Decomposing Non-Redundant Sharing by Complementation

ENEA ZAFFANELLA
Department of Mathematics, University of Parma, Italy (*e-mail:* `zaffanella@cs.unipr.it`)

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

ROBERTO BAGNARA†
Department of Mathematics, University of Parma, Italy (*e-mail:* `bagnara@cs.unipr.it`)

Abstract

Complementation, the inverse of the reduced product operation, is a 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 result 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, a proper decomposition is obtained if the non-redundant version of *SH*, *PSD*, is substituted for *SH*. To establish the results for *PSD*, we define a general schema for subdomains of *SH* that includes *PSD* and *Def* as special cases. This sheds new light on the structure of *PSD* and exposes a natural though unexpected connection between *Def* and *PSD*. 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 (Cortesi, Filé, Giacobazzi, Palamidessi and Ranzato 1997), which is the inverse of the well-known reduced product operation (Cousot and Cousot 1979), can systematically obtain minimal decompositions of complex abstract domains. It has been argued that these decompositions would be useful in finding

* This work was partly supported by EPSRC under grant GR/M05645.
† The work of the first and third authors has been partly supported by MURST project "Certificazione automatica di programmi mediante interpretazione astratta."

space saving representations for domains and to simplify domain verification problems.

In (Filé and Ranzato 1996), Filé and Ranzato presented a new method for computing the complement, which is simpler than the original proposal by Cortesi *et al.* (Cortesi, Filé, Giacobazzi, Palamidessi and Ranzato 1995, Cortesi et al. 1997) 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 (Jacobs and Langen 1992), *SH*, for representing properties of variables such as groundness and sharing. This domain captures the property of set-sharing. Filé and Ranzato illustrated their method by minimally decomposing *SH* into three components; using the words of the authors (Filé and Ranzato 1996, 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. Thus, in order to obtain their non-trivial decomposition of *SH*, they used a different (and somewhat unnatural) definition for an alternative pair-sharing domain, called PS'. The nature of PS' and its connection with *PS* is examined more carefully in Section 6.

We noticed that the reason why Filé and Ranzato obtained this result was not to be found in the definition of *PS*, which accurately represents the property of pair-sharing, but in the use of the domain *SH* to capture the property of pair-sharing. In (Bagnara, Hill and Zaffanella 1997, Bagnara, Hill and Zaffanella 2001), it was observed that, for most (if not all) applications, the property of interest is not set-sharing but pair-sharing. Moreover, it was shown that, for groundness and pair-sharing, *SH* includes redundant elements. By defining an upper closure operator ρ that removed this redundancy, a much smaller domain *PSD*, which was denoted SH^ρ in (Bagnara et al. 1997), was found that captured pair-sharing and groundness with the same precision as *SH*. We show here that using the method given in (Filé and Ranzato 1996), but with this domain instead of *SH* as the reference domain, a proper decomposition can be obtained even when considering the natural definition of the pair-sharing domain *PS*. Moreover, we show that *PS* is *exactly* one of the components obtained by complementation of *PSD*. 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 (Bagnara et al. 1997, Bagnara et al. 2001). We will show that such a problem disappears when *PSD* is used as the reference domain.

Although the primary purpose of this work is to clarify the decomposition of the domain *PSD*, 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 (Armstrong, Marriott, Schachte and Søndergaard 1998). 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 (Codish and Søndergaard 1998). This means that any general result regarding the structure of *SH* is equally applicable to *Pos* and its subdomains.

To establish the results for *PSD*, we define a general schema for subdomains of *SH* that includes *PSD* and *Def* as special cases. This sheds new light on the structure of the domain *PSD*, 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 *PSD*. The results confirm that *PSD* is, in fact, the "appropriate" abstraction of the set-sharing domain *SH* that has to be considered when groundness and pair-sharing are the properties of interest.

The paper, which is an extended version of (Zaffanella, Hill and Bagnara 1999), 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. In Section 6 we present some final remarks and we explain what is, in our opinion, the lesson to be learned from this and other related work. Section 7 concludes.

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*.

A poset $\langle P, \preceq \rangle$ is *totally ordered* with respect to '$\preceq$' if, for each $x, y \in P$, either $x \preceq y$ or $y \preceq x$. A subset S of a poset $\langle P, \preceq \rangle$ is a *chain* if it is totally ordered with respect to '$\preceq$'.

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 a *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,

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

is unique. In this case we write $y = \mathrm{lub}\, S$. *Lower bounds* and *greatest lower bounds* (or glb) are defined dually.

A poset $\langle L, \preceq \rangle$ such that, for each $x, y \in L$, both $\mathrm{lub}\{x, y\}$ and $\mathrm{glb}\{x, y\}$ 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.

As an alternative definition, a lattice is an algebra $\langle L, \wedge, \vee \rangle$ such that $\wedge$ 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 lattice are equivalent. This can be seen by defining:

$$x \preceq y \quad \stackrel{\mathrm{def}}{\iff} \quad x \wedge y = x \quad \stackrel{\mathrm{def}}{\iff} \quad x \vee y = y$$

and

$$\mathrm{glb}\{x, y\} \stackrel{\mathrm{def}}{=} x \wedge y,$$
$$\mathrm{lub}\{x, y\} \stackrel{\mathrm{def}}{=} x \vee y.$$

The existence of an isomorphism between the two lattices L_1 and L_2 is denoted by $L_1 \equiv L_2$.

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

$$\forall x \in P : x \preceq \rho(x).$$

Each upper closure operator ρ over a complete lattice C is uniquely determined by the set of its fixpoints, that is, by its image

$$\rho(C) \stackrel{\mathrm{def}}{=} \bigl\{\, \rho(x) \bigm| x \in C \,\bigr\}.$$

We will often denote upper closure operators by their images. The set of all upper closure operators over a complete lattice C, denoted by $\mathrm{uco}(C)$, forms a complete lattice ordered as follows: if $\rho_1, \rho_2 \in \mathrm{uco}(P)$, $\rho_1 \sqsubseteq \rho_2$ if and only if $\rho_2(C) \subseteq \rho_1(C)$. The *reduced product* of two elements ρ_1 and ρ_2 of $\mathrm{uco}(C)$ is denoted by $\rho_1 \sqcap \rho_2$ and defined as

$$\rho_1 \sqcap \rho_2 \stackrel{\mathrm{def}}{=} \mathrm{glb}\{\rho_1, \rho_2\}.$$

For a more detailed introduction to closure operators, the reader is referred to (Gierz, Hofmann, Keimel, Lawson, Mislove and Scott 1980).

A complete lattice C is *meet-continuous* if for any chain $Y \subseteq C$ and each $x \in C$,

$$x \wedge \Bigl(\bigvee Y\Bigr) = \bigvee_{y \in Y} (x \wedge y).$$

Most domains for abstract interpretation (Cortesi et al. 1997) and, in particular, all the domains considered in this paper are meet-continuous.

Assume that C is a meet-continuous lattice. Then the inverse of the reduced product operation, called *weak relative pseudo-complement*, is well defined and given as follows. Let $\rho, \rho_1 \in \mathrm{uco}(C)$ be such that $\rho \sqsubseteq \rho_1$. Then

$$\rho \sim \rho_1 \stackrel{\mathrm{def}}{=} \mathrm{lub}\{\, \rho_2 \in \mathrm{uco}(C) \mid \rho_1 \sqcap \rho_2 = \rho \,\}.$$

Given $\rho \in \mathrm{uco}(C)$, the *weak pseudo-complement* (or, by an abuse of terminology now customary in the field of Abstract Interpretation, simply *complement*) of ρ is denoted by $id_C \sim \rho$, where id_C is the identity over C. Let $D_i \stackrel{\mathrm{def}}{=} \rho_{D_i}(C)$ with $\rho_{D_i} \in \mathrm{uco}(C)$ 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 $E_k \in \mathrm{uco}(C)$, $D_k \sqsubset E_k$ implies

$$C \sqsubset D_1 \sqcap \cdots \sqcap D_{k-1} \sqcap E_k \sqcap D_{k+1} \sqcap \cdots \sqcap D_n.$$

Assume now that C is a complete lattice. If $X \subseteq C$, then $\mathrm{Moore}(X)$ denotes the *Moore completion of X*, namely,

$$\mathrm{Moore}(X) \stackrel{\mathrm{def}}{=} \left\{\, \bigwedge Y \;\middle|\; Y \subseteq X \,\right\}.$$

We say that C *is meet-generated by* X if $C = \mathrm{Moore}(X)$. 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 $\mathrm{MI}(C)$. Note that $\top \in \mathrm{MI}(C)$. 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 $\mathrm{dAtoms}(C)$. Note that $\mathrm{dAtoms}(C) \subset \mathrm{MI}(C)$. The domain C is *dual-atomistic* if $C = \mathrm{Moore}\bigl(\mathrm{dAtoms}(C)\bigr)$. Thus, if C is dual-atomistic, $\mathrm{MI}(C) = \{\top\} \cup \mathrm{dAtoms}(C)$. The following result holds (Filé and Ranzato 1996, Theorem 4.1).

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

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

Another interesting result is the following (Filé and Ranzato 1996, Corollary 4.5).

