Decomposing Non-Redundant Sharing by Complementation

Enea Zaffanella[1], Patricia M. Hill[2]⋆, and Roberto Bagnara[3]

[1] Servizio IX Automazione, Università degli Studi di Modena, Italy.
`zaffanella.enea@unimo.it`

[2] School of Computer Studies, University of Leeds, Leeds, LS2 9JT, U. K.
`hill@scs.leeds.ac.uk`

[3] Dipartimento di Matematica, Università degli Studi di Parma, Italy.
`bagnara@cs.unipr.it`

Abstract. Complementation, the inverse of the reduced product operation, is a relatively new technique for systematically finding minimal decompositions of abstract domains. Filé and Ranzato advanced the state of the art by introducing a simple method for computing a complement. As an application, they considered the extraction by complementation of the pair-sharing domain PS from the Jacobs and Langen's set-sharing domain SH. However, since the result of this operation was still SH, they concluded that PS was too abstract for this. Here, we show that the source of this difficulty lies not with PS but with SH and, more precisely, with the redundant information contained in SH with respect to ground-dependencies and pair-sharing. In fact, the difficulties vanish if our non-redundant version of SH, SH^ρ, is substituted for SH. To establish the results for SH^ρ, we define a general schema for subdomains of SH that includes SH^ρ and Def as special cases. This sheds new light on the structure of SH^ρ and exposes a natural though unexpected connection between Def and SH^ρ. Moreover, we substantiate the claim that complementation *alone* is not sufficient to obtain *truly minimal* decompositions of domains. The right solution to this problem is to *first* remove redundancies by computing the quotient of the domain with respect to the observable behavior, and only *then* decompose it by complementation.

Keywords: Abstract Interpretation, Domain Decomposition, Complementation, Sharing Analysis.

1 Introduction

Complementation [5], which is the inverse of the well-known reduced product operation [7], can systematically obtain minimal decompositions of complex abstract domains. It was argued that these decompositions would be useful in finding space saving representations for domains and to simplify domain verification problems.

⋆ Part of this work was supported by EPSRC grant GR/M05645.

In [8], Filé and Ranzato presented a new method for computing the complement, which is simpler than the original proposal by Cortesi et al. [4, 5] because it has the advantage that, in order to compute the complement, only a relatively small number of elements (namely the *meet-irreducible* elements of the reference domain) need be considered. As an application of this method, the authors considered the Jacobs and Langen's sharing domain [13], SH, for representing properties of variables such as groundness and sharing. This domain captures the property of set-sharing. However, it has been observed [1] that for most (if not all) applications, the property of interest is not set-sharing but pair-sharing. Filé and Ranzato illustrated their method by minimally decomposing SH into three components; using the words of the authors [8, Section 1]:

> "each representing one of the elementary properties that coexist in the elements of *Sharing*, and that are as follows: (i) the ground-dependency information; (ii) the pair-sharing information, or equivalently variable independence; (iii) the set-sharing information, without variable independence and ground-dependency."

However, this decomposition did not use the usual domain PS for pair-sharing. Filé and Ranzato observed that the complement of the pair-sharing domain PS with respect to SH is again SH and concluded that PS was too abstract to be extracted from SH by means of complementation. In order to overcome this difficulty, and to obtain this non-trivial decomposition of SH, they used a different (and somewhat unnatural) definition for an alternative pair-sharing domain that they called PS'. The nature of PS' and its connection with PS is examined more carefully in Section 6.

We noticed that the difficulty Filé and Ranzato had was not in the definition of PS, which accurately represents the property of pair-sharing, but in the use of the set-sharing domain SH itself since it carries some redundant information. It was shown in [1] that, for groundness and pair-sharing, SH includes redundant information. By defining an upper closure operator ρ that removed this redundancy, a much smaller domain SH^ρ was found that captured pair-sharing with the same precision as SH. We show here that using the method given in [8], but with this domain instead of SH as the reference domain, the difficulties in the decomposition disappear. Moreover, we show that PS is *exactly* one of the components obtained by complementation of SH^ρ. Thus the problem exposed by Filé and Ranzato was, in fact, due to the "information preserving" property of complementation, as any factorization obtained in this way is such that the reduced product of the factors gives back the original domain. In particular, any factorization of SH has to encode the redundant information identified in [1]. We will show that such a problem disappears when SH^ρ is used as the reference domain.

Although the primary purpose of this work is to clarify the decomposition of the domain SH^ρ, the formulation is sufficiently general to apply to other properties that are captured by SH. The domain *Pos* of positive Boolean functions and its subdomain *Def*, the domain of *definite* Boolean functions, are normally used for capturing groundness. Each Boolean variable has the value *true* if the program variable it corresponds to is definitely bound to a ground term. However, the domain *Pos* is isomorphic to *SH* via the mapping from formulas in *Pos* to the set of complements of their models [3]. This means that any general results regarding the structure of *SH* are equally applicable to *Pos* and its subdomains.

To establish the results for SH^ρ, we define a general schema for subdomains of *SH* that includes SH^ρ and *Def* as special cases. This sheds new light on the structure of the domain SH^ρ, which is smaller but significantly more involved than *SH*.[1] Of course, as we have used the more general schematic approach, we can immediately derive (where applicable) corresponding results for *Def* and *Pos*. Moreover, an interesting consequence of this work is the discovery of a natural connection between the abstract domains *Def* and SH^ρ. The results confirm that SH^ρ is, in fact, the "appropriate" domain to consider when pair-sharing is the property of interest.

The paper is structured as follows: In Section 2 we briefly recall the required notions and notations, even though we assume general acquaintance with the topics of lattice theory, abstract interpretation, sharing analysis and groundness analysis. Section 3 introduces the *SH* domain and several abstractions of it. The meet-irreducible elements of an important family of abstractions of *SH* are identified in Section 4. This is required in order to apply, in Section 5, the method of Filé and Ranzato to this family. We conclude in Section 6 with some final remark where, in particular, we explain what is, in our opinion, the lesson to be learned from this and other related works.

2 Preliminaries

For any set S, $\wp(S)$ denotes the power set of S and $\# S$ is the cardinality of S.

A *preorder* $\preceq$ over a set P is a binary relation that is reflexive and transitive. If $\preceq$ is also antisymmetric, then it is called *partial order.* A set P equipped with a partial order $\preceq$ is said to be *partially ordered* and sometimes written $\langle P, \preceq \rangle$. Partially ordered sets are also called *posets.*

Given a poset $\langle P, \preceq \rangle$ and $S \subseteq P$, $y \in P$ is an *upper bound* for S if and only if $x \preceq y$ for each $x \in S$. An upper bound y for S is the *least upper bound* (or lub) of S if and only if, for every upper bound y' for S, $y \preceq y'$. The lub, when it exists, is unique. In this case we write $y = \operatorname{lub} S$. *Lower bounds* and *greatest lower bounds* are defined dually.

