Computer Science at Kent

Three Optimisations for Sharing

Jacob M. Howe and Andy King

Technical Report No: 11-01
Date: August 2001

Published by the Computing Laboratory,
University of Kent, Canterbury, Kent CT2 7NF, UK.

Abstract

In order to improve precision and efficiency sharing analysis should track both freeness and linearity. The abstract unification algorithms for these combined domains are suboptimal, hence there is scope for improving precision. This paper proposes three optimisations for tracing sharing in combination with freeness and linearity, and proves their correctness. Correctness is established in the presence of rational trees in terms of a novel connection between equations and sharing abstractions. A method for pruning intermediate sharing abstractions to gain efficiency is also proposed. The optimisations are lightweight and therefore some, if not all, of these optimisations will be of interest to the implementor.

1 Introduction

A set-sharing analyser will usually also track freeness and linearity. This is because freeness and linearity are cheap to maintain and result in more accurate, that is smaller, sharing abstractions which in turn improve the efficiency of the sharing component of abstract unification. However, current abstract unification algorithms for sharing, freeness and linearity are suboptimal. This paper considers how to improve the precision of sharing with freeness and linearity by considering the interaction of these components. These refinements do not incur a significant computational overhead. To this end three optimisations are given, along with examples of where precision is gained. Their cost is discussed and correctness proved.

The first optimisation follows from the observation that the algorithms for pair-sharing with linearity can sometimes out perform set-sharing with linearity (in terms of which pairs of variables may share). This is because of an independence check which pervades the set-sharing literature (from early work [7] to the most recent and comprehensive [1]). This check is in fact redundant. By removing this, the precision of abstract unification is improved, since linearity can be exploited more frequently.

Precision needs to be balanced with computational tractability. One principle of set-sharing is that the number of sharing groups should be minimised. Even if this does not improve precision (in terms of which variables may share), it can still reduce the size of abstractions and thereby improve efficiency and possibly avoid widening. Filé [4] observed that freeness can be used to decompose a sharing abstraction into a set of sharing abstractions which are mutually exclusive. Abstract unification can then be applied to each component and the resulting abstractions merged. This tactic has not been included in analysers owing to its prohibitive cost. The second optimisation is a lightweight refinement of abstract unification inspired by the decomposition. Though not as precise as the full decomposition, it does achieve the necessary balance between cost and benefit.

Thirdly, an optimisation for pruning sharing groups is presented. This tactic demonstrates that sharing in combination with freeness can improve groundness which, in turn, can improve sharing (even in the presence of rational trees). Put another way, it means that any optimal algorithm for sharing, freeness and linearity will have to consider subtle interactions between sharing, freeness and groundness.

A fourth technique is also proposed which can prune the size of inputs to the abstract unification algorithm by considering the grounding behaviour of sets of equations. Reducing the size of the inputs simplifies abstract unification and can thereby improve performance. Whilst the technique does not improve precision in itself, a precision gain might be achieved if widening is avoided.

The correctness of the refined abstract unification algorithms is proved for rational tree constraint solving (as adopted by SICStus Prolog and Prolog-III). Correctness is expressed in terms of a novel concretisation map which characterises equations as their idempotent most general unifiers. This simplifies the correctness arguments. To the best of the authors' knowledge, this is the first proof of correctness for a sharing, freeness and linearity analysis in the presence of rational trees. (Previous work for rational tree unification has either focused on pair-sharing [6] or set-sharing without freeness and linearity [5]).

In summary, this paper provides the implementor with a number of low-cost techniques for improving the precision and efficiency of sharing analyses.

2 Preliminaries

2.1 Trees and terms

Let ε denote the empty sequence, . denote sequence concatenation, and $\|\alpha\|$ denote the length of a sequence $\alpha \in \mathbb{N}^*$. A tree (or term) over an alphabet of symbols F is a partial map $t : \mathbb{N}^* \to F$ such that $t(\alpha) = t$ if $\alpha = \varepsilon$, otherwise $t(\alpha) = t_i(\beta)$ where $\alpha = i.\beta$ and $t = f(t_1, \ldots, t_n)$. Let $T(F)$ and $T^\infty(F)$ denote the set of finite and possibly infinite trees over F. Let U denote a (denumerable) universe of variables such that $F \cap U = \emptyset$, and let $var(t) = \{u \in U \mid \exists \alpha \in \mathbb{N}^*.t(\alpha) = u\}$ where $t \in T^\infty(F \cup U)$. Finally, $|S|$ denotes the cardinality of the set S.

2.2 Substitutions and equations

A substitution is a (total) map $\theta : U \to T^\infty(F \cup U)$ such that $dom(\theta) = \{u \in U \mid \theta(u) \neq u\}$ is finite. A substitution θ can be represented as a finite set $\{x \mapsto \theta(x) \mid x \in dom(\theta)\}$. Let $rng(\theta) = \cup\{var(\theta(u)) \mid u \in dom(\theta)\}$ and let Sub denote the set of substitutions. If $\theta = \{x_i \mapsto t_i\}_{i=1}^n$ then $\theta(t)$ denotes the tree obtained by simultaneously replacing each occurrence of x_i in t with t_i. For brevity, let $\theta(x, \alpha) = t(\alpha)$ where $\theta(x) = t$. An equation e is a pair $(s = t)$ where $s, t \in T^\infty(F \cup U)$. A finite set of equations is denoted E and Eqn denotes the set of finite sets of equations. Also define $\theta(E) = \{\theta(s) = \theta(t) \mid (s = t) \in E\}$. The map $eqn : Sub \to Eqn$ is defined $eqn(\theta) = \{x = t \mid (x \mapsto t) \in \theta\}$. Where $Y \subseteq U$, projection out and projection onto are respectively defined $\exists Y.\theta = \{x \mapsto t \in \theta \mid x \notin Y\}$ and $\overline{\exists} Y.\theta = \exists(U \setminus Y).\theta$. Composition $\theta \circ \psi$ of two substitutions is defined so that $(\theta \circ \psi)(u) = \theta(\psi(u))$ for all $u \in U$. Composition induces the (more general than) relation $\leq$ defined by $\theta \leq \psi$ iff there exists $\delta \in Sub$ such that $\psi = \delta \circ \theta$. A renaming is a substitution $\rho \in Sub$ that has an inverse, that is, there exists $\rho^{-1} \in Sub$ such that $\rho^{-1} \circ \rho = id$. The set of renamings is denoted $Rename$. A substitution θ is idempotent iff $\theta \circ \theta = \theta$, or equivalently, iff $dom(\theta) \cap rng(\theta) = \emptyset$.

2.3 Solved forms and most general unifiers

A substitution is in rational solved form iff it has no subset $\{x_1 \mapsto x_2, \ldots, x_n \mapsto x_1\}$ where $n \geq 2$. The subset of Sub in rational solved form is denoted $RSub$. The set of unifiers of E is defined by: $unify(E) = \{\theta \in Sub \mid \forall (s = t) \in E.\theta(s) = \theta(t)\}$. The set of most general unifiers (mgus) and the set of idempotent mgus (imgus) are defined: $mgu(E) = \{\theta \in unify(E) \mid \forall \psi \in unify(E).\theta \leq \psi\}$ and $imgu(E) = \{\theta \in mgu(E) \mid dom(\theta) \cap rng(\theta) = \emptyset\}$. Note that $imgu(E) \neq \emptyset$ iff $mgu(E) \neq \emptyset$ [8]. An mgu can be renamed to obtain any other (as can an imgu).

Lemma 2.1 (Proposition 11 from [8]) *Let $\theta \in imgu(E)$. Then $\phi \in imgu(E)$ iff there exists $\{x_i \mapsto y_i\}_{i=1}^n \subseteq \theta$ such that $\phi = \{x_i \mapsto y_i, y_i \mapsto x_i\}_{i=1}^n \circ \theta$.*

One way to obtain an imgu is by considering limits of substitutions.