Theorem 2
If C is dual-atomistic then $\mathrm{uco}(C)$ is pseudo-complemented and for any $\rho \in \mathrm{uco}(C)$

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

Let *Vars* be a denumerable set of variables. For any syntactic object o, $vars(o)$ denotes the set of variables occurring in o. Let $\mathcal{T}_{Vars}$ be the set of first-order terms over *Vars*. If $x \in Vars$ and $t \in \mathcal{T}_{Vars} \setminus \{x\}$, then $x \mapsto t$ is called a *binding*. A *substitution* is a total function $\sigma \colon Vars \to \mathcal{T}_{Vars}$ that is the identity almost everywhere. Substitutions are denoted by the set of their bindings, thus a substitution σ is identified with the (finite) set

$$\left\{\, x \mapsto \sigma(x) \;\middle|\; x \neq \sigma(x) \,\right\}.$$

If $t \in \mathcal{T}_{Vars}$, we write $t\sigma$ to denote $\sigma(t)$. A substitution σ is *idempotent* if, for all $t \in \mathcal{T}_{Vars}$, we have $t\sigma\sigma = t\sigma$. The set of all idempotent substitutions is denoted by Subst.

It should be stressed that this restriction to idempotent substitutions is provided for presentation purposes only. In particular, it allows for a straight comparison of our work with respect to other works appeared in the literature. However, the results proved in this paper do not rely on the idempotency of substitutions and are therefore applicable also when considering substitutions in *rational solved form* (Colmerauer 1982, Colmerauer 1984). Indeed, we have proved in (Hill, Bagnara and Zaffanella 1998) that the usual abstract operations defined on the domain SH, approximating concrete unification over finite trees, also provide a correct approximation of concrete unification over a domain of rational trees.

3 The Sharing Domains

In order to provide a concrete meaning to the elements of the set-sharing domain of D. Jacobs and A. Langen (Jacobs and Langen 1989, Langen 1990, Jacobs and Langen 1992), a knowledge of the finite set $\mathit{VI} \subset \mathit{Vars}$ of variables of interest is required. For example, in the Ph.D. thesis of Langen (Langen 1990) this set is implicitly defined, for each clause being analyzed, as the finite set of variables occurring in that clause. A clearer approach has been introduced in (Cortesi, Filé and Winsborough 1994, Cortesi, Filé and Winsborough 1998) and also adopted in (Bagnara et al. 1997, Bagnara et al. 2001, Cortesi and Filé 1999), where the set of variables of interest is given explicitly as a component of the abstract domain. During the analysis process, this set is *elastic*. That is, it expands (e.g., when solving clause's bodies) and contracts (e.g., when abstract descriptions are projected onto the variables occurring in clause's heads). This technique has two advantages: first, a clear and unambiguous description of those semantic operators that modify the set of variables of interest is provided; second, the definition of the abstract domain is completely independent from the particular program being analyzed. However, since at any given time the set of variables of interest is fixed, we can simplify the presentation by consistently denoting this set by VI. Therefore, in this paper all the abstract domains defined are restricted to a fixed set of variables of interest VI of finite cardinality n; this set is not included explicitly in the representation of the domain elements; also, when considering abstract semantic operators having some arguments in Subst, such as the abstract mgu, the considered substitutions are always taken to have variables in VI. We would like to emphasize that this is done for ease of presentation only: the complete definition of both the domains and the semantic operators can be immediately derived from those given, e.g., in (Bagnara et al. 1997, Bagnara et al. 2001). Note that other solutions are possible; we refer the interested reader to (Cortesi, Filé and Winsborough 1996, Section 7) and (Scozzari 2001, Section 10), where this problem is discussed in the context of groundness analysis.

3.1 The Set-sharing Domain *SH*

Definition 1
(The *set-sharing* domain *SH*.) The domain *SH* is given by

$$SH \stackrel{\text{def}}{=} \wp(SG),$$

where the set of *sharing-groups* *SG* is given by

$$SG \stackrel{\text{def}}{=} \wp(VI) \setminus \{\emptyset\}.$$

SH is partially ordered by set inclusion so that the lub is given by set union and the 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. As usual in the field of abstract interpretation, this ordering provides a formalization of precision where the less precise domain elements are those occurring higher in the partial order. Thus, more precise elements contain less sharing groups.

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

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

In all the examples in this paper, the elements of *SH* are written in a simplified notation, omitting the inner braces. For instance, the set

$$\bigl\{\{x\}, \{x, y\}, \{x, z\}, \{x, y, z\}\bigr\}$$

would be written simply as

$$\{x, xy, xz, xyz\}.$$

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

$$\left.\begin{aligned} s_1 &= \{\ y, z, xy, xz, yz, xyz\}, \\ s_2 &= \{x, \ z, xy, xz, yz, xyz\}, \\ s_3 &= \{x, y, \ xy, xz, yz, xyz\}, \end{aligned}\right\} \quad \text{these lack a singleton;}$$

$$\left.\begin{aligned} s_4 &= \{x, y, z, \ xz, yz, xyz\}, \\ s_5 &= \{x, y, z, xy, \ yz, xyz\}, \\ s_6 &= \{x, y, z, xy, xz, \ xyz\}, \end{aligned}\right\} \quad \text{these lack a pair;}$$

$$s_7 = \{x, y, z, xy, xz, yz \}, \qquad \text{this lacks } VI.$$

The meet-irreducible elements of *SH* are $s_1, \ldots, s_7$, and the top element *SG*.

Definition 2
(Operations over *SH*.) The function $\mathrm{bin} \colon SH \times SH \to SH$, called *binary union*, is given, for each $sh_1, sh_2 \in SH$, by

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

The *star-union* function $(\cdot)^\star \colon SH \to SH$ is given, for each $sh \in SH$, by

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

The *j-self-union* function $(\cdot)^j \colon SH \to SH$ is given, for each $j \geq 1$ and $sh \in SH$, 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 $\mathrm{rel} \colon \wp(VI) \times SH \to SH$ given, for each $V \subseteq VI$ and each $sh \in SH$, 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 $sh \in SH$, $v_x = \{x\}$, $v_t = vars(t)$, 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, for each $sh \in SH$, by

$$\mathrm{proj}(sh, V) \stackrel{\text{def}}{=} \{ S \cap V \mid S \in sh, S \cap V \neq \emptyset \} \cup \bigl\{ \{x\} \bigm| x \in VI \setminus V \bigr\}.$$

Together with lub, the functions proj and amgu are the key operations that make the abstract domain SH suitable for computing static approximations of the substitutions generated by the execution of logic programs. These operators can be combined with simpler ones (e.g., consistent renaming of variables) so as to provide a complete definition of the abstract semantics. Also note that these three operators have been proved to be the *optimal approximations* of the corresponding concrete operators (Cortesi and Filé 1999). The j-self-union operator defined above is new. We show later when it may safely replace the star-union operator. Note that, letting $j = 1$, 2, and n, we have $sh^1 = sh$, $sh^2 = \mathrm{bin}(sh, sh)$, and, as $\# VI = n$, $sh^n = sh^\star$.

3.2 The Tuple-Sharing Domains

To provide a general characterization of domains such as the groundness and pair-sharing domains contained 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 $S \in SG$ and $sh \in SH$, 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

$$\rho_{Con} \stackrel{\mathrm{def}}{=} \rho_{TS_1},$$
$$Con \stackrel{\mathrm{def}}{=} TS_1(SH) = \rho_{Con}(SH).$$

This domain, which represents groundness information, is isomorphic to a 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. From the domain $\mathrm{tuples}_1(Con)$, by set complementation, we obtain the classical domain G (Jones and Søndergaard 1987) for representing the set of variables that are definitely ground (so that we have $TS_1 \stackrel{\mathrm{def}}{=} Con \equiv \mathsf{G}$).

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

$$\rho_{PS} \stackrel{\mathrm{def}}{=} \rho_{TS_2},$$
$$PS \stackrel{\mathrm{def}}{=} TS_2(SH) = \rho_{PS}(SH).$$

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:

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

Note that the bottom of TS_n differs from the top element SG only in that it lacks the sharing group VI. There is no intuitive reading for the information encoded by this element: it describes all but those substitutions $\sigma \in \mathit{Subst}$ such that $\bigcap \bigl\{ \mathit{vars}(x\sigma) \bigm| x \in VI \bigr\} \neq \emptyset$.

Just as for SH, the domain TS_k (where $1 \leq k \leq n$) is 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

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

Example 2

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

$$\begin{aligned} r_1 &= s_1 \cap s_4 \cap s_5 \cap s_7 = \{\; y, z, \qquad\qquad yz\}, \\ r_2 &= s_2 \cap s_4 \cap s_6 \cap s_7 = \{x, \; z, \qquad xz \qquad\}, \\ r_3 &= s_3 \cap s_5 \cap s_6 \cap s_7 = \{x, y, \qquad xy \qquad\qquad\}; \end{aligned}$$

the dual-atoms of PS are

$$\begin{aligned} m_1 &= s_4 \cap s_7 = \{x, y, z, \qquad xz, yz\}, \\ m_2 &= s_5 \cap s_7 = \{x, y, z, xy, \qquad yz\}, \\ m_3 &= s_6 \cap s_7 = \{x, y, z, xy, xz \qquad\}. \end{aligned}$$

It can be seen from the dual-atoms that, for each $j = 1, \ldots, n$, where $j \neq k$, the precision of the information encoded by domains TS_j and TS_k is not comparable. 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 (Bagnara et al. 1997, Bagnara et al. 2001) of the pair-sharing dependency domain *PSD*. 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 $\rho_{TSD_k} : SH \to SH$ is defined as

$$\rho_{TSD_k}(sh) \stackrel{\text{def}}{=} \left\{ S \in SG \;\middle|\; \forall T \subseteq S : \# T < k \implies S = \bigcup \{ U \in sh \mid T \subseteq U \subseteq S \} \right\},$$

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

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

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 $TS_k \subset TSD_k$.

