminus U\} \\
&\quad = \mathrm{mgu}(\{x = t\} \cup \mathrm{Eq}(\delta|_U)) \uplus \beta',
\end{aligned}$$

where $\beta' = \mathrm{mgu}(\{t^y_{\xi_y} = t_y \mid y \in vars(t) \setminus U\})$ and $\mathrm{dom}(\beta') = vars(\{t_y \mid y \in vars(t) \setminus U\})$. It follows that

$$\begin{aligned}
&\alpha_\omega([\mathrm{mgu}(\{x = t\} \cup \mathrm{Eq}(\delta|_U)) \uplus \beta']_V) \\
&\quad = \alpha_\omega([\mathrm{mgu}(\{x = t\} \cup \mathrm{Eq}(\delta|_U))]_V) \\
&\quad = \alpha_\omega(\mathrm{mgu}([\delta]_U, \{x/t\})).
\end{aligned}$$

If $x \notin U$, then

$$\begin{aligned}\mathrm{mgu}(\{x/t\},\delta) &= \mathrm{mgu}(\{x=t\} \cup \mathrm{Eq}(\delta|_U) \cup \mathrm{Eq}(\delta|_{vars(t)\setminus U}) \cup \mathrm{Eq}(\delta|_{\{x\}})\\ &= \mathrm{mgu}(\{x=t\} \cup \mathrm{Eq}(\delta|_U) \cup \{y = t_y \mid y \in vars(t) \setminus U\} \cup \\ &\quad \{t_y = t^y_\xi \mid t(\xi) = y\}\end{aligned}$$

Note that x appears in S' only in the multiset $\{\!\{x\}\!\}$. Moreover, if n is a node labeled by $\{\!\{x\}\!\}$, there is only one edge which departs from n and there are no edges which arrive in n. This means that

- w_n does not appear in any t_y for $y \in V \setminus \{x\}$;
- $\delta(x)$ is linear, since given edges $e \neq e'$, we have that $\mathrm{src}_{E_G}(e) \neq \mathrm{src}_{E_G}(e')$.

As a result, $\delta(x)$ is linear and does not share variables with x/t or the other bindings in δ. The last formula may be rewritten as

$$\mathrm{mgu}(\{x=t\} \cup \mathrm{Eq}(\delta|_U) \cup \{y = t_y \mid y \in vars(t) \setminus U\}) \uplus \beta,$$

where β is a substitution such that $\mathrm{dom}(\beta) = vars(\delta(x)) \subseteq W$. It is obvious that

$$\begin{aligned}&\alpha_\omega([\mathrm{mgu}(\{x=t\} \cup \mathrm{Eq}(\delta|_U) \cup \{y = t_y \mid y \in vars(t) \setminus U\}) \uplus \beta]_V)\\ &\quad = \alpha_\omega([\mathrm{mgu}(\{x=t\} \cup \mathrm{Eq}(\delta|_U) \cup \{y = t_y \mid y \in vars(t) \setminus U\})]_V),\end{aligned}$$

since $\mathrm{dom}(\beta) \cap V = \emptyset$.

Let $U_1 = vars(t) \setminus U$; then

$$\begin{aligned}&\mathrm{mgu}(\{x=t\} \cup \mathrm{Eq}(\delta|_U) \cup \mathrm{Eq}(\delta|_{U_1}))\\ &\quad = \delta|_U \circ \mathrm{mgu}(\delta|_U(\{x=t\} \cup \mathrm{Eq}(\delta|_{U_1})))\\ &\quad = \delta|_U \circ \mathrm{mgu}(\{x = \delta|_U(t)\}) \cup \mathrm{Eq}(\delta|_{U_1})))\\ &\qquad [\text{since } vars(\delta|_{U_1}) \cap vars(\delta|_U) = \emptyset \text{ and } x \notin vars(\delta|_{U_1})]\\ &\quad = \delta|_U \circ \{x/\delta|_U(t)\} \circ \delta|_{U_1}\\ &\qquad [\text{since } \{x\} \notin vars(\delta|_{U_1})].\end{aligned}$$

Note that $\delta|_U \circ \{x/\delta|_U(t)\}$ is $\mathrm{mgu}(\delta|_U, \{x/t\})$. We call $\gamma = \delta|_U \circ \{x/\delta|_U(t)\}$, and we prove that $\alpha_\omega([\gamma]_V) \geqslant_\omega \alpha_\omega([\gamma \circ \delta|_{U_1}]_V)$.

Consider a variable $v \in \mathscr{V}$. If $v \notin vars(\delta|_{U_1})$ there is nothing to prove. If $v \in \mathrm{rng}(\delta|_{U_1})$ we know that v does not occur anywhere else in $\delta|_{U_1}$ and γ. Then $(\gamma \circ \delta|_{U_1})^{-1}(v) = \gamma^{-1}(\{\!\{y, v\}\!\}) = \gamma^{-1}(y) \uplus \{\!\{v\}\!\}$ for the unique y such that $v \in vars(\delta|_{U_1}(y))$. Therefore, since $v \notin V$, the sharing group over V we obtain in $\gamma \circ \delta|_{U_1}$ from v may be obtained in γ from the variable y. If $v \in \mathrm{dom}(\delta|_{U_1})$, then $(\gamma \circ \delta|_{U_1})^{-1}(v) = \{\!\{\}\!\}$, which occurs in every element of $\mathtt{ShLin}^\omega$. □

Theorem 4.31

Let S be a set of ω-sharing groups and x/t be a binding. Then $B \in \mathrm{mgu}_\omega(S, x/t)$ iff there exist $n \in \mathbb{N}^+$, $B_1, \ldots, B_n \in S$ which satisfy the following conditions:

(1) $B = \uplus_{1 \leqslant i \leqslant n} B_i$,
(2) $\sum_{1 \leqslant i \leqslant n} \chi(B_i, x) = \sum_{1 \leqslant i \leqslant n} \chi(B_i, t) \geqslant n - 1$,
(3) either $n = 1$ or $\forall 1 \leqslant i \leqslant n.\ \chi(B_i, x) + \chi(B_i, t) > 0$.

Proof
We first prove that these conditions are necessary. Assume that B is a resultant sharing group for S and x/t, obtained by the sharing graph G. We show that there exist a finite set I and, for each $i \in I$, a multiset $B_i \in S$, which satisfy the above conditions.

Take $I = N_G$ and $B_i = l_G(i)$ for each $i \in I$, so that $B = \uplus_{i\in I} B_i$. Since in-degree of each node then is $\chi(B_i, x)$, the sum of the in-degrees of all the nodes is $\sum_{i\in I} \chi(B_i, x)$, and the sum of the out-degree is $\sum_{i\in I} \chi(B_i, t)$. Both of them must be equal to the number of edges in E_G. Moreover, each connected graph with $|I|$ nodes has at least $|I|-1$ edges. Finally, if a connected graph has more than one node, then every node i has an adjacent edge. Therefore, either $\chi(B_i, x)$ or $\chi(B_i, t)$ is not zero.

Now we prove that the conditions are sufficient. Let $I = \{1, \ldots, n\}$. If $n = 1$ and $\chi(B_i, x)+\chi(B_i, t) = 0$ for the only $i \in I$, simply consider a sharing graph with a single node labeled with B_i and no edges. Otherwise, we partition the set I in three parts:

- $N_x = \{i \in I \mid \chi(B_i, x) = 0\}$;
- $N_t = \{i \in I \mid \chi(B_i, t) = 0\}$;
- $N = \{i \in I \mid \chi(B_i, x) \neq 0, \chi(B_i, t) \neq 0\}$.

Note that this is a partition of I since, by hypothesis, $\forall i \in I.\ \chi(B_i, x) + \chi(B_i, t) > 0$. Now we define a connected labeled multigraph G whose sets of nodes is I and whose labeling function is $\lambda i \in I.B_i$. In order to define the edges, we distinguish two cases.

