Computer Science at Kent

Correctness of Set-Sharing with Linearity

Jacob M. Howe and Andy King

Technical Report No: 3-02
Date: March 2002

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

Abstract

Zaffanella presents an intriguing abstract unification algorithm for tracing set-sharing with linearity and freeness as part of this doctoral thesis. This note provides a short correctness proof for the main novel aspect of this algorithm.

1 Introduction

One challenge in abstract interpretation is the development of analyses that are both useful (precise and tractable) and verifiable (possess convincing correctness arguments). Zaffanella [7] presents an abstract unification algorithm for tracing set-sharing with linearity and freeness in the presence of rational trees that is usually accurate and efficient. Correctness is established in the presence of rational trees [7, pp. 137–149, 155–179]. However, as pointed out in [7], arguments built on alternating paths [1, 4] can be simpler and Zaffenella states "It would be interesting to know whether or not the alternating paths concept (or a small variation of it) could be exploited to obtain simpler correctness proofs for analyses based on the set-sharing domain" [7]. Indeed, Howe and King [3] present a closely related, though less precise abstract unification algorithm for tracking set-sharing with linearity and freeness, whose correctness proof for rational-trees is based on alternating path results. The proof is particularly succinct [3, pp. 8–9] since it exploits a linearity lemma [4] that is slight revision of a classic linearity lemma [1] inspired by alternating path ideas [6]. Superficially it would appear that the proof of [3] cannot be extended to the algorithm of [7], but this note argues the contrary. In fact the correctness argument of [3] can be extended to [7] by (multiple applications of) a linearity lemma. The value of this observation is partly in the brevity of the resulting proof (a strong case for simple proofs is made in [2]), partly in that it increases confidence in the algorithm of [7] (correctness is now established two ways), and partly in that it shows that linearity lemmas devised for pair-sharing are useful for arguing the correctness of set-sharing.

2 Preliminaries

2.1 Terms, substitutions and equations

Let $T(F, V)$ denote the set of (possibly infinite) terms over an alphabet of symbols F and a (denumerable) universe of variables V where $F \cap V = \emptyset$. Let $var(t)$ denote the set of variables occurring in the term t and $|S|$ denote the cardinality of the set S.

A substitution is a (total) map $\theta : V \to T(F, V)$ such that $dom(\theta) = \{u \in V \mid \theta(u) \neq u\}$ is finite. Let $rng(\theta) = \cup\{var(\theta(u)) \mid u \in dom(\theta)\}$ and let Sub denote the set of substitutions. Let $\theta(t)$ denote the term obtained by simultaneously replacing each occurrence of a variable $x \in dom(\theta)$ in t with $\theta(x)$. An equation e is a pair $(s = t)$ where $s, t \in T(F, V)$. 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 $eqn(\theta) = \{x = \theta(x) \mid x \in dom(\theta)\}$. Composition $\theta \circ \psi$ of two substitutions is defined so that $(\theta \circ \psi)(u) = \theta(\psi(u))$ for all $u \in V$. 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 substitution θ is idempotent iff $\theta \circ \theta = \theta$, or equivalently, iff $dom(\theta) \cap rng(\theta) = \emptyset$ [5].

2.2 Most general unifiers

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 $mgu(E) \neq \emptyset$ [5]. The following lemma details how an mgu of an instance of an equation under a substitution, relates to the mgu of the equation and the substitution.

Lemma 2.1 (Lemma 4.3 from [4]) If θ is idempotent and $\delta \in mgu(\theta(E))$ then $\delta \circ \theta \in mgu(E \cup eqn(\theta))$.

2.3 Linearity

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 2.1 The variable multiplicity map $\chi : T(F, V) \to \{0, 1, 2\}$ is defined: $\chi(t) = 0$ if $var(t) = \emptyset$, $\chi(t) = 2$ if there exists a variable that occurs multiply in t, otherwise $\chi(t) = 1$.

