The quotient of an abstract interpretation

Agostino Cortesi a,*, Gilberto Filé b, William Winsborough c

a Dip. di Matematica Applic. e Inform. University of Venezia, Corso di Laurea in Scienze dell’Informazione, via Torino 155, 30170 Mestre-Venezia, Italy
b Dip. di Matematica Pura ed Applicata, Università di Padova, via Belzoni 7, I-35131, Padova, Italy
c Transarc Corporation, USA

Received November 1994; revised December 1996
Communicated by G. Levi

Abstract

Within the abstract interpretation framework, abstract domains are used to represent interesting properties of the concrete domain. For instance, properties that enhance the optimization of the analyzed programs. An abstract domain D expresses, in general, several properties of the concrete domain.

We describe a method for identifying, for any abstract domain D and for each property P expressed by D, the subset of D that is useful for computing P-information. We call it the quotient of D with respect to P. We also give a necessary and sufficient condition for having that the quotient is an abstraction of D. This property seems essential for applications such as that described below.

As an illustration of the usefulness of the notion of quotient, we show that rather sophisticated comparisons between domains, can be carried out using it. Assume to have two abstract domains that both compute some property P, but that also express distinct properties and thus are incomparable as a whole. Such domains can be compared with respect to the precision with which they compute P-information, by comparing their quotients with respect to P.

Using this method, two well-known abstract domains for Prolog programs, Prop and Sharing, are compared with respect to the precision with which they compute groundness information.

© 1998 — Elsevier Science B.V. All rights reserved

1. Introduction

Abstract Interpretation is a general method for defining data-flow analyses. “Ad hoc” analyses, that can be viewed as instances of the abstract interpretation approach, were already used in the 1960s [22], for the optimization of imperative languages. However, the formal foundations of the approach have been given only in 1977 by the Cousots in [5]. In this framework, a data-flow analysis is specified by describing a domain of data-descriptions and operations on these data-descriptions that mimic the
concrete operations of the language. A tuple $D = \langle D, \mu_1, \ldots, \mu_k \rangle$, where $D$ is the set of data-descriptions (called abstract domain) and the $\mu_i$ are the operations on $D$ (called abstract operations), is called an abstract interpretation. The correctness of the data-flow analysis induced by $D$ is guaranteed when some safety condition holds between $D$ and the concrete interpretation $C = \langle C, o_1, \ldots, o_k \rangle$. In this paper we will take as safety condition that:

(i) $D$ and $C$ are complete lattices,

(ii) there is a Galois insertion between $D$ and $C$ (with $\gamma$ and $\alpha$ concretization and abstraction functions, respectively), and

(iii) for any $c \in C$ and for any $d \in D$, $c \sqsubseteq_C \gamma(d) \Rightarrow o_1(c) \sqsubseteq_C \gamma(\mu_1(d))$.

When conditions (i) and (ii) are satisfied, we will say that $D$ abstracts $C$, when also condition (iii) holds then we say that $D$ abstracts $C$. Weaker safety conditions are also sufficient for guaranteeing correctness [18, 12]. However, we need (i - iii) for the purposes of the present paper.

In an abstract interpretation that expresses and thus computes several properties, we want to identify which part is useful for computing each property. Let us be more precise. Consider an abstract interpretation $D$ that expresses a property $P$. We want to identify the subset $\mathcal{P}_D(D)$ of $D$ that expresses exactly the part of $D$ that is useful for computing $P$. We call it the quotient of $D$ with respect to $P$.

The notion of quotient is a tool for obtaining more insight in the functionalities of complex abstract interpretations. Such insight can be useful in several ways. In the present paper we show that quotients are useful for comparing abstract interpretations. Assume we want to compare the relative precision of two abstract interpretations $D$ and $L$ in the computation of a given property $P$. This comparison may be impossible relying on the classical notion of Galois insertion (or connection). In fact, $D$ and $L$ may express information that is irrelevant for computing $P$ and makes them incomparable using the classical formal tools. We show that the comparison can be done by considering the quotients $\mathcal{P}_D(D)$ and $\mathcal{P}_L(L)$. Being able to perform such sophisticated comparisons is obviously useful for choosing the best available abstract interpretation for a particular purpose. Such comparisons are also useful when combining different abstract interpretations in order to obtain a more powerful one, as is suggested in [12, 4]. By appropriate comparisons one may check whether such combinations of $D$ and $L$ are worthwhile at all and also one can “tune” the combination by lowering it only for those properties $P$ for which it may turn out to be profitable: those for which $\mathcal{P}_D(D)$ and $\mathcal{P}_L(L)$ are incomparable.

The main achievements of the paper are described below.

(a) For any abstract interpretation $D$ and any property $P$ expressed by $D$, we define the quotient $\mathcal{P}_D(D)$ of $D$ w.r.t. $P$ and give a condition that guarantees that $\mathcal{P}_D(D)$ is a complete lattice that abstracts $D$.

(b) We show that if $D$ and $L$ are optimal for the computation of a property $P$ and if $\mathcal{P}_D(D)$ abstracts $\mathcal{P}_L(L)$, then $L$ is at least as precise as $D$ for computing $P$. Moreover, if $\mathcal{P}_D(D)$ abstracts $\mathcal{P}_L(L)$ strictly (i.e., $\mathcal{P}_L(L)$ does not abstract $\mathcal{P}_D(D)$), then $L$ is strictly more precise than $D$ for computing $P$. 
(c) As an application of this theory, we compare two abstract interpretations that are well-known for the analysis of Prolog programs: \textbf{Prop}, [19, 9], and \textbf{Sharing}, [16]. We stress the fact that we consider here complete descriptions of \textbf{Prop} and \textbf{Sharing}, i.e., with all the abstract operations needed for their use in static analysis. Although both these interpretations compute the property of variable groundness, they are not directly comparable (i.e. neither one abstracts the other) because \textbf{Prop} computes also possible equivalence through disjunctions and \textbf{Sharing} computes also variable sharing. We compare the (relative) precision of these two analyses in the computation of variable groundness (GR) and we show that \( 2_{GR}(\text{Prop}) \) is \textbf{Prop} itself, whereas \( 2_{GR}(\text{Sharing}) \) is the set of formulas which are conjunctions of formulas of the form: \( \forall W \rightarrow x \), where \( W \) is a set of variables and \( x \) is a variable. Such formulas are called \textit{definite} and form a proper subset, called \textbf{Def} [13, 2], of the formulas in \textbf{Prop}. Hence, \( 2_{GR}(\text{Sharing}) \) strictly abstracts \textbf{Prop} and thus, from (b) above, it follows that \textbf{Prop} is strictly more precise than \textbf{Sharing} for the computation of variable groundness.

The notions and results of points (a) and (b) above are new. The construction of the quotients of \textbf{Prop} and \textbf{Sharing} with respect to \textbf{GR} of point(c) is also new. That \( 2_{GR}(\text{Sharing}) \) coincides with \textbf{Def} is an interesting result on its own right. In fact, both \textbf{Sharing} and \textbf{Def} are well-known domains that have been defined independently and that are both useful for the analysis of logic programs.

The present paper is an extended and improved version of [10]. The main improvement is the introduction of the notion of quotient. In fact in [10], we observed that, in order to compare \( D \) and \( L \), it was necessary to define a reference domain \( R \) and then compare the workings of \( D \) and \( L \) projected on \( R \). Clearly, \( R \) plays the same role as \( 2_{P}(D) \) and \( 2_{P}(L) \) in the present approach, but in [10] we did not have any systematic way to obtain it. Its definition was only based on the insight one had over \( D \) and \( L \).

Another improvement of the present paper concerns the criteria adopted for comparing the precision of interpretations. The criteria used now is strictly stronger (and more natural) than that of [10]. Thus, the relation between \textbf{Prop} and \textbf{Sharing} shown in the present paper is actually stronger than that contained in [10].

The idea of identifying relevant parts of an abstract domain, that has given rise to the notion of quotient, has also inspired the recent work [8] in which the notion of complement between domains is defined. The reader may be interested to know that in that paper, among several examples of application of the complement, the authors consider \textbf{Sharing} and \textbf{Def} (= \( 2_{GR}(\text{Sharing}) \), cf. point(c) above) and characterize the complement of \textbf{Def} w.r.t. \textbf{Sharing}, i.e., the domain that represents what is left of \textbf{Sharing} when \textbf{Def} is taken away from it.

Finally, it is important to remark that, even though we have applied it to the comparison of two abstract interpretations for the data-flow analysis of Prolog programs, the notion of quotient, presented in this paper, is in no way bound to logic programming applications only. On the contrary, it can be useful for studying abstract interpretations for all programming paradigms.
The paper is organized as follows. Section 2 contains the preliminary definitions. The definition of quotient is given in Section 3 together with the general result mentioned in (b) above. The application of the theory to the comparison of Prop and Sharing, mentioned in (c) above, is described in Section 4. The appendix contains some definitions and technical lemmas that are needed for the application part.

2. Preliminaries

This section consists of three parts. The first introduces some classical notions of abstract interpretation theory and some known results (see [5, 6, 1]). The second part contains the definitions of two types of optimality. The final part introduces the criterion for comparing abstract interpretations that will be used in the rest of the paper.

2.1. Galois insertions and their composition

In what follows a function’s domain and range are indicated by subscripts. \(e_{XY}\) is a function from \(X\) to \(Y\). The ordering and the least upper bound operator defined in \(X\) are denoted by \(\sqsubseteq_X\) and \(\sqcup_X\), respectively.

**Definition 2.1 (Galois connection and insertion).** Let \(C\) and \(D\) be posets and consider two functions of the following types: \(\gamma_{DC} : D \to C\) and \(\alpha_{CD} : C \to D\). The 4-tuple \(G_{CD} = (\gamma_{DC}, C, D, \alpha_{DC})\) is a Galois connection if

\[
\forall c \in C\text{ and } \forall d \in D : \alpha_{CD}(c) \sqsubseteq_D d \iff c \sqsubseteq_C \gamma_{DC}(d).
\]

\(G_{CD}\) is a Galois insertion when \(\gamma_{DC}\) is injective or, equivalently, when \(\alpha_{CD}\) is onto. When \(G_{CD}\) is a Galois insertion then we say that \(D\) abstracts \(C\).

In a Galois connection or insertion \(G_{CD}, \gamma_{DC}\) and \(\alpha_{CD}\) are called the concretization and the abstraction function, respectively. The following are well-known properties of these functions, see [6].

**Proposition 2.2.** Let \(G_{CD}\) be a Galois connection/insertion,

1. \(\gamma_{DC} \circ \alpha_{CD}\) is extensive, i.e., \(\forall c \in C : \gamma_{DC}(\alpha_{CD}(c)) \sqsubseteq_C c\);
2. \(\alpha_{CD} \circ \gamma_{DC}\) is reductive, i.e., \(\forall d \in D : \alpha_{CD}(\gamma_{DC}(d)) \sqsubseteq_D d\);
3. \(G_{CD}\) is an insertion if and only if \(\alpha_{CD} \circ \gamma_{DC}\) is the identity.
4. if \(\alpha_{CD}\) and \(\gamma_{DC}\) form a Galois connection, then one of the two functions determines the other one. More precisely, for \(d \in D\), \(\gamma_{DC}(d) = \sqcup_C \{c \in C \mid \alpha_{CD}(c) \sqsubseteq_D d\}\), and similarly, for \(c \in C\), \(\alpha_{CD}(c) = \bigcap_D \{d \in D \mid c \sqsubseteq_C \gamma_{DC}(d)\}\). Each function is called the adjoint of the other one.
5. \(\alpha_{CD} \circ \gamma_{DC} \circ \alpha_{CD} = \alpha_{CD}\) and similarly, \(\gamma_{DC} \circ \alpha_{CD} \gamma_{DC} = \gamma_{DC}\).