$N \neq \emptyset$**:** Let $N = \{b_1, \ldots, b_m\}$ with $m \geqslant 1$ and consider the set of edges

$$\{a \to b_1 \mid a \in N_t\} \cup \{b_1 \to c \mid c \in N_x\} \cup \{b_i \to b_{i+1} \mid i \in \{1, \ldots, m-1\}\}.$$

$N = \emptyset$**:** If $N_t = \emptyset$, then also $N_x = \emptyset$ and there is nothing to prove. We assume that $N_t \neq \emptyset$, and thus $N_x \neq \emptyset$. Let $\bar{a} \in N_t$, $\bar{c} \in N_x$ and consider the set of edges

$$\{\bar{a} \to c \mid c \in N_x\} \cup \{a \to \bar{c} \mid a \in N_t \setminus \{\bar{a}\}\}.$$

Note that in both cases, we obtain a multigraph with the following properties:

(1) It is connected.
(2) It has exactly $n-1$ edges; i.e., it is a tree (if we do not consider the direction of edges).
(3) There is no edge targeted at a node i with $\chi(i, t) = 0$ and no edge whose source is a node i with $\chi(i, x) = 0$.

In the rest of the proof, we call *pre-sharing graph* a multigraph which satisfies the above properties.

If $indeg(i)$ is the in-degree of a node and $outdeg(i)$ the out-degree, we call *unbalancement factor* of the graph the value

$$\sum\{outdeg(i) - \chi(B_i, x) \mid i \in I, outdeg(i) > \chi(B_i, x)\} + \sum\{indeg(i) - \chi(B_i, t) \mid i \in I, indeg(i) > \chi(B_i, t)\}.$$

We prove that given a pre-sharing graph with unbalancement factor k, we can build another pre-sharing graph with unbalancement factor strictly less than k. As a result, there is a pre-sharing graph with unbalancement factor equal to zero.

Assume that the graph has unbalancement factor k. There is at least an unbalanced node. Assume without loss of generality that the unbalanced node is j and that $outdeg(j) > \chi(B_j, x)$. Since $\sum_{i \in I} \chi(B_i, x) \geqslant n - 1$, there exists a node l such that $outdeg(l) < \chi(B_l, x)$. Let e be the unique edge with source j such that if we remove e from the graph, l becomes disconnected from j. Since no edge starts from a node i with $\chi(B_i, x) = 0$, $\chi(B_j, x) > 0$. This means that $outdeg(j) > 1$, and there is at least another edge starting from j. Assume that it is $e' : j \to j'$. Remove this edge and replace it with an edge $l \to j'$. It is obvious that the result is a pre-sharing graph with a smaller unbalancement factor than the original one. The case for $indeg(j) > \chi(B_j, t)$ is symmetric.

Once the unbalancement factor is zero, since $\sum_{i \in I} \chi(B_i, x) = \sum_{i \in I} \chi(B_i, t)$ we can freely add other edges in such a way to complete the graph w.r.t. the condition on the degree of nodes. We obtain a sharing graph G such that $res(G) = B$. □

Appendix B

Proofs of Section 5

In this section we give the proofs of correctness and optimality for the abstract unification operators mgu_2 and mgu_{sl}.

Proposition 5.1
The following properties hold:

(1) $\alpha_2(\biguplus \mathscr{R}) = \biguplus \alpha_2(\mathscr{R})$.
(2) $rel(\gamma_2(S), x, t)) = \gamma_2(rel(S, x, t))$.

Proof
We begin by proving the first property:

$$
\begin{aligned}
&\alpha_2(\uplus\{\!\{B_1, \ldots, B_n\}\!\}) \\
&\quad = \alpha_2\Big(\lambda v \in \bigcup_{1 \leqslant i \leqslant n} [\![B_i]\!].\ \sum_{1 \leqslant i \leqslant n} B_i(v)\Big) \\
&\quad = \lambda v \in \bigcup_{1 \leqslant i \leqslant n} [\![B_i]\!].\ \begin{cases} 1 & \text{if } \sum_{1 \leqslant i \leqslant n} B_i(v) = 1 \\ \infty & \text{otherwise} \end{cases} \\
&\quad = \uplus\{\!\{o_1, \ldots, o_n\}\!\} \text{ where } o_i = \lambda v \in [\![B_i]\!].\ \begin{cases} 1 & \text{if } B_i(v) = 1 \\ \infty & \text{otherwise} \end{cases} \\
&\quad = \uplus\{\!\{\alpha_2(B_1), \ldots, \alpha_2(B_n)\}\!\}.
\end{aligned}
$$

Now we proceed with the proof of the second property:

$$
\begin{aligned}
&rel(\gamma_2(S), x, t)) \\
&\quad = \bigcup \{\gamma_2(o) \mid o \in S, [\![\gamma_2(o)]\!] \cap vars(x = t) \neq \emptyset\} \\
&\quad = \bigcup \{\gamma_2(o) \mid o \in S, [\![o]\!] \cap vars(x = t) \neq \emptyset\} \qquad (\text{since } [\![o]\!] = [\![\gamma_2(o)]\!]) \\
&\quad = \gamma_2(rel(S, x, t))). \quad \square
\end{aligned}
$$

Theorem 5.2
$\langle \alpha_2, \gamma_2 \rangle : \mathtt{ShLin}^2 \leftrightarrows \mathtt{ShLin}^\omega$ is a Galois insertion.

Proof
It is obvious that α_2 and γ_2 are monotone functions and that they are both join-morphisms. Extensionality of $\gamma_2 \circ \alpha_2$ follows from the fact that given an ω-sharing group B, we have $B \in \gamma_2(\alpha_2(B))$. Finally, given a 2-sharing group o, we have $\alpha_2(\gamma_2(o)) = \{o\}$. This implies that $\alpha_2 \circ \gamma_2$ is the identity. $\square$

Theorem 5.5
Given $[S]_U \in \mathtt{ShLin}^\omega$ and the binding x/t with $vars(x/t) \subseteq U$, we have that

$$\begin{aligned} \mathrm{mgu}_2([S]_U, x/t) = [(S \setminus S') \cup \\ \downarrow\{\biguplus Y \mid Y \in \wp_m(S'), n \in \chi(Y, x) \cap \chi(Y, t).\ n \geqslant |Y| - 1\}]_U, \end{aligned}$$

where $S' = rel(S, x, t)$.

Proof
By using Proposition 5.1 point 2 and since $o \neq o' \Rightarrow \gamma_2(o) \cap \gamma_2(o') = \emptyset$, we get

$$\begin{aligned} &\alpha_2(\gamma_2(S) \setminus rel(\gamma_2(S), x, t)) \\ &\quad = \alpha_2(\gamma_2(S) \setminus \gamma_2(rel(S, x, t))) \\ &\quad = \alpha_2(\gamma_2(S \setminus rel(S, x, t))) \\ &\quad = S \setminus rel(S, x, t). \end{aligned}$$

Therefore, we get the equality

$$\begin{aligned} \mathrm{mgu}_2([S]_U, x/t) = \Big[S \setminus rel(S, x, t) \cup \\ \alpha_2\Big(\{\uplus \mathscr{R} \mid \mathscr{R} \in \wp_m(rel(\gamma_2(S), x, t)), \sum_{B \in \mathscr{R}} \chi(B, x) = \sum_{B \in \mathscr{R}} \chi(B, t) \geqslant |\mathscr{R}| - 1\}\Big)\Big]_U. \end{aligned}$$

Now, with simple algebraic manipulations, we obtain

$$\begin{aligned} &\alpha_2(\{\uplus \mathscr{R} \mid \mathscr{R} \in \wp_m(rel(\gamma_2(S), x, t)), \sum_{B \in \mathscr{R}} \chi(B, x) = \sum_{B \in \mathscr{R}} \chi(B, t) \geqslant |\mathscr{R}| - 1\}) \\ &\quad = \alpha_2(\{\uplus \mathscr{R} \mid \mathscr{R} \in \wp_m(\gamma_2(rel(S, x, t))), \sum_{B \in \mathscr{R}} \chi(B, x) = \sum_{B \in \mathscr{R}} \chi(B, t) \geqslant |\mathscr{R}| - 1\}) \\ &\quad = \alpha_2(\{\uplus \{\!\{B_1, \ldots, B_k\}\!\} \mid k \in \mathbb{N}, \\ &\qquad\qquad \forall i. B_i \in \gamma_2(rel(S, x, t)), \sum_{1 \leqslant i \leqslant k} \chi(B_i, x) = \sum_{1 \leqslant i \leqslant k} \chi(B_i, t) \geqslant k - 1\}) \\ &\quad = \alpha_2(\{\uplus \{\!\{B_1, \ldots, B_k\}\!\} \mid k \in \mathbb{N}, \{\!\{o_1, \ldots, o_k\}\!\} \in \wp_m(rel(S, x, t)), \\ &\qquad\qquad \forall i. B_i \in \gamma_2(o_i), \sum_{1 \leqslant i \leqslant k} \chi(B_i, x) = \sum_{1 \leqslant i \leqslant k} \chi(B_i, t) \geqslant k - 1\}) \\ &\quad = \alpha_2(\{\uplus \{\!\{B_1, \ldots, B_k\}\!\} \mid k \in \mathbb{N}, \{\!\{o_1, \ldots, o_k\}\!\} \in \wp_m(rel(S, x, t)), \\ &\qquad\qquad \forall i. \alpha_2(B_i) = o_i, \sum_{1 \leqslant i \leqslant k} \chi(B_i, x) = \sum_{1 \leqslant i \leqslant k} \chi(B_i, t) \geqslant k - 1\}) \\ &\qquad\qquad\qquad \text{(such } o_i\text{'s do always exist, since } rel(S, x, t) \text{ is downward closed)} \end{aligned}$$