A poset $\langle L, \preceq \rangle$ such that, for each $x, y \in L$, both $\operatorname{lub}\{x, y\}$ and $\operatorname{glb}\{x, y\}$ exist, is called a *lattice.* In this case, lub and glb are also called, respectively, the *join* and the *meet* operations of the lattice. A *complete lattice* is a lattice $\langle L, \preceq \rangle$ such that every subset of L has both a least upper bound and a greatest lower bound. The *top* element of a complete lattice L, denoted by $\top$, is such that $\top \in L$ and $\forall x \in L : x \preceq \top$. The *bottom* element of L, denoted by $\bot$, is defined dually.

[1] For the well acquainted with the matter: *SH* is a powerset and hence it is dual-atomistic; this is not the case for SH^ρ.

An algebra $\langle L, \wedge, \vee \rangle$ is also called a *lattice* if $\wedge$ and $\vee$ are two binary operations over L that are commutative, associative, idempotent, and satisfy the following *absorption laws*, for each $x, y \in L$: $x \wedge (x \vee y) = x$ and $x \vee (x \wedge y) = x$. The two definitions of lattices are equivalent. This can be seen by setting up the isomorphism given by: $x \preceq y \overset{\text{def}}{\iff} x \wedge y = x \overset{\text{def}}{\iff} x \vee y = y$, $\operatorname{glb}\{x, y\} \overset{\text{def}}{=} x \wedge y$, and $\operatorname{lub}\{x, y\} \overset{\text{def}}{=} x \vee y$. A complete lattice C is *meet-continuous* if for any chain $Y \subseteq C$ and each $x \in C$, $x \wedge \bigl(\bigvee Y\bigr) = \bigvee_{y \in Y}(x \wedge y)$.

A monotone and idempotent self-map $\rho \colon P \to P$ over a poset $\langle P, \preceq \rangle$ is called a *closure operator* (or *upper closure operator*) if it is also *extensive*, namely $\forall x \in P : x \preceq \rho(x)$. If C is a complete lattice, then each upper closure operator ρ over C is uniquely determined by the set of its fixpoints, that is, by its image $\rho(C) \overset{\text{def}}{=} \{\, \rho(x) \mid x \in C \,\}$. The set of all upper closure operators over a complete lattice C, denoted by $\operatorname{uco}(C)$, form a complete lattice ordered as follows: if $\rho_1, \rho_2 \in \operatorname{uco}(P)$, $\rho_1 \sqsubseteq \rho_2$ if and only if $\rho_2(C) \subseteq \rho_1(C)$. We will often denote upper closure operators by the sets of their fixpoints. The reader is referred to [11] for an extensive treatment of closure operators.

The *reduced product* of two elements ρ_1, ρ_2 of $\operatorname{uco}(C)$ is: $\rho_1 \sqcap \rho_2 \overset{\text{def}}{=} \operatorname{glb}\{\rho_1, \rho_2\}$. Suppose C is a meet-continuous lattice (which is the case for most domains for abstract interpretation [5], here included all the domains considered in this paper). Then the inverse of the reduced product operation, called *complementation*, is well defined and given as follows. If $\rho \sqsubseteq \rho_1$ then $\rho \sim \rho_1 \overset{\text{def}}{=} \operatorname{lub}\{\, \rho_2 \mid \rho_1 \sqcap \rho_2 = \rho \,\}$. Given a meet-continuous lattice C and $\rho \in \operatorname{uco}(C)$, the *pseudo-complement* (or, by an abuse of terminology now customary in the field, simply *complement*) of ρ is denoted by $id_C \sim \rho$, where id_C is the identity over C. Let C be a meet-continuous lattice and $D_i \overset{\text{def}}{=} \rho_{D_i}(C)$ with $\rho_{D_i} \in \operatorname{uco}(C)$ for $i = 1, \ldots, n$. Then $\{\, D_i \mid 1 \leq i \leq n \,\}$ is a *decomposition for* C if $C = D_1 \sqcap \cdots \sqcap D_n$. The decomposition is also called *minimal* if, for each $k \in \mathbb{N}$ with $1 \leq k \leq n$ and each $E_k \in \operatorname{uco}(C)$, $D_k \sqsubset E_k$ implies $C \sqsubset D_1 \sqcap \cdots \sqcap D_{k-1} \sqcap E_k \sqcap D_{k+1} \sqcap \cdots \sqcap D_n$. Let C be a complete lattice and $X \subseteq C$. Then $\operatorname{Moore}(X)$ denotes the *Moore completion of* X, namely, $\operatorname{Moore}(X) \overset{\text{def}}{=} \{\, \bigwedge Y \mid Y \subseteq X \,\}$. We say that C *is meet-generated by* X if $C = \operatorname{Moore}(X)$. An element $x \in C$ is *meet-irreducible* if $\forall y, z \in C : \bigl((x = y \wedge z) \implies (x = y \text{ or } x = z)\bigr)$. The set of meet-irreducible elements of a complete lattice C is denoted by $\operatorname{MI}(C)$. Note that $\top \in \operatorname{MI}(C)$. An element $x \in C$ is a *dual-atom* if $x \neq \top$ and, for each $y \in C$, $x \leq y < \top$ implies $x = y$. The set of dual-atoms is denoted by $\operatorname{dAtoms}(C)$. Note that $\operatorname{dAtoms}(C) \subset \operatorname{MI}(C)$. The domain C is *dual-atomistic* if $C = \operatorname{Moore}\bigl(\operatorname{dAtoms}(C)\bigr)$. Thus, if C is dual-atomistic, $\operatorname{MI}(C) = \{\top\} \cup \operatorname{dAtoms}(C)$.

The following result holds [8, Theorem 4.1].

Theorem 1. *If C is meet-generated by $\operatorname{MI}(C)$ then $\operatorname{uco}(C)$ is pseudo-complemented and for any $\rho \in \operatorname{uco}(C)$*

$$id_C \sim \rho = \operatorname{Moore}\bigl(\operatorname{MI}(C) \setminus \rho(C)\bigr).$$

Another useful result from lattice theory is the following.

Theorem 2. *All continuous lattices are meet-generated by their meet-irreducible elements.*

Hence we have the following corollary [8, Corollary 4.5].

Corollary 1. *If C is dual-atomistic then* $\mathrm{uco}(C)$ *is pseudo-complemented and for any $\rho \in \mathrm{uco}(C)$*

$$id_C \sim \rho = \mathrm{Moore}\bigl(\mathrm{dAtoms}(C) \setminus \rho(C)\bigr).$$

Domains such as *SH* are normally defined over a denumerable set of variables *Vars* and then a finite subset of *Vars* is used to define a subset of *SH* that is restricted to these variables. In this paper, we assume there is a fixed and finite set of variables of interest $\mathit{VI} \subset \mathit{Vars}$ of cardinality n.