A function \(\alpha : C \to D\), where \(C\) and \(D\) are posets, is additive when \(\forall X \subseteq C\) such that \(\sqcup_C X\) exists, \(\alpha(\sqcup_C X) = \sqcup_D \{\alpha(x) \mid x \in X\}\).
Proposition 2.3. Let $\pi_{CD}: C \to D$, where $C$ and $D$ are complete lattices. The function $\pi_{CD}$ is additive if $\pi_{CD}$, together with its adjoint concretization, forms a Galois connection between $C$ and $D$.

Proof. The ($\Rightarrow$) direction is shown in [6, Proposition 7]. For the other direction, assume that $\pi_{CD}$ and its adjoint $\gamma_{DC}$ form a Galois connection. We must show that

$$\forall X \subseteq C, \quad \pi_{CD}(\sqcup_C X) = \sqcup_D \{ \pi_{CD}(x) \mid x \in X \}.$$ 

Clearly, $\pi_{CD}(\sqcup_C X)$ is an upper bound of $\{ \pi_{CD}(x) \mid x \in X \}$ because $\forall x \in X, \sqcup_C X \supseteq x$. We show now that it is the least upper bound. Let $d$ be an upper bound of $\{ \pi_{CD}(x) \mid x \in X \}$, i.e., $\forall x \in X, \pi_{CD}(x) \sqsubseteq_D d$. Thus, by definition of Galois connection, $\forall x \in X, x \subseteq_C \gamma_{DC}(d)$. This implies that $\gamma_{DC}(d) \supseteq_C \sqcup_C X$. It suffices now to apply $\pi_{CD}$ to both members of the inequality, to obtain: $\pi_{CD} \circ \gamma_{DC}(d) \sqsubseteq_D \pi_{CD}(\sqcup_C X)$, and thus, by Proposition 2.2(2), we have that, $d \sqsubseteq_D \pi_{CD}(\sqcup_C X)$. \qed

It is well-known, see [6], that in place of considering two domains $C$ and $D$, where $D$ abstracts $C$, one can view $D$ as a particular subset of $C$: the subset of $C$ containing the fixpoints of an upper closure operator on $C$.

Definition 2.4 (Upper closure operator). Given a poset $C$, an upper closure operator (uco) on $C$, is a function $\rho : C \to C$, that is monotonic, idempotent and extensive (i.e. $\forall c \in C, \rho(c) \supseteq_C c$). The set of fixpoints of $\rho$ is $\{ c \in C \mid \rho(c) = c \}$. This set is indicated by $\rho(C)$.

The following result shows that Galois insertions and uco’s are two equivalent ways of representing abstractions. The proof can be found in [6]. In what follows, if $G_{CD}$ is a Galois connection/insertion, then $\gamma_{DC}(D) = \{ \gamma_{DC}(d) \mid d \in D \}$.

Proposition 2.5. Let $G_{CD} = (\gamma_{DC}, C, D, \pi_{DC})$ be a Galois insertion. Then $\gamma_{DC} \circ \pi_{CD}$ is an uco on $C$ whose set of fixpoints is $\gamma_{DC}(D)$. Vice Versa, if $\rho$ is an uco on $C$, then $\rho(C)$ can be viewed as a new domain that abstracts $C$ via a Galois insertion that has $\rho$ as abstraction and the identity as concretization.

The following result, shown in [6], will be useful in the sequel.

Proposition 2.6. If $C$ is a complete lattice and $D$ is a poset and there exists a Galois insertion between them, then also $D$ is a complete lattice.

The idea of the previous result is that joins and meets on $D$ “can be computed on $C$ and then abstracted on $D$”. More precisely, if $G_{CD} = (\gamma_{DC}, C, D, \pi_{DC})$ is a Galois insertion, then $\forall X \subseteq D, \sqcup_D X = \pi_{DC}(\sqcup_C \{ \gamma_{DC}(d) \mid d \in X \})$. A similar relation holds for the meet.

In this paper we will always consider domains that are complete lattices and will assume the existence of Galois insertions between them.
Definition 2.7 (Interpretation). An interpretation is a tuple $D = (D, \mu_D)$ where $D$ is a complete lattice and $\mu_D$ is a continuous function of type $D \rightarrow D$.

In general, an interpretation contains more than one operation, and the operations may have arity greater than one and may take some other arguments besides elements of $D$. For the sake of clarity, we consider the simplest setting, as the generalization of definitions and results is immediate.

Definition 2.8 (Abstraction). An interpretation $D = (D, \mu_D)$ abstracts an interpretation $C = (C, \mu_C)$ if

1. there is a Galois insertion $(\gamma_{DC}, C, D, \alpha_{CD})$ and
2. $\forall c \in C : \forall d \in D : c \subseteq C \gamma_{DC}(d) = \mu_C(c) \subseteq C \gamma_{DC}(\mu_D(d))$.

We say that $D$ properly abstracts $C$ if $D$ abstracts $C$ but $C$ does not abstract $D$.

An abstract interpretation is intended to report information about a program’s execution behavior. When $L$ abstracts $D$ we know that the analysis induced by $D$ is at least as informative as the analysis induced by $L$. In the rest of the paper we denote domains by capital letters $C, D, L, R$ possibly subscripted, and we denote interpretations by boldface capital letters $\mathbf{C}, \mathbf{D}, \mathbf{L}, \mathbf{R}$.

It is well known that if $R$ abstracts $D$ and $D$ abstracts $C$, then $R$ abstracts $C$. The proofs of the propositions listed below are straightforward.

Proposition 2.9 (Same-order composition). Let $G_{DR}$ and $G_{CD}$ be Galois insertions. Their composition, denoted $G_{CD} \circ G_{DR}$ is the Galois insertion $G_{CR} = (\gamma_{RC}, C, R, \alpha_{CR})$, where $\gamma_{RC} = \gamma_{DC} \circ \gamma_{RD}$ and $\alpha_{CR} = \alpha_{DR} \circ \alpha_{CD}$.

The following two propositions are shown in [10].

Proposition 2.10 (Opposite-order composition). Assume that $G_{CL}$ and $G_{CD}$ are Galois insertions. Let $\epsilon_{DL} = \alpha_{CL} \circ \gamma_{DC}$ and $\epsilon_{LD} = \alpha_{CD} \circ \gamma_{LC}$. The following holds:

(i) $\epsilon_{DL}$ and $\epsilon_{LD}$ are monotone;
(ii) $\forall d \in D, \forall \ell \in L$ $\epsilon_{LD}(\epsilon_{DL}(d)) \supseteq D d$ and $\epsilon_{DL}(\epsilon_{LD}(\ell)) \supseteq L \ell$.

Proposition 2.11. Assume that $G_{CL}$ and $G_{CD}$ are Galois insertions. The following conditions are equivalent:

(i) $G_{DL} = (\alpha_{CD} \circ \gamma_{LC}, D, L, \alpha_{CL} \circ \gamma_{DC})$ is a Galois insertion;
(ii) $\gamma_{LC}(L) \subseteq \gamma_{DC}(D)$;
(iii) $\forall c_1, c_2 \in C. \alpha_{CD}(c_1) = \alpha_{CD}(c_2) \Rightarrow \alpha_{CL}(c_1) = \alpha_{CL}(c_2)$.

Definition 2.12. If $G_{CL}$, $G_{CD}$ and $G_{DL} = (\alpha_{CD} \circ \gamma_{LC}, D, L, \alpha_{CL} \circ \gamma_{DC})$ are Galois insertions, we say that $G_{DL}$ is coherent with respect to $C$.

Coherency and same-order composition are strongly related.
Proposition 2.13. Let \( G_{CD} \) \( G_{CL} \) and \( G_{DL} \) be Galois insertions. Then \( G_{DL} \) is coherent with respect to \( C \) if and only if \( G_{CL} = G_{CD} \circ G_{DL} \).

Proof. \((\Rightarrow)\) We want to show that \( \gamma_{LC} = \gamma_{DC} \circ \gamma_{LD} \), which, since \( G_{DL} \) is coherent w.r.t. \( C \) and thus \( \gamma_{LD} = \alpha_{CD} \circ \gamma_{LC} \), can be rewritten in

\[
\gamma_{LC} = \gamma_{DC} \circ \alpha_{CD} \circ \gamma_{LC}
\]

This equality is verified because, by Proposition 2.11(ii), \( \forall l \in L, \gamma_{LC}(l) \in \gamma_{DC}(D) \) and thus, by Proposition 2.2(5), \( \forall l \in L, \gamma_{DC} \circ \alpha_{CD}(\gamma_{LC}(l)) = \gamma_{LC}(l) \).

That \( \alpha_{CL} = \alpha_{DL} \circ \alpha_{CD} \) is shown as follows. By coherency, \( \alpha_{DL} \circ \alpha_{CD} = \alpha_{CL} \circ \gamma_{DC} \circ \alpha_{CD} \). Since \( \forall c \in C, \alpha_{CD}(c) = \alpha_{CD}(\gamma_{DC} \circ \alpha_{CD}(c)) \), cf. Proposition 2.2(5), using Proposition 2.11(iii), one obtains that \( \forall c \in C, \alpha_{CL}(c) = \alpha_{CL}(\gamma_{DC} \circ \alpha_{CD}(c)) \).

\((\Leftarrow)\) By hypothesis, \( \gamma_{DC} \circ \gamma_{LD} = \gamma_{LC} \). It suffices to apply \( \alpha_{CD} \) to both members of this equation and use Proposition 2.2(3), to obtain the desired relation

\[
\gamma_{LD} = \alpha_{CD} \circ \gamma_{LC}.
\]

That also \( \alpha_{DL} = \alpha_{CD} \circ \gamma_{DC} \) can be shown similarly. \( \square \)

2.2. Optimalities

As usual, an abstract interpretation is optimal if it mimics the concrete one in the best possible way. We also introduce a weaker notion in which we project the result of the operation on a more abstract domain.

Definition 2.14 (Optimalities). Consider the interpretations \( D = \langle D, \mu_D \rangle \) and \( C = \langle C, \mu_C \rangle \). Assume that \( D \) abstracts \( C \). \( D \) is optimal if \( \forall d \in D : \mu_D(d) = \alpha_{CD}(\mu_C(\gamma_{DC}(d))) \).

Let now \( R \) be a domain which abstracts \( D \), and let \( \alpha_{CR} = \alpha_{DR} \circ \alpha_{CD} \). We say that \( D \) is \( R \)-optimal if \( \forall d \in D : \mu_R(d) = \alpha_{CR}(\mu_C(\gamma_{DC}(d))) \).