$$
\begin{aligned}
&= \downarrow\{\alpha_2(\uplus\{\!\{B_1,\dots,B_k\}\!\}) \mid k \in \mathbb{N}, \{\!\{o_1,\dots,o_k\}\!\} \in \wp_m(rel(S,x,t)),\\
&\qquad \forall i.\alpha_2(B_i) = o_i, \sum_{1\leqslant i\leqslant k} \chi(B_i,x) = \sum_{1\leqslant i\leqslant k} \chi(B_i,t) \geqslant k-1\}\\
&= \downarrow\{\uplus\{\!\{\alpha_2(B_1),\dots,\alpha_2(B_k)\}\!\} \mid k \in \mathbb{N}, \{\!\{o_1,\dots,o_k\}\!\} \in \wp_m(rel(S,x,t)),\\
&\qquad \forall i.\alpha_2(B_i) = o_i, \sum_{1\leqslant i\leqslant k} \chi(B_i,x) = \sum_{1\leqslant i\leqslant k} \chi(B_i,t) \geqslant k-1\}\\
&\qquad\qquad \text{(by Proposition 5.1 point 5.1)}\\
&= \downarrow\{\uplus\{\!\{o_1,\dots,o_k\}\!\} \mid k \in \mathbb{N}, \{\!\{o_1,\dots,o_k\}\!\} \in \wp_m(rel(S,x,t)),\\
&\qquad \forall i.\alpha_2(B_i) = o_i, \sum_{1\leqslant i\leqslant k} \chi(B_i,x) = \sum_{1\leqslant i\leqslant k} \chi(B_i,t) \geqslant k-1\}\}\\
&= \downarrow\{\uplus\{\!\{o_1,\dots,o_k\}\!\} \mid k \in \mathbb{N}, \{\!\{o_1,\dots,o_k\}\!\} \in \wp_m(rel(S,x,t)),\\
&\qquad \forall i.\alpha_2(B_i) = o_i, \forall i.\alpha_2(B'_i) = o_i, \sum_{1\leqslant i\leqslant k} \chi(B_i,x) = \sum_{1\leqslant i\leqslant k} \chi(B'_i,t) \geqslant k-1\}\\
&\qquad\qquad \text{(we discuss later why this is faithful)}\\
&= \downarrow\{\uplus\{\!\{o_1,\dots,o_k\}\!\} \mid k \in \mathbb{N}, \{\!\{o_1,\dots,o_k\}\!\} \in \wp_m(rel(S,x,t)), n \geqslant k-1,\\
&\qquad n \in \{\sum_{1\leqslant i\leqslant k} B_i(x) \mid \forall i.\alpha_2(B_i) = o_i\} \cap \{\sum_{1\leqslant i\leqslant k} \chi(B'_i,t) \mid \forall i.\alpha_2(B'_i) = o_i\}\}\\
&= \downarrow\{\uplus\{\!\{o_1,\dots,o_k\}\!\} \mid k \in \mathbb{N}, \{\!\{o_1,\dots,o_k\}\!\} \in \wp_m(rel(S,x,t)), n \geqslant k-1\\
&\qquad n \in [\sum_{1\leqslant i\leqslant k} o_{im}(x), \sum_{1\leqslant i\leqslant k} o_i(x)] \cap \{\sum_{1\leqslant i\leqslant k} \chi(B'_i,t) \mid \forall i.\alpha_2(B'_i) = o_i\}\}.
\end{aligned}
$$

The move from a single family $\{B_i\}_{1\leqslant i\leqslant k}$ to different families $\{B_i\}_{1\leqslant i\leqslant k}$ and $\{B'_i\}_{1\leqslant i\leqslant k}$ is possible since if

$$\forall i.\alpha_2(B_i) = o_i \text{ and } \forall i.\alpha_2(B'_i) = o_i \text{ and } \sum_{1\leqslant i\leqslant k} \chi(B_i,x) = \sum_{1\leqslant i\leqslant k} \chi(B'_i,t) \geqslant k-1,$$

we may define a family $\{C_i\}_{1\leqslant i\leqslant k}$ such that $C_i(x) = B_i(x)$ and $C_i(v) = B'(v)$ for each $v \neq x$. It is immediate to check that the C_i's satisfy the condition

$$\forall i.\alpha_2(C_i) = o_i \text{ and } \sum_{1\leqslant i\leqslant k} \chi(C_i,x) = \sum_{1\leqslant i\leqslant k} \chi(C_i,t) \geqslant k-1.$$

If we denote with $c(\{\!\{o_1,\dots,o_k\}\!\},t)$ the set $\{\sum_{1\leqslant i\leqslant k} \chi(B_i,t) \mid \forall i.\alpha_2(B_i) = o_i\}$, what remains to prove is that

$$
\begin{aligned}
&\downarrow\{\biguplus X \mid X \in \wp_m(rel(S,x,t)), n \in \chi(X,x) \cap c(X,t).n \geqslant |X|-1\}\\
&= \downarrow\{\biguplus X \mid X \in \wp_m(rel(S,x,t)), n \in \chi(X,x) \cap \chi(X,t), n \geqslant |X|-1\},
\end{aligned}
$$

where the only difference is that we have replaced $c(X,t)$ with $\chi(X,t)$.

We begin by examining the relationship between $c(X,t)$ and $\chi(X,t)$. First of all, it is obvious that $c(X,t) \subseteq \chi(X,t)$; therefore we only need to prove half of the equality.

If there exists $o \in X$ such that $\chi_M(o,t) = \infty$, then $c(X,t)$ is an infinite set. We call n its least element. Under the same conditions, $\chi(X,t)$ is the interval $[n,\infty]$. If there is no $o \in X$ such that $\chi_M(o,t) = \infty$, then $c(X,t) = \chi(X,t)$ and they are both singletons.

In the same way, if there exists some $o \in X$ such that $o(x) = \infty$, then $\chi(X, x)$ is an interval of the kind $[n, \infty)$. However, if there is no such o, then $\chi(X, x)$ is a singleton, whose unique element is $|\{o \in X \mid o(x) = 1\}|$.

Assume that we have $X \in \wp_m(rel(S, x, t))$ such that there exists $n \in \chi(X, x) \cap \chi(X, t)$ with $n \geqslant |X| - 1$. We want to prove that we may find a multiset $Y \in \wp_m(rel(S, x, t))$ such that there exists $m \geqslant |Y| - 1$ with $m \in \chi(Y, x) \cap c(Y, t)$ and $\biguplus X \leqslant \biguplus Y$. This is enough to complete the proof of the theorem.

We distinguish several cases.