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

Lemma 2.2 (Linearity lemma from [4]) If $\theta \in mgu(\{s = t\})$, $x \neq y$ and $var(\theta(x)) \cap var(\theta(y)) \neq \emptyset$ then either:

$$\begin{array}{lcllcl} x \in var(s) \text{ and } y \in var(t) & or & x, y \in var(t) & \text{and} & \chi(s) = 2 & or \\ x \in var(t) \text{ and } y \in var(s) & or & x, y \in var(s) & \text{and} & \chi(t) = 2. & \end{array}$$

2.4 Linearity and sharing abstractions

Let X denote a finite subset of V. The sharing and linearity domains are defined over X as follows:

Definition 2.2 *$Lin_X = \wp(X)$ and $Sh_X = \{S \subseteq \wp(X) \mid \emptyset \in S\}$.*

These domains are ordered by $\subseteq$ and connect to the concrete domain of sets of equations by Galois connections induced by the following concretisation maps.

Definition 2.3 ([3]) The concretisation maps $\gamma_X^{Lin} : Lin_X \to \wp(Eqn)$ and $\gamma_X^{Sh} : Sh_X \to \wp(Eqn)$ are defined by:

$$\begin{array}{rcl} \gamma_X^{Lin}(L) & = & \{E \in Eqn \mid \exists \theta \in imgu(E).\forall x \in L.\chi(\theta(x)) \leq 1\} \\ \gamma_X^{Sh}(S) & = & \{E \in Eqn \mid \exists \theta \in imgu(E).\alpha_X^{Sh}(\theta) \subseteq S\} \end{array}$$

where $\alpha_X^{Sh}(\theta) = \{occ(\theta, u) \cap X | u \in V\}$ and $occ(\theta, y) = \{u \in V | y \in var(\theta(u))\}$.

Note that γ_X^{Lin} and γ_X^{Sh} are well-defined though formulated in terms of an arbitrary imgu [3]. Couching the definition in terms of an arbitrary imgu (rather than a specific imgu [4]) simplifies the correctness proofs. Note also that an equation may possibly characterise a rational tree.

Finally, the following auxiliary operations will be used to express the algorithm and state its correctness. 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 $S_1 \uplus S_2 = \{G_1 \cup G_2 \mid G_1 \in S_1 \wedge G_2 \in S_2\}$. An abstract multiplicity map $\chi : T(F, X) \times Sh_X \times Lin_X \to \{1, 2\}$ is also assumed, defined so that if $E \in \gamma_X^{Sh}(S) \cap \gamma_X^{Lin}(L)$ and $\theta \in imgu(E)$ then $\chi(\theta(t)) \leq \chi(t, S, L)$ [3].

3 Correctness result

Theorem 3.1 Suppose $E \in \gamma_X^{Sh}(S) \cap \gamma_X^{Lin}(L)$, $var(s) \cup var(t) \subseteq X$ and $\chi(s, S, L) = \chi(t, S, L) = 1$. Then $E \cup \{s = t\} \in \gamma_X^{Sh}(S')$ where

$$S' = (S \setminus (S_s \cup S_t)) \cup (S_s \cup (S_s \uplus S_{st}^*)) \uplus (S_t \cup (S_t \uplus S_{st}^*))$$

$S_s = rel(s, S)$, $S_t = rel(t, S)$ and $S_{st} = S_s \cap S_t$.

Proof 3.1 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 $\delta \in imgu(\theta(E'))$. By lemma 2.1, $\delta \circ \theta \in mgu(E' \cup eqn(\theta)) = mgu(E' \cup E)$. Since $\delta \in imgu(\theta(E'))$, it follows that $rng(\delta) \subseteq (var(E') \setminus dom(\theta)) \cup rng(\theta)$. Since $dom(\theta) \cap rng(\theta) = \emptyset$, $dom(\theta) \cap rng(\delta) = \emptyset$, hence $\delta \circ \theta \in imgu(E \cup E')$. To show $\alpha_X^{Sh}(\delta \circ \theta) \subseteq S'$, let $y \in V$ and consider $occ(\delta \circ \theta, y) \cap X$.

1. Suppose $y \notin rng(\delta \circ \theta)$. Proceed as in [3] to show $occ(\delta \circ \theta, y) \cap X \in S \setminus (S_s \cup S_t) \subseteq S'$.

2. Suppose $y \in rng(\delta \circ \theta) \setminus var(\theta(E'))$. Proceed as in [3] to again show $occ(\delta \circ \theta, y) \cap X \in S \setminus (S_s \cup S_t) \subseteq S'$.