Definition 2.1 Let $\{t_n \mid n \in \mathbb{N}\} \subseteq T^\infty(F \cup U)$. Then $t = \lim_{n\to\infty} t_n$ iff for all $k \in \mathbb{N}$ there exists $l \in \mathbb{N}$ such that for all $m \geq l$ and $\|\alpha\| \leq k$, $t(\alpha) = t_m(\alpha)$. Furthermore, if $\{\theta_n \mid n \in \mathbb{N}\} \subseteq Sub$ then $\lim_{n\to\infty} \theta_n = \lambda x. \lim_{n\to\infty} \theta_n(x)$.

Note that $\lim_{n\to\infty} \theta^n$ exists iff $\theta \in RSub$ [6]. The following lemmas detail how limits of substitutions and composition of substitutions relate to an mgu.

Lemma 2.2 (Lemmas 2.2, 4.3 and 4.4 from [6])

1. $\theta^\infty \in mgu(eqn(\theta))$ if $\theta \in RSub$.

2. $\delta \circ \theta^\infty \in mgu(E \cup eqn(\theta))$ if $\delta \in mgu(\theta^\infty(E))$.

3. $\exists(dom(\theta) \setminus rng(\theta)).\delta \in mgu(\theta(E))$ if $\delta \circ \theta \in mgu(E)$.

2.4 Linearity and alternating paths

Variable multiplicity is defined in order to formalise linearity. The significance of linearity is that unification of linear terms enables sharing to be described by more precise sharing abstractions (even in the presence of rational trees).

Definition 2.2 The variable multiplicity map $\chi : T^\infty(F \cup U) \to \{0,1,2\}$ is defined: $\chi(t) = \max(\{\chi(x,t) \mid x \in U\})$ where $\chi(x,t) = \min(2, |\{\alpha \mid t(\alpha) = x\}|)$.

If $\chi(t) = 0$, t is ground; if $\chi(t) = 1$, t is linear; and if $\chi(t) = 2$, t is non-linear. The next lemma details the forms of sharing barred by the unification of linear terms.

Lemma 2.3 (Proposition 3.1 from [6]) If $\theta \in mgu(\{s = t\})$, $x \neq y$ and $var(\theta(x)) \cap var(\theta(y)) \neq \emptyset$ then either: $x \in var(s)$ and $y \in var(t)$; or $x, y \in var(t)$ and $\chi(s) = 2$; or $x \in var(t)$ and $y \in var(s)$; or $x, y \in var(s)$ and $\chi(t) = 2$.

The correctness arguments for abstract unification require lemma 2.3 to be augmented with a new result – lemma 2.5. Alternating paths build towards this result.

Definition 2.3 (Alternating paths) *If $E \in Eqn$, then the binary relations $-_E, \sim_E \subseteq (E \times \mathbb{N}^*)^2$ are the least symmetric relations such that:*

1. $\langle e, 1.\alpha_1 \rangle -_E \langle e, 2.\alpha_2 \rangle$ *if* $e \in E$, $e(1.\alpha_1) \in U$ *and* $e(2.\alpha_2) \in U$

2. $\langle e_1, \alpha_1 \rangle \sim_E \langle e_2, \alpha_2 \rangle$ *if* $e_1, e_2 \in E$, $e_1(\alpha_1) = e_2(\alpha_2) \in U$ *and* ($e_1 \neq e_2$ *or* $\alpha_1 \neq \alpha_2$).

where $e(1.\alpha) = s(\alpha)$ *and* $e(2.\alpha) = t(\alpha)$ *if* $e = (s = t)$. *A sequence* $\langle e_1, \alpha_1 \rangle . \langle e_1, \beta_1 \rangle \ldots \langle e_n, \alpha_n \rangle . \langle e_n, \beta_n \rangle \in APath_E$ *iff* $\langle e_i, \alpha_i \rangle -_E \langle e_i, \beta_i \rangle$ *for all* $1 \leq i \leq n$ *and* $\langle e_i, \beta_i \rangle \sim_E \langle e_{i+1}, \alpha_{i+1} \rangle$ *for all* $1 \leq i < n$ *where* $n \in \mathbb{N}$.

One key result on alternating paths is that a rational solved form θ exists for an equation E (if one exists at all) whose alternating paths all occur in E. Specifically, if there exists an alternating path in θ whose ends points connect the variables x and y, then the same must be true of E. This is formalised below.

Lemma 2.4 (Lemmas 3.1 and 3.2 from [6])

1. Let $E \in Eqn$ and $unify(E) \neq \emptyset$. There exists $\theta \in RSub$ such that $mgu(E) = mgu(eqn(\theta))$. Moreover, if $\langle e_1', \alpha_1' \rangle \ldots \langle e_m', \beta_m' \rangle \in APath_{eqn(\theta)}$, there exists $\langle e_1, \alpha_1 \rangle \ldots \langle e_n, \beta_n \rangle \in APath_E$ with $e_1(\alpha_1) = e_1'(\alpha_1')$ and $e_n(\beta_n) = e_m'(\beta_m')$.

2. If $\langle e, \alpha_1 \rangle \ldots \langle e, \beta_n \rangle \in APath_{\{e\}}$ where $e = (s = t)$ then either: $\alpha_1 = 1.\alpha$ and $\beta_n = 2.\beta$; or $\alpha_1 = 2.\alpha$, $\beta_n = 2.\beta$ and $\chi(s) = 2$; or $\alpha_1 = 2.\alpha$ and $\beta_n = 1.\beta$; or $\alpha_1 = 1.\alpha$, $\beta_n = 1.\beta$ and $\chi(t) = 2$.

For brevity (to avoid detailing a solved form algorithm) part 1 is presented slightly differently from Lemma 3.1 of [6]. Part 1 and part 2 provide the building blocks for the following lemma which details how non-linearity arises.

Lemma 2.5 If $\theta \in mgu(\{s = t\})$ and $\chi(\theta(x)) = 2$ then either: $x \in var(s) \cap var(t)$; or $x \in var(t)$ and $\chi(s) = 2$; or $x \in var(s)$ and $\chi(t) = 2$.

Proof 2.1 *Let $E = \{s = t\}$. By part 1 of lemma 2.4, there exists $\psi \in RSub$ such that $mgu(E) = mgu(eqn(\psi))$. By part 1 of lemma 2.2, $\psi^\infty \in mgu(E)$, thus there exists $\rho \in Rename$ such that $\rho \circ \theta = \psi^\infty$. Hence $\chi(\psi^\infty(x)) = 2$ and therefore there exists $y \in U$ and $\beta \neq \beta'$ such that $\psi^\infty(x, \beta) = y = \psi^\infty(x, \beta')$. There exists $m \in \mathbb{N}$ such that $\psi^m(x, \beta) = y = \psi^m(x, \beta')$ so there exists $e_1{}' = (x = t_1)$, $e_2{}' = (x_2 = t_2)$, $\ldots$, $e_m{}' = (x_m = t_m)$, $e_1{}'' = (x = t'_1)$, $e_2{}'' = (x'_2 = t'_2)$, $\ldots$, $e_m{}'' = (x'_m = t'_m) \in eqn(\psi)$ where $t_1(\beta_1) = x_2$, $\ldots$, $t_{m-1}(\beta_{m-1}) = x_m$, $t_m(\beta) = y$ and $t'_1(\beta'_1) = x'_2$, $\ldots$, $t'_{m-1}(\beta'_{m-1}) = x'_m$, $t'_m(\beta') = y$. Hence $\langle e_1{}', 1\rangle.\langle e_1{}', 2.\beta_1\rangle \ldots \langle e_m{}', 1\rangle.\langle e_m{}', 2.\beta\rangle.\langle e_m{}'', 2.\beta'\rangle.\langle e_m{}'', 1\rangle \ldots \langle e_1{}'', 2.\beta'_1\rangle.\langle e_1{}'', 1\rangle \in APath_{eqn(\psi)}$. By part 1 of lemma 2.4, there exists $\langle e_1, \alpha_1\rangle \ldots \langle e_n, \beta_n\rangle \in APath_{\{s=t\}}$ with $e_1(\alpha_1) = x = e_n(\beta_n)$. The result follows by part 2 of lemma 2.4.*

2.5 Groundness and sharing abstractions

The abstract domains of interest in this paper are represented either as Boolean functions, or as sets or as sets of sets. Let X denote a finite subset of U. The set of propositional formulae over X is denoted by $Bool_X$ and Y abbreviates the formula $\wedge Y$. The (bijective) map $model_X : Bool_X \to \wp(\wp(X))$ is defined by $model_X(f) = \{M \subseteq X \mid \psi_X(M) \models f\}$ where $\psi_X(M) = M \wedge \wedge\{\neg y \mid y \in X \backslash M\}$. The groundness, sharing, freeness and linearity domains over X are defined as follows:

Definition 2.4 *$Pos_X = \{f \in Bool_X \mid X \models f\}$, $Sh_X = \wp(\wp(X))$, $Fr_X = \wp(X)$ and $Lin_X = \wp(X)$.*

If $S \in Sh_X$, then each $G \in S$ is referred to as a sharing group.