- $\chi(X, x)$ and $\chi(X, t)$ are both infinite. In this case, $c(X, t)$ is infinite. Moroever, since $\chi(X, x)$ is an interval, there are infinite natural numbers in $\chi(X, x) \cap c(X, t)$. We may take $Y = X$.
- $\chi(X, t)$ is infinite and $\chi(X, x)$ is a singleton $\{v\}$; then $v = |\{o \in X \mid o(x) = 1\}| \leqslant k$. Since it must be $v \geqslant k - 1$, there are only two choices: either $v = k$ or $v = k - 1$. We distinguish the two subcases:
  - (i) $v = k - 1$. In this case, there exists $o \in X$ such that $\chi_m(o, t) = 0$ and $o(x) = 1$; otherwise it is not possible that $v \geqslant \chi_m(X, t)$. Since $\chi(X, t)$ is infinite, the same holds for $c(X, t)$; hence we may find an $n \in c(X, t)$ such that $n \geqslant v$. Consider $Y = X \uplus (n - v) * \{\!\{o\}\!\}$. We have $\chi(Y, x) = \{v + (n - v)\} = n$, $c(Y, t) = c(X, t)$, and $|Y| = |X| + n - v = n + 1$. Therefore $n \in c(Y, x) \cap c(Y, t)$ and $n \geqslant |Y| - 1$. $\biguplus Y$ is a valid result, and $\biguplus X \leqslant \biguplus Y$.
  - (ii) $v = k$. If there is an $o \in X$ such that $\chi_m(o, t) = 0$, the proof proceeds as in the previous case. Otherwise, $\chi_m(X, t) \geqslant k$, and since it should be $v = k \geqslant \chi_m(X, t)$, we have $\chi_m(X, t) = k$. Therefore $k \in c(X, t)$ too, since $\min c(X, t) = \min \chi(X, t)$, and we may take $Y = X$.
- If $\chi(X, t)$ is finite, then $\chi(X, t) = c(X, t)$, and we take $Y = X$. □

Theorem 5.7
Given $[S]_U$ in $\mathtt{ShLin}^2$ and the binding x/t, let $V = \{v_1, \ldots, v_n\}$ be $vars(x/t) \setminus U$. Then,

$$\mathrm{mgu}_2([S]_U, x/t) = \mathrm{mgu}_2([S \cup \{v_1, \ldots, v_n\}]_{U \cup V}, x/t).$$

Proof
First of all, given a finite set of variables V, let us define the extension operator $\mathrm{ext}_V : \mathtt{ShLin}^\omega \to \mathtt{ShLin}^\omega$ such that $\mathrm{ext}_V([S]_U) = [S \cup \{\{\!\{v\}\!\} \mid v \in V \setminus U\}]_{U \cup V}$. Given $V = vars(x/t) \setminus U = \{v_1, \ldots, v_n\}$, we have that

$$\begin{aligned}\mathrm{mgu}_2([S]_U, x/t) &= \alpha_2(\mathrm{mgu}_\omega(\gamma_2([S]_U), x/t)) \\ &= \alpha_2(\mathrm{mgu}_\omega(\mathrm{ext}_V(\gamma_2([S]_U)), x/t)).\end{aligned}$$

We also know that

$$\begin{aligned}&\mathrm{mgu}_2([S \cup \{v_1, \ldots, v_n\}]_{U \cup V}, x/t) \\ &\quad = \alpha_2(\mathrm{mgu}_\omega(\gamma_2([S \cup \{v_1, \ldots, v_n\}]_{U \cup V}), x/t)).\end{aligned}$$

Hence, it is enough to prove that

$$\mathrm{ext}_V(\gamma_2([S]_U)) = \gamma_2([S \cup \{v_1, \ldots, v_n\}]_{U \cup V}).$$

By definition of γ_2, we have that

$$\begin{aligned}
&\gamma_2([S \cup \{v_1,\dots,v_n\}]_{U\cup V})\\
&\quad = \left[\bigcup\{\gamma_2(o) \mid o \in S \cup \{v_1,\dots,v_n\}\}\right]_{U\cup V}\\
&\quad = \left[\bigcup\{\gamma_2(o) \mid o \in S\} \cup \{\{\!\{v_1\}\!\},\dots,\{\!\{v_n\}\!\}\}\right]_{U\cup V} \qquad [\text{since } \gamma_2(v_i) = \{\!\{v_i\}\!\}]\\
&\quad = \mathrm{ext}_V(\gamma_2([S]_U)),
\end{aligned}$$

which completes the proof. □

Theorem 5.9
Given $[S]_U \in \mathtt{ShLin}^2$ and the binding x/t with $vars(x/t) \subseteq U$, we have

$$\mathrm{mgu}_2([S]_U, x/t) = [(S \setminus S') \cup \downarrow \bigcup_{X \subseteq S'} res(X,x,t)]_U,$$

where $S' = rel(S,x,t)$ and $res(X,x,t)$ is defined as follows:

(1) if X is nonlinear for x and t, then $res(X,x,t) = \{\biguplus X^2\}$;
(2) if X is nonlinear for x and linear for t, $|X_x| \leqslant 1$ and $|X_t| \geqslant 1$, then we have $res(X,x,t) = \{(\biguplus X_x) \uplus (\biguplus X^2_{xt}) \uplus (\biguplus X^2_t)\}$;
(3) if X is linear for x and strongly nonlinear for t, $|X_x| \geqslant 1$ and $|X_t| \leqslant 1$, then we have $res(X,x,t) = \{(\biguplus X^2_x) \uplus (\biguplus X^2_{xt}) \uplus (\biguplus X_t)\}$;
(4) if X is linear for x and not strongly nonlinear for t, $|X_t| \leqslant 1$, then we have

$$res(X,x,t) = \{(\biguplus Z) \uplus (\biguplus X^2_{xt}) \uplus (\biguplus X_t) \mid \begin{array}{l} Z \in \wp_m(X_x),\\ |Z| = \chi_M(X_t,t) = \chi_m(X_t,t),\\ \|Z\| = X_x\}; \end{array}$$

(5) otherwise $res(X,x,t) = \emptyset$.

Proof
By Theorem 5.5, we only need to show that

$$\downarrow\{\biguplus Y \mid Y \in \wp_m(S'), n \in \chi(Y,x) \cap \chi(Y,t).\ n \geqslant |Y|-1\} = \downarrow \bigcup_{X \subseteq S'} res(X,x,t), \quad \text{(B1)}$$

where $S' = rel(S,x,t)$. We prove the two different inclusions separately.

Left to right inclusion. Let $\bar{o} \in res(X,x,t)$ for some $X \subseteq rel(S,x,t)$. We want to prove that there exist $Y \in \wp_m(S')$ and $n \in \chi(Y,x) \cap \chi(Y,t)$ such that $n \geqslant |Y|-1$ and $\uplus Y = \bar{o}$. We distinguish several cases.

- If X is nonlinear for x and t, it is $\uplus X^2 = \bar{o}$. We distinguish two subcases:
  (i) if $\chi_M(X,t) = \infty$, it is enough to take $Y = X \uplus X$.
  (ii) if $\chi_M(X,t)$ is finite, since X is nonlinear for t, there exists $o' \in X$ such that $\chi_m(o',t) > 1$. Since S' is downward closed, consider $o \in S$ such that $o(x) = \min(o'(x),1)$ and $o(v) = o'(v)$ if $v \neq x$. We show that there exists a natural number n such that for $Y = X \uplus X \uplus n\{\!\{o\}\!\}$, we have $\chi_m(Y,t) \geqslant \chi_m(Y,x)$ and $\chi_m(Y,t) \geqslant |Y|-1$. Since $\chi_m(Y,x) \leqslant 2\chi_m(X,x)+n$, we need to solve the inequalities $2\chi_m(X,t) + n\chi_m(o,t) \geqslant 2\chi_m(X,x)+n$ and $2\chi_m(X,t) + n\chi_m(o,t) \geqslant 2|X| + n$. Since $\chi_m(o,t) \geqslant 2$, there always exists a

solution for n. Since $\chi_M(X,x) = \infty$, we have that $\biguplus Y = \bar{o}$ is on the left-hand side of (B1).