As an immediate consequence of Propositions 1 and 2 we have that that TSD_k is not less precise than $TS_1 \sqcap \cdots \sqcap TS_k$.

Corollary 1

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 $\rho_{TSD_k}\left(sh^k\right) = sh^\star$.

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

Definition 8

(The *ground dependency* domain *Def*.**)** The domain *Def* is induced by the upper closure operator $\rho_{Def} \colon SH \to SH$. They are defined as

$$\rho_{Def} \stackrel{\text{def}}{=} \rho_{TSD_1},$$
$$Def \stackrel{\text{def}}{=} TSD_1 = \rho_{Def}(SH).$$

By Proposition 3, we have, for all $sh \in SH$, $\rho_{TSD_1}(sh) = sh^\star$ so that TSD_1 is a representation of the domain *Def* used for capturing groundness. It also provides evidence for the fact that the computation of the star-union is not needed for the elements in *Def*.

Definition 9

(The *pair-sharing dependency* domain *PSD*.**)** The upper closure operator $\rho_{PSD} \colon SH \to SH$ and the corresponding domain *PSD* are defined as

$$\rho_{PSD} \stackrel{\text{def}}{=} \rho_{TSD_2},$$
$$PSD \stackrel{\text{def}}{=} TSD_2 = \rho_{PSD}(SH).$$

Then, it follows from (Bagnara et al. 1997, Theorem 7) that *PSD* corresponds to the domain SH^ρ defined for capturing pair-sharing. By Proposition 3 we have, for all $sh \in SH$, that $\rho_{PSD}(sh^2) = sh^\star$, so 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. This was also proved in (Bagnara et al. 1997, Theorem 11). Furthermore, Corollary 1 confirms the observation made in (Bagnara et al. 1997) that *PSD* also captures groundness.

Finally, letting $k = n$, we observe that $TSD_n = SH$. Figure 1 summarizes the relations between the tuple-sharing and the tuple-sharing dependency domains.

As already discussed at the start of this section, the set of variables of interest *VI* is fixed and, to simplify the notation, omitted. In (Bagnara et al. 1997, Bagnara et al. 2001) the domains *SS* and SS^ρ (corresponding to *SH* and *PSD*, respectively) are instead obtained by explicitly adding to each domain element a new component, representing the set of variables of interest. It is shown that SS^ρ is as good as *SS* for both representing and propagating pair-sharing and it is also proved that any weaker domain does not satisfy these properties, so that SS^ρ is the quotient (Cortesi et al. 1994, Cortesi et al. 1998) of *SS* with respect to the pair-sharing property *PS*.

We now generalize and strengthen the results in (Bagnara et al. 1997, Bagnara et al. 2001) and show that, for each $k \in \{1, \dots, n\}$, TSD_k is the quotient of *SH* with respect to the reduced product $TS_1 \sqcap \cdots \sqcap TS_k$. These results are proved at the end of this section.

Theorem 3

Let $sh_1, sh_2 \in SH$ and $1 \leq k \leq n$. If $\rho_{TSD_k}(sh_1) = \rho_{TSD_k}(sh_2)$ then, for each

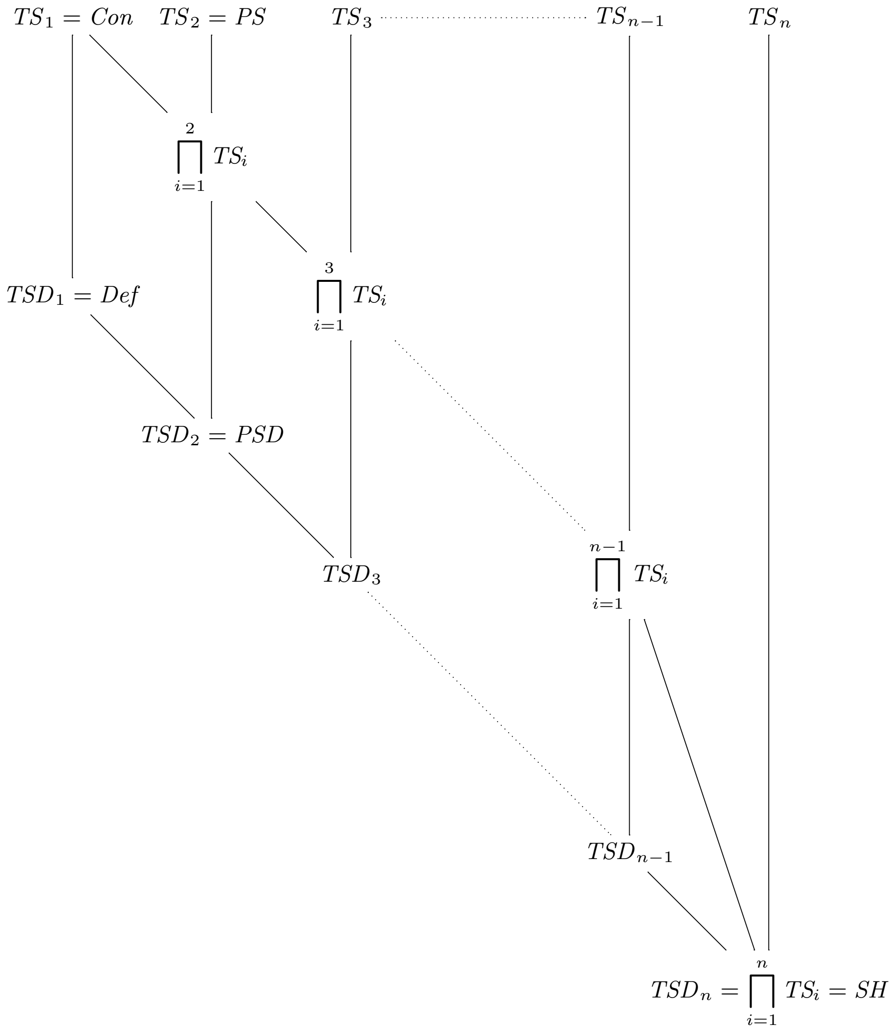

Fig. 1. The set-sharing domain SH and some of its abstractions.

$\sigma \in \mathit{Subst}$, each $sh' \in SH$, and each $V \in \wp(VI)$,

$$\rho_{TSD_k}\big(\mathrm{amgu}(sh_1, \sigma)\big) = \rho_{TSD_k}\big(\mathrm{amgu}(sh_2, \sigma)\big),$$
$$\rho_{TSD_k}(sh' \cup sh_1) = \rho_{TSD_k}(sh' \cup sh_2),$$
$$\rho_{TSD_k}\big(\mathrm{proj}(sh_1, V)\big) = \rho_{TSD_k}\big(\mathrm{proj}(sh_2, V)\big).$$

Theorem 4

Let $1 \leq k \leq n$ For each $sh_1, sh_2 \in SH$, $\rho_{TSD_k}(sh_1) \neq \rho_{TSD_k}(sh_2)$ implies

$$\exists \sigma \in \mathit{Subst}, \exists j \in \{1, \ldots, k\} \,.\, \rho_{TS_j}\big(\mathrm{amgu}(sh_1, \sigma)\big) \neq \rho_{TS_j}\big(\mathrm{amgu}(sh_2, \sigma)\big).$$

3.4 Proofs of Theorems 3 and 4

In what follows we use the fact that ρ_{TSD_k} is an upper closure operator so that, for each $sh_1, sh_2 \in SH$,

$$sh_1 \subseteq \rho_{TSD_k}(sh_2) \iff \rho_{TSD_k}(sh_1) \subseteq \rho_{TSD_k}(sh_2). \tag{1}$$

In particular, since $(\cdot)^\star = \rho_{TSD_1}$, we have

$$sh_1 \subseteq sh_2^\star \iff sh_1^\star \subseteq sh_2^\star. \tag{2}$$

Lemma 1
For each $sh \in SH$ and each $V \in \wp(VI)$,

$$\rho_{TSD_k}(sh) \setminus \mathrm{rel}\bigl(V, \rho_{TSD_k}(sh)\bigr) = \rho_{TSD_k}\bigl(sh \setminus \mathrm{rel}(V, sh)\bigr).$$

Proof
By Definition 7,

$$\begin{aligned} &S \in \rho_{TSD_k}\bigl(sh \setminus \mathrm{rel}(V, sh)\bigr) \\ &\quad\iff \forall T \subseteq S : \Bigl(\#T < k \implies S = \bigcup\bigl\{\, U \in sh \setminus \mathrm{rel}(V, sh) \bigm| T \subseteq U \subseteq S \,\bigr\}\Bigr) \\ &\quad\iff \forall T \subseteq S : \Bigl(\#T < k \implies S = \bigcup\{\, U \in sh \mid T \subseteq U \subseteq S \,\}\Bigr) \\ &\qquad\quad \wedge S \cap V = \emptyset \\ &\quad\iff S \in \rho_{TSD_k}(sh) \setminus \mathrm{rel}\bigl(V, \rho_{TSD_k}(sh)\bigr). \quad \square \end{aligned}$$

Lemma 2
For each $sh_1, sh_2 \in SH$, each $V \in \wp(VI)$ and each $k \in \mathbb{N}$ with $1 < k \leq n$,

$$\rho_{TSD_k}(sh_1) \subseteq \rho_{TSD_k}(sh_2) \implies \mathrm{rel}(V, sh_1)^\star \subseteq \mathrm{rel}(V, sh_2)^\star.$$

Proof
We prove that

$$sh_1 \subseteq \rho_{TSD_k}(sh_2) \implies \mathrm{rel}(V, sh_1) \subseteq \mathrm{rel}(V, sh_2)^\star.$$

The result then follows from Eqs. (1) and (2).