These domains are connected to the concrete domain of sets of equations by Galois connections induced by the concretisation maps. This approach leads to succinct statements of correctness. To obtain well defined concretisations, maps abstracting substitutions are introduced. It is then observed that the abstractions for equivalent idempotent substitutions are the same.

Definition 2.5 *The abstraction maps $\alpha^{Pos} : Sub \to Pos_U$ and $\alpha^{Sh}_X : Sub \to Sh_X$ are defined: $\alpha^{Pos}(\theta) = \wedge\{x \leftrightarrow var(t) \mid x \mapsto t \in \theta\}$, $\alpha^{Sh}_X(\theta) = \{occ(\theta, u) \cap X \mid u \in U\}$ and $occ(\theta, y) = \{u \in U \mid y \in var(\theta(u))\}$.*

Lemma 2.6 *Let $\theta, \phi \in imgu(E)$. Then $\alpha^{Pos}(\theta) = \alpha^{Pos}(\phi)$, $\alpha^{Sh}_X(\theta) = \alpha^{Sh}_X(\phi)$, $\theta(x) \in U$ iff $\phi(x) \in U$ and $\chi(\theta(x)) \leq 1$ iff $\chi(\phi(x)) \leq 1$.*

Proof 2.2 *By lemma 2.1 there exists $\{x_i \mapsto y_i\}_{i=1}^n \subseteq \theta$ such that $\phi = \rho \circ \theta$ where $\rho = \{x_i \mapsto y_i, y_i \mapsto x_i\}_{i=1}^n$.*

1. *Let $x \mapsto t \in \theta$. Observe that $\{x \mapsto \rho(t), y_1 \mapsto x_1, \ldots, y_n \mapsto x_n\} \subseteq \rho \circ \theta$ and $y_i \in var(t)$ iff $x_i \in var(\rho(t))$, thus $\alpha^{Pos}(\phi) \models x \leftrightarrow var(\rho(t)) \wedge (\wedge_{i=1}^n y_i \leftrightarrow x_i) \models x \leftrightarrow var(t)$. Hence $\alpha^{Pos}(\phi) \models \alpha^{Pos}(\theta)$. The other direction is similar.*

2. *Observe that $occ(\rho \circ \theta, y_i) = occ(\theta, x_i)$, $occ(\rho \circ \theta, x_i) = occ(\theta, y_i)$ and $occ(\rho \circ \theta, u) = occ(\theta, u)$ for all $u \notin dom(\rho) \cup rng(\rho)$. Hence $\alpha^{Sh}_X(\theta) = \alpha^{Sh}_X(\phi)$.*

3. and 4. Immediate.

Instead of defining concretisation in terms of a particular imgu (the limit of a rational solved form [6]), an arbitrary imgu is used. This new approach simplifies correctness proofs.

Definition 2.6 *The concretisation maps $\gamma_X^{Pos} : Pos_X \to \wp(Eqn)$, $\gamma_X^{Sh} : Sh_X \to \wp(Eqn)$, $\gamma_X^{Fr} : Fr_X \to \wp(Eqn)$ and $\gamma_X^{Lin} : Lin_X \to \wp(Eqn)$ are respectively defined by:*

$$\begin{aligned} \gamma_X^{Pos}(f) &= \{E \in Eqn \mid \exists \theta \in imgu(E).\alpha^{Pos}(\theta) \models f\} \\ \gamma_X^{Sh}(S) &= \{E \in Eqn \mid \exists \theta \in imgu(E).\alpha_X^{Sh}(\theta) \subseteq S\} \\ \gamma_X^{Fr}(F) &= \{E \in Eqn \mid \exists \theta \in imgu(E).\forall x \in F.\, \theta(x) \in U\} \\ \gamma_X^{Lin}(L) &= \{E \in Eqn \mid \exists \theta \in imgu(E).\forall x \in L.\chi(\theta(x)) \leq 1\} \end{aligned}$$

This paper is concerned with combined domains and the following combined concretisation maps will be useful: $\gamma_X^{SF}(\langle S, F\rangle) = \gamma_X^{Sh}(S) \cap \gamma_X^{Fr}(F)$ and $\gamma_X^{SFL}(\langle S, F, L\rangle) = \gamma_X^{SF}(\langle S, F\rangle) \cap \gamma_X^{Lin}(L)$.

A connection is established in [3] which sheds light on the relationship between sharing and Boolean functions. The corollary (also observed in the long version of [1]) explains how this can be used to improve precision of combined domains.

Lemma 2.7 (Observation 4.1 and lemma 5.1 from [3]) $\{X \setminus G \mid G \in \alpha_X^{Sh}(\theta)\} \subseteq model_X(\alpha^{Pos}(\theta))$ *where θ is idempotent.*

Corollary 2.1 $\gamma_X^{Pos}(f) \cap \gamma_X^{Sh}(S) = \gamma_X^{Pos}(f) \cap \gamma_X^{Sh}(trim_X(f, S))$ *where $trim_X(f, S) = \{G \in S \mid X \setminus G \in model_X(f)\}$.*

Finally, the following auxiliary operations will be used throughout the paper. Let $S, S_i \in Sh_X$. The relevance map is defined $rel(t, S) = \{G \in S | var(t) \cap G \neq \emptyset\}$; closure is defined $S^* = \cap\{S' \mid S \subseteq S' \wedge \forall G_1, G_2 \in S'.G_1 \cup G_2 \in S'\}$; and pair-wise union is defined $S_1 \uplus S_2 = \{G_1 \cup G_2 \mid G_1 \in S_1 \wedge G_2 \in S_2\}$.

3 Independence check in set-sharing

The following example demonstrates that pair-sharing can sometimes detect independence when standard set-sharing unification algorithms cannot.

Example 3.1 *Let $X = \{u, v, w, x, y, z\}$ and consider $E \in \gamma_X^{SFL}(\langle S, F, L\rangle)$ where $S = \{\emptyset, \{u, w\}, \{v, w\}, \{x, y\}, \{x, z\}, \{w, x\}\}$, $F = \emptyset$ and $L = X$. Let $\theta' \in imgu(E \cup \{w = x\})$. The set-sharing unification algorithms of [7, 1] give the following abstraction $S' = \{\emptyset\} \cup (S_w^* \uplus S_x^*)$ for θ' where $S_w = \{\{u, w\}, \{v, w\}, \{w, x\}\}$ and $S_x = \{\{x, y\}, \{x, z\}, \{w, x\}\}$. Observe that $\{u, v, w\} \in S_w^*$ and $\{x, y, z\} \in S_x^*$ and therefore S' does not assert the independence of u and v (similarly y and z). However, if S is interpreted as a set of pairs, then the pair-sharing abstract unification algorithms of [2, 6] both give the abstraction $S \cup \{\{w\}, \{x\}, \{u, x\}, \{u, y\}, \{u, z\}, \{v, x\}, \{v, y\}, \{v, z\}, \{w, y\}, \{w, z\}\}$ which states the independence of u and v (and similarly y and z).*

One intriguing difference between pair-sharing and set-sharing algorithms is that the former does not require the terms in the equation to be independent to exploit linearity. Lemmas 2.3 and 2.5 detail the forms of sharing that can arise in $mgu(\{s = t\})$ rational (and finite) tree unification. Observe that s and t are not required to be independent. Abstract unification algorithms with the independence check are safe. However, this check is not fundamental to combining sharing with linearity. By observing how to exploit linearity more fully, a more precise abstract unification algorithm can be obtained. This algorithm also explains why algorithms with the independence check are safe. The following abstract operator is used to approximate the multiplicity map in abstract unification. Lemma 3.1 asserts its correctness.