If t is a first-order term over VI, then $\mathit{vars}(t)$ denotes the set of variables occurring in t. *Bind* denotes the set of equations of the form $x = t$ (sometimes written $x \mapsto t$) where $x \in \mathit{VI}$ and t is a first-order term over VI. Note that we do not impose the *occur-check* condition $x \notin \mathit{vars}(t)$, since we have proved in [12] that this is not required to ensure correctness of the operations of *SH* and its derivatives. We also define $\mathit{Subst} \stackrel{\mathrm{def}}{=} \wp(\mathit{Bind})$.

3 The Sharing Domains

3.1 The Set-sharing Domain *SH*

In this paper, since the set of relevant variables is fixed, it is assumed that *SH*, *Def* and *PS* are each restricted to the finite set of variables $\mathit{VI} \subset \mathit{Vars}$. Therefore, the domain elements for *SH* (and hence for all their subdomains such as *PS* and *Def*) do not include the set of variables explicitly.

Definition 1. (The *set-sharing* domain *SH*.) *The domain SH is given by $\mathit{SH} \stackrel{\mathrm{def}}{=} \wp(\mathit{SG})$ where $\mathit{SG} \stackrel{\mathrm{def}}{=} \bigl\{\, S \in \wp(\mathit{VI}) \bigm| S \neq \emptyset \,\bigr\}$. SH is partially ordered by set inclusion so that the lub is given by set union and glb by set intersection.*

Note that, as we are adopting the upper closure operator approach to abstract interpretation, all the domains we define here are ordered by subset inclusion. In the following examples, the elements of *SH* will be always written in a simplified notation, omitting the inner braces. For instance, the set $\bigl\{\{x\}, \{x, y\}, \{x, z\}, \{x, y, z\}\bigr\}$ will be written simply as $\{x, xy, xz, xyz\}$.

For the purpose of this paper, we just need the following operations over *SH*. See [1] for a precise definition of all the operations used for an analysis.

Definition 2. (Some operations over *SH*.) *The function* $\mathrm{bin}\colon \mathit{SH} \times \mathit{SH} \to \mathit{SH}$, *called* binary union, *is given by*

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

The star-union *function* $(\cdot)^\star \colon SH \to SH$ *is given by*

$$sh^\star \stackrel{\text{def}}{=} \Bigl\{\, S \in SG \Bigm| \exists sh' \subseteq sh \mathrel{.} S = \bigcup sh' \,\Bigr\}.$$

For each $j \geq 1$, the j-self-union *function* $(\cdot)^j \colon SH \to SH$ *is given by*

$$sh^j \stackrel{\text{def}}{=} \Bigl\{\, S \in SG \Bigm| \exists sh' \subseteq sh \mathrel{.} \bigl(\# sh' \leq j, S = \bigcup sh'\bigr) \,\Bigr\}.$$

The extraction of the relevant component *of an element of SH with respect to a subset of VI is encoded by the function* $\mathrm{rel} \colon \wp_{\mathrm{f}}(VI) \times SH \to SH$ *given by*

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

The function amgu *captures the effects of a binding $x \mapsto t$ on an element of SH. Let $v_x = \{x\}$, $v_t = vars(t)$ and $v_{xt} = v_x \cup v_t$. Then*

$$\mathrm{amgu}(sh, x \mapsto t) \stackrel{\text{def}}{=} \bigl(sh \setminus (\mathrm{rel}(v_{xt}, sh)\bigr) \cup \mathrm{bin}\bigl(\mathrm{rel}(v_x, sh)^\star, \mathrm{rel}(v_t, sh)^\star\bigr).$$

We also define the extension $\mathrm{amgu} \colon SH \times Subst \to SH$ *by*

$$\begin{aligned} \mathrm{amgu}(sh, \emptyset) &\stackrel{\text{def}}{=} sh, \\ \mathrm{amgu}\bigl(sh, \{x \mapsto t\} \cup \sigma\bigr) &\stackrel{\text{def}}{=} \mathrm{amgu}\bigl(\mathrm{amgu}(sh, x \mapsto t), \sigma \setminus \{x \mapsto t\}\bigr). \end{aligned}$$

The function $\mathrm{proj} \colon SH \times \wp(VI) \to SH$ *that* projects *an element of SH onto a subset $V \subseteq VI$ of the variables of interest is given by*

$$\mathrm{proj}(sh, V) \stackrel{\text{def}}{=} \{\, S \cap V \mid S \in sh, S \cap V \neq \emptyset \,\}.$$

The j-self-union operator is new. We show later when it may be safely used to replace the star-union operator. Note, in particular that, letting $j = 1$, 2, and n, we have $sh^1 = sh$, $sh^2 = \mathrm{bin}(sh, sh)$, and, as $\# VI = n$, $sh^n = sh^\star$.

Since SH is a power set, SH is dual-atomistic and

$$\mathrm{dAtoms}(SH) = \bigl\{\, SG \setminus \{S\} \bigm| S \in SG \,\bigr\}.$$

Example 1. Suppose $VI = \{x, y, z\}$. Then the seven dual-atoms of SH are:

$$\begin{aligned} &\left.\begin{aligned} s_1 &= \{\ y, z, xy, xz, yz, xyz\}, \\ s_2 &= \{x, \ z, xy, xz, yz, xyz\}, \\ s_3 &= \{x, y, \ xy, xz, yz, xyz\}, \end{aligned}\right\} && \text{these lack a singleton;} \\ &\left.\begin{aligned} s_4 &= \{x, y, z, \ xz, yz, xyz\}, \\ s_5 &= \{x, y, z, xy, \ yz, xyz\}, \\ s_6 &= \{x, y, z, xy, xz, \ xyz\}, \end{aligned}\right\} && \text{these lack a pair;} \\ &\quad s_7 = \{x, y, z, xy, xz, yz \}, && \text{this lacks } VI. \end{aligned}$$

Then the meet-irreducible elements of SH are $s_1, \ldots, s_7$ together with SG, the top element.

3.2 The Tuple-sharing Domains

To provide a general characterization of domains such as the groundness and pair-sharing domains in SH, we first identify the sets of elements that have the same cardinality.

Definition 3. (Tuples of cardinality k.) *For each $k \in \mathbb{N}$ with $1 \leq k \leq n$, the overloaded functions* $\mathrm{tuples}_k \colon SG \to SH$ *and* $\mathrm{tuples}_k \colon SH \to SH$ *are defined as*

$$\mathrm{tuples}_k(S) \stackrel{\mathrm{def}}{=} \bigl\{\, T \in \wp(S) \bigm| \# T = k \,\bigr\},$$
$$\mathrm{tuples}_k(sh) \stackrel{\mathrm{def}}{=} \bigcup \bigl\{\, \mathrm{tuples}_k(S') \bigm| S' \in sh \,\bigr\}.$$

In particular, if $S \in SG, sh \in SH$, let

$$\mathrm{pairs}(S) \stackrel{\mathrm{def}}{=} \mathrm{tuples}_2(S),$$
$$\mathrm{pairs}(sh) \stackrel{\mathrm{def}}{=} \mathrm{tuples}_2(sh).$$

The usual domains that represent groundness and pair-sharing information will be shown to be special cases of the following more general domain.