Suppose $S \in \mathrm{rel}(V, sh_1)$. Then, $S \in sh_1$ and $V \cap S \neq \emptyset$. By the hypothesis, $S \in \rho_{TSD_k}(sh_2)$. Let $x \in V \cap S$. Then, by Definition 7, we have

$$\begin{aligned} S &= \bigcup\bigl\{\, U \in sh_2 \bigm| \{x\} \subseteq U \subseteq S \,\bigr\} \\ &= \bigcup\bigl\{\, U \in \mathrm{rel}(V, sh_2) \bigm| \{x\} \subseteq U \subseteq S \,\bigr\}. \end{aligned}$$

Thus $S \in \mathrm{rel}(V, sh_2)^\star$. $\square$

Lemma 3
For each $sh_1, sh_2 \in SH$, each $\sigma \in \mathit{Subst}$ and each $k \in \mathbb{N}$ with $1 \leq k \leq n$,

$$\rho_{TSD_k}(sh_1) = \rho_{TSD_k}(sh_2) \implies \rho_{TSD_k}\Bigl(\mathrm{amgu}\bigl(sh_1, \sigma\bigr)\Bigr) = \rho_{TSD_k}\Bigl(\mathrm{amgu}\bigl(sh_2, \sigma\bigr)\Bigr).$$

Proof

If $\sigma = \emptyset$, the statement is obvious from the definition of amgu. In the other cases, the proof is by induction on the size of σ. The inductive step, when σ has more than one binding, is straightforward. For the base case, when $\sigma = \{x \mapsto t\}$, we have to show that

$$sh_1 \subseteq \rho_{\scriptscriptstyle TSD_k}(sh_2) \implies \mathrm{amgu}\bigl(sh_1, \{x \mapsto t\}\bigr) \subseteq \rho_{\scriptscriptstyle TSD_k}\Bigl(\mathrm{amgu}\bigl(sh_2, \{x \mapsto t\}\bigr)\Bigr).$$

The result then follows from Eq. (1).

Let $v_x \stackrel{\mathrm{def}}{=} \{x\}$, $v_t \stackrel{\mathrm{def}}{=} \mathit{vars}(t)$, and $v_{xt} \stackrel{\mathrm{def}}{=} v_x \cup v_t$. Suppose

$$S \in \mathrm{amgu}\bigl(sh_1, \{x \mapsto t\}\bigr).$$

Then, by definition of amgu,

$$S \in \bigl(sh_1 \setminus \mathrm{rel}(v_x \cup v_t, sh_1)\bigr) \cup \mathrm{bin}\bigl(\mathrm{rel}(v_x, sh_1)^\star, \mathrm{rel}(v_t, sh_1)^\star\bigr).$$

There are two cases:

1. $S \in sh_1 \setminus \mathrm{rel}(v_x \cup v_t, sh_1)$. Then, by hypothesis, $S \in \rho_{\scriptscriptstyle TSD_k}(sh_2)$. Hence we have $S \in \rho_{\scriptscriptstyle TSD_k}(sh_2) \setminus \mathrm{rel}\bigl(v_x \cup v_t, \rho_{\scriptscriptstyle TSD_k}(sh_2)\bigr)$. Thus, by Lemma 1,

$$S \in \rho_{\scriptscriptstyle TSD_k}\bigl(sh_2 \setminus \mathrm{rel}(v_x \cup v_t, sh_2)\bigr).$$

2. $S \in \mathrm{bin}\bigl(\mathrm{rel}(v_x, sh_1)^\star, \mathrm{rel}(v_t, sh_1)^\star\bigr)$. Then we must have $S = T \cup R$ where $T \in \mathrm{rel}(v_x, sh_1)^\star$ and $R \in \mathrm{rel}(v_t, sh_1)^\star$.

The proof here splits into two branches, 2a and 2b, depending on whether $k > 1$ or $k = 1$.

2a. We first assume that $k > 1$. Then, by Lemma 2 we have that $T \in \mathrm{rel}(v_x, sh_2)^\star$ and $R \in \mathrm{rel}(v_t, sh_2)^\star$. Hence,

$$S \in \mathrm{bin}\bigl(\mathrm{rel}(v_x, sh_2)^\star, \mathrm{rel}(v_t, sh_2)^\star\bigr).$$

Combining case 1 and case 2a we obtain

$$S \in \rho_{\scriptscriptstyle TSD_k}\bigl(sh_2 \setminus \mathrm{rel}(v_x \cup v_t, sh_2)\bigr) \cup \mathrm{bin}\bigl(\mathrm{rel}(v_x, sh_2)^\star, \mathrm{rel}(v_t, sh_2)^\star\bigr).$$

Hence as $\rho_{\scriptscriptstyle TSD_k}$ is extensive and monotonic

$$S \in \rho_{\scriptscriptstyle TSD_k}\Bigl(\bigl(sh_2 \setminus \mathrm{rel}(v_x \cup v_t, sh_2)\bigr) \cup \mathrm{bin}\bigl(\mathrm{rel}(v_x, sh_2)^\star, \mathrm{rel}(v_t, sh_2)^\star\bigr)\Bigr),$$

and hence, when $k > 1$, $S \in \rho_{\scriptscriptstyle TSD_k}\Bigl(\mathrm{amgu}\bigl(sh_2, \{x \mapsto t\}\bigr)\Bigr)$.

2b. Secondly suppose that $k = 1$. In this case, we have, by Proposition 3:

$$\rho_{\scriptscriptstyle TSD_1}(sh_2) = sh_2^\star$$

and that

$$\rho_{\scriptscriptstyle TSD_1}\Bigl(\mathrm{amgu}\bigl(sh_2, \{x \mapsto t\}\bigr)\Bigr) = \mathrm{amgu}\bigl(sh_2, \{x \mapsto t\}\bigr)^\star.$$

Thus, by the hypothesis,

$$\begin{aligned} S &\in \operatorname{bin}\bigl(\operatorname{rel}(v_x, sh_2^\star)^\star, \operatorname{rel}(v_t, sh_2^\star)^\star\bigr), \\ &= \operatorname{bin}\bigl(\operatorname{rel}(v_x, sh_2^\star), \operatorname{rel}(v_t, sh_2^\star)\bigr). \end{aligned}$$

Therefore we can write

$$S = T_{\text{-}} \cup T_x \cup R_{\text{-}} \cup R_t$$

where

$$\begin{aligned} T_{\text{-}} \cup T_x &\in \operatorname{rel}(v_x, sh_2^\star), \\ R_{\text{-}} \cup R_t &\in \operatorname{rel}(v_t, sh_2^\star), \\ T_{\text{-}}, R_{\text{-}} &\in \bigl(sh_2 \setminus \operatorname{rel}(v_{xt}, sh_2)\bigr)^\star, \\ T_x &\in \operatorname{rel}(v_x, sh_2)^\star \setminus \emptyset, \\ R_t &\in \operatorname{rel}(v_t, sh_2)^\star \setminus \emptyset. \end{aligned}$$

Thus

$$\begin{aligned} S &\in \Bigl(\bigl(sh_2 \setminus \operatorname{rel}(v_{xt}, sh_2)\bigr) \cup \operatorname{bin}\bigl(\operatorname{rel}(v_x, sh_2)^\star, \operatorname{rel}(v_t, sh_2)^\star\bigr)\Bigr)^\star \\ &= \operatorname{amgu}\bigl(sh_2, \{x \mapsto t\}\bigr)^\star. \end{aligned}$$

Combining case 1 and case 2b for $k = 1$, the result follows immediately by the monotonicity and extensivity of $(\cdot)^\star$. □

Lemma 4
For each $sh_1, sh_2 \in SH$,

$$\rho_{TSD_k}(sh_1 \cup sh_2) = \rho_{TSD_k}\bigl(\rho_{TSD_k}(sh_1) \cup \rho_{TSD_k}(sh_2)\bigr).$$

Proof
This is a classical property of upper closure operators (Gierz et al. 1980). □

Lemma 5
For each $sh_1, sh_2 \in SH$ and each $V \subseteq VI$,

$$\rho_{TSD_k}(sh_1) = \rho_{TSD_k}(sh_2) \implies \rho_{TSD_k}\bigl(\operatorname{proj}(sh_1, V)\bigr) = \rho_{TSD_k}\bigl(\operatorname{proj}(sh_2, V)\bigr).$$

Proof
We show that

$$sh_1 \subseteq \rho_{TSD_k}(sh_2) \implies \operatorname{proj}(sh_1, V) \subseteq \rho_{TSD_k}\bigl(\operatorname{proj}(sh_2, V)\bigr).$$

The result then follows from Eq. (1).