Lemma 2.15. Let \( C, D, \) and \( R \) as in the previous definition. If \( D \) is optimal then \( D \) is also \( R \)-optimal.

Proof. Let \( d \in D: \)

\[
\mu_D(d) = \alpha_{CD}(\mu_C(\gamma_{DC}(d))) \quad \text{as} \ D \text{ is optimal}
\]

\[
\Rightarrow \alpha_{DR}(\mu_D(d)) = \alpha_{DR}(\alpha_{CD}(\mu_C(\gamma_{DC}(d)))) \quad \text{applying} \ \alpha_{DR} \text{ to both sides}
\]

\[
\Rightarrow \alpha_{RD}(\mu_D(d)) = \alpha_{CR}(\mu_C(\gamma_{DC}(d))) \quad \text{by definition of} \ \alpha_{CR}. \quad \square
\]

Observe that the notion of \( R \)-optimality is a generalization of the notion of optimality. In fact, \( D \) is optimal iff \( D \) is \( D \)-optimal.

2.3. General comparison criterion

Let \( P \) be an abstract domain that represents a property we are interested in. Assume that the two interpretations \( L \) and \( D \) also represent this property. This fact is modeled
by the assumption that $P$ abstracts both $L$ and $D$. We want to compare the precision of $L$ and $D$ with respect to the way they compute $P$, according to the following intuitive idea. $L$ is at least as precise as $D$ with respect to $P$ if every sequence of concrete operations is better or equally approximated by $L$ than by $D$ when considering only the information representable in $P$.

**Definition 2.16 (Comparison criterion).** Let $D = \langle D, \mu_D \rangle$ and $L = \langle L, \mu_L \rangle$ be interpretations abstracting $C = \langle C, \mu_C \rangle$. Let $P$ be a domain abstracting both $D$ and $L$. Let also $\mu^i_C$ denote the $i$th fold composition of $\mu_C$ and $\mu^i_D$ and $\mu^i_L$ the corresponding sequences of operators of $D$ and $L$.

- $L$ is at least as precise as $D$ with respect to $P$ if
  \[ \forall c \in C, \forall i \geq 0 : z_{DP}(\mu^i_L(z_{CL}(c))) \subseteq_P z_{DP}(\mu^i_D(z_{CL}(c))). \]

- $L$ is strictly more precise than $D$ with respect to $P$ if $L$ is at least as precise as $D$ but the converse does not hold.

### 3. The quotient of an interpretation

This section consists of two parts. In the first one, two domains $D$ and $P$ are considered, where $P$ represents a particular property expressed by $D$. An equivalence relation $r_P$ on $D$ is defined that identifies the classes of elements of $D$ that are equivalent w.r.t. the computation of $P$-information. It is shown that, when $r_P$ is additive, it is possible to define an abstraction of $D$, called the *quotient of $D$ w.r.t. $P$*, that represents exactly the information of $D$ that is used to compute $P$-information. In the second part of the section we show the relevance of quotients for comparing the precision with which different interpretations compute a given property.

Throughout this section, we always assume that $G_{CP}, G_{CD}, G_{DP}$ are Galois insertions with $G_{DP}$ coherent with respect to $C$ (i.e., $G_{CP} = G_{CD} \circ G_{DP}$), and that the interpretation $D = \langle D, \mu_D \rangle$ abstracts the interpretation $C = \langle C, \mu_C \rangle$ with $\mu_D$ optimal.

#### 3.1. Definition and properties of the quotient

First, we characterize elements of $D$ that are equivalent, with respect to $P$, in any computation sequence.

**Definition 3.1 (Associated relation).** The equivalence relation $r_P$ on $D$ associated to $P$ is defined by
\[(d_1, d_2) \in r_P \iff \forall i \geq 0 : z_{DP}(\mu^i_D(d_1)) = z_{DP}(\mu^i_D(d_2)).\]

Observe in particular that $(d_1, d_2) \in r_P$ implies $z_{DP}(d_1) = z_{DP}(d_2)$. In the sequel, $[d]_P$ denotes the set \[ \{ d' \in D \mid (d, d') \in r_P \} \].
Clearly, the intuition suggests that the quotient of $D$ w.r.t. $P$ should be a set $Q$ that has one element corresponding to each equivalence class of $D$ w.r.t. $r_P$. Unfortunately, this is not always the case. In fact, it is easy to find equivalence relations on $D$ for which such a $Q$ is not an abstraction of $D$ (whereas we want the quotient of $D$ to abstract it). Below we will show that if $r_P$ is additive, then such a $Q$ is an abstraction of $D$ and it will, in fact, be the quotient we are looking for. After having proven this fact, we will show in Theorem 3.7 that the additivity of $r_P$ is equivalent to the additivity of the abstraction function that connects $D$ to $Q$. This relationship should not be surprising in view of Proposition 2.3 that, in the present context, shows that the additivity of the abstraction function implies the existence of a Galois insertion between $D$ and $Q$.

**Definition 3.2 (Additivity).** The relation $r_P$ is additive when $\forall S \subseteq r_P$, if $S_1 = \{ a \mid (a,b) \in S \}$ and $S_2 = \{ b \mid (a,b) \in S \}$, it is true that $(\cup_P S_1, \cup_P S_2) \in r_P$.

In what follows some important consequences of the additivity of $r_P$ are shown.

**Lemma 3.3.** If the relation $r_P$ on $D$ associated to $P$ is additive, then

$$\forall d \in D : (\cup_P [d], d) \in r_P, \text{ i.e. } \cup_P [d] \subseteq [d].$$

**Proof.** It is sufficient to observe that, by additivity of $r_P$, if $[d] = \{ d_i : i \in I \}$ then $(\cup_P \{ d_i : i \in I \}, \cup_P \{ d \}) = (\cup_P \{ d_i : i \in I \}, d) \in r_P$. $\Box$

**Lemma 3.4.** If the relation $r_P$ on $D$ associated to $P$ is additive, then

$$\forall d_1, d_2 \in D : d_1 \subseteq_D d_2 \Rightarrow \cup_P [d_1] \subseteq_D \cup_P [d_2].$$

**Proof.** Let $\hat{d}_1 = \cup_P [d_1]$ and $\hat{d}_2 = \cup_P [d_2]$. By Lemma 3.3, $(d_1, \hat{d}_1) \in r_P$ and $(d_2, \hat{d}_2) \in r_P$. Thus, by additivity of $r_P$, $(d_1 \cup_P d_2, \hat{d}_1 \cup_P \hat{d}_2) \in r_P$. By hypothesis, $d_1 \subseteq_D d_2$, so we get $(d_2, \hat{d}_1 \cup_P \hat{d}_2) \in r_P$, i.e. $\hat{d}_1 \subseteq_P \hat{d}_2 \in [d_2]$. Therefore, by the definition of $\hat{d}_2$, $\hat{d}_1 \cup_P \hat{d}_2 \subseteq \hat{d}_2$, and thus $d_1 \subseteq_D d_2$. $\Box$

Let us give now the definition of quotient. After that we will show that the additivity of $r_P$ implies that the quotient enjoys all the properties we wanted and in particular that it abstracts $D$.

**Definition 3.5 (Quotient).** The quotient of $D$ with respect to $P$ is the set $\mathcal{Q}_P(D)$ defined by

$$\mathcal{Q}_P(D) = \{ \cup_P [d] \mid d \in D \}.$$

$\mathcal{Q}_P(D)$ is a subset of $D$ and thus it is partially ordered.
Theorem 3.6. If the associated relation $r_p$ is additive, the following facts hold.

(i) $\mathcal{P}(D)$ is a complete lattice that abstracts $D$;
(ii) $P$ abstracts $\mathcal{P}(D)$ coherently w.r.t. $C$;

Proof. (i) We will show that $\mathcal{P}(D)$ is the set of fixpoints of the following $uco$ $\rho_D$ on $D$:

$$\forall d \in D. \quad \rho_D(d) : = \sqcup_D [d]_p.$$ 

Thus we must show that $\rho_D$ is extensive, idempotent and monotone:
- it is extensive: by definition;
- it is idempotent: using Lemma 3.3 it is simple to see that $\forall d \in D. \sqcup_D [\sqcup_D [d]_p] = \sqcup_D [d]_p$;
- it is monotone: immediate from Lemma 3.4.

Obviously $\mathcal{P}(D) = \rho_D(D)$ and thus, from Proposition 2.5, it follows that there is a Galois insertion between $D$ and $\mathcal{P}(D)$ with abstraction $\rho_D$ and the identity as concretization. Moreover, from Proposition 2.6, we have that $\mathcal{P}(D)$ is a complete lattice.

(ii) In order to show that $P$ abstracts $\mathcal{P}(D)$, by Proposition 2.11(ii), it suffices to show that $\gamma_pD(P) \subseteq \rho_D(D)$. Precisely, we want to show that

$$\forall b \in P. \quad \gamma_pD(b) = \sqcup_D [\gamma_pD(b)]_p$$

Let $a = \sqcup_D [\gamma_pD(b)]_p$. Obviously, the following point (1) holds: $a \sqsubseteq_D \gamma_pD(b)$. Moreover, by Lemma 3.3, $\gamma_pD(a) = \gamma_pD(b) = b$ ($\iota \circ \gamma$ is the identity) from which, applying $\gamma_pD$ on both sides, one obtains:

$$a \sqsubseteq_D \gamma_pD \circ \gamma_pD(a) = \gamma_pD(b) \quad (\gamma \circ \iota$ is extensive).

This together with (1) shows what we wanted. Thus, by Proposition 2.11(i), there is a Galois insertion $G_{QP} = (\gamma_pD \circ \gamma_pD, \mathcal{P}(D), P, \gamma_pD \circ \gamma_pD)$ that is coherent w.r.t. $D$. Thus, by Proposition 2.13, $G_{DP} = G_{DQ} \circ G_{QP}$. Since, by hypothesis, $G_{DP}$ is coherent w.r.t. $C$, it is the case that, $G_{DP} = G_{DC} \circ G_{DP}$, and thus, $G_{CP} = G_{CD} \circ G_{DQ} \circ G_{QP} = G_{CQ} \circ G_{QP}$ which, by Proposition 2.13, proves that $G_{QP}$ is coherent w.r.t. $C$. □

It is easy to see that the join on $\mathcal{P}(D)$, denoted $\sqcup_Q$, is as follows: $u_1 \sqcup u_2 = \sqcup_D [u_1 \sqcup_D u_2]_p$. The meet is defined similarly. As already announced, the additivity of $r_Q$ is equivalent to that of the abstraction function $\rho_D$ (defined in the proof of Theorem 3.6). Recall that $\rho_D : D \rightarrow \mathcal{P}(D)$ is additive if $\forall X \subseteq D. \quad \rho_D(\sqcup_D X) = \sqcup_Q \{ \rho_D(x) \mid x \in X \}$.

Theorem 3.7. $r_p$ is additive iff $\rho_D$ is additive.