Definition 4. (The *tuple-sharing* domains TS_k.) *For each $k \in \mathbb{N}$ such that $1 \leq k \leq n$, the function $\rho_{TS_k} \colon SH \to SH$ is defined as*

$$\rho_{TS_k}(sh) \stackrel{\mathrm{def}}{=} \bigl\{\, S \in SG \bigm| \mathrm{tuples}_k(S) \subseteq \mathrm{tuples}_k(sh) \,\bigr\}$$

and, as $\rho_{TS_k} \in \mathrm{uco}(SH)$, it induces the lattice

$$TS_k \stackrel{\mathrm{def}}{=} \rho_{TS_k}(SH).$$

Note that $\rho_{TS_k}\bigl(\mathrm{tuples}_k(sh)\bigr) = \rho_{TS_k}(sh)$ and that there is a one to one correspondence between TS_k and $\wp\bigl(\mathrm{tuples}_k(VI)\bigr)$. The isomorphism is given by the functions $\mathrm{tuples}_k \colon TS_k \to \wp\bigl(\mathrm{tuples}_k(VI)\bigr)$ and $\rho_{TS_k} \colon \wp\bigl(\mathrm{tuples}_k(VI)\bigr) \to TS_k$. Thus the domain TS_k is the smallest domain that can represent properties characterized by sets of variables of cardinality k. We now consider the tuple-sharing domains for the cases when $k = 1$, 2, and n.

Definition 5. (The *groundness* domain Con.) *The upper closure operator $\rho_{Con} \colon SH \to SH$ and the corresponding domain Con are defined as $\rho_{Con} \stackrel{\mathrm{def}}{=} \rho_{TS_1}$ and $Con \stackrel{\mathrm{def}}{=} TS_1 = \rho_{Con}(SH)$.*

This domain, which represents groundness information, is isomorphic to the domain of conjunctions of Boolean variables. The isomorphism tuples_1 maps each element of Con to the set of variables that are possibly non-ground. The usual domain (also normally called Con) for representing groundness can be obtained (as for Pos and Def) by set complementation.

Definition 6. (The *pair-sharing* domain PS.) *The upper closure operator $\rho_{PS} \colon SH \to SH$ and the corresponding domain PS are defined as $\rho_{PS} \overset{\mathrm{def}}{=} \rho_{TS_2}$ and $PS \overset{\mathrm{def}}{=} TS_2 = \rho_{PS}(SH)$.*

This domain represents pair-sharing information and the isomorphism tuples_2 maps each element of PS to the set of pairs of variables that may be bound to terms that share a common variable. The domain for representing variable independence can be obtained by set complementation. Finally, in the case when $k = n$ we have a domain consisting of just two elements:

$$TS_n = \bigl\{ SG, SG \setminus \{ VI \} \bigr\}.$$

Just as for SH, the domains TS_k are dual-atomistic and:

$$\mathrm{dAtoms}(TS_k) = \Bigl\{ \bigl(SG \setminus \{ U \in SG \mid T \subseteq U \} \bigr) \Bigm| T \in \mathrm{tuples}_k(VI) \Bigr\}.$$

Thus we have

$$\begin{aligned}
\mathrm{dAtoms}(Con) &= \Bigl\{ \bigl(SG \setminus \{ U \in SG \mid x \in U \} \bigr) \Bigm| x \in VI \Bigr\}, \\
\mathrm{dAtoms}(PS) &= \Bigl\{ \bigl(SG \setminus \{ U \in SG \mid x, y \in U \} \bigr) \Bigm| x \neq y \in VI \Bigr\}.
\end{aligned}$$

Example 2. Consider Example 1. Then the dual atoms of Con are

$$\begin{array}{l}
\{ \quad y, z, \qquad\quad yz \}, \\
\{ x, \quad z, \quad\ xz \quad \}, \\
\{ x, y, \quad xy \qquad \},
\end{array}$$

and the dual atoms of PS are

$$\begin{array}{l}
\{ x, y, z, \qquad xz, yz \}, \\
\{ x, y, z, xy, \qquad yz \}, \\
\{ x, y, z, xy, xz \qquad \}.
\end{array}$$

It can be seen from the dual atoms, that the precision of the information encoded by domains TS_j and TS_k is not comparable when $j \neq k$. Also, we note that, if $j < k$, then $\rho_{TS_j}(TS_k) = \{SG\}$ and $\rho_{TS_k}(TS_j) = TS_j$.

3.3 The Tuple-Sharing Dependency Domains

We now need to define domains that capture the propagation of groundness and pair-sharing; in particular, the dependency of these properties on the further instantiation of the variables. In the same way as with TS_k for Con and PS, we first define a general subdomain TSD_k of SH. This must be safe with respect to the tuple-sharing property represented by TS_k when performing the usual abstract operations. This was the motivation behind the introduction in [1] of the pair-sharing dependency domain SH^{ρ}. We now generalize this for tuple-sharing.

Definition 7. The *tuple-sharing dependency* domain (TSD_k.) *For each k where $1 \leq k \leq n$, the function $\rho_{\scriptscriptstyle TSD_k} \colon SH \to SH$ is defined as*

$$\begin{aligned}\rho_{\scriptscriptstyle TSD_k}&(sh)\\ &\stackrel{\text{def}}{=} \Bigl\{\, S \in SG \Bigm| \forall T \subseteq S : \# T < k \implies S = \bigcup \{\, U \in sh \mid T \subseteq U \subseteq S \,\} \,\Bigr\},\end{aligned}$$

and, as $\rho_{\scriptscriptstyle TSD_k} \in \mathrm{uco}(SH)$, it induces the tuple-sharing dependency *lattice*

$$TSD_k \stackrel{\text{def}}{=} \rho_{\scriptscriptstyle TSD_k}(SH).$$

It follows from the definitions that the domains TSD_k form a strict chain.

Proposition 1. *For $j, k \in \mathbb{N}$ with $1 \leq j < k \leq n$, we have $TSD_j \subset TSD_k$.*

Moreover, TSD_k is not less precise than TS_k.

Proposition 2. *For $k \in \mathbb{N}$ with $1 \leq k \leq n$, we have $TS_k \subseteq TSD_k$. Furthermore, if $n > 1$ then $TS_k \subset TSD_k$.*

A consequence of Propositions 1 and 2 is that TSD_k is not less precise than $TS_1 \sqcap \cdots \sqcap TS_k$.

Corollary 2. *For $j, k \in \mathbb{N}$ with $1 \leq j \leq k \leq n$, we have $TS_j \subseteq TSD_k$.*

It also follows from the definitions that, for the TSD_k domain, the star-union operator can be replaced by the k-self-union operator.

Proposition 3. *For $1 \leq k \leq n$, we have $\rho_{\scriptscriptstyle TSD_k}(sh^k) = sh^\star$.*

We consider the tuple-sharing dependency domains for the cases when $k = 1$, 2, and n.