Definition 3.1

$$\chi(t, S, L) = \begin{cases} 2 & \text{if } \exists x \in var(S).\chi(x,t) = 2 \\ 2 & \text{if } \exists x \in var(S).x \notin L \\ 2 & \text{if } \exists x, y \in var(t).\exists G \in S.x \neq y \wedge x, y \in G \\ 1 & \text{otherwise} \end{cases}$$

Lemma 3.1 If $E \in \gamma_X^{Sh}(S) \cap \gamma_X^{Lin}(L)$ and $\theta \in imgu(E)$ then $\chi(\theta(t)) \leq \chi(t, S, L)$.

Proof 3.1 *Suppose $\chi(\theta(t)) = 2$. One of the following holds:*

- *There exists $x \in var(t)$ such that $\chi(x,t) = 2$ and $var(\theta(x)) \neq \emptyset$. Then $x \in var(S)$ so that $\chi(t, S, L) = 2$.*
- *There exists $x \in var(t)$ such that $\chi(\theta(x)) = 2$. Then $x \in var(S)$ and $x \notin L$ so that $\chi(t, S, L) = 2$.*
- *There exist $x, y \in var(t)$ such that $x \neq y$ and $var(\theta(x)) \cap var(\theta(y)) \neq \emptyset$. Then there exists $G \in S$ such that $x, y \in G$ so that $\chi(t, S, L) = 2$.*

The revised abstract unification algorithm (with the independence check removed) is detailed in definition 3.2, and theorem 3.1 establishes its correctness.

Definition 3.2 (Abstract unification 1) *Abstract unification $amgu_1(\langle S, F, L\rangle, s, t) = \langle S', F', L'\rangle$ is defined:*

$$S_s = rel(s, S) \quad S_t = rel(t, S) \quad S' = (S \setminus (S_s \cup S_t)) \cup S'' \quad G' = X \setminus var(S')$$

$$S'' = \begin{cases} S_s \uplus S_t & \text{if } s \in F \vee t \in F \\ (S_s^* \uplus S_t) \cap (S_s \uplus S_t^*) & \text{if } \chi(s, S, L) = \chi(t, S, L) = 1 \\ S_s^* \uplus S_t & \text{if } \chi(s, S, L) = 1 \\ S_s \uplus S_t^* & \text{if } \chi(t, S, L) = 1 \\ S_s^* \uplus S_t^* & \text{otherwise} \end{cases}$$

$$F' = \begin{cases} F & \text{if } s \in F \wedge t \in F \\ F \setminus var(S_s) & \text{if } s \in F \\ F \setminus var(S_t) & \text{if } t \in F \\ F \setminus var(S_s \cup S_t) & \text{otherwise} \end{cases}$$

$$L' = F' \cup G' \cup \begin{cases} L \setminus (var(S_s) \cap var(S_t)) & \text{if } \chi(s, S, L) = 1 \wedge \chi(t, S, L) = 1 \\ L \setminus var(S_s) & \text{if } \chi(s, S, L) = 1 \\ L \setminus var(S_t) & \text{if } \chi(t, S, L) = 1 \\ L \setminus var(S_s \cup S_t) & \text{otherwise} \end{cases}$$

A precision gain over previous algorithms follows since a closure is avoided if s is linear but not t (or vice versa) and s and t are not independent. When both s and t are linear, but not independent, two closures are required (as previously), but the resulting sharing abstraction may contain fewer elements owing to the pruning effect of intersection. When the independence check is satisfied, that is $S_s \cap S_t = \emptyset$, it follows that $(S_s^* \uplus S_t) \cap (S_s \uplus S_t^*) = S_s \uplus S_t$. This explains why algorithms with the independence check are safe. Note that if s and t are both linear, but not independent, an implementor might trade precision for efficiency by computing $S_s^* \uplus S_t$ if $|S_s| \leq |S_t|$ and $S_s \uplus S_t^*$ otherwise.

Theorem 3.1 (Correctness of abstract unification 1) Let $E \in \gamma_X^{SFL}(\langle S, F, L\rangle)$, $var(s) \cup var(t) \subseteq X$ and $amgu_1(\langle S, F, L\rangle, s, t) = \langle S', F', L'\rangle$. Then $E \cup \{s = t\} \in \gamma_X^{SFL}(\langle S', F', L'\rangle)$.

Proof 3.2 *Put $E' = \{s = t\}$. Let $\theta \in imgu(E)$ and $\theta' \in imgu(E \cup E')$. Observe that $unify(\theta(E')) \supseteq unify(\theta(E') \cup eqn(\theta)) = unify(E' \cup eqn(\theta)) = unify(E \cup E') \neq \emptyset$. Thus let $\delta \in imgu(\theta(E')) = imgu(\theta^\infty(E'))$. By part 2 of lemma 2.2, $\delta \circ \theta^\infty \in mgu(eqn(\theta) \cup E') = mgu(E \cup E')$. Since $dom(\theta) \cap rng(\delta) = \emptyset$, $\delta \circ \theta^\infty = \delta \circ \theta \in imgu(E \cup E')$.*

1. *To show $\alpha_X^{Sh}(\delta \circ \theta) \subseteq S'$, let $y \in U$ and consider $occ(\delta \circ \theta, y)$.*
   (a) *Suppose $y \notin rng(\delta \circ \theta)$.*
      i. *Suppose $y \notin dom(\delta \circ \theta)$, that is, $\delta \circ \theta(y) = y$. Thus $\theta(y) = y'$ and $\delta(y') = y$. Suppose $y \neq y'$. Then $y \in dom(\theta)$, thus $y \notin rng(\delta)$ which is a contradiction. Therefore $y = y'$, giving $\theta(y) = y$ and $\delta(y) = y$.*
         A. *Suppose $y \notin var(\theta(s))$ and $y \notin var(\theta(t))$. Hence $y \notin dom(\delta)$ and $y \notin rng(\delta)$, so that $occ(\delta \circ \theta, y) \cap X = occ(\theta, y) \cap X \in S$. But $var(s) \cap occ(\theta, y) = \emptyset$ and similarly $var(t) \cap occ(\theta, y) = \emptyset$, so that $occ(\delta \circ \theta, y) \cap X \in S'$.*
         B. *Suppose $y \in var(\theta(s))$ and $y \notin var(\theta(t))$. Since $\delta(y) = y$, it follows that $y \in var(\delta \circ \theta(s)) = var(\delta \circ \theta(t))$. Suppose $y \in rng(\delta)$, then $y \notin dom(\theta)$, hence $y \in rng(\delta \circ \theta)$ which is a contradiction. Therefore $y \notin rng(\delta)$, thus $y \in var(\theta(t))$ which is a contradiction.*
         C. *Suppose $y \notin var(\theta(s))$ and $y \in var(\theta(t))$. Analogous to the previous case.*
         D. *Suppose $y \in var(\theta(s))$ and $y \in var(\theta(t))$. Since $\delta(y) = y$ and $y \notin rng(\delta \circ \theta)$, $y \notin rng(\theta)$. Thus $y \in var(s)$ and $y \in var(t)$. Since $y \notin rng(\theta)$, it follows that $y \notin dom(\theta)$, therefore $y \notin rng(\delta)$. Thus, $occ(\delta \circ \theta, y) = occ(\theta, y)$. Therefore $occ(\delta \circ \theta, y) \cap X \in S_s$ since $var(s) \subseteq X$ and $occ(\delta \circ \theta, y) \cap X \in S_t$ since $var(t) \subseteq X$. Thus $occ(\delta \circ \theta, y) \cap X \in S'$.*
      ii. *Suppose $y \in dom(\delta \circ \theta)$. Since $y \notin rng(\delta \circ \theta)$, $occ(\delta \circ \theta, y) \cap X = \emptyset \in S'$.*
   (b) *Suppose $y \in rng(\delta \circ \theta) \setminus var(\theta(E'))$. Then $y \notin dom(\delta)$ and $y \notin rng(\delta)$ so that $occ(\delta \circ \theta, y) = occ(\theta, y)$. Moreover, since $y \notin var(\theta(E'))$ it follows that $occ(\delta \circ \theta, y) \cap X \in S \setminus (S_s \cup S_t) \subseteq S'$.*
   (c) *Suppose $y \in rng(\delta \circ \theta) \cap var(\theta(E'))$. Since $occ(\delta, y) \subseteq var(\theta(s)) \cup var(\theta(t))$, $occ(\delta \circ \theta, y) \cap X = \cup\{occ(\theta, u) \cap X \mid u \in occ(\delta, y)\} = (\cup R_s) \cup (\cup R_t)$, where $R_s = \cup\{occ(\theta, v) \cap X \mid v \in var(\theta(s)) \cap occ(\delta, y)\}$ and $R_t = \cup\{occ(\theta, w) \cap X \mid w \in var(\theta(t)) \cap occ(\delta, y)\}$. Since $var(s) \subseteq X$, $R_s \subseteq S_s$ and since $var(t) \subseteq X$, $R_t \subseteq S_t$.*
      i. *Suppose $s \in F$. Thus $\theta(s) \in U$, hence $|R_s| = |var(\theta(s))| = 1$. Moreover $\chi(\theta(s)) \leq 1$. Suppose $|R_t| > 1$. Thus there exists $u \neq v$ such that $u, v \in var(\theta(t))$ and $var(\delta(u)) \cap var(\delta(v)) \neq \emptyset$. This contradicts lemma 2.3, hence $|R_t| \leq 1$. Therefore $occ(\delta \circ \theta, y) \cap X \in S_s \uplus S_t$.*