Proof. By the additivity of $r_p$, it is true that

$$\forall X \subseteq D. \quad \mathcal{P}(D) \subseteq \sqcup_D [\{ \sqcup_D [a]_p \mid a \in X \}]_p.$$ 

Using the definition of $r_p$, this is equivalent to

$$\sqcup_D [\sqcup_D X]_p = \sqcup_D [\sqcup_D \{ \sqcup_D [a]_p \mid a \in X \}]_p.$$
It suffices now to observe that
\[ \sqcup_D [\sqcup_D X]_p = \rho_\varepsilon (\sqcup_D X) \]
by definition of \( \rho_\varepsilon \),
and that
\[ \sqcup_D [\sqcup_D \{ a \in X \}]_p = \sqcup_\varnothing \{ \rho_\varepsilon (a) \mid a \in X \} \]
by definition of \( \rho_\varepsilon \) and of \( \sqcup_\varnothing \)
given before this theorem. \( \Box \)

3.2. Comparison of quotients

The results of this subsection show the important role that the notion of quotient plays in the comparison of two abstract interpretations. For the sake of clarity, in ASS below we summarize the notation and the hypotheses that we will use in the following theorems.

ASS (a) \( D \) and \( L \) are abstract interpretations, \( C \) is the concrete interpretation and \( P \) is the abstract domain that represents the property that is being studied.
(b) \( G_{CD}, G_{CL}, G_{CP}, G_{DP} \) and \( G_{LP} \) are Galois insertions. The last two are coherent with respect to \( C \).
(c) \( R_1 = \varepsilon (D) \) and \( R_2 = \varepsilon (L) \). By Theorem 3.6, there are Galois insertions \( G_{R_1 P} \)
and \( G_{R_2 P} \) that are coherent with respect to \( C \).
(d) \( D \) and \( L \) are, respectively, \( R_1 \)- and \( R_2 \)-optimal.
(e) \( R_1 \) abstracts \( R_2 \) coherently with respect to \( C \).

The following fact is a simple consequence of the assumptions ASS above.

**Lemma 3.8.** \( G_{R_2 P} = G_{R_1 R_1} \circ G_{R_2 P} \) and thus, in particular, \( \varkappa_{R_2 P} = \varkappa_{R_2 R_1} \circ \varkappa_{R_1 P} \).

Fig. 1 summarizes the relations existing among all domains considered. The arrows correspond to \( \gamma \)-functions.

![Diagram](image)

Fig. 1. Domain abstractions in Theorems 3.10 and 3.11.
Theorem 3.9. Assume ASS above. Then $L$ is at least as precise as $D$ for computing $P$.

Proof. Consider any sequence $\pi_D$ of operations of $D$. From the fact that $D$ is $R_1$-optimal and from the construction of the quotient $R_1$, it is true that for any $d \in D$, the computation $\pi_D(d)$ can be "read" over $R_1$, as far as the $P$-information is concerned. More precisely, assume that $d_i$ is the result after the first $i \geq 0$ steps of the computation $\pi_D(d)$, and let $t_i = \pi_{D,R_1}(d_i)$, then, for each $i$, $\alpha_{DP}(d_i) = \alpha_{R_1,P}(t_i)$. A similar fact holds for any computation $\pi_L$ on $L$. For such a $\pi_L$, let $l_0, l_1, \ldots, l_i$ be the intermediate results and $k_0, \ldots, k_i$ be the corresponding values in $R_2$.

We will use the above fact and the notation introduced in the sequel of the proof. Let for any concrete value $c \in C$, $d_0 = \alpha_{CD}(c)$ and $l_0 = \alpha_{CL}(c)$. Let also $\pi_D$ and $\pi_L$ be sequences of corresponding operations of $D$ and $L$. In what follows, the $c'_i, t_i, l_i$, and $k_i$ are in the relation explained above. In order to prove the result, it suffices to show the following fact:

\[(\star) \quad \forall i \geq 0, \quad \gamma_{R,C}(t_i) \sqsupseteq_C \gamma_{R,C}(k_i).\]

In fact, from (\star) it follows that $\alpha_{R_1,R_1}(k_i) \sqsubseteq_R t_i$. By Lemma 3.8 above and the monotonicity of $\alpha_{R_1,P} \circ \alpha_{R_1,R_1}(k_i) = \alpha_{P,R_1,R_1}(k_i) \sqsubseteq_R \alpha_{R_1,P}(t_i)$. We proceed by induction on $i$.

Basis. Let us first consider $i = 0$. We want to show that

\[(1) \quad \gamma_{R,C}(t_0) \sqsupseteq_C \gamma_{R,C}(k_0).\]

By Proposition 2.9, $t_0 = \alpha_{CR_1}(c)$ and $k_0 = \alpha_{CR_1}(c)$. Since $R_1$ abstracts $R_2$ coherently w.r.t. $C$, the following two points hold:

(a) $\alpha_{CR_1} = \alpha_{CR_1} \circ \alpha_{CR_2}$. Hence, $t_0 = \alpha_{R_1,R_1}(k_0)$.

(b) $\gamma_{R,C} = \gamma_{R,C} \circ \gamma_{R_1,R_2}$.

From point (a) and (b), using the extensivity of $\gamma_{R_1,R_1} \circ \alpha_{R_2,R_1}$ (cf. Section 2.1), we get immediately that

$\gamma_{R,C}(t_0) = \gamma_{R,C} \circ \gamma_{R_1,R_1} \circ \alpha_{R_2,R_1}(k_0) \supseteq_C \gamma_{R,C}(k_0)$.

Thus (1) holds.

Step. Let us now prove (\star) for $i > 0$. We want to show that

\[(2) \quad \gamma_{R,C}(t_i-1) \sqsupseteq_C \gamma_{R,C}(k_{i-1}) \quad \Rightarrow \quad (3) \quad \gamma_{R,C}(t_i) \sqsupseteq_C \gamma_{R,C}(k_i).\]

By the assumption of $R_1$- and $R_2$-optimality of $D$ and $L$, we know that

$t_i = \alpha_{CR_1}(\mu_C(\gamma_{R,C}(t_{i-1})))$

where $\mu_C$ is the concrete operation corresponding to the $i$th operation of $\pi_D$ and $\pi_L$. A similar relation holds for $k_i$.

By assumption (2) and the monotonicity of $\mu_C$ it follows that

$\mu_C(\gamma_{R,C}(t_{i-1})) \sqsubseteq_C \mu_C(\gamma_{R,C}(k_{i-1}))$

and thus, by the same reasoning used for the case $i = 0$, we have that (3) holds. \(\square\)
**Theorem 3.10.** Assume, in addition to ASS, that \( R_1 \) properly abstracts \( R_2 \). Then \( L \) is strictly more precise than \( D \) for computing \( P \).

**Proof.** By assumption it is true that

\[
(1) \quad \gamma_{R_1 C}(R_1) \subseteq \gamma_{R_1 C}(R_2) \quad \text{and} \quad \gamma_{R_1 C}(R_2) \not\supseteq \gamma_{R_1 C}(R_1).
\]

Hence, there is \( r_0 \in R_2 \) such that \( \gamma_{R_1 C}(r_0) \not\subset \gamma_{R_1 C}(R_1) \). Let \( c_0 = \gamma_{R_1 C}(r_0) \), and \( l_0 = \gamma_{R_1 L}(r_0) \).

Note that, since \( \gamma_{R_1 C} = \gamma_{I C} \circ \gamma_{R_1 L} \), \( \gamma_{I C}(l_0) = c_0 \).

Let now \( \hat{c} = \gamma_{R_1 C} \circ \alpha_{C R_1}(c_0) \). Clearly, by their definitions, \( \hat{c} \supseteq C c_0 \) and \( \alpha_{C R_1}(c_0) = \alpha_{C R_1}(\hat{c}) \), from which the following fact (2) holds:

**Fact (2).** \( \alpha_{C D}(c_0) \) and \( \alpha_{C D}(\hat{c}) \) are elements of \( D \) that are equivalent w.r.t. the computation of \( P \), i.e., they are in the same equivalence class of the relation on \( D \) associated to \( P \).

Note also that, by (1), \( \hat{c} \in \gamma_{R_1 C}(R_2) \). Let us now abstract \( \hat{c} \) into \( L \) and \( R_2 \). We call \( l = \alpha_{C L}(\hat{c}) \) and \( r_2 = \alpha_{L R_1}(l) \).

By assumption, we know that for all corresponding sequences \( s_D \) and \( s_L \) of operations of \( D \) and \( L \), respectively, it is true that

\[
\alpha_{DP}(s_D(\alpha_{C D}(\hat{c}))) \supseteq \alpha_{LP}(s_L(\alpha_{C L}(\hat{c}))).
\]

From this, using Fact (2), we obtain

\[
(3) \quad \alpha_{DP}(s_D(\alpha_{C D}(c_0))) = \alpha_{DP}(s_D(\alpha_{C D}(\hat{c}))) \supseteq \alpha_{LP}(s_L(\alpha_{C L}(\hat{c}))).
\]

Observe now that, since \( \hat{c} \supseteq C c_0 \), and both are in \( \gamma_{R_1 C}(R_2) \), it is true that, \( r_2 \supseteq L r_0 \).

Thus, there exists a computation sequence \( \pi_L \) of operations of \( L \) such that

\[
\alpha_{DP}(\pi_L(l)) \neq \alpha_{DP}(\pi_L(\hat{c})).
\]

Since \( \pi_L \) is composed of continuous (and thus monotone) operations, we have

\[
(4) \quad \alpha_{DP}(\pi_L(l)) \supseteq \alpha_{DP}(\pi_L(\hat{c})).
\]

It suffices now to put together (3) and (4) to show the thesis:

\[
\alpha_{DP}(\pi_D(\alpha_{C D}(c_0))) \supseteq \alpha_{DP}(\pi_L(\alpha_{C L}(\hat{c}))) = \alpha_{DP}(\pi_L(l)) \supseteq \alpha_{DP}(\pi_L(\hat{c}))
\]

\[
= \alpha_{DP}(\pi_L(\alpha_{C L}(c_0))). \quad \square
\]

**4. Applications**

We apply the theory developed in the previous section for comparing two well-known abstract interpretations for logic programming: \textbf{Prop} \cite{2, 9, 11, 19} and \textbf{Sharing} \cite{16}. This section is organized as follows. After some preliminary definitions concerning substitutions, in Section 4.2, we recall from \cite{11} the concrete interpretation \textbf{Rsub} and the two abstract interpretations we wish to compare. The domain
GR representing groundness and the characterizations of the quotients $\mathcal{Z}_{GR}(\text{Sharing})$ and $\mathcal{Z}_{GR}(\text{Prop})$ are described in Section 4.3. The main result of the application part is in Section 4, where we apply Theorems 3.9 and 3.10 for proving that Prop is strictly more precise than Sharing with respect to the precision in computing groundness.

We point out that the interpretations Prop and Sharing that we compare are complete interpretations, in the sense that they include all the operations needed for the static analysis, viz. forward/backward unification, least upper bound, and projection. In order to describe in a simple way all the operations (and, in particular, projection), we adopt the approach introduced in [11]. In this approach, the (non trivial) values of a domain are pairs in which the first component is “the usual value”, and the second component explicitly specifies the variables about which the first component provides information. These variables are often called the variables of interest.