Definition 8. (The *ground dependency* domain Def.) *The upper closure operator $\rho_{\scriptscriptstyle Def} \colon SH \to SH$ and the corresponding domain Def are defined as $\rho_{\scriptscriptstyle Def} \stackrel{\text{def}}{=} \rho_{\scriptscriptstyle TSD_1}$ and $\mathit{Def} \stackrel{\text{def}}{=} TSD_1 = \rho_{\scriptscriptstyle Def}(SH)$.*

By Proposition 3, we have for all $sh \in SH$, $\rho_{\scriptscriptstyle TSD_1}(sh) = sh^\star$ so that TSD_1 is a representation of the domain Def used for capturing groundness. It also confirms the well-known result that the star-union operator is redundant for elements in Def.

Definition 9. (The *pair-sharing dependency* domain PSD.) *The upper closure operator $\rho_{\scriptscriptstyle PSD} \colon SH \to SH$ and the corresponding domain PSD are defined as $\rho_{\scriptscriptstyle PSD} \stackrel{\text{def}}{=} \rho_{\scriptscriptstyle TSD_2}$ and $PSD \stackrel{\text{def}}{=} TSD_2 = \rho_{\scriptscriptstyle PSD}(SH)$.*

Then, it follows from [1, Theorem 7] that PSD is in fact the domain, corresponding to SH^ρ, defined for capturing pair-sharing. By Proposition 3, we have for all $sh \in SH$, $\rho_{\scriptscriptstyle PSD}(sh^2) = sh^\star$; thus confirming the result in [1] that for elements in PSD, the star-union operator $sh^\star$ can be replaced by the 2-self-union

$sh^2 = \mathrm{bin}(sh, sh)$ without any loss of precision. Furthermore, Corollary 2 confirms the result established in [1] that *PSD* also captures groundness. Finally, letting $k = n$, we observe that $\mathit{TSD}_n = \mathit{SH}$.

In [1], the *PSD* domain is shown to be as good as *SH* for both representing and propagating pair-sharing. It is also proved that any weaker domain does not satisfy these properties, so that *PSD* is the quotient [6] of *SH* with respect to the pair-sharing property *PS*. In the view of recent results on abstract domain completeness [9], this also means that *PSD* is the *least fully-complete extension* (*lfce*) of *PS* with respect to *SH*.

From a *purely theoretical* point of view, the quotient of an abstract interpretation with respect to a property of interest and the least fully-complete extension of an upper closure operator are not equivalent. It is known [6] that the quotient may not exist, while the lfce is always defined. However, it is also known [10] that when the quotient exists it is exactly the same as the lfce. Moreover, it should be noted that the quotient will exist as long as we consider a semantics where *at least one* of the domain operators is additive and this is almost always the case (just consider the merge-over-all-paths operator, usually implemented as the lub of the domain). Therefore, for the domains considered here, these two approaches to the completeness problem in abstract interpretation are equivalent.

We now generalize and strengthen the result in [1] and show that, for each $k \in \{1, \ldots, n\}$, TSD_k is the quotient of *SH* with respect to the reduced product $\mathit{TS}_1 \sqcap \cdots \sqcap \mathit{TS}_k$ (equivalently, the lfce of $\mathit{TS}_1 \sqcap \cdots \sqcap \mathit{TS}_k$ with respect to *SH*).

The following results can be obtained by generalizing the corresponding results in [1].

Theorem 3. *Let $sh_1, sh_2 \in \mathit{SH}$ and $1 \leq k \leq n$. If $\rho_{\mathit{TSD}_k}(sh_1) = \rho_{\mathit{TSD}_k}(sh_2)$ then, for each $\sigma \in \mathit{Subst}$, each $sh' \in \mathit{SH}$, and each $V \in \wp_{\mathrm{f}}(\mathit{VI})$,*

$$\begin{aligned}
\rho_{\mathit{TSD}_k}\bigl(\mathrm{amgu}(sh_1, \sigma)\bigr) &= \rho_{\mathit{TSD}_k}\bigl(\mathrm{amgu}(sh_2, \sigma)\bigr), \\
\rho_{\mathit{TSD}_k}(sh' \cup sh_1) &= \rho_{\mathit{TSD}_k}(sh' \cup sh_2), \\
\rho_{\mathit{TSD}_k}\bigl(\mathrm{proj}(sh_1, V)\bigr) &= \rho_{\mathit{TSD}_k}\bigl(\mathrm{proj}(sh_2, V)\bigr).
\end{aligned}$$

Theorem 4. *Let $1 \leq k \leq n$ For each $sh_1, sh_2 \in \mathit{SH}$, $\rho_{\mathit{TSD}_k}(sh_1) \neq \rho_{\mathit{TSD}_k}(sh_2)$ implies*

$$\exists \sigma \in \mathit{Subst}, \exists j \in \{1, \ldots, k\} \;.\; \rho_{\mathit{TS}_j}\bigl(\mathrm{amgu}(sh_1, \sigma)\bigr) \neq \rho_{\mathit{TS}_j}\bigl(\mathrm{amgu}(sh_2, \sigma)\bigr).$$

4 The Meet-Irreducible Elements

In Section 5, we use the method of Filé and Ranzato [8] to decompose the dependency domains TSD_k. In preparation for this, in this section, we identify the meet-irreducible elements for the domains and state some general results.

We have already observed that TS_k and $\mathit{TSD}_n = \mathit{SH}$ are dual-atomistic. However, TSD_k, for $k < n$, is not dual-atomistic and we need to identify the meet-irreducible elements. In fact, the set of dual-atoms for TSD_k is

$$\mathrm{dAtoms}(\mathit{TSD}_k) = \bigl\{\, \mathit{SG} \setminus \{S\} \bigm| S \in \mathit{SG}, \# S \leq k \,\bigr\}.$$

Note that $\#\,\mathrm{dAtoms}(\mathit{TSD}_k) = \sum_{j=1}^{k} \binom{n}{j}$. Specializing this for $k = 1$ and $k = 2$, respectively, we have:

$$\begin{aligned}
\mathrm{dAtoms}(\mathit{Def}) &= \bigl\{\, \mathit{SG} \setminus \{\{x\}\} \bigm| x \in \mathit{VI} \,\bigr\}, \\
\mathrm{dAtoms}(\mathit{PSD}) &= \bigl\{\, \mathit{SG} \setminus \{S\} \bigm| S \in \mathrm{pairs}(\mathit{VI}) \,\bigr\} \cup \mathrm{dAtoms}(\mathit{Def}),
\end{aligned}$$

and that $\#\,\mathrm{dAtoms}(\mathit{Def}) = n$ and $\#\,\mathrm{dAtoms}(\mathit{PSD}) = n(n+1)/2$. We present as an example of this the dual atoms for *Def* and *PSD* when $n = 3$.