- If X is nonlinear for x and linear for t. We need to find m such that if we take $Y = X_x \uplus 2X_{xt} \uplus 2mX_t$, we have $\chi_m(Y,t) \geqslant \chi_m(Y,x)$. In other words, we need to solve the disequation $2\chi_m(X_{xt},t) + 2m\chi_m(X_t,t) \geqslant \chi_m(X_x,x) + 2\chi_m(X_{xt},x)$, which is always possible, since $|X_t| \geqslant 1$. Since $|Y| \leqslant 1 + 2|X_{xt}| + 2m|X_t|$ we have $\chi_m(X,t) \geqslant |Y| - 1$.
- If X is linear for x and strongly nonlinear for t, we distinguish two subcases:
  (i) $\chi_M(X,t) = \infty$. Let $n = 2\chi_m(X_{xt},t) + \chi_m(X_t,t)$ and consider any number m such that $2m|X_x| + 2|X_{xt}| \geqslant n$. (Such an m always exists, since $|X_x| \geqslant 1$.) Then, consider the multiset $Y = 2mX_x \uplus 2X_{xt} \uplus X_t$, and we have that $\chi_m(Y,x) = \chi_M(Y,x) = 2m|X_x| + 2|X_{xt}| \geqslant \chi_m(Y,t)$ by construction. Moreover $\chi_M(Y,t) = \infty$ and $|Y| \leqslant 2m|X_x| + 2|X_{xt}| + 1$. Then $\biguplus Y \in res(X,x,t)$ is a valid resultant sharing group.
  (ii) $\chi_M(X,t)$ is finite. Let $o \in X_{xt}$ be a sharing group such that $\chi_M(o,t) > 1$ and o' be a generic sharing group in X_x. We need to find two natural numbers n and m such that if we take $Y = 2X_x \uplus 2X_{xt} \uplus X_t \uplus m\{\!\{o\}\!\} \uplus n\{\!\{o'\}\!\}$, we obtain $\chi_m(Y,x) = \chi_m(Y,t)$ (from which $\chi_M(Y,x) = \chi_M(Y,t)$ immediately follows) and $\chi_m(Y,x) \geqslant |Y| - 1$. This means we need to solve the equations

  $$2|X_x| + 2|X_{xt}| + m + n = 2\chi_m(X_{xt},t) + \chi_m(X_t,t) + m\chi_m(o,t),$$
  $$2|X_x| + 2|X_{xt}| + m + n \geqslant 2|X_x| + 2|X_{xt}| + |X_t| + m + n - 1.$$

  Since $|X_t| \leqslant 1$, the second equation is always satisfied. A solution for the first equation always exists, since the greatest common divisor of $\chi_m(o,t) - 1$ and 1 is 1.
- If X is linear for x and X is not strongly nonlinear for t, consider the multiset $Y = Z \uplus X_{xt} \uplus X_{xt} \uplus X_t$. Then $\chi_m(Y,x) = \chi_M(Y,x) = |Z| + 2|X_{xt}|$ and $\chi_m(Y,t) = \chi_M(Y,t) = 2|X_{xt}| + \chi_m(X_t,t)$. Since $|Z| = \chi_m(X_t,t)$, we have that $\chi_m(Y,x) = \chi_m(Y,t)$. Moreover, $|Y| = |Z| + 2|X_{xt}| + |X_t| \leqslant \chi_m(X_t,t) + 2|X_{xt}| + 1 = \chi_m(Y,t) + 1$.

Right to left inclusion. Let $o = \biguplus X$, where $X \in \wp_m(S')$ and there exists $n \geqslant |X| - 1$ such that $n \in \chi(X,x) \cap \chi(X,t)$. We show that there exists $Y \sqsubseteq S'$ and $o' \in res(Y,x,t)$ such that $o' \geqslant_2 o$. Let $k = |X|$. We partition X in three multisets $X_x = X|_{\{o \mid \chi_M(o,t)=0\}}$, $X_t = X|_{\{o \mid \chi_M(o,x)=0\}}$, and $X|_{xt} = X|_{\{o \mid \chi_M(o,t)>0 \wedge \chi_M(o,t)>0\}}$. Note that X_x, X_t, and X_{xt} here are multisets and not ordinary set as in the definition of mgu_2. We distinguish several cases.

- If $\llbracket X \rrbracket$ is linear for x and strongly nonlinear for t, then $\chi_m(X,x) = \chi_M(X,x) = |X_x| + |X_{xt}| \leqslant k$. Since $\chi_m(X,x) \geqslant k - 1$, there are two cases: either $|X_x| + |X_{xt}| = k - 1$ or $|X_x| + |X_{xt}| = k$, which implies that $|X_t| \leqslant 1$.
  Since $\llbracket X \rrbracket$ is strongly nonlinear for t, there exists $o'' \in X_t \uplus X_{xt}$ such that $\chi_M(o'',t) \geqslant 2$, and thus $\chi_m(X,t) \geqslant 2$. Therefore $\chi_m(X,t) > |X_{xt}|$. Since $\chi_m(X,x) = \chi_M(X,x) \geqslant \chi_m(X,t)$, we have that $|X_x| \geqslant 1$. It follows that $o = \biguplus(X_x \uplus X_{xt} \uplus X_t) \leqslant_2 (\biguplus \llbracket X_x \rrbracket)^2 \uplus (\biguplus \llbracket X_{xt} \rrbracket)^2 \uplus (\biguplus \llbracket X_t \rrbracket) \in res(\llbracket X \rrbracket, x, t)$.

- If $\llbracket X \rrbracket$ is linear for x and not strongly nonlinear for t, then, as in the previous case we have $|X_t| \leqslant 1$. Since X is not strongly nonlinear for t, $\chi_M(X,t) = \chi_m(X,t) = |X_{xt}| + \chi_M(X_t,t)$. Moreover, $\chi_M(X,x) = \chi_m(X,x) = |X_x| + |X_{xt}|$. By the condition $n \in \chi(X,x) \cap \chi(X,t)$, we get $\chi_M(X_t,t) = |X_x|$. Therefore $o \leqslant_2 \biguplus \llbracket X_x \rrbracket \uplus (\biguplus \llbracket X_{xt} \rrbracket)^2 \uplus (\biguplus X_t) \in res(\llbracket X \rrbracket, x, t)$.
- If $\llbracket X \rrbracket$ is nonlinear for x and t, then $o \leqslant_2 (\biguplus \llbracket X \rrbracket)^2 \in res(\llbracket X \rrbracket, x, t)$.
- If $\llbracket X \rrbracket$ is nonlinear for x and linear t, the proof is symmetric to the one of the first case. $\square$

Theorem 5.12
Given $[S]_U \in \mathtt{ShLin}^2$ and the binding x/t with $vars(x/t) \subseteq U$, we have

$$\mathrm{mgu}_2([S]_U, x/t) = [(S \setminus S') \cup \downarrow \bigcup_{X \sqsubseteq \max S'} (res(X,x,t) \cup res'(X,x,t))]_U,$$

where $S' = rel(S,x,t)$ and

$$res'(X,x,t) = \begin{cases} \{\biguplus X^2\} & \text{if } X = X_{xt} \text{ and } l(X) \text{ is linear for } t, \\ \emptyset & \text{otherwise.} \end{cases}$$

Proof
It clearly holds that

$$\mathrm{mgu}_2([S]_U, x/t) \supseteq [(S \setminus S') \cup \downarrow \bigcup_{X \sqsubseteq \max S'} (res(X,x,t) \cup res'(X,x,t))]_U, \tag{2}$$

since, for each $X \sqsubseteq \max S'$, if $res'(X,x,t)$ is nonempty, then $\biguplus X^2$ may be generated by the characterization in Theorem 5.9. It is enough to take $X' = \{l(o) \mid o \in X\}$; hence $\biguplus X' = \biguplus X^2 \in res(X',x,t)$ according to the last case of Theorem 5.9.

We prove the opposite inclusion. Let $X \sqsubseteq S'$ and assume that $X \not\sqsubseteq \max S'$. There exists $X' \sqsubseteq \max S'$ obtained by replacing each $a \in X$ with $b \in \max S'$ such that $a \leqslant_2 b$. We have that $|X'| \leqslant |X|$, since two different elements in X may be replaced with the same maximal element in X'. We want to prove that either $res(X,x,t) = \emptyset$ or $res(X,x,t) \subseteq \downarrow res(X',x,t)$ or $res(X,x,t) \subseteq \downarrow res'(X',x,t)$. Therefore, we assume that $res(X,x,t) \neq \emptyset$ and compare the linearity properties (linear, nonlinear, strongly nonlinear) of X' w.r.t. those of X.

If they coincide, then it follows that $res(X,x,t) \subseteq \downarrow res(X',x,t)$. This happens because both $res(X,x,t)$ and $res(X',x,t)$ are obtained by the same case of Theorem 5.9. However, note that X' may have less elements than X, and therefore some variable which is nonlinear in $res(X,x,t)$ could be linear in $res(X',x,t)$. Actually, this never happens, since the elements in X' which are not explicitly delinearized are either elements of the multiset Z in the third case of Theorem 5.9 (and therefore may appear multiple times) or elements of X_t (X_x) subject to the condition $|X_t| \leqslant 1$ ($|X_x| \leqslant 1$).