3. 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 $R_s = \{occ(\theta, v) \cap X \mid v \in var(\theta(s)) \cap occ(\delta, y)\}$ and $R_t = \{occ(\theta, w) \cap X \mid w \in var(\theta(t)) \cap occ(\delta, y)\}$. Because $\theta \in imgu(E)$ and $E \in \gamma_X^{Sh}(S)$, then $\{occ(\theta, u) \cap X \mid u \in V\} = \alpha_X^{Sh}(\theta) \subseteq S$, hence $R_s, R_t \subseteq S$. Since $var(s) \subseteq X$, $R_s \subseteq S_s$ and since $var(t) \subseteq X$, $R_t \subseteq S_t$. If $R_s = \emptyset$, then $y \notin var(\delta \circ \theta(s)) = var(\delta \circ \theta(t))$, hence $R_t = \emptyset$ and $occ(\delta \circ \theta, y) \cap X = \emptyset \in S'$. Likewise $occ(\delta \circ \theta, y) \cap X = \emptyset \in S'$ if $R_t = \emptyset$. Thus suppose $R_s \neq \emptyset$ and $R_t \neq \emptyset$. Since $\chi(s, S, L) = 1$ and $\chi(t, S, L) = 1$, it follows that $\chi(\theta(s)) = 1$ and $\chi(\theta(t)) = 1$. Suppose $|R_t \setminus R_s| > 1$. Thus there exists $u \neq v$ such that $u, v \in var(\theta(t)) \setminus var(\theta(s))$ and $var(\delta(u)) \cap var(\delta(v)) \neq \emptyset$. This contradicts lemma 2.2 (when instantiated with $\theta(s)$ and $\theta(t)$ rather than s and t), hence $|R_t \setminus R_s| \leq 1$. Likewise $|R_s \setminus R_t| \leq 1$.

   (a) Suppose $|R_t \setminus R_s| = 0$ and $|R_s \setminus R_t| = 0$. Then $R_s = R_t$ so that $occ(\delta \circ \theta, y) \cap X \in S_{st}^* \subseteq S_s \uplus S_{st}^* \uplus S_t$.

   (b) Suppose $|R_t \setminus R_s| = 0$ and $|R_s \setminus R_t| = 1$. Then there exists $G_s \in R_s$ such that $R_s = \{G_s\} \cup R_t$. Hence $occ(\delta \circ \theta, y) \cap X \in S_s \uplus S_{st}^* \subseteq S_s \uplus S_{st}^* \uplus S_t$.

   (c) Suppose $|R_t \setminus R_s| = 1$ and $|R_s \setminus R_t| = 0$. Analogous to previous case.

   (d) Suppose $|R_t \setminus R_s| = 1$ and $|R_s \setminus R_t| = 1$. If $R_s \cap R_t = \emptyset$ then $occ(\delta \circ \theta, y) \cap X \in S_s \uplus S_t$ whereas if $R_s \cap R_t \neq \emptyset$ then $occ(\delta \circ \theta, y) \cap X \in S_s \uplus S_{st}^* \uplus S_t$.

   Finally observe $(S_s \uplus S_t) \cup (S_s \uplus S_{st}^* \uplus S_t) = (S_s \cup (S_s \uplus S_{st}^*)) \uplus (S_t \cup (S_t \uplus S_{st}^*))$.

Acknowledgements

We thank Enea Zaffanella for his thought provoking doctoral defence and Harald Søndergaard for rekindling our interest in linearity.

References

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

[2] R. A. De Millo, R. J. Lipton, and A. J. Perlis. Social processes and proofs of theorems and programs. *Communications of the ACM*, 22(5):271–280, 1979.

[3] J. M. Howe and A. King. Three Optimisations for Sharing. *Theory and Practice of Logic Programming*, January 2003. http://arXiv.org/abs/cs.PL/0203022.

[4] A. King. Pair-Sharing over Rational Trees. *Journal of Logic Programming*, 46(1–2):139–155, 2000. http://www.cs.ukc.ac.uk/pubs/2000/1052.

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

[6] H. Søndergaard. An application of the abstract interpretation of logic programs: occur-check reduction. In *European Symposium on Programming*, volume 213 of *Lecture Notes in Computer Science*, pages 327–338. Springer-Verlag, 1986.

[7] E. Zaffanella. *Correctness, Precision and Efficiency in the Sharing Analysis of Real Logic Languages*. PhD thesis, University of Leeds, 2001. http://www.cs.unipr.it/~zaffanella.