4.1. Preliminaries

Let $V$ be a countable set of variables. $FP(V)$ denotes the set of finite subsets of variables of $V$. A substitution $\sigma$ is a function in that maps variables in $\mathcal{V}$ to terms over $\mathcal{V}$ and an alphabet of function symbols, and such that $\sigma x \neq x$ only for a finite number of variables $x$. The set of support of $\sigma$ is given by $\text{supp}(\sigma) = \{x \mid \sigma x \neq x\}$. The variable range of $\sigma$ is given by $\text{var-range}(\sigma) = \bigcup \{\text{Var}(\sigma x) \mid x \in \text{supp}(\tau)\}$, where $\text{Var}(t)$ denotes the set of variables occurring in $t$. The set of variables occurring in $\sigma$ is given by $\text{Var}(\sigma) = \text{supp}(\sigma) \cup \text{var-range}(\sigma)$. A substitution is typically specified by listing its non-trivial bindings. So $\sigma = \{x/\sigma x \mid x \in \text{supp}(\sigma)\}$.

Consider two substitutions $\sigma_1$ and $\sigma_2$. If there exists $\vartheta$ such that $\sigma_2 = \vartheta \circ \sigma_1$, then $\sigma_1$ is more general than $\sigma_2$, which we write $\sigma_2 \leq \sigma_1$. In this case, we say that $\sigma_2$ an instance of $\sigma_1$.

We write $\text{Subst}$ for the set of idempotent substitutions. Although $\text{Subst}$ is not closed under composition, in a step of the execution of a logic program in which $\vartheta \circ \sigma$ is constructed, it is always the case that, $\text{var-range}(\vartheta) \cap \text{supp}(\sigma) = \emptyset$, which, provided that $\vartheta$ and $\sigma$ are idempotent, ensures that $\vartheta \circ \sigma$ is also idempotent.

As we will consider sequences of concrete/abstract operations of “real” domains, i.e. containing not only unary operations, as it was assumed in Definition 2.7 for the sake of simplicity, it is necessary to make precise this notion for any set of operations. Assume to have an interpretation $D = \langle D, \mu_1, \ldots, \mu_k \rangle$. A derived operator over $D$ is a term $t$ constructed using the symbols in $\mu_1, \ldots, \mu_k$, the values in $D$, values $a$’ any other domain that may be required by the operations (for instance, substitutions are needed in the unification operations), and exactly one variable. The following example illustrates this notion for $Z = \langle Z, +, * \rangle$, where $Z$ represents the set of all integers completed with top and bottom elements.

**Example 4.1.** A derived operator for $Z$ is $t = +(*(x, 3), +(2, 1))$. 
Clearly, a derived operator \( t \) is a function \( t : D \rightarrow D \). Intuitively, the result of the function \( t \) for a given value \( d \in D \) is obtained by evaluating \( t(d) \) interpreting the function symbols in \( t \) according to \( D \). In the above example, \( t(0) = 3 \) and \( t(2) = 9 \).

4.2. The interpretations \( \text{Rsub}, \text{Prop}, \text{and Sharing} \)

The interpretations \( \text{Rsub}, \text{Prop}, \text{and Sharing} \) consist each of a domain and three operations: unification, projection, and least upper bound. Since some of the operations are quite technical, we have chosen to recall them in the appendix, and to describe here only the domains, their partial orders, and the concretization/abstraction functions relating them.

4.2.1. The “concrete” domain \( \text{Rsub} \)

The “concrete” domain \( \text{Rsub} \) \cite{11} is the complete lattice

\[
\text{Rsub} = \left( \phi(\text{Subst}) \times \text{FP}(V) \right) \cup \{ \top_{R}, \bot_{R} \}.
\]

\( \text{Rsub} \) stands for restricted substitutions. The partial order of \( \text{Rsub} \) is defined, on non-trivial elements, by \( [\Sigma_1, U_1] \sqsubseteq [\Sigma_2, U_2] \) iff \( U_1 = U_2 \) and \( \Sigma_1 \subseteq \Sigma_2 \). The operations of \( \text{Rsub} \) are described in the appendix.

4.2.2. The domain \( \text{Prop} \)

For any set of variables \( U \in \text{FP}(V) \), by \( \wedge U \) we denote the formula consisting of the conjunction of the variables in \( U \). For any \( U \in \text{FP}(V) \), a positive formula \cite{2, 20} on \( U \) is any propositional formula containing only variables in \( U \) and that is satisfied by the truth-assignment that assigns \text{true} to all variables in \( U \). The set of positive formulas on \( U \) is denoted \( \text{Pos}_U \). From now on, in order to avoid burdensome notation, we simply write \( f \) for the class of formulas equivalent to \( f \) and assume that \( \text{Pos}_U \) consists of classes of equivalent formulas. We also adopt the usual convention of representing a truth-assignment \( a \) on \( U \) as the set \( \{ x \in U | a(x) = \text{true} \} \).

Notice that for any \( U \in \text{FP}(V) \), \( \text{Pos}_U \cup \{ \text{true} \} \) is a complete lattice with least upper bound and greatest lower bound, respectively, \( \lor \) (logical disjunction) and \( \wedge \) (logical conjunction), appropriately extended to classes of equivalent formulas.

The domain \( \text{Prop} \) is as follows:

\[
\text{Prop} = \{ \langle f, U \rangle : U \in \text{FP}(V), f \in \text{Pos}_U \cup \{ \text{true} \} \} \cup \{ \top_{Pr}, \bot_{Pr} \}.
\]

\( \text{Prop} \) is partially ordered: \( \top_{Pr} \) is the largest element and \( \bot_{Pr} \) is the smallest; for the other elements, \( \langle f_1, U_1 \rangle \leq_{Pr} \langle f_2, U_2 \rangle \) if and only if \( U_1 = U_2 \) and \( f_1 \vdash f_2 \).

Fig. 2 depicts the domain \( \text{Prop} \) for \( V = \{ x, y \} \). The lines represent the ordering relation among the (equivalence classes of) formulas.

That positive formulas are useful for computing variable groundness in logic programs is well-known, see \cite{2, 9, 11, 19, 12}. The intuition behind the relation between positive formulas and substitutions, is as follows. Each substitution defines a truth-assignment, and, since groundness is a property closed under instantiation, we say
that a formula approximates a substitution when it is true w.r.t. the truth-assignments defined by all instances of that substitution.

The truth-assignment of substitution $\sigma$ is $\text{assign } \sigma : \text{assign } \sigma x = \text{true iff } \sigma \text{ grounds } x$. The concretization expressing the relation between Prop and Rsub, is

$$\gamma_{PR,Rs} : \text{Prop} \rightarrow \text{Rsub},$$

$$\gamma_{PR,Rs}(d) = \begin{cases} \top_{Rs} & \text{if } d = \top_{Pr}; \\ \bot_{Rs} & \text{if } d = \bot_{Pr}; \\ \{ \sigma \in \text{Subst} \mid \forall \sigma' \subseteq \sigma \cdot \text{assign } \sigma' \models f, U \} & \text{if } d = [f, U]. \end{cases}$$

The function $\varkappa_{Rs,Pr} : \text{Rsub} \rightarrow \text{Prop}$ is the usual adjoint [6] of $\gamma_{PR,Rs}$, i.e., $\varkappa_{Rs,Pr}(c) = \cap_{Pr}\{d \in \text{Prop} \mid \gamma_{PR,Rs}(d) \supseteq \bot_{Rs} c\}$. The tuple $(\gamma_{PR,Rs}, \varkappa_{Rs,Pr}, \text{Prop}, \varkappa_{Rs,Pr})$ is a Galois insertion [11]. In [11] it is also shown that the operations of Prop (see the appendix) are optimal.
Lemma 4.2 (Cortesi et al. [11]). Prop is optimal.

The definite formulas in $U$, denoted $\text{Def}_U$ [2, 13], consist of the formulas $f \in \text{Pos}_U$ that satisfy the following model intersection property: consider any two models $M_1$ and $M_2$ of $f$, if $M = M_1 \cap M_2$, then $M \models f$. The name “definite” for such formulas comes from the following well-known syntactical characterization: for each $f \in \text{Def}_U$ there is a formula on $U$ equivalent to $f$ and consisting of a conjunction of definite implications of the form $\land W \rightarrow x$.

Observe that the set $\text{Def}_U$ is properly included in $\text{Pos}_U$. For instance, the formula $x \lor y$ of $\text{Pos}_{\{x,y\}}$ does not belong to $\text{Def}_{\{x,y\}}$. In fact, let $M_1 = \{x\}$, and $M_2 = \{y\}$ it is immediate to see that both $M_1$ and $M_2$ are models of $x \lor y$, whereas $M = M_1 \cap M_2 = \emptyset$ is not a model of this formula.

In the same way as positive formulas were used for defining Prop, it is possible to define a domain using the definite formulas:

$$\text{Def} = \{[f, U] \mid U \in \text{FP}(V), f \in \text{Def}_U \cup \{\bot\} \cup \{\top_{\text{Def}}, \bot_{\text{Def}}\}\}.$$  

Obviously, Def abstracts Prop with the identity as concretization and its adjoint as abstraction. From this it also follows that Def also abstracts $\text{RSub}$ with the same concretization as Prop.

4.2.3. The domain Sharing

The abstract domain Sharing proposed by Jacobs and Langen in [16] in order to represent variable aliasing, covering, and groundness is defined by

$$\text{Sharing} = \{[A, U] \mid A \subseteq \wp(U), A \neq \emptyset \Rightarrow \emptyset \in A, U \in \text{FP}(V) \} \cup \{\top_{\text{Sh}}, \bot_{\text{Sh}}\}.$$  

Sharing is partially ordered: $\top_{\text{Sh}}$ is the largest element and $\bot_{\text{Sh}}$ is the smallest one; for the other elements, $[A_1, U_1] \subseteq_{\text{Sh}} [A_2, U_2]$ iff $U_1 \subseteq U_2$ and $A_1 \subseteq A_2$. The domain Sharing for $V = \{x, y\}$ is depicted in Fig. 3. Even though the lattice structure is similar to that of Prop, the two domains represent different informations of the concrete domain RSub.

Jacobs and Langen [16] proved that Sharing enjoys a Galois insertion into $\wp(\text{Subst})$. This can be immediately extended to our concrete domain RSub. We recall briefly the construction of the abstraction of this insertion. For $x \in V$, $U \subseteq V$, and $\sigma \in \text{Subst}$, let $\text{share}(\sigma, x, U)$ be the set of variables in $U$ whose images under $\sigma$ contain the variable $x$, i.e. $\text{share}(\sigma, x, U) = \{y \in U \mid x \in \text{Var}(\sigma y)\}$. For $[\Sigma, U] \in \text{RSub}$,

$$\alpha_{\text{RSub}}([\Sigma, U]) = \{\text{share}(\sigma, x, U) \mid \sigma \in \Sigma, x \in V\}, U\}.$$  

The concretization $\gamma_{\text{ShRb}}$ is the usual adjoint of the abstraction. Let $[A, U] = \alpha_{\text{RSub}}([\sigma], U)$. Each $S \in A$ is a set of variables that under $\sigma$ share a variable. Every variable $x \in U$ such that $\text{Var}(\sigma x) = \emptyset$ will not appear in $A$. In [7] it is shown that the operations of Sharing are optimal.