Assume that the linearity properties of X and X' do not coincide. The only interesting case is when X is linear for x and not strongly nonlinear for t. In all the other cases, it is immediate from the definition that $res(X,x,t) \subseteq \downarrow res(X',x,t)$.

If X' is not linear for x and for t, then it holds $res(X,x,t) \subseteq \downarrow res(X',x,t)$ by definition.

If X' is linear for x and strongly nonlinear for t, then it is immediate from the definition that $res(X,x,t) \subseteq \downarrow res(X',x,t)$, provided that $|X_x| \geqslant 1$. Otherwise, it must be $|X_t| = 0$, and therefore, in order to be $res(X,x,t) \neq \emptyset$, we have $X = X_{xt}$ and $\chi_M(X,t) = 1$, which means $l(X') = X$ is linear for t. It follows that $res(X,x,t) = \{\biguplus X^2\} = res'(X',x,t)$.

If X' is not linear for x and linear for t, we show that $|X_x| \leqslant 1$. Assume, by contradiction, that $|X_x| > 1$. Since X' is linear for t and $|X_t| \leqslant 1$, $\chi_M(X_t,t) = \chi_m(X_t,t) \leqslant 1$, while $|\llbracket Z \rrbracket| = |X_x| > 1$, which is a contradiction. Thus it must be $|X_x| \leqslant 1$. If $|X_x| = 0$, then $|X_t| = 0$; hence $res(X,x,t) = \{\biguplus X^2\}$ and $res(X,x,t) = res'(X,x,t)$. If $|X_x| = 1$, since X' is linear for t, it follows that $|Z| = 1$. Thus $res(X,x,t) \subseteq \downarrow res(X',x,t)$. □

Theorem 5.16
The operator mgu_{sl} in Definition 5.15 is correct and optimal w.r.t. mgu, when $vars(x/t) \subseteq U$.

Proof
It is enough to prove that mgu_{sl} is correct and optimal w.r.t. mgu_2, namely, that

$$\mathrm{mgu}_{sl}([S,L,U]),x/t) = \alpha_{sl}(\mathrm{mgu}_2(\gamma_{sl}([S,L,U]),x/t)).$$

Let $\gamma_{sl}([S,L,U]) = [T]_U$. By Theorem 5.12, it holds that

$$\begin{aligned}
&\alpha_{sl}(\mathrm{mgu}_2(\gamma_{sl}([S,L,U]),x/t)\\
&\quad = \alpha_{sl}([(T \setminus T') \cup \downarrow \bigcup_{Y \subseteq \max T'} (res(Y,x,t) \cup res'(Y,x,t)\})]_U\\
&\quad = \alpha_{sl}\big([T \setminus T']_U \sqcup_2 \bigsqcup_{Y \subseteq \max T'} ([\downarrow res(Y,x,t)]_U \sqcup_2 [\downarrow res'(Y,x,t)\}]_U)\big),
\end{aligned}$$

where $T' = rel(T,x,t)$ and $\sqcup_2$ is the lowest upper bound in $\mathtt{ShLin}^2$. By additivity of α_{sl}, this is equivalent to

$$\alpha_{sl}([T \setminus T']_U) \sqcup_{sl} \bigsqcup_{Y \subseteq \max T'} (\alpha_{sl}([res(Y,x,t)]_U) \sqcup_{sl} \alpha_{sl}([res'(Y,x,t)]_U)). \qquad (3)$$

Let X, L', U', and K as in Definition 5.15, we have that $\mathrm{mgu}_{sl}([S,L,U],x/t)$ is equivalent to

$$[(S \setminus X) \cup K, U' \cup L', U]. \qquad (4)$$

We need to prove that equations (3) and (4) do coincide. In the rest of the paper, we assume that the result of (3) is $[S'',L'',U]$.

Sharing. We first prove that the `Sharing` components of the two equations are equal, i.e., $S'' = (S \setminus X) \cup K$. Given $B \in S''$, there are several cases. If $B = \llbracket o \rrbracket$ with $o \in T \setminus T'$, then $B \in S \setminus X$.

If $B = \llbracket o \rrbracket$, for $o \in res'(Y,x,t)$ with $Y \subseteq \max T'$, then $B = \bigcup\{\llbracket o \rrbracket \mid o \in Y\}$ with $Y = Y_{xt}$ and $l(Y)$ is linear for t. If $x \in L$, then B is generated by $(X^U_{xt})^+$, since $l(Y)$ is linear for t. If $x \notin L$ there are two cases: if Y is linear for t, then it is generated by $(X^{=1}_{xt})^+$, otherwise by $\mathrm{bin}(X^{>1}_t \cup X^{>1}_{xt}, X_x \cup X_{xt}, X^*)$. Thus $B \in K$.

Now, assume that $B = \llbracket o \rrbracket$ with $o \in res(Y,x,t)$ and $\emptyset \neq Y \subseteq \max T'$. Then $B = \bigcup W$, where $W = \{\llbracket o \rrbracket \mid o \in Y\}$. Since Y is made of maximal elements and

$[T]_U = \gamma_2([S, L, U])$, we have that Y is linear for x iff $x \in L$. For the same reason, Y is linear for t iff (W, L) is linear for t. As a consequence, if Y is nonlinear for t, then (X, L) is nonlinear for t.

We proceed by cases.

Y **nonlinear for** x **and** t**.** Then $res(Y, x, t) = \{\biguplus Y^2\}$. Since (X, L) is nonlinear for x and t, we have $X_t^{>1} \cup X_{xt}^{>1} \neq \emptyset$ and $X_x \cup X_{xt} \neq \emptyset$. Thus $B \in \text{bin}(X_t^{>1} \cup X_{xt}^{>1}, X_x \cup X_{xt}, X^*) \subseteq K$.

Y **nonlinear for** x **and linear for** t**.** By hypothesis $|Y_x| \leqslant 1$ and $|Y_t| \geqslant 1$; hence $o = (\biguplus Y_x) \uplus (\biguplus Y_{xt}^2) \uplus (\biguplus Y_t^2)$ and

$$B \in \text{bin}((X_t^{=1})^+, X_x \cup X_{xt}^{=1}, (X_{xt}^{=1})^*) \subseteq K.$$

In particular, $B \in \text{bin}((X_t^{=1})^+, X_x, (X_{xt}^{=1})^*)$ when $|Y_x| = 1$; otherwise $B \in \text{bin}((X_t^{=1})^+, X_{xt}^{=1}, (X_{xt}^{=1})^*)$.

Y **linear for** x **and strongly nonlinear for** t**.** In this case we have that $o = (\biguplus Y_x^2) \uplus (\biguplus Y_{xt}^2) \uplus (\biguplus Y_t)$ with $|Y_x| \geqslant 1$ and $|Y_t| \leqslant 1$. By definition of strong nonlinearity, we have two cases:

- There exists $o \in Y_{xt}$ such that $\chi_M(o, t) > 1$. In this case
$$B \in \text{bin}(X_t \cup \{\emptyset\}, X_{xt}^{>1}, X_x^+, X_{xt}^*) \subseteq K.$$
- There exists $o \in Y_t$ such that $\chi_M(o, t) = \infty$. In this case
$$B \in \text{bin}(X_t^{=\infty}, X_x^+, X_{xt}^*) \subseteq K.$$

Y **linear for** x **and non strongly nonlinear for** t**.** In this case

$$o = (\biguplus Z') \uplus (\biguplus Y_{xt}^2) \uplus (\biguplus Y_t),$$

with $|Y_t| = 1$, for some $Z' \in \wp_m(Y_x)$ such that $|Z'| = \chi_m(Y_t, t)$ and $\llbracket Z' \rrbracket = Y_x$. It is obvious that

$$B \in \text{bin}(\{\{o\} \cup (\cup Z) \mid o \in X_t^{\in \mathbb{N}}, Z \subseteq X_x, 1 \leqslant |Z| \leqslant \chi_M^L(o, t)\}, (X_{xt}^{=1})^*) \subseteq K,$$

by choosing $Z = \{\llbracket o \rrbracket \mid o \in Z'\}$.

This proves that if $B \in S''$, then $B \in (S \setminus X) \cup K$. Now, we need to prove the converse implication. If $B \in S \setminus X$, then $B = \llbracket o \rrbracket$ for some $o \in T$, and it is obvious that $o \in T \setminus T'$; hence $B \in S''$.