Suppose $sh_1 \subseteq \rho_{TSD_k}(sh_2)$ and $S \in \operatorname{proj}(sh_1, V)$. Then, as proj is monotonic, we have $S \in \operatorname{proj}\bigl(\rho_{TSD_k}(sh_2), V\bigr)$. We distinguish two cases.

1. There exists $x \in V$ such that $S = \{x\}$. Then $S \in \operatorname{proj}(sh_2, V)$ and hence, by Definition 7, $S \in \rho_{TSD_k}\bigl(\operatorname{proj}(sh_2, V)\bigr)$.
2. Otherwise, by definition of proj and Definition 7, there exists $S' \in \rho_{TSD_k}(sh_2)$ such that $S = S' \cap V$ and

$$\forall T \subseteq S' : \Bigl(\# T < k \implies S = \bigcup \{\, U \in sh_2 \mid T \subseteq U \subseteq S' \,\} \cap V\Bigr).$$

Hence

$$\forall T \subseteq S : \Bigl(\# T < k \implies S = \bigcup \bigl\{\, U \in \mathrm{proj}(sh_2, V) \bigm| T \subseteq U \subseteq S \,\bigr\} \Bigr),$$

and thus $S \in \rho_{\mathit{TSD}_k}\bigl(\mathrm{proj}(sh_2, V)\bigr)$.

□

Proof of Theorem 3.
Statements 1, 2 and 3 follow from Lemmas 3, 4 and 5, respectively. □

The following lemma is also proved in (Bagnara et al. 1997, Bagnara et al. 2001) but we include it here for completeness.

Lemma 6
Let $\sigma \stackrel{\mathrm{def}}{=} \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$, where, for each $i = 1, \dots, n$, t_i is a ground term. Then, for all $sh \in SH$ we have

$$\mathrm{amgu}(sh, \sigma) = sh \setminus \mathrm{rel}\bigl(\{x_1, \dots, x_n\}, sh\bigr).$$

Proof
If $n = 0$, so that $\sigma = \emptyset$, the statement can be easily verified after having observed that $\mathrm{rel}(\emptyset, sh) = \emptyset$. Otherwise, if $n > 0$, we proceed by induction on n. For the base case, let $n = 1$. Then

$$\begin{aligned} \mathrm{amgu}(sh, x_1 \mapsto t_1) &= sh \setminus \mathrm{rel}\bigl(\{x_1\}, sh\bigr) \cup \mathrm{bin}\Bigl(\mathrm{rel}\bigl(\{x_1\}, sh\bigr)^{\star}, \mathrm{rel}\bigl(\emptyset, sh\bigr)^{\star}\Bigr) \\ &= sh \setminus \mathrm{rel}\bigl(\{x_1\}, sh\bigr). \end{aligned}$$

For the inductive step, let $n > 1$ and let

$$\sigma' \stackrel{\mathrm{def}}{=} \{x_1 \mapsto t_1, \dots, x_{n-1} \mapsto t_{n-1}\}.$$

By definition of amgu we have

$$\begin{aligned} \mathrm{amgu}(sh, \sigma) &= \mathrm{amgu}\bigl(sh, \{x_n \mapsto t_n\} \cup \sigma'\bigr) \\ &= \mathrm{amgu}\Bigl(\mathrm{amgu}\bigl(sh, \{x_n \mapsto t_n\}\bigr), \sigma'\Bigr) \\ &= \mathrm{amgu}\Bigl(sh \setminus \mathrm{rel}\bigl(\{x_n\}, sh\bigr), \sigma'\Bigr) \\ &= \Bigl(sh \setminus \mathrm{rel}\bigl(\{x_n\}, sh\bigr)\Bigr) \setminus \mathrm{rel}\Bigl(\{x_1, \dots, x_{n-1}\}, sh \setminus \mathrm{rel}\bigl(\{x_n\}, sh\bigr)\Bigr) \\ &= sh \setminus \Bigl(\mathrm{rel}\bigl(\{x_n\}, sh\bigr) \cup \mathrm{rel}\Bigl(\{x_1, \dots, x_{n-1}\}, sh \setminus \mathrm{rel}\bigl(\{x_n\}, sh\bigr)\Bigr)\Bigr) \\ &= sh \setminus \mathrm{rel}\bigl(\{x_1, \dots, x_n\}, sh\bigr). \quad \square \end{aligned}$$

Proof of Theorem 4.
We assume that $S \in \rho_{\mathit{TSD}_k}(sh_1) \setminus \rho_{\mathit{TSD}_k}(sh_2)$. (If such an S does not exist we simply swap sh_1 and sh_2.)

Let C denote a ground term and let

$$\sigma \stackrel{\mathrm{def}}{=} \{\, x \mapsto C \mid x \in VI \setminus S \,\}.$$

Then, by Lemma 6, for $i = 1, 2$, we define $\mathrm{amgu}(sh_i, \sigma) \stackrel{\mathrm{def}}{=} sh_i^S$ where

$$sh_1^S \stackrel{\mathrm{def}}{=} \{\, T \subseteq S \mid T \in sh_1 \,\},$$
$$sh_2^S \stackrel{\mathrm{def}}{=} \{\, T \subset S \mid T \in sh_2 \,\}.$$

Now, if $\# S = j$ and $j \leq k$, then we have $S \in sh_1 \setminus sh_2$. Hence $S \in sh_1^S \setminus sh_2^S$ and we can easily observe that $S \in \rho_{TS_j}(sh_1^S)$ but $S \notin \rho_{TS_j}(sh_2^S)$.

On the other hand, if $\# S = j$ and $j > k$, then by Definition 7 there exists T with $\# T < k$ such that

$$S = \bigcup \{\, U \in sh_1^S \mid T \subseteq U \,\}$$

but

$$S \supset \bigcup \{\, U \in sh_2^S \mid T \subseteq U \,\} \stackrel{\mathrm{def}}{=} S'.$$

Let $x \in S \setminus S'$. We have $h \stackrel{\mathrm{def}}{=} \#\bigl(T \cup \{x\}\bigr) \leq k$ and thus we can observe that $T \cup \{x\} \in \rho_{TS_h}(sh_1^S)$ but $T \cup \{x\} \notin \rho_{TS_h}(sh_2^S)$. □

4 The Meet-Irreducible Elements

In Section 5, we will use the method of Filé and Ranzato (Filé and Ranzato 1996) 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 $TSD_n = SH$ 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

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

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

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

and we have $\# \mathrm{dAtoms}(\mathit{Def}) = n$ and $\# \mathrm{dAtoms}(PSD) = n(n+1)/2$. 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 the sharing group xyz such as $\{x\}$ and $\bot = \rho_{\mathit{Def}}(\bot) = \emptyset$ 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 set M_k of the 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 $S \in SG$ such that $\# S > k$ and each tuple $T \subset S$ with $\# T = k$. Such an element is obtained from SG by removing all the sharing groups U such that $T \subseteq U \subseteq S$. Formally, for $1 \leq k \leq n$,

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

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 $\# \mathrm{MI}(\mathit{TSD}_k) = \sum_{j=0}^{k-1} \binom{n}{j} + \binom{n}{k} 2^{n-k}$.

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

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

The proof of this theorem is included at the end of this section. Here, we illustrate the result for 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, besides SG, are the following (see Figure 2):

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

Next, consider the domain *PSD*. The only meet-irreducible elements that are not dual-atoms, beside SG, are the following (see Figure 3):

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

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

Looking at Examples 2 and 4, it can be seen that all the dual-atoms of the domains *Con* and *PS* are meet-irreducible elements of the domains *Def* and *PSD*, respectively. Indeed, the following general result shows that the dual-atoms of the domain TS_k are meet-irreducible elements for the domain TSD_k.

Corollary 2
Let $k \in \mathbb{N}$ with $1 \leq k \leq n$. Then

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

For the decomposition, we need to identify which meet-irreducible elements of TSD_k are in TS_j. Using Corollaries 1 and 2 we have the following result.

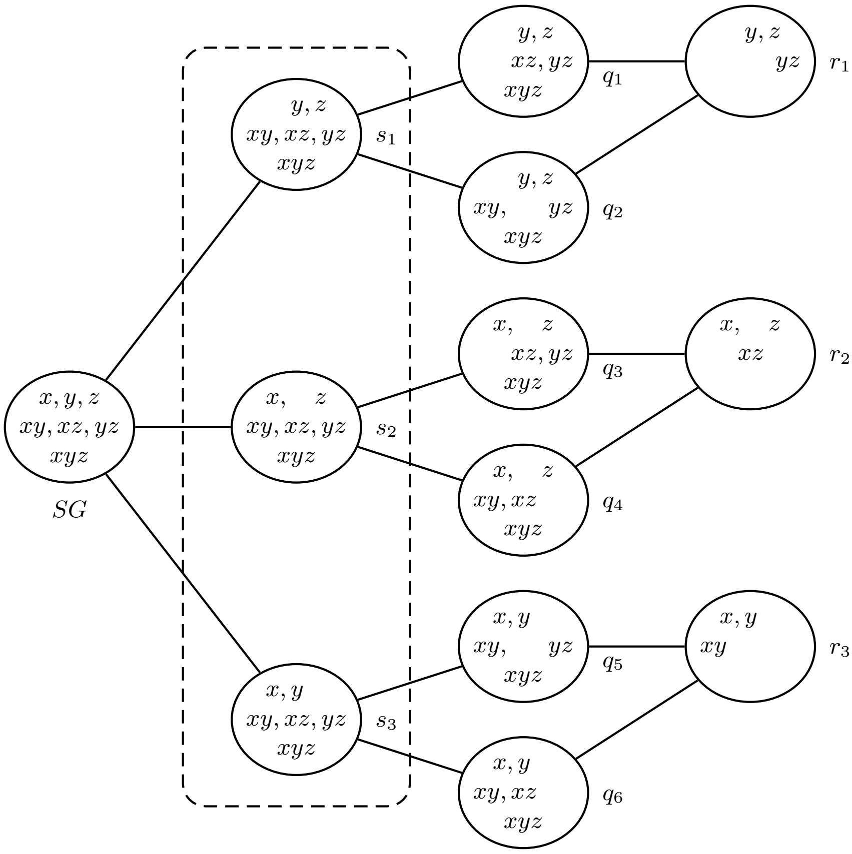

Fig. 2. The meet-irreducible elements of *Def* for $n = 3$, with dual-atoms emphasized.