Lemma 4.3 (Cortesi and Filé [7]). Sharing is optimal.
4.3. Quotients with respect to groundness

The interpretations Sharing and Prop are incomparable with respect to the notion of abstraction [10]. The intuition behind this result is the following. On the one hand, by means of disjunctions, Prop represents also possible equivalence (and thus also groundness), whereas Sharing does not. On the other hand, Sharing represents variable independence that is not expressible in Prop. However, as both interpretations compute groundness information, we are interested in comparing their precision in the computation of groundness.

4.3.1. The domain GR

The simplest domain that represents variable groundness is GR as follows. Given an element \([\Sigma, U] \in \text{Rsub}\), its groundness information can be represented by the set of
variables grounded by every substitution in $\Sigma$:

$$\mathcal{GR} = \{ [A, U] : A \in \wp(U) \cup \{ \bot \}, \ U \in FP(V) \} \cup \{ \top_{\mathcal{GR}}, \bot_{\mathcal{GR}} \}. $$

The set $\mathcal{GR}$ is partially ordered as follows. $\top_{\mathcal{GR}}$ is the top element, and $\bot_{\mathcal{GR}}$ the bottom one. $[B_1, U_1] \sqsubseteq_{\mathcal{GR}} [B_2, U_2]$ if $U_1 = U_2$ and $B_1 \supseteq B_2$. Obviously, $\mathcal{GR}$ is a complete lattice. The least upper bound of two elements $[B_1, U_1]$ and $[B_2, U_2]$ is defined as

$$[B_1, U_1] \sqcup_{\mathcal{GR}} [B_2, U_2] = \begin{cases} [B_1 \cap B_2, U_1] & \text{if } U_1 = U_2, \\ \top_{\mathcal{GR}} & \text{otherwise}. \end{cases}$$

It is easy to see that there are Galois insertion between $\mathcal{GR}$ on the one side and $\mathcal{R}_{\mathcal{sub}}, \mathcal{Prop}, \mathcal{Def}$ and $\mathcal{Sharing}$ on the other. We only specify the concretization and abstraction functions, as proving that each pair of functions forms a Galois insertion, is an easy exercise:

$$\begin{align*}
\gamma_{\mathcal{GR}}([B, U]) &= [\Sigma, U], \text{ where } \Sigma = \{ \sigma \in \text{Subst} \mid \forall x \in B, \text{Var}(\sigma x) = \emptyset \}, \\
\pi_{\mathcal{GR}}([\Sigma, U]) &= \{ x \in U \mid \text{Var}(\sigma x) = \emptyset \}, \ U. \end{align*}$$
\[
\begin{align*}
\gamma_{GRP}([B, U]) &= \gamma_{GRF}([B, U]) = [\land B, U], \\
\alpha_{PGR}([f, U]) &= \alpha_{DF}([f, U]) = \{ x \mid f \models x \}, U, \\
\gamma_{GRS}([B, U]) &= [\psi(U \setminus B), U], \\
\alpha_{SGR}([A, U]) &= [U \setminus (\cup A), U].
\end{align*}
\]

These Galois insertions together with those that connect Prop, Def, and Sharing to Rsub, defined above, are coherent with respect to Rsub.

4.3.2. The quotient of Prop with respect to GR

In Section 3 we have proved that the quotient of an interpretation with respect to a given domain is a domain abstracting the starting domain, provided the associated relation is additive.

The quotient of Prop with respect to GR is Prop itself. This is due to the fact that none of the formulas of Prop is irrelevant for the computation of groundness [11].

Lemma 4.4. Let \([f_1, U], [f_2, U] \in \text{Prop}\). If \(f_1 \neq f_2\), there exists a derived operator \(t\) using the operations of Prop, such that \(\alpha_{PGR}(t([f_1, U])) \neq \alpha_{PGR}(t([f_2, U]))\).

Corollary 4.5. \(r_{Pr}\) is the identity on Prop, and thus it is obviously additive.

Theorem 4.6. \(\varepsilon_{GR}(\text{Prop}) = \text{Prop}\).

Proof. Follows immediately from Corollary 4.5 and Theorem 3.6. \qed

4.3.3. The quotient of Sharing with respect to GR

Let \(r_{SH}\) be the relation on Sharing associated to GR. It will be shown that, differently from what we just saw for Prop, the equivalence classes of \(r_{SH}\) are not singletons. However, \(r_{SH}\) is additive and thus \(\varepsilon_{GR}(\text{Sharing})\) exists. Moreover, we will show that \(\varepsilon_{GR}(\text{Sharing})\) is isomorphic to Def. Because of this fact, the comparison between \(\varepsilon_{GR}(\text{Sharing})\) and \(\varepsilon_{GR}(\text{Prop})\) will be extremely simple. Some results of this section need rather technical proofs. For the readability sake these proofs are given in the appendix.

That Sharing expresses information about groundness is well-known, cf. [16]. A formalization of this intuition was first attempted in [10] where it is shown that between Def and Sharing there is a Galois connection. The following even stronger result has been shown recently in [8].

Theorem 4.7. The domain Def abstracts Sharing with the following abstraction and concretization functions:

For \([A, U] \in \text{Sharing}\), let

\[
\mathcal{G}([A, U]) = \bigwedge \{ \forall W_1 \rightarrow x \mid W_1 \subseteq U, \ x \in U, \ \text{and} \ \forall N \in A : x \in N \Rightarrow (W_1 \cap N) \neq \emptyset \},
\]
\[ \alpha_{ShDf}(A, U) = \begin{cases} \{F, U\} & \text{if } A = \emptyset, \\ \emptyset([A, U]), U & \text{otherwise}. \end{cases} \]

\[ \gamma_{DF, Sh}([f, U]) = \{U \setminus M \mid M \models f\}, U \].

This Galois insertion is coherent w.r.t. \( R_{sub}. \)

According to Definition 3.1, the relation on \( \text{Sharing} \) associated to \( GR \) is defined as follows. Let \( S_1, S_2 \in \text{Sharing} \), and let \( t \) be any derived operator on the operations of \( \text{Sharing} \):

\( (S_1, S_2) \in r_{Sh} \iff \alpha_{ShGR}(t(S_1)) = \alpha_{ShGR}(t(S_2)). \)

The following theorem characterizes \( r_{Sh} \) using the abstractions of the elements of \( \text{Sharing} \) into \( \text{Def} \). Its proof is in the appendix.

**Theorem 4.8.** Let \( S_1, S_2 \in \text{Sharing} \), where \( S_1, S_2 \in \text{Sharing} \):

\( (S_1, S_2) \in r_{Sh} \iff \alpha_{ShDf}(S_1) = \alpha_{ShDf}(S_2). \)

The existence of the quotient \( Z_{GR}(\text{Sharing}) \) is guaranteed by the following result.

**Theorem 4.9.** \( r_{Sh} \) is additive.

**Proof.** Consider \( X \subseteq r_{Sh} \). Let for \( i \in \{1, 2\}, X_i = \{S_i \mid (S_1, S_2) \in X\} \). We want to show that

\( (\sqcup_{Sh} X_1, \sqcup_{Sh} X_2) \in r_{Sh}. \)

By Theorem 4.8,

\( (\sqcup_{Sh} X_1, \sqcup_{Sh} X_2) \in r_{Sh} \iff \alpha_{ShDf}(\sqcup_{Sh} X_1) = \alpha_{ShDf}(\sqcup_{Sh} X_2). \)

Since, by Theorem 4.7, \( \alpha_{ShDf} \) together with its adjoint, forms a Galois insertion, by Proposition 2.3, it is additive and therefore, the following holds:

\[ \alpha_{ShDf}(\sqcup_{Sh} X_1) = \sqcup_{Df} \{\alpha_{ShDf}(x) \mid x \in X_1\} \quad \text{by the additivity} \]

\[ = \sqcup_{Df} \{\alpha_{ShDf}(x) \mid x \in X_2\} \quad \text{by definition of } X_1 \text{ and } X_2 \]

\[ = \alpha_{ShDf}(\sqcup_{Sh} X_2) \quad \text{again by the additivity}. \]

The following theorem characterizes the quotient of \( \text{Sharing} \) w.r.t. \( GR \).

**Theorem 4.10.** \( Z_{GR}(\text{Sharing}) = \text{Def}. \)

**Proof.** It is easy to show that the sets \( \text{Def} \) and \( \{\sqcup_{Sh}[S]_{r_{Sh}} \mid S \in \text{Sharing}\} \) are isomorphic: the abstraction \( \alpha_{ShDf} \) is a bijection that preserves the orders of the two sets. In fact,
Fig. 5. The quotient $\mathcal{J}_{\text{GR}(\text{Sharing})}$ for $V = \{x, y\}$.

on the one hand, if $[A_1, U] \sqsubseteq_{SB} [A_2, U]$ then $A_1 \subseteq A_2$ and thus $\mathcal{C}([A_1, U]) \models \mathcal{G}([A_2, U])$ (see Theorem 4.7 for the definition of $\mathcal{G}$). On the other hand, if $f_1 \models f_2$, with $[f_1, U]$ and $[f_2, U]$ in Def, then $f_1$ has less models than $f_2$, and thus,

$$\{U \setminus M | M \models f_1\} \setminus \{U \setminus M | M \models f_2\}.$$

Hence, $\gamma_{DF, SB}([f_1, U]) \sqsubseteq_{SB} \gamma_{DF, SB}([f_2, U])$. □

Fig. 5 depicts, for the case that $V = \{x, y\}$, the quotient of Sharing with respect to GR, which is the domain Def. Observe that the elements $[[\{x\}, \{y\}, \{x, y\}], \{x, y\}]$ and $[[\{x\}, \{y\}, \{x, y\}]$ belong to the same equivalence class $[true, \{x, y\}]$ via $r_{SB}$. The only difference between these two elements is that the first one represents also substitutions $\sigma$ such that $\sigma x$ and $\sigma y$ share a common variable. However, this distinction is irrelevant when considering only groundness computation. In fact, both elements simply say that $x$ and $y$ are completely unrelated with respect to groundness.
4.4. Comparison of Prop and Sharing w.r.t. GR.

We can finally compare the two interpretations Prop and Sharing using the new theory developed in Section 3.

Theorem 4.11. Prop is strictly more precise than Sharing with respect to the domain GR representing groundness.

Proof. It suffices to show that Theorems 3.9 and 3.10 are applicable, that is, we have to show that all the assumptions ASS (a)–(e) of Section 3.2 are satisfied:
- Point (a) and (b): have been stated in Sections 4.2.2, 4.2.3 and 4.3.1.
- Point (c): is shown in Theorems 4.6 and 4.10.
- Point (d): is stated in Lemmas 4.2 and 4.3.
- Point (e): is stated in Section 4.2.2. □

5. Conclusions

In this paper we addressed the problem of exactly characterizing the part of an abstract domain which is useful for the computation of a given property. To this end, we introduced the notion of quotient of an abstract interpretation. We showed that the comparison of abstract interpretations w.r.t. a common property can be performed by comparing their quotients w.r.t. that property. As an example, we applied this technique to the comparison of two well-known abstract interpretations for logic programs: Prop and Sharing.