Therefore, assume that $B \in K$, and consider the case in which $x \in L$ and $B \in \text{bin}(X_t^{=\infty}, X_x^+, X_{xt}^*)$. We have that $B = A \cup (\cup A') \cup (\cup A'')$ for some $A \in X_t^{=\infty}$, A' nonempty subset of X_x and $A'' \subseteq X_{xt}$. We may find $o' \in \max T'$, $Y', Y'' \subseteq \max T'$ such that $\llbracket o' \rrbracket = A$, $\llbracket Y' \rrbracket = A'$, and $\llbracket Y'' \rrbracket = A''$. We have that $Y''' = \{o'\} \cup Y' \cup Y''$ is linear for x and strongly nonlinear for t (due to the element o'), with $|Y_x'''| \geqslant 1$ and $|Y_t'''| \leqslant 1$. Therefore, we may apply the definition of res to obtain $res(Y''', x, t) = \{o\}$ with $\llbracket o \rrbracket = B$; hence $B \in S''$.

With similar reasonings, we may prove that for every $B \in K$, we have $B \in S''$. In particular, the second line of (6) corresponds to the case in which we choose a Y''' which is linear for x and strongly nonlinear for t, due to an element $o \in Y_{xt}'''$ which

$\chi_M(o,t) > 1$; the third line of (6) corresponds to the case that Y''' is linear for X and is not strongly nonlinear for t; the first line of (7) corresponds to the case that Y''' is nonlinear for both x and t; the second line of (7) corresponds to the case that Y''' is linear for t and nonlinear for x.

Finally, if $x \notin L$ and $B \in (X_{xt}^{=1})^+$, it is possible that B cannot be obtained as $res(Y''', x, t)$ for any $Y''' \subseteq \max T'$. However, B may be obtained as $res'(Y''', x, t)$, choosing Y''' as in the previous cases. The same happens if $x \in L$ and $B \in (X_{xt}^U)^+$.

Linearity. We want to prove that $L'' = L' \cup U'$. First of all, let us define $L''_g = U \setminus vars(\text{mgu}_2([T]_U, x/t))$ the set of ground variables in $\text{mgu}_2([T]_U, x/t)$; hence $L''_g \subseteq L''$. We are going to prove that $U' = L''_g$ and $L' \setminus U' = L'' \setminus L''_g$. The first equality trivially follows from the fact that the sharing component of mgu_{sl} is optimal; hence a variable occurs in a sharing group of $S \setminus S \cup K$ iff it occurs in a 2-sharing group of $\text{mgu}_2([T]_U, x/t)$.

Now, we consider a variable $v \in U \setminus U'$ and prove that $v \in L'$ iff $v \in L''$. There are several cases. If we assume that $v \notin L$, by (8) we have $v \notin L'$. Moreover, if $Y \in \max T'$ and $v \in \llbracket Y \rrbracket$, by maximality of Y we have $Y(v) = \infty$. Hence, by Theorem 5.12, we have $v \notin L''$. If we assume that $v \notin X$, by (8) we have $v \in L'$ iff $v \in L$. Since $vars(X) = vars(T)$, we also have $v \in L''$ iff $v \in L$ and therefore $v \in L'$ iff $v \in L''$.

The only case it remains to prove is $v \in vars(X) \cap L$ which, combined with the condition $v \notin U'$, gives $v \in vars(K) \cap L$. First of all, note that if $v \in vars(X_{xt})$, then $v \notin L'$ (by definition of L') and $v \notin L''$ (since X_{xt}) appears delinearized in every 2-sharing group resulting from *res* or *res'*. If $v \notin vars(X_{xt})$, we distinguish four subcases.

- $x \in L$ and (S, L) linear for t. Given $Y \subseteq \max T'$, checking the fourth case of Theorem 5.9 when $\chi_M(Y_t, t) = 1$, we have that $res(Y, x, t)$ is not linear for v iff $v \in vars(Y_{xt})$ or $v \in vars(Y_x) \cap vars(Y_t)$. Note that there exists $Y \subseteq \max T'$ s.t. $v \in vars(Y_{xt}) \cup (vars(Y_x) \cap vars(Y_t))$ iff $v \in vars(T'_{xt}) \cup (vars(T'_x) \cap vars(T'_t))$. Finally $v \in L''$ iff $v \in vars(T'_{xt}) \cup (vars(T'_x) \cap vars(T'_t))$ iff $v \in (X_{xt} \cup (X_x \cap X_t))$ iff $v \in L'$.
- $x \in L$ and (S, L) not linear for t. Given $Y \subseteq \max T'$, checking the third and fourth cases (when $\chi_M(Y, t) > 1$) of Theorem 5.9, we have that $res(Y, x, t)$ nonlinear for v implies $v \in vars(Y_{xt})$ or $v \in vars(Y_x)$, which is equivalent to $v \in X_{xt} \cup X_x$, i.e., $v \notin L'$. On the other hand, if $v \in X_x$, we distinguish the following cases:
  - (i) (S, L) strongly nonlinear for t. There exists $o \in T'$ such that $\chi_M(o, t) = \infty$ or $o \in T'_{xt}$ such that $\chi_M(o, t) > 1$. Moreover, there exists $o' \in T'_x$ such that $v \in \llbracket o' \rrbracket$. If we take $Y = \{o, o'\}$, we have that $res(Y, x, t)$ is not linear for v; hence $v \notin L''$.
  - (ii) (S, L) is not strongly nonlinear for t. There exists $o \in T'_t$ such that $1 < \chi_M(o, t) < \infty$. Moreover, there exists $o' \in T'_x$ such that $v \in \llbracket o' \rrbracket$. If we take $Y' = \{o, o'\}$, by the fourth case in the definition of *res*, we have $res(Y, x, t)$ is not linear for v, i.e., $v \notin L''$.

- $x \notin L$ and (S, L) linear for t. If $v \notin L''$, then $v \in vars(Y_{xt})$ or $v \in vars(Y_t)$. This implies $v \in X_{xt} \cup X_t$, i.e., $v \notin L'$. On the other hand, if $v \in X_t$, there exist $o \in T'_x$ such that $\chi_M(o, x) = \infty$ and $o' \in T'_t$ such that $v \in \llbracket o' \rrbracket$. By definition of *res*, we have that $res(\{o, o'\}, x, y)$ is not linear for v; hence $v \notin L''$.
- $x \notin L$ and (S, L) nonlinear for t. Since $L' = L \setminus X$, it is obvious that $v \notin L'$. Moreover, there exist $o \in T'$ such that $\chi_M(o, x) = \infty$, $o' \in T'$ such that $\chi_M(o, t) > 1$ and $o'' \in T'$ such that $v \in \llbracket o'' \rrbracket$. Note that it is possible that $o = o' = o''$. By definition, we have $res(\{o, o', o''\}, x, t)$ is not linear for v; hence $v \notin L''$. □

Theorem 5.19
The operator mgu_{sl} in Definition 5.18 is the optimal abstraction of mgu.

Proof
First of all, given a finite set of variables V, let us define the extension operator $\text{ext}_V : \texttt{ShLin}^2 \to \texttt{ShLin}^2$ such that $\text{ext}_V([S]_U) = [S \cup \{v \mid v \in V \setminus U\}]_{U \cup V}$. Given $V = vars(x/t) \setminus U$, we have that

$$\alpha_{sl}(\text{mgu}_2(\gamma_{sl}([S, L, U], x/t)) = \alpha_{sl}(\text{ext}_V(\text{mgu}_2(\gamma_{sl}([S, L, U])), x/t)).$$

By Theorem 5.16 we have that

$$\begin{aligned} \text{mgu}_{sl}([S, L, U], x/t) &= \text{mgu}_{sl}([S \cup V, L \cup V, U \cup V], x/t) \\ &= \alpha_{sl}(\text{mgu}_2(\gamma_{sl}([S \cup V, L \cup V, U \cup V], x/t))). \end{aligned}$$

Hence, it is enough to prove that

$$\text{ext}_V(\gamma_{sl}([S, L, U]) = \gamma_2([S \cup V, L \cup V, U \cup V]).$$

By definition of γ_2, we have that

$$\begin{aligned} &\gamma_{sl}([S \cup V, L \cup V, U \cup V]) \\ &\quad = [\{B_{L \cup V} \mid B \in S\} \cup \{B_{L \cup V} \mid B \in V\}]_{U \cup V} \\ &\quad = [\{B_L \mid B \in S\} \cup V]_{U \cup V} \quad [\text{since } v_{L \cup V} = v] \\ &\quad = \text{ext}_V(\gamma_{sl}([S, L, U]), \end{aligned}$$

which completes the proof. □

References

AMATO, G. AND SCOZZARI, F. 2002. Optimality in goal-dependent analysis of sharing. In *Proc. of the Joint Conference on Declarative Programming (AGP'02)*, J. J. Moreno-Navarro and J. Mariño-Carballo, Eds. Universidad Politécnica de Madrid, Madrid, 189–205.