Example 3. Consider Example 1. Then the 3 dual-atoms for *Def* are s_1, s_2, s_3 and the 6 dual atoms for *PSD* are $s_1, \ldots, s_6$. Note that these are not all the meet-irreducible elements since sets that do not contain xyz such as $\bot = \rho_{\mathit{Def}}(\bot) = \emptyset$ and $\{x\}$ cannot be obtained by the meet (which is set intersection) of a set of dual-atoms. Thus, unlike *Con* and *PS*, neither *Def* nor *PSD* are dual-atomistic.

Consider next the subset M_k of meet-irreducible elements of TSD_k that are neither the top element *SG* nor dual atoms. M_k has an element for each sharing group $S \in \mathit{SG}$ such that $\#\,S > k$ and each tuple $T \subset S$ with $\#\,T = k$. Such an element is obtained from *SG* by removing all sharing groups U such that $T \subseteq U \subseteq S$. Formally,

$$M_k \stackrel{\mathrm{def}}{=} \bigl\{\, \mathit{SG} \setminus \{\, U \in \mathit{SG} \mid T \subseteq U \subseteq S \,\} \bigm| T, S \in \mathit{SG}, T \subset S, \#\,T = k \,\bigr\}.$$

Note that, as there are $\binom{n}{k}$ possible choices for T and $2^{n-k} - 1$ possible choices for S, we have $\#\,M_k = \binom{n}{k}(2^{n-k} - 1)$ and $\#\,\mathrm{MI}(\mathit{TSD}_k) = \sum_{j=0}^{k-1} \binom{n}{j} + \binom{n}{k} 2^{n-k}$. Again, we illustrate this definition with the case when $n = 3$.

Example 4. Consider again Example 3. First, consider the domain *Def*. The meet-irreducible elements which are not dual-atoms, beside *SG*, are the following:

$$\begin{aligned}
q_1 &= \{\quad y, z, \quad\;\; xz, yz, xyz\} \subset s_1, & & \\
q_2 &= \{\quad y, z, xy, \quad\;\; yz, xyz\} \subset s_1, & r_1 &= \{\quad y, z, \qquad\qquad yz\} \subset q_1 \cap q_2, \\
q_3 &= \{x, \quad z, \quad\;\; xz, yz, xyz\} \subset s_2, & & \\
q_4 &= \{x, \quad z, xy, xz, \quad\;\; xyz\} \subset s_2, & r_2 &= \{x, \quad z, \qquad xz \qquad\} \subset q_3 \cap q_4, \\
q_5 &= \{x, y, \quad xy, \quad\;\; yz, xyz\} \subset s_3, & & \\
q_6 &= \{x, y, \quad xy, xz, \quad\;\; xyz\} \subset s_3, & r_3 &= \{x, y, \quad xy \qquad\quad\} \subset q_5 \cap q_6.
\end{aligned}$$

Next, consider the domain *PSD*. The only meet-irreducible elements that are not dual-atoms, beside *SG*, are the following:

$$\begin{aligned}
m_1 &= \{x, y, z, \quad\;\; xz, yz \qquad\} \subset s_4 \\
m_2 &= \{x, y, z, xy, \quad\;\; yz \qquad\} \subset s_5 \\
m_3 &= \{x, y, z, xy, xz \qquad\quad\} \subset s_6.
\end{aligned}$$

Each of these lack a pair and none contains the sharing group xyz.

We now show that we have identified precisely all the meet-irreducible elements of TSD_k.

Theorem 5. *If $k \in \mathbb{N}$ with $1 \leq k \leq n$, then*

$$\mathrm{MI}(\mathit{TSD}_k) = \{\mathit{SG}\} \cup \mathrm{dAtoms}(\mathit{TSD}_k) \cup M_k.$$

As a consequence, we have the following result.

Corollary 3. *Let $k \in \mathbb{N}$ with $1 \leq k \leq n$. Then*

$$\mathrm{dAtoms}(\mathit{TS}_k) = \bigl\{\, \mathit{sh} \in \mathrm{MI}(\mathit{TSD}_k) \bigm| \mathit{VI} \notin \mathit{sh} \,\bigr\}.$$

For the decomposition, we need to identify which meet-irreducible elements of TSD_k are in TS_j. Using Corollaries 2 and 3 we have the following result.

Corollary 4. *If $j, k \in \mathbb{N}$ with $1 \leq j < k \leq n$, then* $\mathrm{MI}(\mathit{TSD}_k) \cap \mathit{TS}_j = \{\mathit{SG}\}$.

By combining Proposition 1 with Theorem 5 we can identify the meet-irreducible elements of TSD_k that are in TSD_j, where $j < k$.

Corollary 5. *If $j, k \in \mathbb{N}$ with $1 \leq j < k \leq n$, then*

$$\mathrm{MI}(\mathit{TSD}_k) \cap \mathit{TSD}_j = \mathrm{dAtoms}(\mathit{TSD}_j).$$

5 The Decomposition of the Domains

5.1 Removing the Tuple-Sharing Domains

We first consider the decomposition of TSD_k with respect to TS_j. It follows from Theorem 1 and Corollaries 2 and 4 that, for $1 \leq j < k \leq n$, we have

$$\mathit{TSD}_k \sim \mathit{TS}_j = \mathit{TSD}_k. \tag{1}$$

Since $\mathit{SH} = \mathit{TSD}_n$, we have, using Eq. (1) and setting $k = n$, that, if $j < n$,

$$\mathit{SH} \sim \mathit{TS}_j = \mathit{SH}. \tag{2}$$

Thus, in general, TS_j is too abstract to be removed from SH by means of complementation. (Note that here it is required $j < n$, because we have $\mathit{SH} \sim \mathit{TS}_n \neq \mathit{SH}$.) In particular, letting $j = 1, 2$ (assuming $n > 2$) in Eq. (2), we have

$$\mathit{SH} \sim \mathit{PS} = \mathit{SH} \sim \mathit{Con} = \mathit{SH}, \tag{3}$$

showing that Con and PS are too abstract to be removed from SH by means of complementation. Also, by Eq. (1), letting $j = 1$ and $k = 2$ it follows that the complement of Con in PSD is PSD.

Now consider decomposing TSD_k using TS_k. It follows from Theorem 1, Proposition 2 and Corollary 3 that, for $1 \leq k \leq n$, we have

$$\begin{aligned} \mathit{TSD}_k \sim \mathit{TS}_k &= \mathrm{Moore}\bigl(\mathrm{MI}(\mathit{TSD}_k) \setminus \rho_{\mathit{TS}_k}(\mathit{TSD}_k)\bigr) \\ &= \{\, \mathit{sh} \in \mathit{TSD}_k \mid \mathit{VI} \in \mathit{sh} \,\}. \end{aligned} \tag{4}$$

Thus we have

$$\mathit{TSD}_k \sim (\mathit{TSD}_k \sim \mathit{TS}_k) = \mathit{TS}_k. \tag{5}$$

We have therefore extracted *all* the domain TS_k from TSD_k. So by letting $k = 1$, 2 in Eq. (5), we have found the complements of Con in Def and PS in PSD:

$$\begin{aligned}\mathit{Def} \sim \mathit{Con} &= \{\, \mathit{sh} \in \mathit{Def} \mid \mathit{VI} \in \mathit{sh} \,\},\\ \mathit{PSD} \sim \mathit{PS} &= \{\, \mathit{sh} \in \mathit{PSD} \mid \mathit{VI} \in \mathit{sh} \,\}.\end{aligned}$$

Thus if we denote the domains induced by these complements as $\mathit{Def}^{\oplus}$ and $\mathit{PSD}^{\oplus}$, respectively, we have the following result.