Other algebraic operators on abstract domains and abstract interpretations have been proposed in the literature, namely the reduced product [6], the open product [12], the powerset [15], and the complement [7]. An interesting subject for future work is the study of the interaction between the quotient and these operators. For instance, one may wonder whether the quotient of a reduced product is the reduced product of the
quotients and also whether the quotient of the powerset of a domain \( D \) is the powerset of the quotient of \( D \).

Another question that deserves further study is what one can do for comparing two domains when the present framework cannot be applied, for instance, when one of the associated relations is not additive. In this case it may still be possible to perform a comparison by lifting the domains to their powersets and comparing their quotients. In fact, quotients always exist for domains obtained through the powerset operation.

Appendix

The appendix consists of four parts. In the first three, we formally define the operations in \( \text{Rsub, Prop, and Sharing} \). Then, we show some technical lemmas that lead to the proof of Theorem 4.8.

A.1. Operations in \( \text{Rsub} \)

Let \( E \) be a set of term equations. If a substitution \( \sigma \) makes \( \sigma(t_1) \) syntactically identical to \( \sigma(t_2) \) for each \( (t_1 = t_2) \in E \), \( \sigma \) is called a unifier of \( E \). A \emph{most general unifier} of \( E \) is a unifier \( \sigma \) of \( E \) that is more general than any other unifier of \( E \). We denote by \( \text{mgu}(E) \) any idempotent most general unifier of \( E \). It is not necessary to specify which most general unifier is considered, because, from the relationship existing among the idempotent most general unifiers of a given set of equations \( 17 \), it is immediate to see that each of them carries the same information about the properties we are interested in, namely, variable groundness and sharing.

A set of equations \( E \) is in \emph{solved form} if it has the form \( \{x_1 = t_1, \ldots, x_n = t_n\} \), where each \( x_i \) is a distinct variable occurring in none of the terms \( t_j \). Given a set of equations \( E = \{x_1 = t_1, \ldots, x_n = t_n\} \) in solved form, the substitution \( \sigma = \{x_1/t_1, \ldots, x_n/t_n\} \) is an idempotent most general unifier of \( E \); we denote \( E \) by \( \text{Eq}(\sigma) \).

Least upper bound: The operation \( \sqcup_{\text{Rsub}} \), which produces the least upper bound of two elements of \( \text{Rsub} \), is as follows: for any \( k \in \text{Rsub} \), \( \sqcup_{\text{Rsub}} \  k = \sqcap_{\text{Rsub}} \  k \), \( \sqcup_{\text{Rsub}} \  k = k \), for the other elements,

\[
[S_1, U_1] \sqcup_{\text{Rsub}} [S_2, U_2] = \begin{cases} [S_1 \cup S_2, U_1] & \text{if } U_1 = U_2, \\ [S_1, \sqcap_{\text{Rsub}} U_1] & \text{otherwise}. \end{cases}
\]

Projection: The concrete projection \( \pi_{\text{Rsub}} : \text{Rsub} \times \text{FP}(V) \rightarrow \text{Rsub} \) maps \( ([\Sigma, U_1], U_2) \) to \( [\Sigma, U_1 \cap U_2] \).

Thus, projection only changes the second component leaving the first one unchanged. One may think that the fact that the projection does not eliminate from the substitutions the variables that are projected out, may cause problems of variable capture. As usual, variable captures can be avoided using appropriate renamings.
Unification: In order to define the concrete unification \( U_{R^s} \), it is convenient to introduce first the following function \( u_{R^s} \):

\[
u_{R^s} : \text{Subst} \times \text{Subst} \times \text{Subst} \rightarrow \text{Subst}
\]

\[
(\sigma_1, \sigma_2, \delta) \mapsto \text{mgu}(\text{Eq}(\sigma_1) \cup \text{Eq}(\sigma_2) \cup \text{Eq}(\delta)).
\]

\( U_{R^s} \) is strict: if either of the first two arguments is \( \bot_{R^s} \), the result is \( \bot_{R^s} \). Otherwise, if one of these is \( \top_{R^s} \), the result is \( \top_{R^s} \). The other cases are as follows:

\[
U_{R^s} : \text{Rsub} \times \text{Rsub} \times \text{Subst} \rightarrow \text{Rsub}
\]

\[
([\Sigma_1, U_1], [\Sigma_2, U_2], \delta) \mapsto \begin{cases} 
\{u_{R^s}(\sigma_1, \sigma_2, \delta) | \sigma_1 \in \Sigma_1 \land \sigma_2 \in \Sigma_2 \}, \\
U_1 \cup U_2 \cup \text{Var}(\delta). 
\end{cases}
\]

A.2. Operations in Prop

Least upper bound: For all \( d \in \text{Prop} \), \( \top_{Pr} \sqcup_{Pr} d = \top_{Pr} \) and \( \bot_{Pr} \sqcup_{Pr} d = d \); for the other elements,

\[
[f_1, U_1] \sqcup_{Pr} [f_2, U_2] = \begin{cases} 
[f_1 \lor f_2, U_1] & \text{if } U_1 = U_2, \\
\top_{Pr} & \text{otherwise}.
\end{cases}
\]

Projection: The abstract projection \( \pi_{Pr} \) amounts to existentially quantifying a formula [8, 20]. The existential quantification of a propositional formula obeys \( \exists x. f \equiv f(x/T) \lor f(x/F) \).

\[
\pi_{Pr}([f, U], V) = [\exists(u \in V). f \land U \lor V].
\]

Unification: The abstract unification is obtained by means of logical conjunctions. For \( \delta = \{x_i/t_i \mid 1 \leq i \leq n\} \in \text{Subst} \), let \( \varphi_{\delta} = \bigwedge \{x_i \rightarrow (\land \text{Var}(t_i)) \mid 1 \leq i \leq n\} \).

\[
U_{Pr} : \text{Prop} \times \text{Prop} \times \text{Subst} \rightarrow \text{Prop}
\]

\[
([f_1, U_1], [f_2, U_2], \delta) \mapsto [f_1 \land f_2 \land \varphi_{\delta}, U_1 \cup U_2 \cup \text{Var}(\delta)].
\]

A.3. Operations in sharing

Least upper bound: The least upper bound of any two nontrivial elements \([A_1, U_1]\) and \([A_2, U_2]\) is defined by

\[
[A_1, U_1] \sqcup_{Sh} [A_2, U_2] = \begin{cases} 
[A_1 \cup A_2, U_1] & \text{if } U_1 = U_2, \\
\top_{Sh} & \text{otherwise}.
\end{cases}
\]

Projection: The projection on \( \text{Share} \) is the identity on the bottom and top elements. In the other cases it is defined by means of set intersection:

\[
\pi_{Sh} : \text{Sharing} \times \text{FP}(V) \rightarrow \text{Sharing}
\]

\[
([S_1, U_1], U_2) \mapsto \{A \cap U_2 \mid A \in S_1\}, U_1 \cap U_2.
\]
Unification: In order to define the abstract unification function \( U_{Sh} \) for the Sharing domain, we need the following auxiliary functions [16]:

- The closure under union of \( A \in \wp(\wp(V)) \), denoted \( A^\ast \), is the smallest superset of \( A \) satisfying \( X \in A^\ast \) and \( Y \in A^\ast \) implies \( (X \cup Y) \in A^\ast \).
- The part of \( A \in \wp(\wp(V)) \) that is relevant to a term \( t \), denoted \( rel(A,t) \), is the set \( \{ S \in A \mid \text{Var}(t) \cap S \neq \emptyset \} \).
- If \( A, A' \in \wp(\wp(V)) \) then \( A \otimes A' = \{(S \cup S') \mid S \in A \text{ and } S' \in A' \} \).
- The basic unification step is performed by

\[
\psi_{Sh} : \wp(\wp(V)) \times \text{Subst} \rightarrow \wp(\wp(V)),
\]
\[
\forall A_0 \in \wp(\wp(V)), \forall \delta \in \text{Subst}, \delta = \{ x_1/t_1, \ldots, x_m/t_m \}
\]
\[
\psi_{Sh}(A_0, \delta) = \text{amgu}(\text{arg}(x_1, \ldots, x_m), [t_1, \ldots, t_m], A_0).
\]

\[
\text{amgu}(\text{arg}(x_1, \ldots, x_m), [t_1, \ldots, t_m], B) = B.
\]

\[
\text{amgu}(x, t, B) = (B \setminus (\text{rel}(x,B) \cup \text{rel}(t,B))) \cup (\text{rel}(x,B) \setminus \text{rel}(t,B)).
\]

- The backward/forward unification \( U_{Sh} : \text{Sharing} \times \text{Sharing} \times \text{Subst} \rightarrow \text{Sharing} \) is defined as follows. Let \([A, U], [A', U'] \in \text{Sharing}\), with \( U \cap U' = \emptyset \), and let \( \delta \in \text{Subst} \) such that \( \text{Var}(\delta) \subseteq U \cup U' \).

\[
U_{Sh}([A, U], [A', U'], \delta) = [\psi_{Sh}(A \cup A', \delta), U \cup U'].
\]

A.4. Technical results

The goal of this section is to show Theorem 4.8. Some preliminary results are necessary.

Lemma A.1. Let \( S_1, S_2 \in \text{Sharing} \):

\[
\zeta_{SD_f}(S_1) = \zeta_{SD_f}(S_2) \Rightarrow \zeta_{SG_r}(S_1) = \zeta_{SG_r}(S_2).
\]

Proof. It suffices to observe that \( \zeta_{SG_r} = \zeta_{DF_r} \circ \zeta_{SD_f} \), cf. Section 4.3.1 for the definition of the abstraction functions. \( \square \)

In what follows we will use the notion of \( \& \) introduced in Section 4.3.3. For the sake of clarity, we recall its definition below: for \([A, U] \in \text{Sharing}\),

\[
\&([A, U]) = \bigwedge \{ \forall W_1 \rightarrow x \mid W_1 \subseteq U, x \in U, \text{ and } \forall N \in A : x \in N \Rightarrow (W_1 \cap N) \neq \emptyset \}.
\]

Recall also that \( \zeta_{SD_f}([A, U]) = [\&([A, U]), U] \).

The following lemma plays a central role in this section: it expresses the \( \& \) meaning of the result of the \( U_{Sh} \) operation in terms of the conjunction of the \( \& \) meaning of its arguments.
Lemma A.2. Let $S_1 = [A_1, U_1]$, $S_2 = [A_2, U_2] \in \text{Sharing}$, with $U_1 \cap U_2 = \emptyset$, and let $\delta = \{x_1/t_1, \ldots, x_n/t_n\}$ be any substitution in $\text{Subst}$ with $\text{Var}(\delta) \subseteq U_1 \cup U_2$. Let $\varphi_\delta = \bigwedge \{x_i \mapsto (\bigwedge \text{Var}(t_i)) \mid 1 \leq i \leq n\}$.

$$\mathcal{C}(U_{\text{Sh}}(S_1, S_2, \delta)) = \mathcal{C}(S_1) \land \mathcal{C}(S_2) \land \varphi_\delta.$$ 

Proof. We show that the relation holds for the case that $\delta = \{x/t\}$. The general result easily follows by applying the basis case $n$ times.