ii. *Suppose $t \in F$. Analogous to the previous case.*

iii. *Suppose $\chi(s, S, L) = 1$. Thus $\chi(\theta(s)) \leq 1$. As with case 1(c)i, it follows that $|R_t| \leq 1$. Thus $occ(\delta \circ \theta, y) \cap X \in S_s^* \uplus S_t$.*

iv. *Suppose $\chi(t, S, L) = 1$. Analogous to the previous case.*

v. *Otherwise $occ(\delta \circ \theta, y) \cap X \in S_s^* \uplus S_t^*$.*

2. *It is straightforward to show $\delta \circ \theta(x) \in U$ for all $x \in F'$.*

3. *To show $\chi(\delta \circ \theta(x)) \leq 1$ for all $x \in L'$. Observe $\chi(\delta \circ \theta(x)) = 0$ if $x \in G'$ and $\chi(\delta \circ \theta(x)) = 1$ if $x \in F'$. Hence, let $x \in L \subseteq X$ and suppose $\chi(\delta \circ \theta(x)) = 2$.*

   (a) *Suppose $\chi(s, S, L) = 1$. By lemma 3.1, $\chi(\theta(s)) \leq 1$.*

      i. *Suppose there exist $u, v \in var(\theta(x))$, $u \neq v$ such that $var(\delta(u)) \cap var(\delta(v)) \neq \emptyset$. By lemma 2.3 either:*

         A. *$u \in var(\theta(s))$ and $v \in var(\theta(t))$, hence $x \in occ(\theta, u) \cap X \in S_s$, and therefore $x \notin L'$.*

         B. *$u \in var(\theta(t))$ and $v \in var(\theta(s))$, hence $x \in occ(\theta, v) \cap X \in S_s$, and therefore $x \notin L'$.*

         C. *$u, v \in var(\theta(s))$. Hence $x \in occ(\theta, v) \cap X \in S_s$, and thus $x \notin L'$.*

      ii. *Suppose there exists $u \in var(\theta(x))$ such that $\chi(\delta(u)) = 2$. By lemma 2.5, $u \in var(\theta(s))$, thus $x \in occ(\theta, u) \cap X \in S_s$ and therefore $x \notin L'$.*

   (b) *Suppose $\chi(t, S, L) = 1$. Analogous to the previous case.*

   (c) *Otherwise observe that either:*

      i. *There exist $u, v \in var(\theta(x))$, $u \neq v$ such that $var(\delta(u)) \cap var(\delta(v)) \neq \emptyset$. Thus $u \in var(\theta(E'))$ and $x \in occ(\theta, u) \cap X \in S_s \cup S_t$. Hence $x \notin L'$.*

      ii. *There exists $u \in var(\theta(x))$ such that $\chi(\delta(u)) = 2$. Thus $u \in var(\theta(E'))$ and $x \in occ(\theta, u) \cap X \in S_s \cup S_t$. Hence $x \notin L'$.*

Example 3.2 *Consider again example 3.1. Observe that $amgu_1(\langle S, F, L\rangle, w, x) = \langle S', F', L'\rangle$ where $S' = \{\emptyset\} \cup (S_w^* \uplus S_x) \cap (S_w \uplus S_x^*) = \{\emptyset, \{u, w, x\}, \{u, w, x, y\}, \{u, w, x, z\}, \{v, w, x\}, \{v, w, x, y\}, \{v, w, x, z\}, \{w, x\}, \{w, x, y\}, \{w, x, z\}\}$, $F' = \emptyset$ and $L' = \emptyset$. This asserts the independence of u and v (similarly y and z), as required.*

The following example, adapted from [7], illustrates that closure can be required to abstract the unification of linear terms.

Example 3.3 *Let $X = \{w, x, y, z\}$ and observe $E \in \gamma_X^{SFL}(\langle S, F, L\rangle)$ where $E = \{w = f(x, y, z)\}$, $S = \{\emptyset, \{w, x\}, \{w, y\}, \{w, z\}\}$, $F = \emptyset$ and $L = \{w, x, y, z\}$. Let $E' = \{w = f(z, x, y)\}$ and note that $\theta' \in imgu(E \cup E')$ where $\theta' = \{w \mapsto f(z, z, z), x \mapsto z, y \mapsto z\}$. Thus $E \cup E' \in \gamma_X^{SFL}(\langle S', F', L'\rangle)$ where $S' = \{\emptyset, \{w, x, y, z\}\}$, $F' = \emptyset$ and $L' = \{x, y, z\}$. Indeed, if $S_s = rel(w, S) = \{\{w, x\}, \{w, y\}, \{w, z\}\}$ and $S_t = rel(f(z, x, y), S) = \{\{w, x\}, \{w, y\}, \{w, z\}\}$ then $(S_s^* \uplus S_t) \cap (S_s \uplus S_t^*) = \{\{w, x\}, \{w, y\}, \{w, z\}, \{w, x, y\}, \{w, x, z\}, \{w, y, z\}, \{w, x, y, z\}\}$, thus $amgu_1(\langle S, F, L\rangle, w, f(z, x, y))$ yields a safe, though conservative, abstraction. Closure is required to construct the $\{w, x, y, z\}$ sharing group.*

4 Decomposition of set-sharing

Filé [4] observes that different sharing and freeness abstractions can represent the same equations, that is, $\gamma_X^{SF}(\langle S_1, F\rangle) = \gamma_X^{SF}(\langle S_2, F\rangle)$ does not imply that $S_1 = S_2$. Therefore the relationship between $Sh \times Fr$ and the concrete domain is a Galois connection rather than an insertion. An insertion is constructed by using F to decompose S into a set of sharing abstractions $K_F(S)$ such that each $B \in K_F(S)$ does not include sharing groups that definitely arise from different computational paths. The following definition and lemma from [4] formalises this decomposition, henceforth referred to as the Filé decomposition.

Definition 4.1 *The map $K_F(S) : Sh \to \wp(Sh)$ is defined by:*

$$K_F(S) = \left\{ B \;\middle|\; \begin{array}{c} B \subseteq S \quad \wedge \quad F \subseteq var(B) \quad \wedge \\ \forall G_1, G_2 \in B.(G_1 \neq G_2 \to G_1 \cap G_2 \cap F = \emptyset) \end{array} \right\}$$

Lemma 4.1 $\gamma_X^{SF}(\langle S, F\rangle) = \cup\{\gamma_X^{SF}(\langle B, F\rangle) \mid B \in K_F(S)\}$.