Corollary 3
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 4
If $j, k \in \mathbb{N}$ with $1 \leq j < k \leq n$, then

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

4.1 Proof of Theorem 5

Proof of Theorem 5.
We prove the two inclusions separately.

1. $\mathrm{MI}(\mathit{TSD}_k) \supseteq \{\mathit{SG}\} \cup \mathrm{dAtoms}(\mathit{TSD}_k) \cup M_k$.
   Let m be in the right-hand side. If $m \in \{\mathit{SG}\} \cup \mathrm{dAtoms}(\mathit{TSD}_k)$ there is nothing to prove. Therefore we assume $m \in M_k$. We need to prove that if

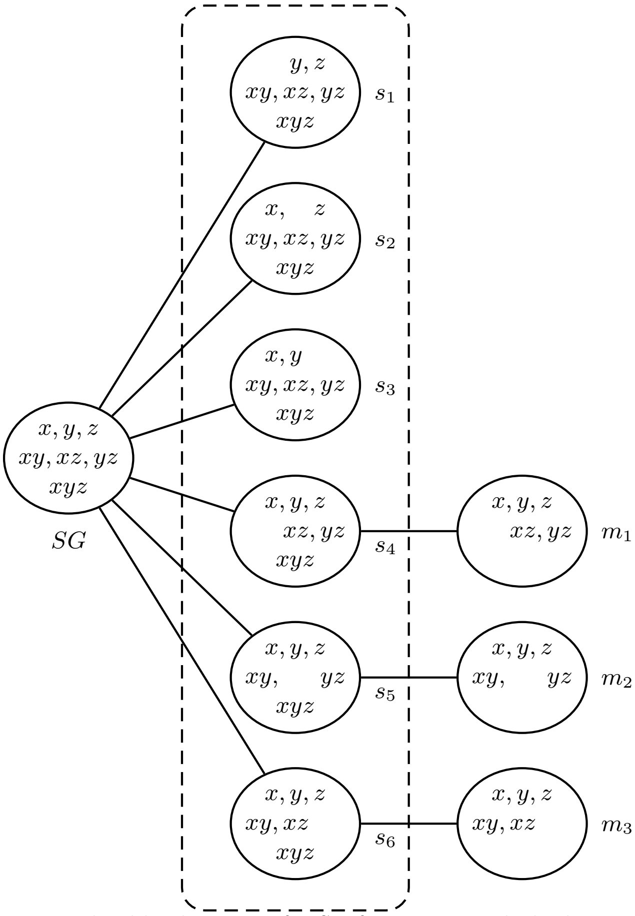

Fig. 3. The meet-irreducible elements of PSD for $n = 3$, with dual-atoms emphasized.

$sh_1, sh_2 \in TSD_k$ and

$$m = sh_1 \wedge sh_2 \stackrel{\text{def}}{=} sh_1 \cap sh_2$$

then $m = sh_1$ or $m = sh_2$. Obviously, we have $m \subseteq sh_1$ and $m \subseteq sh_2$. Moreover, by definition of M_k, there exist $T, S \in SG$ where $\# T = k$ and $T \subset S$ such that

$$m = SG \setminus \{\, U \in SG \bigm| T \subseteq U \subseteq S \,\}.$$

Since $S \notin m$, we have $S \notin sh_1$ or $S \notin sh_2$. Let us consider the first case (the other is symmetric). Then, applying the definition of TSD_k, there is a $T' \subset S$ with $\# T' < k$ such that

$$\bigcup \{\, U' \in sh_1 \mid T' \subseteq U' \subseteq S \,\} \neq S.$$

Since $\# T' < \# T$, there exists x such that $x \in T \setminus T'$. Thus $T' \subset S \setminus \{x\}$ and $S \setminus \{x\} \in m$. Hence, as $m \subseteq sh_1$, we have $S \setminus \{x\} \in sh_1$. Consider an arbitrary $U \in SG$ where $T \subseteq U \subseteq S$. Then $x \in U$. Thus, since $S = \bigl(S \setminus \{x\}\bigr) \cup U$ and $S \notin sh_1$, $U \notin sh_1$. Thus, as this is true for all such U, $sh_1 \subseteq m$.

2. $\mathrm{MI}(TSD_k) \subseteq \{SG\} \cup \mathrm{dAtoms}(TSD_k) \cup M_k$.
Let $sh \in TSD_k$. We need to show that sh is the meet of elements in the right-hand side. If $sh = SG$ then there is nothing to prove. Suppose $sh \neq SG$. For each $S \in SG$ such that $S \notin sh$, we will show there is an element m_S in the right-hand side such that $S \notin m_S$ and $sh \subseteq m_S$. Then $sh = \bigcap\{\, m_S \mid S \notin sh \,\}$. There are two cases.

2a. $\# S \leq k$; Let $m_S = SG \setminus \{S\}$. Then $m_S \in \mathrm{dAtoms}(TSD_k)$ and $sh \subseteq m_S$.

2b. $\# S > k$; in this case, applying the definition of TSD_k, there must exist a set $T' \subset S$ with $\# T' < k$ such that

$$\bigcup\{\, U' \in sh \mid T' \subset U' \subseteq S \,\} \subset S.$$

However, since $T' \subset S$, we have $S = \bigcup\bigl\{\, T' \cup \{x\} \bigm| x \in S \setminus T' \,\bigr\}$. Thus, for some $x \in S \setminus T'$, if U is such that $T' \cup \{x\} \subseteq U \subseteq S$ then $U \notin sh$. Choose $T \in SG$ so that $T' \cup \{x\} \subseteq T$ and $\# T = k$ and let $m_S = SG \setminus \{\, U \in SG \mid T \subseteq U \subseteq S \,\}$. Then $m_S \in M_k$, $S \notin m_S$, and $sh \subseteq m_S$.

□

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 1 and 3 that, for $1 \leq j < k \leq n$, we have

$$\begin{aligned} TSD_k \sim TS_j &= \mathrm{Moore}\bigl(\mathrm{MI}(TSD_k) \setminus \rho_{TS_j}(TSD_k)\bigr) \\ &= \mathrm{Moore}\bigl(\mathrm{MI}(TSD_k) \setminus TS_j\bigr) \\ &= TSD_k. \end{aligned} \tag{3}$$

Since $SH = TSD_n$, we have, using Eq. (3) and setting $k = n$, that, if $j < n$,

$$SH \sim TS_j = SH. \tag{4}$$

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 $SH \sim TS_n \neq SH$.) In particular, letting $j = 1, 2$ (assuming $n > 2$) in Eq. (4), we have

$$SH \sim PS = SH \sim Con = SH, \tag{5}$$

showing that Con and PS are too abstract to be removed from SH by means of complementation. Also, by Eq. (3), 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 2 that, for $1 \leq k \leq n$, we have

$$\begin{aligned} TSD_k \sim TS_k &= \mathrm{Moore}\bigl(\mathrm{MI}(TSD_k) \setminus \rho_{TS_k}(TSD_k)\bigr) \\ &= \mathrm{Moore}\bigl(\mathrm{MI}(TSD_k) \setminus TS_k\bigr) \\ &= \{\, sh \in TSD_k \mid VI \in sh \,\}. \end{aligned} \tag{6}$$

Thus we have

$$TSD_k \sim (TSD_k \sim TS_k) = TS_k. \tag{7}$$

We have therefore extracted *all* the domain TS_k from TSD_k. So by letting $k = 1$, 2 in Eq. (6), we have found the complements of *Con* in *Def* and *PS* in *PSD*:

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

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 4, the complement of TSD_j in TSD_k, where $1 \leq j < k \leq n$, is given as follows:

$$\begin{aligned} TSD_k \sim TSD_j &= \mathrm{Moore}\bigl(\mathrm{MI}(TSD_k) \setminus \rho_{TSD_j}(TSD_k)\bigr) \\ &= \mathrm{Moore}\bigl(\mathrm{MI}(TSD_k) \setminus TSD_j\bigr) \\ &= \bigl\{\, sh \in TSD_k \bigm| \forall S \in SG : \# S \leq j \implies S \in sh \,\bigr\}. \end{aligned} \tag{8}$$