Let in what follows, $U_{\text{Sh}}(S_1, S_2, \delta) = S' = [A', U]$, where $U = U_1 \cup U_2$. Let also $A = A_1 \cup A_2$ and $S = [A, U]$.

($\Rightarrow$) First, we show that $\mathcal{C}(S') \models \mathcal{C}(S_1) \land \mathcal{C}(S_2) \land (x \mapsto \bigwedge \text{Var}(t))$. The fact that $\mathcal{C}(S') \models x \mapsto \bigwedge \text{Var}(t)$ follows easily from the definition of $U_{\text{Sh}}$. Consider $\text{rel}(A, x)$ and $\text{rel}(A, t)$. If one of them is empty, then $A'$ contains no variable in $\{x\} \cup \text{Var}(t)$. Thus, $\mathcal{C}(S') \models x \mapsto \bigwedge \text{Var}(t)$ and, therefore, $\mathcal{C}(S') \models x \mapsto \bigwedge \text{Var}(t)$.

On the other hand, if both $\text{rel}(A, x)$ and $\text{rel}(A, t)$ are nonempty, then, by definition of $U_{\text{Sh}}$, $A'$ satisfies the following condition: $\forall N \in A'$, $x \in N \iff \text{Var}(t) \cap N \neq \emptyset$. Thus, $\mathcal{C}(S') \models x \mapsto \bigwedge \text{Var}(t)$.

Let us now prove that $\mathcal{C}(S') \models \mathcal{C}(S_1)$. The proof for $\mathcal{C}(S_2)$ is analogous. Consider any definite formula $f = \bigwedge W \to y$ such that

1. $\forall N \in A_1$, $y \in N \Rightarrow W \cap N \neq \emptyset$.

It is easy to see that this condition is satisfied also if we replace $A_1$ with $A'$: each element $N \in A'$ is either an element of $A_1$ (that satisfies (1) by assumption), or it is an element of $A_2$ (that satisfies (1) trivially because $U_1 \cap U_2 = \emptyset$ and $\{y\} \cup W \subseteq U_1$), or it is the union of some elements of $A_1$ and of $A_2$ ((1) is satisfied by the above arguments).

($\Leftarrow$) We will show that $\mathcal{C}(S') \land \varphi_\delta \models \mathcal{C}(S')$. This proves the thesis. In fact, $\mathcal{C}(S) = \mathcal{C}(S_1) \land \mathcal{C}(S_2)$, because in $A$ the elements of $A_1$ and $A_2$ do not interact since they are pairwise disjoint.

We proceed by contradiction. Assume that there is $f = \bigwedge W \to y$ such that $\mathcal{C}(S') = f$, but $\mathcal{C}(S) \land \varphi_\delta \not\models f$. Then it must be the case that $\mathcal{C}(S) \not\models f$ and thus, there must be $B \in A$ such that $y \in B$, but $W \cap B = \emptyset$. Surely, $B \cap \text{Var}(x = t) \neq \emptyset$, otherwise $B \in A'$, that contradicts the hypothesis. Thus, $B \in \text{rel}(A, x) \cup \text{rel}(A, t)$. From this, it follows that

2. $\mathcal{C}(S) \models ((\bigwedge W) \land x \land (\bigwedge \text{Var}(t))) \to y$

because for each $B \in A$, if $y \in B$, but $W \cap B = \emptyset$, then $\{(x) \cup \text{Var}(t)) \cap B \neq \emptyset$.

Assume that $B \in \text{rel}(A, t)$, the case that $B \in \text{rel}(A, x)$ is analogous. By definition of $U_{\text{Sh}}$, $A'$ contains $B \otimes \text{rel}(A, x)$. By the initial assumption, $\forall R \in \text{rel}(A, x)$ it must be that $R \cap W \neq \emptyset$. From this it follows that

3. $\mathcal{C}(S) \models \bigwedge W \to x$
Thus, from (2) and (3) we have
\[ \mathcal{C}(S) \land (x \leftrightarrow \bigwedge \{ \text{Var}(t) \}) \models [(x \leftrightarrow \bigwedge \{ \text{Var}(t) \})] \land [\bigwedge W \rightarrow x] \land [(\bigwedge W) \land x \land (\bigwedge \text{Var}(t)) \rightarrow y], \]
from which one easily obtains
\[ \mathcal{C}(S) \land (x \leftrightarrow \bigwedge \{ \text{Var}(t) \}) \models \bigwedge W \rightarrow y. \quad \square \]

In the following three lemmas, we will show, for each one of the three operations of \textit{Sharing}, that the \( \mathcal{C} \) meaning of its result is completely determined by the \( \mathcal{C} \) meaning of its arguments.

**Lemma A.3** (Unification). \textit{Let} \( S_1 = [A_1, U], S_2 = [A_2, U] \) and \( S' = [A', U'] \) \textit{be elements of Sharing such that} \( U \cap U' = \emptyset \). \textit{Let also} \( \delta \in \text{Subst} \) \textit{such that } \text{Var}(\delta) \subseteq U \cup U'. \textit{Then}
\[ \mathcal{C}(S_1) = \mathcal{C}(S_2) \Rightarrow \mathcal{C}(U_{\text{Sh}}(S', S_1, \delta)) = \mathcal{C}(U_{\text{Sh}}(S', S_2, \delta)). \]

**Proof.** By Lemma A.2, we know that
\[ \mathcal{C}(U_{\text{Sh}}(S', S_1, \delta)) = \mathcal{C}(S') \land \mathcal{C}(S_1) \land \varphi_\delta, \]
by assumption
\[ = \mathcal{C}(S') \land \mathcal{C}(S_2) \land \varphi_\delta, \]
\[ = \mathcal{C}(U_{\text{Sh}}(S', S_2, \delta)). \quad \square \]

**Lemma A.4** (Projection). \textit{Let} \( S_1 = [A_1, U], S_2 = [A_2, U] \in \text{Sharing}, \) and \( U' \) \textit{be a finite set of variables.}
\[ \mathcal{C}(S_1) = \mathcal{C}(S_2) \Rightarrow \mathcal{C}(\pi_{\text{Sh}}(S_1, U')) = \mathcal{C}(\pi_{\text{Sh}}(S_2, U')). \]

**Proof.** Straightforward. \( \square \)

**Lemma A.5** (Lub). \textit{Let} \( S_1 = [A_1, U], S_2 = [A_2, U] \) \textit{be elements of Sharing with} \( A_1, A_2 \neq \emptyset \), and \( S' = [A', U'] \in \text{Sharing}: \)
\[ \mathcal{C}(S_1) = \mathcal{C}(S_2) \Rightarrow \mathcal{C}(S_1 \cup_{\text{Sh}} S') = \mathcal{C}(S_2 \cup_{\text{Sh}} S'). \]

**Proof.** Recall that, for any \( S = [A, U] \in \text{Sharing}, \alpha_{\text{ShDF}}(S) = \mathcal{C}(S), U \]. Thus, we can use the additivity of \( \alpha_{\text{ShDF}} \), obtaining the following:
\[ \alpha_{\text{ShDF}}(S_1 \cup_{\text{Sh}} S') = \alpha_{\text{ShDF}}(S_1) \cup_{\text{DF}} \alpha_{\text{ShDF}}(S') = \alpha_{\text{ShDF}}(S_2) \cup_{\text{DF}} \alpha_{\text{ShDF}}(S') = \alpha_{\text{ShDF}}(S_2 \cup_{\text{Sh}} S') \]
which shows the thesis. \( \square \)

Finally, we show Theorem 4.8.
Theorem 4.8. Let $S_1, S_2 \in \text{Sharing}$, where $S_1 = [A_1, U_1], S_2 = [A_2, U_2]$ and $A_1, A_2 \neq \emptyset$.

$$(S_1, S_2) \in r_{Sh} \iff \pi_{ShDf}(S_1) = \pi_{ShDf}(S_2).$$

Proof.

$$(\Rightarrow)$$ From the definition of $r_{Sh}$ it follows immediately that $U_1 = U_2$: it is sufficient to consider the empty sequence of operations to obtain

$$([A_1, U_1], [A_2, U_2]) \in r_{Sh} \Rightarrow \pi_{ShGr}([A_1, U_1]) = \pi_{ShGr}([A_2, U_2]) \Rightarrow U_1 = U_2.$$  

It remains to show that $\Psi(S_1) = \Psi(S_2)$. Assume the converse. This means that there exists a definite formula $\psi = \bigwedge W \rightarrow x$ such that $\Psi(S_1) \models \psi$ and $\Psi(S_2) \not\models \psi$ (or vice versa). Notice that this means that there exists $N \in A_2$ such that $x \in N$ but $W \cap N = \emptyset$.

Consider $S_0 = [\{\emptyset\}, U_0]$ with $U_0 \cap U_1 = \emptyset$ (and thus $U_0 \cap U_2 = \emptyset$). Call $U = U_0 \cup U_1$.

Consider also the substitution $\delta = \{x/a \mid x \in W\}$. Let

$$U_{Sh}(S_0, S_1, \delta) = [R_1, U] \quad \text{and} \quad U_{Sh}(S_0, S_2, \delta) = [R_2, U].$$

By the definition of $U_{Sh}$, $R_1 = A_1 \setminus \{H \in A_1 \mid H \cap W \neq \emptyset\}$. Thus, by the hypothesis, $x \notin R_1$ and thus, by definition of $\pi_{ShGr}$ (cf. Section 4.3.1), $x$ is in the first component of $\pi_{ShGr}([R_1, U])$.

On the other hand, $R_2 = A_2 \setminus \{H \in A_2 \mid H \cap W \neq \emptyset\}$ and thus $N \in R_2$ from which, $x \in R_2$. From this it follows that $x$ does not belong to the first component of $\pi_{ShGr}([R_2, U])$. Thus we arrived to a contradiction of our initial hypothesis.

$$(\Leftarrow)$$ Since, by definition, cf. Theorem 4.7,

$$\pi_{ShDf}(S_1) = \pi_{ShDf}(S_2) \iff U_1 = U_2 \quad \text{and} \quad \Psi(S_1) = \Psi(S_2)$$

it suffices to show that for any derived operator $t$ on $\{\cup_{Sh}, \pi_{Sh}, U_{Sh}\}$,

$$\Psi(S_1) = \Psi(S_2) \land U_1 = U_2 \Rightarrow \pi_{ShGr}(t(S_1)) = \pi_{ShGr}(t(S_2))$$

By structural induction on the derived operator $t$, using Lemmas A.3–A.5, one can show that $\Psi(t(S_1)) = \Psi(t(S_2))$. From this it follows that

$$(1) \quad \pi_{ShDf}(t(S_1)) = \pi_{ShDf}(t(S_2))$$

because $U_1 = U_2$ by hypothesis, and the same derived operator $t$ is applied to $S_1$ and $S_2$ and thus the two results will have equal second components. By Lemma A.1. (1) implies the thesis. $\square$

References