Theorem 6.

$$\begin{aligned}\mathit{Def} \sim \mathit{Con} &= \mathit{Def}^{\oplus}, &\qquad \mathit{Def} \sim \mathit{Def}^{\oplus} &= \mathit{Con},\\ \mathit{PSD} \sim \mathit{PS} &= \mathit{PSD}^{\oplus}, &\qquad \mathit{PSD} \sim \mathit{PSD}^{\oplus} &= \mathit{PS}.\end{aligned}$$

Moreover, Con and $\mathit{Def}^{\oplus}$ form a minimal decomposition for Def and, similarly, PS and $\mathit{PSD}^{\oplus}$ form a minimal decomposition for PSD.

5.2 Removing the Dependency Domains

First we note that, by Theorem 5, Proposition 1 and Corollary 5 the complement of TSD_j in TSD_k, where $1 \leq j < k \leq n$, is given as follows:

$$\begin{aligned}\mathit{TSD}_k \sim \mathit{TSD}_j &= \mathrm{Moore}\big(\mathrm{MI}(\mathit{TSD}_k) \setminus \rho_{\mathit{TSD}_j}(\mathit{TSD}_k)\big)\\ &= \big\{\, \mathit{sh} \in \mathit{TSD}_k \;\big|\; \forall S \in \mathit{SG} : \# S \leq j \implies S \in \mathit{sh} \,\big\}.\end{aligned} \tag{6}$$

It therefore follows from Eq. (6) and setting $k = n$ that the complement of ρ_{TSD_j} in SH for $j < n$ is:

$$\begin{aligned}\mathit{SH} \sim \mathit{TSD}_j &= \big\{\, \mathit{sh} \in \mathit{SH} \;\big|\; \forall S \in \mathit{SG} : \# S \leq j \implies S \in \mathit{sh} \,\big\}\\ &\overset{\text{def}}{=} \mathit{SH}^{+}_{j}.\end{aligned} \tag{7}$$

In particular, in Eq. (7) when $j = 1$, we have the following result for Def (also proved in [8, Lemma 5.4]):

$$\mathit{SH} \sim \mathit{Def} = \big\{\, \mathit{sh} \in \mathit{SH} \;\big|\; \forall x \in \mathit{VI} : \{x\} \in \mathit{sh} \,\big\}.$$

This complement is denoted by $\mathit{SH}^{+}_{\mathit{Def}}$. Also, in Eq. (7) when $j = 2$, we have the following result for PSD:

$$\mathit{SH} \sim \mathit{PSD} = \big\{\, \mathit{sh} \in \mathit{SH} \;\big|\; \forall S \in \mathit{SG} : \# S \leq 2 \implies S \in \mathit{sh} \,\big\}.$$

This complement is denoted by $\mathit{SH}^{+}_{\mathit{PSD}}$.

We next construct the complement of PSD with respect to Def. By Eq. (6),

$$\begin{aligned}\mathit{PSD} \sim \mathit{Def} &= \big\{\, \mathit{sh} \in \mathit{PSD} \;\big|\; \forall x \in \mathit{VI} : \{x\} \in \mathit{sh} \,\big\}\\ &\overset{\text{def}}{=} \mathit{PSD}^{+}.\end{aligned}$$

Then the complement factor $\mathit{Def}^{-} \overset{\text{def}}{=} \mathit{PSD} \sim \mathit{PSD}^{+}$ is exactly the same as $\mathit{SH} \sim \mathit{SH}^{+}_{\mathit{Def}}$ and PSD and SH behave similarly for Def.

5.3 Completing the Decomposition

Just as for SH, the complement of SH^{+}_{Def} using PS (or, more generally, TS_j where $1 < j < n$) is SH^{+}_{Def}. By Corollary 3 and Theorem 1, as PS is dual-atomistic, the complement of PS in PSD^{+} is given as follows.

Theorem 7.

$$\begin{aligned} PSD^{\ddagger} &\stackrel{\mathrm{def}}{=} PSD^{+} \sim PS \\ &= \bigl\{\, sh \in PSD \bigm| VI \in sh, \forall x \in VI : \{x\} \in sh \,\bigr\}, \\ PSD^{+} \sim PSD^{\ddagger} &= PS. \end{aligned}$$

So, we have extracted *all* the domain PS from PSD^{+} and we have the following result.

Corollary 6. *Def^{-}, PS, and $PSD^{\ddagger}$ form a minimal decomposition for PSD.*

6 Conclusion

In [1], we said that $PSD \sim PS \neq PSD$. This paper now clarifies that statement. We have provided a minimal decomposition for PSD whose components include Def^{-} and PS. Moreover, we have shown that PSD is *not* dual-atomistic. The meet-irreducible elements of PSD have been completely specified. As a consequence, it can be seen that the dual-atoms of PSD are precisely those elements which have the form $SG \setminus \{S\}$ where $\# S \leq 2$.

By studying the sharing domain SH in a more general framework, we have been able to show that the domain PSD has natural place in a scheme of domains based on SH. Moreover, by means of this approach we have highlighted the close relationship between Def and PSD and the many properties they share.

Our starting point was the work of Filé and Ranzato. In [8], they noted, as we have, that $SH^{+}_{Def} \sim PS = SH^{+}_{Def}$ so that none of the domain PS could be extracted from SH^{+}_{Def}. They noticed that ρ_{PS} maps all dual-atoms that contain VI to SG and thus lose all pair-sharing information. To avoid this, they replaced PS with the domain PS' where, for all $sh \in SH^{+}_{Def}$, $\rho_{PS'}(sh) = \rho_{PS}(sh) \setminus \bigl(\{VI\} \setminus sh\bigr)$, and noted that $SH^{+}_{Def} \sim PS' = \{\, sh \in SH^{+}_{Def} \mid VI \in sh \,\}$. To understand the nature of this new domain PS', we first observe that, PS' is simply the reduced product of PS and TS_n. This is because $TS_n = \mathrm{MI}(TS_n) = \bigl\{SG \setminus \{VI\}, SG\bigr\}$. In addition, $SH^{+}_{Def} \sim TS_n = \{\, sh \in SH^{+}_{Def} \mid VI \in sh \,\}$, which is precisely the same as $SH^{+}_{Def} \sim PS'$. Thus, since $SH^{+}_{Def} \sim PS = SH^{+}_{Def}$, it is not surprising that it is precisely the *added* component TS_n that is removed when we compute the complement for SH^{+}_{Def} with respect to PS'.

