Abstract Domains for Universal and Existential Properties

Andrew Heaton[1] and Patricia M. Hill[2] and Andy King[3]

[1] School of Computer Studies, University of Leeds, LS2 9JT, UK,
heaton@scs.leeds.ac.uk,
tel: +44 113 233 5322, fax: +44 113 233 5468.

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

[3] Computing Laboratory, University of Kent at Canterbury, CT2 7NF, UK,
amk@ukc.ac.uk.

Abstract. Abstract interpretation theory has successfully been used for constructing algorithms to statically determine run-time properties of programs. Central is the notion of an abstract domain, describing certain properties of interest about the program. In logic programming, program analyses typically fall into two different categories: either they detect program points where the property definitely holds (universal analyses) or possibly holds (existential analyses). We study the relation between such analyses in the case where the concrete domain is a lattice join-generated by its set of join-irreducible elements. Although our intended application is for logic programming, the theory is sufficiently general for possible applications to other languages.

1 Introduction

Abstract interpretation theory has successfully been used for constructing algorithms to statically determine run-time properties of programs. Traditionally, the semantics of the program is specified with a concrete domain. The central notion is to approximate program semantics by defining an abstract domain whose operations mimic those of the concrete domain. The abstract domain describes certain properties of interest about the program. Each element of the abstract domain specifies information about a possibly infinite number of concrete states. Thus, in order to construct an abstract domain tracing a property of the program, the property needs to be considered as a property over sets of concrete states.

Our aim is to provide new techniques for the construction of new abstract domains from given ones. Many operations have been designed for systematically constructing new domains. Domain operators studied include reduced product [8,4], reduced power [8] and disjunctive completion [8,11]. Linear refinement is introduced in [13] as an extension of the Heyting completion studied in [14]. In [15], a new domain for freeness analysis of logic programs is defined using linear refinement. In this paper, we suppose that the concrete domain is a lattice join-generated by its set of join-irreducible elements. In this case, given any property p defined over each individual concrete state, p can always be uniformly extended to a property over sets of concrete states.

For example, in logic programming it is standard to define the concrete domain as the powerset of substitutions, $\wp(Sub)$, partially ordered by set inclusion. $\wp(Sub)$ is join-generated by Sub. For many properties of logic programs, it is natural to first define the property on substitutions and then lift the property to include sets of substitutions. Consider the property of groundness. A variable x is ground under a substitution $\theta \in Sub$ if θ binds x to a term with no variables. Letting X be the set of variables of interest, the mapping $gr : Sub \to \wp(X)$ is defined:

$$gr(\theta) = \{x \in X \mid var(\theta(x)) = \emptyset\}.$$

Suppose we now want to consider groundness as a property with domain $\wp(Sub)$. We can consider either definite (universal) groundness or possible (existential) groundness. For definite groundness, $Gr^{\forall} : \wp(Sub) \to \wp(X)$ is defined:

$$Gr^{\forall}(\Theta) = \bigcap\{gr(\theta) \mid \theta \in \Theta\}.$$

For possible groundness, $Gr^{\exists} : \wp(Sub) \to \wp(X)$ is defined:

$$Gr^{\exists}(\Theta) = \bigcup\{gr(\theta) \mid \theta \in \Theta\}.$$

Note that definite groundness traces positive information about the groundness of program variables, whereas possible groundness traces negative information. Knowledge of both positive and negative information about program properties such as groundness is particularly useful for debugging applications.

In general, given a concrete domain C, an abstract domain D and a property p mapping the join-irreducible elements of C to D, p is extended to C using the join operation of D. We name this extension of p the D-lattice property of p. For example, $Gr^{\forall}$ is the $D^{\forall}_{gr}$-lattice property of gr where $D^{\forall}_{gr}$ is the lattice $\wp(Sub)$, partially ordered by $\supseteq$ with set intersection as the join operation. $Gr^{\exists}$ is the $D^{\exists}_{gr}$-lattice property of gr where $D^{\exists}_{gr}$ is the lattice $\wp(Sub)$, partially ordered by $\subseteq$ with set union as the join operation.

The main theoretical results shown are as follows:

- Given a Galois connection (C, α, D, γ) (where C is completely distributive and join-generated by its set of join-irreducible elements) specifying an analysis tracing positive information of p, we show how to construct a mirror Galois connection $(C, \alpha^m, D^d, \gamma^m)$ (where D^d is the dual lattice of D) specifying an analysis tracing negative information of p.
- Suppose $op : C \to C$ is a concrete operation and $\langle D, op' \rangle$ is a correct abstract interpretation of $\langle C, op \rangle$ specified by (C, α, D, γ). We find conditions on $\langle D, op' \rangle$ and $\langle C, op \rangle$ which ensure that $\langle D^d, op' \rangle$ is a correct abstract interpretation of $\langle C, op \rangle$ specified by $(C, \alpha^m, D^d, \gamma^m)$.

The paper is organised as follows: in Section 3 we define the notion of lattice properties and mirror properties. Section 4 considers some applications with the well-known domains *Pos* and *Sharing* of logic programming. In Section 5 we consider the safe approximation of concrete functions in analyses for mirror properties. Finally, Section 6 gives some concluding remarks and directions for future work.

2 Preliminaries

Throughout the paper, we assume familiarity with the basic notions of lattice theory ([3]) and abstract interpretation ([7–9]). Below we introduce notation and recall some of the central notions.