It therefore follows from Eq. (8) and setting $k = n$ that the complement of ρ_{TSD_j} in SH for $j < n$ is:

$$\begin{aligned} SH \sim TSD_j &= \bigl\{\, sh \in SH \bigm| \forall S \in SG : \# S \leq j \implies S \in sh \,\bigr\} \\ &\stackrel{\mathrm{def}}{=} SH_j^{+}. \end{aligned} \tag{9}$$

In particular, in Eq. (9) when $j = 1$, we have the following result for *Def*, also proved in (Filé and Ranzato 1996, Lemma 5.4):

$$\begin{aligned} SH \sim \mathit{Def} &= \bigl\{\, sh \in SH \bigm| \forall x \in \mathit{VI} : \{x\} \in sh \,\bigr\} \\ &\stackrel{\mathrm{def}}{=} SH_{\mathit{Def}}^{+}. \end{aligned}$$

Also, in Eq. (9) when $j = 2$, we have the following result for *PSD*:

$$\begin{aligned} SH \sim \mathit{PSD} &= \bigl\{\, sh \in SH \bigm| \forall S \in SG : \# S \leq 2 \implies S \in sh \,\bigr\} \\ &\stackrel{\mathrm{def}}{=} SH_{\mathit{PSD}}^{+}. \end{aligned}$$

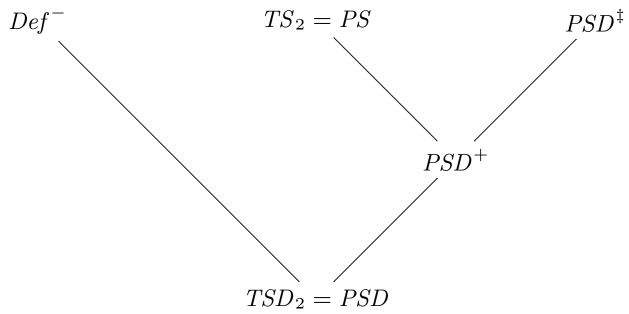

Fig. 4. A non-trivial decomposition of PSD.

We next construct the complement of PSD with respect to Def. By Eq. (8),

$$\begin{aligned} PSD \sim Def &= \{\, sh \in PSD \mid \forall x \in VI : \{x\} \in sh \,\} \\ &\stackrel{\text{def}}{=} PSD^+. \end{aligned}$$

Then the complement factor $Def^- \stackrel{\text{def}}{=} PSD \sim PSD^+$ is exactly the same thing as $SH \sim SH^+_{Def}$ so that 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 2 and Theorem 1, as PS is dual-atomistic, the complement of PS in PSD^+ is given as follows.

Theorem 7

$$\begin{aligned} PSD^\ddagger &\stackrel{\text{def}}{=} PSD^+ \sim PS \\ &= \{\, sh \in PSD \mid VI \in sh, \forall x \in VI : \{x\} \in sh \,\}, \\ PSD^+ \sim PSD^\ddagger &= PS. \end{aligned}$$

So, we have extracted *all* the domain PS from PSD^+ and we have the following result (see Figure 4).

Corollary 5
Def^-, PS, and $PSD^\ddagger$ form a minimal decomposition for PSD.

6 Discussion

By studying the sharing domain SH in a more general framework, we have been able to show that the domain PSD has a natural place in a scheme of domains based on SH. Since the well-known domain Def for groundness analysis is an instance of this scheme, we have been able to highlight the close relationship between Def and PSD and the many properties they share. In particular, it was somehow unexpected that these domains could both be obtained as instances of a single parametric construction. As another contribution, we have generalized and strengthened the results in (Cortesi et al. 1994, Cortesi et al. 1998) and (Bagnara et al. 1997, Bagnara et al. 2001) stating that

- *Def* is the quotient of *SH* with respect to the groundness domain $G \equiv \mathit{Con}$; and
- *PSD* is the quotient of *SH* with respect to the reduced product $\mathit{Con} \sqcap \mathit{PS}$ of groundness and pair-sharing.

In the view of recent results on abstract domain completeness (Giacobazzi and Ranzato 1997), these points can be restated by saying that *Def* and *PSD* are the *least fully-complete extensions* (lfce's) of *Con* and $\mathit{Con} \sqcap \mathit{PS}$ with respect to *SH*, respectively.

From a theoretical point of view, the quotient of an abstract domain with respect to a property of interest and the least fully-complete extension of this same property with respect to the given abstract domain are not equivalent. While the lfce is defined for any semantics given by means of continuous operators over complete lattices, it is known (Cortesi et al. 1994, Cortesi et al. 1998) that the quotient may not exist. However, it is also known (Giacobazzi, Ranzato and Scozzari 1998b) that when the quotient exists it is exactly the same as the lfce, so that the latter has also been called *generalized quotient*. In particular, for all the domains considered in this paper, these two approaches to the completeness problem in abstract interpretation are equivalent.

In (Bagnara et al. 1997, Bagnara et al. 2001), we wrote that $\mathit{PSD} \sim \mathit{PS} \neq \mathit{PSD}$. 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 *Def* and *PSD* are *not* dual-atomistic and we have completely specified their meet-irreducible elements. Our starting point was the work of Filé and Ranzato. In (Filé and Ranzato 1996), they noted, as we have, that $\mathit{SH}^{+}_{\mathit{Def}} \sim \mathit{PS} = \mathit{SH}^{+}_{\mathit{Def}}$ so that nothing of the domain *PS* could be extracted from $\mathit{SH}^{+}_{\mathit{Def}}$. They observed that ρ_{PS} maps all dual-atoms that contain the sharing group *VI* to the top element *SG* and thus lose all pair-sharing information. To avoid this, they replaced the classical pair-sharing domain *PS* with the domain PS' where, for all $sh \in \mathit{SH}^{+}_{\mathit{Def}}$,

$$\rho_{\mathit{PS}'}(sh) = \rho_{\mathit{PS}}(sh) \setminus \bigl(\{\mathit{VI}\} \setminus sh\bigr),$$

and noted that $\mathit{SH}^{+}_{\mathit{Def}} \sim \mathit{PS}' = \{\, sh \in \mathit{SH}^{+}_{\mathit{Def}} \mid \mathit{VI} \in sh \,\}$. To understand the nature of this new domain PS', we first observe that,

$$\mathit{PS}' = \mathit{PS} \sqcap \mathit{TS}_n.$$

This is because $\mathit{TS}_n = \mathrm{MI}(\mathit{TS}_n) = \bigl\{\mathit{SG} \setminus \{\mathit{VI}\}, \mathit{SG}\bigr\}$. In addition,

$$\mathit{SH}^{+}_{\mathit{Def}} \sim \mathit{TS}_n = \{\, sh \in \mathit{SH}^{+}_{\mathit{Def}} \mid \mathit{VI} \in sh \,\},$$

which is precisely the same as $\mathit{SH}^{+}_{\mathit{Def}} \sim \mathit{PS}'$. Thus, since $\mathit{SH}^{+}_{\mathit{Def}} \sim \mathit{PS} = \mathit{SH}^{+}_{\mathit{Def}}$, it is not surprising that it is precisely the *added* component TS_n that is removed when we compute the complement for $\mathit{SH}^{+}_{\mathit{Def}}$ with respect to PS'.

We would like to point out that, in our opinion, the problems outlined above are not the consequence of the particular domains considered. Rather, they are mainly related to the methodology for decomposing a domain. As shown here, 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 (Cortesi et al. 1994, Cortesi et al. 1998, Giacobazzi and Ranzato 1997, Giacobazzi et al. 1998b) 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 the domain with respect to the observable behavior) and only *then* decompose it by complementation. This is precisely what is done here.

We conclude our discussion about complementation with a few remarks. It is our opinion that, from a theoretical point of view, complementation is an excellent concept to work with: by allowing the splitting of complex domains into simpler components, avoiding redundancies between them, it really enhances our understanding of the domains themselves.

However, as things stand at present, complementation has never been exploited from a practical point of view. 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 together with its corresponding best approximations of the concrete semantic operators), otherwise precision would be lost and the theoretical results would not apply.

Moreover, complementation appears to have little relevance when trying to design or evaluate better implementations of a known abstract domain. In particular, this reasoning applies to the use of complementation as a tool for obtaining space saving representations for domains. As a notable example, the GER representation for *Pos* (Bagnara and Schachte 1999) is a well-known domain decomposition that does enable significant memory and time savings with no precision loss. This is not (and could not be) based on complementation. Observe that the complement of G with respect to Pos is Pos itself. This is because of the isomorphisms $Pos \equiv SH$ (Codish and Søndergaard 1998) and $G \equiv Con \stackrel{\mathrm{def}}{=} TS_1$ so that, by Eq. (5), $Pos \sim G = Pos$. It is not difficult to observe that the same phenomenon happens if one considers the groundness equivalence component E, that is, $Pos \sim E = Pos$. Intuitively, each element of the domain E defines a partition of the variable of interest VI into groundness equivalence classes. In fact, it can be shown that two variables $x, y \in VI$ are ground-equivalent in the abstract element $sh \in SH \equiv Pos$ 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).