Using the above, abstract unification can be refined to $\cup\{amgu(\langle B, F, L\rangle, s, t) | B \in K_F(S)\}$. Abstract unification computed in this way does not merge sharing groups arising from different computational paths, and thereby improves precision. Calculating $K_F(S)$ is expensive and the number of calls to *amgu* is $|K_F(S)|$ (which is potentially exponential in $|S|$). However, this tactic suggests lightweight refinements to closure ($*$) and pair-wise union ($\uplus$) that recover some precision at little cost. Since two distinct sharing groups which contain a common free variable must arise from different computational paths, they cannot describe the same equation and therefore need not be combined. Definition 4.2 details the refined abstract unification algorithm and theorem 4.1 builds on lemma 4.2 to establish correctness.

Definition 4.2 (Abstract unification 2) *Abstract unification $amgu_2(\langle S, F, L\rangle, s, t) = \langle S', F', L'\rangle$ is defined:*

$$S'' = \begin{cases} (S_s^{*_F} \uplus_F S_t) \cap (S_s \uplus_F S_t^{*_F}) & \text{if } \chi(s, S, L) = \chi(t, S, L) = 1 \\ S_s^{*_F} \uplus_F S_t & \text{if } \chi(s, S, L) = 1 \\ S_s \uplus_F S_t^{*_F} & \text{if } \chi(t, S, L) = 1 \\ S_s^{*_F} \uplus_F S_t^{*_F} & \text{otherwise} \end{cases}$$

$$S_1 \uplus_F S_2 = \bigcup \left\{ G_1 \cup G_2 \;\middle|\; \begin{array}{l} G_1 \in S_1 \;\wedge\; G_2 \in S_2 \quad \wedge \\ G_1 \neq G_2 \to G_1 \cap G_2 \cap F = \emptyset \end{array} \right\}$$

$$S^{*_F} = \bigcap \left\{ S' \;\middle|\; \begin{array}{c} S \subseteq S' \quad \wedge \quad \forall G_1, G_2 \in S'. \\ G_1 \cap G_2 \cap F = \emptyset \to G_1 \cup G_2 \in S' \end{array} \right\}$$

where S', S_s, S_t, F' and L' are defined as in definition 3.2.

Notice that the use of freeness is completely absorbed into $*^F$ and $\uplus^F$. The following lemma demonstrates that $\uplus_F$ and $*_F$ coincide with $\uplus$ and $*$ for each element of the Filé decomposition. The correctness of abstract unification ($amgu_2$) follows from this result.

Lemma 4.2

1. *If $B \in K_F(S)$ and $R \subseteq B$, then $R^* = R^{*_F}$.*

2. *If $B \in K_F(S)$ and $R_1, R_2 \subseteq B$, then $R_1 \uplus R_2 = R_1 \uplus_F R_2$, $R_1^* \uplus R_2 = R_1^* \uplus_F R_2$, $R_1 \uplus R_2^* = R_1 \uplus_F R_2^*$ and $R_1^* \uplus R_2^* = R_1^* \uplus_F R_2^*$.*

Proof 4.1

1. *Proof by induction.*

   (a) *Suppose* $R = \emptyset$. *Then* $R^* = \emptyset = R^{*_F}$.

   (b) *Suppose* $R = \{G\} \cup R'$. *By the hypothesis,* $R'^* = R'^{*_F}$. *Since* $R \subseteq B$, *then for all* $G' \in R'$, $G' \cap G \cap F = \emptyset$. *Hence* $R^* = R^{*_F}$.

2. (a) *To show* $R_1 \uplus R_2 = R_1 \uplus_F R_2$. *Let* $G_i \in R_i$. *If* $G_1 \cap G_2 \cap F \neq \emptyset$ *then* $G_1 = G_2$. *Hence* $G_1 \cup G_2 \in R_1 \uplus_F R_2$.

   (b) *To show* $R_1^* \uplus R_2 = R_1^* \uplus_F R_2$. *Let* $G_1 \in R_1^*$ *and* $G_2 \in R_2$. *Then* $G_1 = \cup Q_1$ *for some* $Q_1 \subseteq R_1$. *Put* $Y = G_1 \cap G_2 \cap F$, $Q_1' = \{G \in Q_1 \mid G \cap Y = \emptyset\}$ *and* $Q_1'' = Q_1 \setminus Q_1'$. *Observe that* $|Q_1''| \leq 1$ *and* $Q_1'' \subseteq \{G_2\}$. *Thus* $G_1 \cup G_2 = (\cup Q_1') \cup G_2$. *Since* $(\cup Q_1') \cap G_2 \cap F = \emptyset$ *it follows that* $G_1 \cup G_2 \in R_1^* \uplus_F R_2$.

   (c) *To show* $R_1 \uplus R_2^* = R_1 \uplus_F R_2^*$. *Analogous to the previous case.*

   (d) *To show* $R_1^* \uplus R_2^* = R_1^* \uplus_F R_2^*$. *Let* $G_1 \in R_1^*$ *and* $G_2 \in R_2^*$. *Then* $G_i = \cup Q_i$ *for some* $Q_i \subseteq R_i$. *Put* $Y = G_1 \cap G_2 \cap F$, $Q_i' = \{G \in Q_i \mid G \cap Y = \emptyset\}$ *and* $Q_i'' = Q_i \setminus Q_i'$. *Observe that* $|Q_i''| \leq 1$.

      i. *Suppose* $|Q_1''| = \emptyset$ *or* $|Q_2''| = \emptyset$. *Then* $G_1 \cap G_2 \cap F = \emptyset$, *hence* $G_1 \cup G_2 \in R_1^* \uplus_F R_2^*$.

      ii. *Suppose* $|Q_1''| = |Q_2''| = 1$. *Hence* $Q_1'' = Q_2''$, *thus* $G_1 \cup G_2 = G_1 \cup (\cup Q_2')$. *Since* $G_1 \cap (\cup Q_2') \cap F = \emptyset$ *it follows that* $G_1 \cup G_2 \in R_1^* \uplus_F R_2^*$.

Theorem 4.1 (Correctness of abstract unification 2) Let $E \in \gamma_X^{SFL}(\langle S, F, L\rangle)$, $var(s) \cup var(t) \subseteq X$ and $amgu_2(\langle S, F, L\rangle, s, t) = \langle S', F', L'\rangle$. Then $E \cup \{s = t\} \in \gamma_X^{SFL}(\langle S', F', L'\rangle)$.

Proof 4.2 *Observe* $E \in \gamma_X^{SF}(\langle S, F\rangle)$ *and* $E \in \gamma_X^{L}(L)$. *By lemma 4.1, there exists* $B \in K_F(S)$ *such that* $E \in \gamma_X^{SF}(\langle B, F\rangle)$, *hence* $E \in \gamma_X^{SFL}(\langle B, F, L\rangle)$. *Observe that if* $s \in F$ *then* $S_s^{*_F} = S_s$ *(and likewise for* $t \in F$*) and hence by lemma 4.2,* $amgu_1(\langle B, F, L\rangle, s, t) = amgu_2(\langle B, F, L\rangle, s, t)$. *By theorem 3.1,* $E \cup \{s = t\} \in \gamma_X^{SFL}(amgu_1(\langle B, F, L\rangle, s, t)) = \gamma_X^{SFL}(amgu_2(\langle B, F, L\rangle, s, t))$, *thus* $E \cup \{s = t\} \in \gamma_X^{SFL}(amgu_2(\langle S, F, L\rangle, s, t))$.

The proof explains why the standard freeness tactic is a specialised version of the Filé decomposition.

This refinement is only worthwhile if redundant sharing groups are introduced in analysis. Although it can be shown that projection and join do not introduce redundancy, the following example indicates that redundant sharing groups can arise in abstract unification ($amgu_1$) and that the refined abstract unification ($amgu_2$) can avoid some of these redundant sharing groups.