We also note that, in [8], the fact that one of the components found by decomposition (here called TS_n) has only two elements is seemingly regarded as a positive thing, because [8, Section 1] "This shows that domain decomposition can lead to great gains in the size of the representation." In our humble opinion, if one of the components is very small (e.g., only 2 elements in this case) this means that almost all the complexity remained elsewhere. If the objective is to decompose in order to find space saving representations for domains and to simplify domain verification problems, then "balanced" decompositions (that is, into parts of nearly equal complexity) should instead be regarded as the best one can hope for.

It should be stressed that the problems outlined above are not dependent on the particular domain chosen and, in our opinion, they are mainly related to the methodology for decomposing a domain. Indeed, we argue that complementation *alone* is not sufficient to obtain *truly minimal* decompositions of domains. The reason being that complementation only depends on the domain's data (that is, the domain elements and the partial order relation modeling their intrinsic precision), while it is completely independent from the domain operators that manipulate that data. In particular, if the concrete domain contains elements that are redundant with respect to its operators (because the observable behavior of these elements is exactly the same in all possible program contexts) then any factorization of the domain obtained by complementation will encode this redundancy. However, the theoretical solution to this problem is well known [6, 9, 10] and it is straightforward to improve the methodology so as to obtain *truly minimal* decompositions: *first* remove all redundancies from the domain (this can be done by computing the quotient of a domain with respect to the observable behavior) and only *then* decompose it by complementation. This is *exactly* what is done here.

We conclude with some remarks on complementation. There are a number of reasons why we believe this is important. First of all, complementation is really an excellent concept to work with from a theoretical point of view: it allows the splitting of complex domains into simpler components, avoiding redundancies between them. However, as things stand at present, complementation has never been exploited. This may be because it is easier to implement a single complex domain than to implement several simpler domains and integrate them together. Note that complementation requires the implementation of a full-integration between components (i.e., the reduced product), otherwise precision would be lost and the theoretical results would not apply.

One notable example of domain decomposition that does enable significant memory and time savings with no precision loss is the GER decomposition of *Pos* [2], and this is not based on complementation. Observe that the complement of G with respect to Pos is Pos itself. This is because Pos is isomorphic to SH [3] and $G \equiv \mathit{Con} \stackrel{\mathrm{def}}{=} \mathit{TS}_1$ so that, by Eq. (3), $\mathit{Pos} \sim G = \mathit{Pos}$. It is not difficult to observe that the same phenomenon happens if one considers the groundness equivalence component E, that is, $\mathit{Pos} \sim E = \mathit{Pos}$. In fact, it can be shown that two variables x and y are ground-equivalent in $sh \in \mathit{SH} \equiv \mathit{Pos}$ if and only if $\mathrm{rel}\bigl(\{x\}, sh\bigr) = \mathrm{rel}\bigl(\{y\}, sh\bigr)$. In particular, this implies both $\{x\} \notin sh$ and $\{y\} \notin sh$. Thus, it can be easily observed that in all the dual-atoms of Pos no variable is ground-equivalent to another variable (because each dual-atom lacks just a *single* sharing-group).

References

1. R. Bagnara, P. M. Hill, and E. Zaffanella. Set-sharing is redundant for pair-sharing. In P. Van Hentenryck, editor, *Static Analysis: Proceedings of the 4th International Symposium*, volume 1302 of *Lecture Notes in Computer Science*, pages 53–67, Paris, France, 1997. Springer-Verlag, Berlin.
2. R. Bagnara and P. Schachte. Factorizing equivalent variable pairs in ROBDD-based implementations of *Pos*. In A. M. Haeberer, editor, *Proceedings of the "Seventh International Conference on Algebraic Methodology and Software Technology (AMAST'98)"*, volume 1548 of *Lecture Notes in Computer Science*, pages 471–485, Amazonia, Brazil, 1999. Springer-Verlag, Berlin.
3. M. Codish and H Søndergaard. The Boolean logic of set sharing analysis. In C. Palamidessi, H. Glaser, and K. Meinke, editors, *Principles of Declarative Programming*, volume 1490 of *Lecture Notes in Computer Science*, pages 89–100, Pisa, Italy, 1998. Springer-Verlag, Berlin.
4. A. Cortesi, G. Filé, R. Giacobazzi, C. Palamidessi, and F. Ranzato. Complementation in abstract interpretation. In A. Mycroft, editor, *Static Analysis: Proceedings of the 2nd International Symposium*, volume 983 of *Lecture Notes in Computer Science*, pages 100–117, Glasgow, UK, 1995. Springer-Verlag, Berlin.
5. A. Cortesi, G. Filé, R. Giacobazzi, C. Palamidessi, and F. Ranzato. Complementation in abstract interpretation. *ACM Transactions on Programming Languages and Systems*, 19(1):7–47, 1997.
6. A. Cortesi, G. Filé, and W. Winsborough. The quotient of an abstract interpretation for comparing static analyses. *Theoretical Computer Science*, 202(1&2):163–192, 1998.
7. P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Programming Languages*, pages 269–282, 1979.
8. G. Filé and F. Ranzato. Complementation of abstract domains made easy. In M. Maher, editor, *Logic Programming: Proceedings of the Joint International Conference and Symposium on Logic Programming*, MIT Press Series in Logic Programming, pages 348–362, Bonn, Germany, 1996. The MIT Press.
9. R. Giacobazzi and F. Ranzato. Completeness in abstract interpretation: a domain perspective. In M. Johnson, editor, *Proceedings of the 6th International Conference on Algebraic Methodology and Software Technology (AMAST'97)*, volume 1349 of *Lecture Notes in Computer Science*, pages 231–245, Sydney, Australia, 1997. Springer-Verlag, Berlin.
10. R. Giacobazzi, F. Ranzato, and F. Scozzari. Complete abstract interpretations made constructive. In J. Gruska and J. Zlatuska, editors, *Proceedings of 23rd International Symposium on Mathematical Foundations of Computer Science (MFCS'98)*, volume 1450 of *Lecture Notes in Computer Science*, pages 366–377. Springer-Verlag, Berlin, 1998.
11. G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott. *A Compendium of Continuous Lattices*. Springer-Verlag, Berlin, 1980.
12. P. M. Hill, R. Bagnara, and E. Zaffanella. The correctness of set-sharing. In G. Levi, editor, *Static Analysis: Proceedings of the 5th International Symposium*, volume 1503 of *Lecture Notes in Computer Science*, pages 99–114, Pisa, Italy, 1998. Springer-Verlag, Berlin.
13. D. Jacobs and A. Langen. Static analysis of logic programs for independent AND parallelism. *Journal of Logic Programming*, 13(2&3):291–314, 1992.