AMATO, G. AND SCOZZARI, F. 2003. A general framework for variable aliasing: Towards optimal operators for sharing properties. In *Logic Based Program Synthesis and Transformation 12th International Workshop, LOPSTR 2002, Madrid, Spain, September 17–20, 2002. Revised Selected Papers*, M. Leuschel, Ed. Lecture Notes in Computer Science, vol. 2664. Springer, Berlin Heidelberg, 52–70.

AMATO, G. AND SCOZZARI, F. 2005. *On Abstract Unification for Variable Aliasing*, Technical Report TR-05-08. Dipartimento di Informatica, Università di Pisa.

Amato, G. and Scozzari, F. 2009. Optimality in goal-dependent analysis of sharing. *Theory and Practice of Logic Programming 9*, 5 (September), 617–689.

Armstrong, T., Marriott, K., Schachte, P. and Søndergaard, H. 1994. Boolean functions for dependency analysis: Algebraic properties and efficient representation. In *Static Analysis, 1st International Static Analysis Symposium, SAS'94 Namur, Belgium, September 28–30, 1994, Proc.*, B. Le Charlier, Ed. Lecture Notes in Computer Science, vol. 864. Springer, Berlin Heidelberg, 266–280.

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

Bagnara, R., Zaffanella, E. and Hill, P. M. 2005. Enhanced sharing analysis techniques: A comprehensive evaluation. *Theory and Practice of Logic Programming 5*, 1–2 (January), 1–43.

Bueno, F., Cabeza, D., Carro, M., Hermenegildo, M. V., López-García, P. and Puebla, G. 1997. *The Ciao Prolog System: Reference Manual* [online], Technical Report CLIP3/97. School of Computer Science, Technical University of Madrid (UPM). Accessed 14 July 2009. URL: `http://www.ciaohome.org/`

Bueno, F. and García de la Banda, M. J. 2004. Set-sharing is not always redundant for pair-sharing. In *Functional and Logic Programming, 7th International Symposium, FLOPS 2004, Nara, Japan, April 7–9, 2004, Proc.*, Y. Kameyama and P. J. Stuckey, Eds. Lecture Notes in Computer Science, vol. 2998. Springer, Berlin Heidelberg, 117–131.

Codish, M., Dams, D. and Yardeni, E. 1991. Derivation and safety of an abstract unification algorithm for groundness and aliasing analysis. In *Logic Programming, Proc. of the Eighth International Conference*, K. Furukawa, Ed. MIT Press, Cambridge, MA, 79–93.

Codish, M., Lagoon, V. and Bueno, F. 2000. An algebraic approach to sharing analysis of logic programs. *The Journal of Logic Programming 42*, 2 (February), 110–149.

Codish, M., Marriott, K. and Taboch, C. 2000. Improving program analyses by structure untupling. *The Journal of Logic Programming 43*, 3 (June), 251–263.

Codish, M., Søndergaard, H. and Stuckey, P. J. 1999. Sharing and groundness dependencies in logic programs. *ACM Transactions on Programming Languages and Systems 21*, 5 (September), 948–976.

Cortesi, A. and Filé, G. 1999. Sharing is optimal. *The Journal of Logic Programming 38*, 3 (March), 371–386.

Cortesi, A., Filé, G., Giacobazzi, R., Palamidessi, C. and Ranzato, F. 1997. Complementation in abstract interpretation. *ACM Transactions on Programming Languages and Systems 19*, 1 (January), 7–47.

Cortesi, A., Filé, G. and Winsborough, W. W. 1998. The quotient of an abstract interpretation. *Theoretical Computer Science 202*, 1–2 (July), 163–192.

Cousot, P. and Cousot, R. 1979. Systematic design of program analysis frameworks. In *POPL '79: Proc. of the 6th ACM SIGACT-SIGPLAN Symposium on Principles of programming languages*. ACM Press, New York, 269–282.

Cousot, P. and Cousot, R. 1992a. Abstract interpretation and applications to logic programs. *The Journal of Logic Programming 13*, 2–3 (July), 103–179.

Cousot, P. and Cousot, R. 1992b. Abstract interpretation frameworks. *Journal of Logic and Computation 2*, 4 (August), 511–549.

Cousot, P. and Cousot, R. 1992c. Comparing the Galois connection and widening/narrowing approaches to abstract interpretation [Invited paper]. In *Programming Language Implementation and Logic Programming, 4th International Symposium, PLILP '92, Leuven, Belgium, August 26–28, 1992, Proc.*, M. Bruynooghe and M. Wirsing, Eds. Lecture Notes in Computer Science, vol. 631. Springer, Berlin Heidelberg, 269–295.

Hans, W. and Winkler, S. 1992. *Aliasing and Groundness Analysis of Logic Programs through Abstract Interpretation and Its Safety* [online], Technical Report. 92–27.

Technical University of Aachen (RWTH Aachen). Accessed 10 July 2009. URL: http://sunsite.informatik.rwth-aachen.de/Publications/AIB

Hermenegildo, M. V. and Rossi, F. 1995. Strict and nonstrict independent and-parallelism in logic programs: Correctness, efficiency, and compile-time conditions. *The Journal of Logic Programming 22*, 1 (January), 1–45.

Hill, P. M., Zaffanella, E. and Bagnara, R. 2004. A correct, precise and efficient integration of set-sharing, freeness and linearity for the analysis of finite and rational tree languages. *Theory and Practice of Logic Programming 4*, 3 (May), 289–323.

Howe, J. M. and King, A. 2003. Three optimisations for sharing. *Theory and Practice of Logic Programming 3*, 2 (January), 243–257.

Jacobs, D. and Langen, A. 1992. Static analysis of logic programs for independent AND parallelism. *The Journal of Logic Programming 13*, 2–3 (July), 291–314.

Jones, N. D. and Søndergaard, H. 1987. A semantics-based framework for the abstract interpretation of Prolog. In *Abstract Interpretation of Declarative Languages*, S. Abramsky and C. Hankin, Eds. Ellis Horwood, Chichester, UK, 123–142.

King, A. 1994. A synergistic analysis for sharing and groundness which traces linearity. In *Programming Languages and Systems – ESOP '94, 5th European Symposium on Programming, Edinburg, U.K., April 11–13, 1994, Proc.*, D. Sannella, Ed. Lecture Notes in Computer Science, vol. 788. Springer, Berlin Heidelberg, 363–378.

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

Lagoon, V. and Stuckey, P. J. 2002. Precise pair-sharing analysis of logic programs. In *PPDP '02: Proc. of the 4th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming*. ACM Press, New York, 99–108.

Langen, A. 1990. *Static Analysis for Independent And-Parallelism in Logic Programs*, PhD thesis. University of Southern California, Los Angeles.

Levi, G. and Spoto, F. 2003. Pair-independence and freeness analysis through linear refinement. *Information and Computation 182*, 1 (April), 14–52.

Mac Lane, S. 1988. *Categories for the Working Mathematician*, 2nd ed. Graduate Texts in Mathematics, vol. 5. Springer, Berlin Heidelberg.

Marriott, K., Søndergaard, H., and Jones, N. D. 1994. Denotational abstract interpretation of logic programs. *ACM Transactions on Programming Languages and Systems 16*, 3 (May), 607–648.

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

Søndergaard, H. 1986. An application of abstract interpretation of logic programs: Occur check reduction. In *ESOP '86, European Symposium on Programming, Saarbrcken, Federal Republic of Germany, March 17–19, 1986, Proc.*, B. Robinet and R. Wilhelm, Eds. Lecture Notes in Computer Science, vol. 213. Springer, Berlin Heidelberg, 327–338.

Sterling, L. and Shapiro, E. Y. 1994. *The Art of Prolog: Advanced Programming Techniques*, 2nd ed. MIT Press, Cambridge, MA.