Example 4.1 *Let* $X = \{x, y, z\}$, $S = \{\emptyset, \{x, y\}, \{y, z\}\}$, $F = \{y\}$ *and* $L = \emptyset$. *Observe that* $amgu_1(\langle S, F, L\rangle, x, z) = \langle\{\emptyset, \{x, y, z\}\}, \emptyset, \emptyset\rangle$, *but* $amgu_2(\langle S, F, L\rangle, x, z) = \langle\{\emptyset\}, \emptyset, \emptyset\rangle$.

The following example demonstrates that $amgu_2$ is not as precise as the full Filé decomposition.

Example 4.2 *Let* $X = \{x, y, z\}$, $S = \{\emptyset, \{x\}, \{z\}, \{x, y\}, \{y, z\}\}$, $F = \{x, y, z\}$ *and* $L = \emptyset$. *Then* $amgu_2(\langle S, F, L\rangle, x, z) = \langle\{\emptyset, \{x, z\}, \{x, y, z\}\}, F, L\rangle$ *includes the redundant sharing group* $\{x, z\}$, *whereas the Filé decomposition introduces no redundancy.*

5 Pruning of set-sharing

Pruning sharing groups is advantageous for efficiency and precision. By reducing the size of an abstraction, abstract unification works on smaller objects and is therefore faster, even if no precision is gained. Of course, the benefit of pruning for efficiency needs to outweigh its cost.

5.1 Pruning with freeness via groundness

Surprisingly, combined sharing and freeness information can improve groundness propagation and sharing even for rational tree unification. For example, the equation $x = f(y, z)$ can be abstracted by $(x \leftrightarrow z) \wedge (x \leftrightarrow y)$ if x and y are free variables that share. This is because, in this circumstance, finite tree unification fails for $x = f(y, z)$ whereas rational tree unification binds x and y to $f(f(\ldots, z), z)$. Abstract unification can use the freeness of variables in the equation to extract hidden groundness information (for distinct computational paths) and thereby prune sharing groups and improve precision. The proof of theorem 5.1 again uses the Filé decomposition.

Definition 5.1 (Abstract unification 3) *Abstract unification $amgu_3(\langle S, F, L\rangle, s, t) = \langle S', F', L'\rangle$ is defined:*

$$S' = (S \setminus (S_s \cup S_t)) \cup \begin{cases} \bigcup_{G \in S_s} trim_X(s \leftrightarrow Y, \{G\} \uplus_F S_t) & if\, s \in F \wedge t \notin U \\ \bigcup_{G \in S_t} trim_X(Z \leftrightarrow t, S_s \uplus_F \{G\}) & if\, t \in F \wedge s \notin U \\ S'' & otherwise \end{cases}$$

where $Y = var(t) \setminus (G \cap F)$, $Z = var(s) \setminus (G \cap F)$, S_s, S_t, S'', F' and L' are defined as in definition 4.2.

Theorem 5.1 (Correctness of abstract unification 3) Let $E \in \gamma_X^{SFL}(\langle S, F, L\rangle)$, $var(s) \cup var(t) \subseteq X$ and $amgu_3(\langle S, F, L\rangle, s, t) = \langle S', F', L'\rangle$. Then $E \cup \{s = t\} \in \gamma_X^{SFL}(\langle S', F', L'\rangle)$.

Proof 5.1 *Suppose $s \in F$. By lemma 4.1, there exists $B \in K_F(S)$ such that $E \in \gamma_X^{SFL}(\langle B, F\rangle)$ and by theorem 4.1, $E \cup \{s = t\} \in \gamma_X^{Sh}(B')$ where $B' = (B \setminus (B_s \cup B_t)) \cup (B_s \uplus_F B_t)$, $B_s = rel(s, B)$ and $B_t = rel(t, B)$. Let $\theta \in imgu(E)$. Since $s \in F$, $\theta(s) = x$ for some $x \in U$. Furthermore, $s \in G$ for all $G \in S_s$. Since $s \in F$, $B_s = \{G\}$ where $G = occ(\theta, x)$. Observe that $\theta(y) = x$ for all $y \in G \cap F$. Since $t \notin U$, $\theta(t) \notin U$, hence $\alpha_X^{Pos}(\{\theta(s) = \theta(t)\}) \models s \leftrightarrow Y$. Moreover, $mgu(E \cup \{s = t\}) = mgu(eqn(\theta) \cup \{s = t\}) = mgu(eqn(\theta) \cup \{\theta(s) = \theta(t)\})$. Thus $\alpha_X^{Pos}(E \cup \{s = t\}) \models \alpha_X^{Pos}(\{\theta(s) = \theta(t)\}) \models s \leftrightarrow Y$. The result follows by corollary 2.1. The $t \in F$ case is analogous and the otherwise case follows immediately from theorem 4.1.*

The following example illustrates the gain of precision. Note that even the Filé decomposition cannot match this level of precision.

Example 5.1 Let $X = \{x, y, z\}$, $E = \{x = f(y, z)\}$, $S = \{\emptyset, \{x, y\}, \{y\}, \{z\}\}$, $F = \{x, y\}$ and $L = \emptyset$. Then $amgu_2(\langle S, F, L\rangle, x, f(y, z)) = \langle S', \emptyset, \emptyset\rangle$ where $S' = \{\emptyset, \{x, y\}, \{x, y, z\}\}$, whereas $amgu_3(\langle S, F, L\rangle, x, f(y, z)) = \langle trim_X(S', x \leftrightarrow z), \emptyset, \emptyset\rangle = \langle\{\emptyset, \{x, y, z\}\}, \emptyset, \emptyset\rangle$.

This pruning tactic suggests that any optimal abstract unification algorithm for sharing, freeness and linearity, in the presence of groundness, will have to consider subtle interactions between the components.

5.2 Early pruning with groundness

Sharing abstractions can always be pruned by removing sharing groups which contain ground variables. Common practice is to schedule the solving of equations so as to first apply abstract unification to equations on ground terms [7]. This can involve repeated searching. This section proposes a revision of this tactic that applies groundness to the complete set of equations and then uses the resulting groundness information to prune sharing before abstract unification is applied. The gain is that searching and scheduling are no longer required and abstract unification can potentially operate on smaller abstractions. Observe that groundness will normally be tracked anyway, thus the computational overhead is negligible. To formulate this strategy, abstract unification is lifted to sets of equations as follows:

Definition 5.2 *The map* $amgu_i(T, E) = \{T' \mid \langle T, E\rangle \rightsquigarrow^\star \langle T', \emptyset\rangle\}$ *is defined by the least relation* $\rightsquigarrow \subseteq (Share_X \times Fr_X \times Lin_X)^2$ *such that* $\langle T, \{s = t\} \cup E\rangle \rightsquigarrow \langle amgu_i(T, s, t), E\rangle$.

The following theorem states correctness of the early pruning using groundness for $amgu_1$, $amgu_2$ and $amgu_3$.

Theorem 5.2 Let $E \in \gamma_X^{Pos}(f) \cap \gamma_X^{SFL}(\langle S, F, L\rangle)$, $E \cup E' \in \gamma_X^{Pos}(f')$, $Y = \{y \in X \mid f' \models y\}$, $S' = trim_X(f \wedge Y, S)$, $F' = F \setminus var(rel(Y, S))$, $L' = L \cup Y$, $var(E) \subseteq X$ and $T' \in amgu_i(\langle S', F', L'\rangle, E')$. Then $E \cup E' \in \gamma_X^{SFL}(T')$.