A new domain for pair-sharing analysis has been defined in (Scozzari 2000) as

$$\mathsf{Sh}^{\mathsf{PSh}} = \mathit{PSD}^{+} \sqcap A,$$

where the A component is a strict abstraction of the well-known groundness domain *Pos*. It can be seen from the definition that $\mathsf{Sh}^{\mathsf{PSh}}$ is a close relative of *PSD*. This new domain is obtained, just as in the case for *PSD*, by a construction that starts from the set-sharing domain $\mathit{SH} \equiv \mathsf{Sh}$ and aims at deriving the pair-sharing information encoded by $\mathit{PS} \equiv \mathsf{PSh}$. However, instead of applying the generalized quotient operator used to define *PSD*, the domain $\mathsf{Sh}^{\mathsf{PSh}}$ is obtained by applying a new domain-theoretic operator that is based on the concept of *optimal semantics* (Giacobazzi, Ranzato and Scozzari 1998a).

When comparing $\mathsf{Sh}^{\mathsf{PSh}}$ and *PSD*, the key point to note is that $\mathsf{Sh}^{\mathsf{PSh}}$ is neither an abstraction nor a concretization of the starting domain *SH*. On the one hand $\mathsf{Sh}^{\mathsf{PSh}}$ is strictly more precise for computing pair-sharing, since it contains formulas of *Pos* that are not in the domain *SH*. On the other hand *SH* and *PSD* are strictly more precise for computing groundness, since $\mathsf{Sh}^{\mathsf{PSh}}$ does not contain all of *Def*: in particular, it does not contain any of the elements in *Con*.

While these differences are correctly stated in (Scozzari 2000), the informal discussion goes further. For instance, it is argued in (Scozzari 2000, Section 6.1) that

> "in [(Bagnara et al. 2001)] the domain *PSD* is compared to its proper abstractions only, which is a rather restrictive hypothesis ... "

This hypothesis is not one that was made in (Bagnara et al. 2001) but is a distinctive feature of the generalized quotient approach itself. Moreover, such an observation is not really appropriate because, when devising the *PSD* domain, the goal was to *simplify* the starting domain *SH* without losing precision on the observable *PS*. This is the objective of the generalized quotient operator and, in such a context, the "rather restrictive hypothesis" is not restrictive at all.

The choice of the generalized quotient can also provide several advantages that have been fully exploited in (Bagnara et al. 2001). Since an implementation for *SH* was available, the application of this operator resulted in an *executable* specification of the simpler domain *PSD*. By just optimizing this executable specification it was possible to arrive at a much more efficient implementation: exponential time and space savings have been achieved by removing the redundant sharing groups from the computed elements and by replacing the star-union operator with the 2-self-union operator. Moreover, the executable specification inherited all the correctness results readily available for that implementation of *SH*, so that the only new result that had to be proved was the correctness of the optimizations.

These advantages do not hold for the domain $\mathsf{Sh}^{\mathsf{PSh}}$. In fact, the definition of a feasible representation for its elements and, a fortiori, the definition of an executable specification of the corresponding abstract operators seem to be open issues.[2] Most importantly, the required correctness results cannot be inherited from those of SH. All the above reasons indicate that the generalized quotient was a sensible choice when looking for a domain simpler than SH while preserving precision on PS.

Things are different if the goal is *to improve the precision of a given analysis* with respect to the observable, as was the case in (Scozzari 2000). In this context the generalized quotient would be the wrong choice, since by definition it cannot help, whereas the operator defined in (Scozzari 2000) could be useful.

7 Conclusion

We have addressed the problem of deriving a non-trivial decomposition for abstract domains tracking groundness and sharing information for logic languages by means of complementation. To this end, we have defined a general schema of domains approximating the set-sharing domain of Jacobs and Langen and we have generalized and strengthened known completeness and minimality results. From a methodological point of view, our investigation has shown that, in order to obtain truly minimal decompositions of abstract interpretation domains, complementation should be applied to a reference domain already enjoying a minimality result with respect to the observable property.

Acknowledgment

We recognize the hard work required to review technical papers such as this one and would like to express our real gratitude to the Journal referees for their critical reading and constructive suggestions for preparing this improved version.

References

Armstrong, T., Marriott, K., Schachte, P. and Søndergaard, H. (1998). Two classes of Boolean functions for dependency analysis, *Science of Computer Programming* **31**(1): 3–45.

Bagnara, R. and Schachte, P. (1999). Factorizing equivalent variable pairs in ROBDD-based implementations of *Pos*, *in* A. M. Haeberer (ed.), *Proceedings of the "Seventh International Conference on Algebraic Methodology and Software Technology (AMAST'98)"*, Vol. 1548 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Amazonia, Brazil, pp. 471–485.

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

Bagnara, R., Hill, P. M. and Zaffanella, E. (2001). Set-sharing is redundant for pair-sharing, *Theoretical Computer Science.* To appear.

[2] In (Scozzari 2000), the only representation given for the elements of $\mathsf{Sh}^{\mathsf{PSh}}$ is constituted by infinite sets of substitutions.

Codish, M. and Søndergaard, H. (1998). The Boolean logic of set sharing analysis, *in* C. Palamidessi, H. Glaser and K. Meinke (eds), *Principles of Declarative Programming*, Vol. 1490 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Pisa, Italy, pp. 89–100.

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

Colmerauer, A. (1984). Equations and inequations on finite and infinite trees, *Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'84)*, ICOT, Tokyo, Japan, pp. 85–99.

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

Cortesi, A., Filé, G. and Winsborough, W. (1994). The quotient of an abstract interpretation for comparing static analyses, *in* M. Alpuente, R. Barbuti and I. Ramos (eds), *Proceedings of the "1994 Joint Conference on Declarative Programming (GULP-PRODE'94)"*, Peñíscola, Spain, pp. 372–397. An extended version has been published as (Cortesi et al. 1998).

Cortesi, A., Filé, G. and Winsborough, W. (1996). Optimal groundness analysis using propositional logic, *Journal of Logic Programming* **27**(2): 137–167.

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

Cortesi, A., Filé, G., Giacobazzi, R., Palamidessi, C. and Ranzato, F. (1995). Complementation in abstract interpretation, *in* A. Mycroft (ed.), *Static Analysis: Proceedings of the 2nd International Symposium*, Vol. 983 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Glasgow, UK, pp. 100–117.

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

Cousot, P. and Cousot, R. (1979). Systematic design of program analysis frameworks, *Proceedings of the Sixth Annual ACM Symposium on Principles of Programming Languages*, pp. 269–282.

Filé, G. and Ranzato, F. (1996). Complementation of abstract domains made easy, *in* M. Maher (ed.), *Logic Programming: Proceedings of the Joint International Conference and Symposium on Logic Programming*, MIT Press Series in Logic Programming, The MIT Press, Bonn, Germany, pp. 348–362.

Giacobazzi, R. and Ranzato, F. (1997). Completeness in abstract interpretation: a domain perspective, *in* M. Johnson (ed.), *Proceedings of the 6th International Conference on Algebraic Methodology and Software Technology (AMAST'97)*, Vol. 1349 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Sydney, Australia, pp. 231–245.

Giacobazzi, R., Ranzato, F. and Scozzari, F. (1998a). Building complete abstract interpretations in a linear logic-based setting, *in* Levi (1998), pp. 215–229.

Giacobazzi, R., Ranzato, F. and Scozzari, F. (1998b). Complete abstract interpretations made constructive, *in* J. Gruska and J. Zlatuska (eds), *Proceedings of 23rd International Symposium on Mathematical Foundations of Computer Science (MFCS'98)*, Vol. 1450 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 366–377.

Gierz, G., Hofmann, K. H., Keimel, K., Lawson, J. D., Mislove, M. and Scott, D. S. (1980). *A Compendium of Continuous Lattices*, Springer-Verlag, Berlin.

Hill, P. M., Bagnara, R. and Zaffanella, E. (1998). The correctness of set-sharing, *in* Levi (1998), pp. 99–114.

Jacobs, D. and Langen, A. (1989). Accurate and efficient approximation of variable aliasing in logic programs, *in* E. L. Lusk and R. A. Overbeek (eds), *Logic Programming: Proceedings of the North American Conference*, MIT Press Series in Logic Programming, The MIT Press, Cleveland, Ohio, USA, pp. 154–165.

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

Jones, N. D. and Søndergaard, H. (1987). A semantics-based framework for the abstract interpretation of Prolog, *in* S. Abramsky and C. Hankin (eds), *Abstract Interpretation of Declarative Languages*, Ellis Horwood Ltd, West Sussex, England, chapter 6, pp. 123–142.

Langen, A. (1990). *Advanced Techniques for Approximating Variable Aliasing in Logic Programs*, PhD thesis, Computer Science Department, University of Southern California. Printed as Report TR 91-05.

Levi, G. (ed.) (1998). *Static Analysis: Proceedings of the 5th International Symposium*, Vol. 1503 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Pisa, Italy.

Scozzari, F. (2000). Abstract domains for sharing analysis by optimal semantics, *in* J. Palsberg (ed.), *Static Analysis: 7th International Symposium, SAS 2000*, Vol. 1824 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Santa Barbara, CA, USA, pp. 397–412.

Scozzari, F. (2001). Logical optimality of groundness analysis, *Theoretical Computer Science*. To appear.

Zaffanella, E., Hill, P. M. and Bagnara, R. (1999). Decomposing non-redundant sharing by complementation, *in* A. Cortesi and G. Filé (eds), *Static Analysis: Proceedings of the 6th International Symposium*, Vol. 1694 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Venice, Italy, pp. 69–84.