Proof 5.2 Let $\theta \in imgu(E)$ and $\theta' \in imgu(E \cup E')$. Since $\theta' \in unify(E)$, $\theta \leq \theta'$ and there exists $\zeta \in Sub$ such that $\zeta \circ \theta = \theta'$. Since $\theta' \in unify(E')$, $\zeta \in unify(\theta(E'))$ so that $mgu(\theta(E')) \neq \emptyset$. Let $\delta \in imgu(\theta(E')) = imgu(\theta^\infty(E'))$. By part 2 of lemma 2.2, $\delta \circ \theta = \delta \circ \theta^\infty \in mgu(eqn(\theta) \cup E') = mgu(E \cup E')$. Thus there exists $\rho \in Rename$ such that $\rho \circ \delta \circ \theta = \theta'$. Now $var(\theta'(y)) = \emptyset$ for all $y \in Y$, hence $var(\delta \circ \theta(y)) = \emptyset$ for all $y \in Y$. Put $Z = \cup\{var(\theta(y)) \mid y \in Y\}$, $\phi = \overline{\exists} Z.\delta$ and $\psi = \exists Z.\delta$. Let $z \in Z$. Then there exists $y \in Y$ such that $z \in var(\theta(y))$. But $var(\delta \circ \theta(y)) = \emptyset$, hence $rgn(\phi) = \emptyset$ and $\delta = \psi \circ \phi$. Thus $\psi \circ \phi \in mgu(\theta(E'))$ and by lemma 2.2 part 3, $\exists(dom(\phi) \setminus rng(\phi)).\psi \in mgu(\phi \circ \theta(E'))$. Furthermore, $\exists(dom(\phi) \setminus rng(\phi)).\psi = \psi$ hence $\psi \in mgu(\phi \circ \theta(E'))$. Since $\phi \circ \theta$ is idempotent, $\psi \in mgu((\phi \circ \theta)^\infty(E'))$. By lemma 2.2, part 2, $\psi \circ \phi \circ \theta = \psi \circ (\phi \circ \theta)^\infty \in mgu(eqn(\phi \circ \theta) \cup E')$. Thus $\theta' \in imgu(eqn(\phi \circ \theta) \cup E')$.

To show $eqn(\phi \circ \theta) \in \gamma_X^{Sh}(trim(f \wedge Y, S))$. Let $u \in U$. If $occ(\phi \circ \theta, u) = \emptyset$ then $occ(\phi \circ \theta, u) \cap X \in S$ trivially. If $occ(\phi \circ \theta, u) \neq \emptyset$ then $occ(\phi \circ \theta, u) = occ(\theta, u)$ since $rng(\phi) = \emptyset$. Thus $occ(\phi \circ \theta, u) \cap X \in S$. Therefore $eqn(\phi \circ \theta) \in \gamma_X^{Sh}(S)$. By lemma 2.2, part 2, $\delta \circ \theta \in mgu(E \cup eqn(\theta))$. But $\theta' \in mgu(E \cup eqn(\theta))$ and therefore there exists $\rho \in Rename$ such that $\rho \circ \delta \circ \theta = \theta'$. Thus $\alpha^{Pos}(\delta \circ \theta) \models \alpha^{Pos}(\rho \circ \delta \circ \theta) = \alpha^{Pos}(\theta') \models Y$. Observe that if $\alpha^{Pos}(\delta \circ \theta) \models u$ then $\alpha^{Pos}(\phi \circ \theta) \models u$ hence $\alpha^{Pos}(\phi \circ \theta) \models Y$. Since $\alpha^{Pos}(\phi \circ \theta) \models \alpha^{Pos}(\theta) \models f$, it follows that $\alpha^{Pos}(\phi \circ \theta) \models f \wedge Y$. Therefore $eqn(\phi \circ \theta) \in \gamma_X^{Pos}(f \wedge Y)$. By corollary 2.1, $eqn(\phi \circ \theta) \in \gamma_X^{Sh}(trim(f \wedge Y, S))$.

To show $\phi \circ \theta(x) \in U$ for all $x \in F'$. Let $x \in F$ and $x \notin var(rel(Y, S))$. Since $x \notin var(rel(Y, S))$, $x \notin occ(\theta, u) \cap X$ or $y \notin occ(\theta, u) \cap X$ for all $u \in U$ and $y \in Y$. Since $x \in X$ and $Y \subseteq X$, $var(\theta(x)) \cap var(\theta(y)) = \emptyset$ for all $y \in Y$. Hence $\theta(x) \notin Z$, thus $\theta(x) \notin dom(\phi)$, therefore $\phi \circ \theta(x) \in U$. Thus $eqn(\phi \circ \theta) \in \gamma_X^{Fr}(F')$.

To show $\chi(\phi \circ \theta(x)) \leq 1$ for all $x \in L'$. Since $rng(\phi) = \emptyset$, $\chi(\phi \circ \theta(x)) \leq 1$ for all $x \in L$. Moreover, $\alpha^{Pos}(\phi \circ \theta) \models Y$ and therefore $\chi(\phi \circ \theta(x)) \leq 1$ for all $x \in Y$. Thus $eqn(\phi \circ \theta) \in \gamma_X^{Lin}(L')$. The result then follows by induction on E and theorems 3.1, 4.1 and 5.1.

The following example illustrates the computational advantages of early pruning.

Example 5.2 Let $X = \{u, v, x, y\}$, $S = \{\emptyset, \{x\}, \{y\}, \{u\}, \{v\}\}$, $F = \emptyset$, $L = \emptyset$ and $f = x \vee y$. Let $E' = \{x = f(u,v), x = y\}$ so that $f' = (x \vee y) \wedge (x \leftrightarrow (u \wedge v)) \wedge (x \leftrightarrow y) = x \wedge y \wedge u \wedge v$. Then $Y = \{x, y, u, v\}$ so that $f \wedge Y = x \wedge y \wedge u \wedge v$ and $S' = trim_X(f \wedge Y, S) = \{\emptyset\}$. Hence $amgu_3(\langle S, F, L\rangle, E')$ reduces to $amgu_3(\langle S', F, L\rangle, E') = \langle\{\emptyset\}, \emptyset, \emptyset\rangle$. Without this tactic, no equation of E' will possess a ground argument and both calls to $amgu_3$ will involve non-trivial sharing group manipulation.

6 Conclusion

This paper has given correctness proofs for sharing analysis with freeness and linearity which hold in the presence of rational trees. The abstract unification algorithms are themselves novel – incorporating optimisations for both precision and efficiency. Specifically, the independence check which can prevent linearity from be exploited has been removed. In addition, refined closure and pair-wise union operations have been derived from the Filé decomposition. These refinements have been chosen to balance precision against efficiency. A further precision optimisation has been presented which exploits an interaction between sharing, freeness and groundness, which shows the subtlety that an optimal algorithm will need to address. This work provides the implementor with a suite of new optimisations for abstract unification algorithms for sharing, freeness and linearity.

Acknowledgements

We thank Gilberto Filé for kindly sending us a copy of his technical report. This work was supported, in part, by EPSRC grant GR/MO8769.

References

[1] R. Bagnara, E. Zaffanella, and P. Hill. Enhanced Sharing Analysis Techniques: A Comprehensive Evaluation. In *Proceedings of Principles and Practice of Declarative Programming*, pages 103–114. ACM Press, 2000. Long version available at http://www.comp.leeds.ac.uk/hill.

[2] M. Codish, D. Dams, and E. Yardeni. Derivation and Safety of an Abstract Unification Algorithm for Groundness and Aliasing Analysis. In *Proceedings of the International Conference on Logic Programming*, pages 79–93. MIT Press, 1991.

[3] M. Codish, H. Søndergaard, and P. Stuckey. Sharing and Groundness Dependencies in Logic Programs. *Transactions on Programming Languages and Systems*, 21(5):948–976, 1999.

[4] G. Filé. Sharing $\times$ Free: Simple and Correct. Technical Report 15, Dipartimento di Matematica, Università Degli Studi di Padova, 1994.

[5] P. Hill, R. Bagnara, and E. Zaffanella. The Correctness of Set-Sharing. In *Proceedings of Symposium on Static Analysis*, volume 1503 of *Lecture Notes in Computer Science*, pages 99–114. Springer-Verlag, 1998.

[6] A. King. Pair-Sharing over Rational Trees. *Journal of Logic Programming*, 46(1–2):139–155, 2000.

[7] A. Langen. *Advanced Techniques for Approximating Variable Aliasing in Logic Programs.* PhD thesis, University of Southern California, Los Angeles, 1991.

[8] J-L. Lassez, M. Maher, and K. Marriott. Unification Revisited. In *Foundations of Deductive Databases and Logic Programming*, pages 587–625. Morgan Kaufmann, 1988.