2.1 Lattice Theory

In the following, we assume $\langle A, \sqsubseteq_A, \sqcap_A, \sqcup_A, \top_A, \bot_A \rangle$ is a complete lattice. The *dual* lattice $\langle A, \sqsubseteq_A^d, \sqcap_A^d, \sqcup_A^d, \top_A^d, \bot_A^d \rangle$ is defined such that:

1. $\forall a, b \in A. a \sqsubseteq_A^d b$ iff $b \sqsubseteq_A a$;
2. $\sqcap_A^d = \sqcup_A$;
3. $\sqcup_A^d = \sqcap_A$;
4. $\top_A^d = \bot_A$;
5. $\bot_A^d = \top_A$.

We will often write A^d to denote the dual lattice $\langle A, \sqsubseteq_A^d, \sqcap_A^d, \sqcup_A^d, \top_A^d, \bot_A^d \rangle$. Given a mapping $f : A_1 \to A_2$, we will sometimes abuse notation by also writing f to denote the dual mapping $f^d : A_1^d \to A_2^d$ such that $f(a) = f^d(a)$ for all $a \in A_1$.

An element $a \in A$ is *join-irreducible* if, for any $S \subseteq A$, $a = \sqcup_A S$ implies $a \in S$. The set of join-irreducible elements of A is denoted by $JI(A)$. Letting $S \subseteq A$, then A is *join-generated* by S if, for all $a \in A$, $a = \sqcup_A \{x \in S \mid x \sqsubseteq_A a\}$. For convenience, we assume $\bot_A = \sqcup_A \emptyset$. An element $a \in A$ is an *atom* if a covers $\bot_A$, i.e. $a \neq \bot_A$ and $\forall x \in A.(\bot_A \sqsubset_A x \sqsubseteq_A a) \Rightarrow (x = a)$. We denote by $atom_A$ the set of atoms of A. Note that $atom_A \subseteq JI(A)$. A is *atomistic* if A is *join-generated* by $atom_A$. A is *dual-atomistic* if A^d is atomistic.

A complete lattice A is *completely distributive* if, for any $\{x_{i,k} \mid i \in I, k \in K(i)\} \subseteq A$, the following identity holds:

$$\bigsqcap_{i \in I} \bigsqcup_{k \in K(i)} x_{i,k} = \bigsqcup_{f \in I \rightsquigarrow K} \bigsqcap_{i \in I} x_{i,f(i)},$$

where for any $i \in I$, $K(i)$ is a set of indices, and $I \rightsquigarrow K$ is the set of all functions f from I to $\bigcup_{i \in I} K(i)$ such that $\forall i \in I. f(i) \in K(i)$.

Example 1. The powerset of any set S, $\wp(S)$, ordered with set-theoretic inclusion, is completely distributive and join-generated by S. In this case $\wp(S)$ is also an atomistic lattice where the atoms are the elements of S.

The key property of completely distributive lattices we shall use is:

Lemma 1 ([2]). Let A be a completely distributive lattice. Then, $x \in JI(A)$ iff for any $S \subseteq A$, $x \sqsubseteq_A \bigsqcup_A S$ implies $x \sqsubseteq_A s$ for some $s \in S$.

2.2 Galois Connections

If C and D are posets and $\alpha : C \to D$, $\gamma : D \to C$ functions such that $\forall c \in C.\forall d \in D.\alpha(c) \sqsubseteq_D d \Leftrightarrow c \sqsubseteq_C \gamma(d)$, then (C, α, D, γ) is a *Galois connection* between C and D. If in addition γ is 1-1, or, equivalently, α is onto then (C, α, D, γ) is a *Galois insertion* of D in C. In the setting of abstract interpretation, C and D are called the *concrete* and *abstract* domains, respectively. Given a Galois connection (C, α, D, γ), α and γ are uniquely determined by each other. A practical consequence of this is that an abstract interpretation can be performed by defining only one of α or γ. We assume that every concrete domain C and abstract domain D form complete lattices. Given a concrete domain C and an abstract domain D, a property is defined as a (partial) mapping from C to D. Every Galois connection (C, α, D, γ) can be viewed as a specification of the property $\alpha : C \to D$.

An important property of Galois connections is the preservation of bounds. Suppose C, D are complete lattices. A mapping $\alpha : C \to D$ is *additive* if it preserves least upper bounds. Thus if $S \subseteq C$ then $\alpha(\bigsqcup_C S) = \bigsqcup_D \{\alpha(c) \mid c \in S\}$. A mapping $\alpha : C \to D$ is *co-additive* if $\alpha : C^d \to D^d$ is additive. If (C, α, D, γ) is a Galois connection, then α is additive. The converse is also true, i.e. if α is additive then α entirely determines a unique Galois connection (C, α, D, γ). Thus in order to define a Galois connection (C, α, D, γ) (where C, D are complete lattices), it is sufficient to define an additive α.

One way of defining new Galois connections is by composition. Given two Galois connections $(C, \alpha_A, A, \gamma_A)$ and $(A, \alpha_D, D, \gamma_D)$, $(C, \alpha_A \circ \alpha_D, D, \gamma_D \circ \gamma_A)$ is a Galois connection. We call $(C, \alpha_A \circ \alpha_D, D, \gamma_D \circ \gamma_A)$ the *composition* of $(C, \alpha_A, A, \gamma_A)$ and $(A, \alpha_D, D, \gamma_D)$.

Suppose (C, α, D, γ) is a Galois connection and $op_C : C \to C$, $op_D : D \to D$ are operations on C and D, respectively. $\langle D, op_D \rangle$ is a *correct abstract interpretation* of $\langle C, op_C \rangle$ specified by (C, α, D, γ) if $\alpha(op_C(\gamma(d))) \sqsubseteq_D op_D(d)$ for all $d \in D$. $\langle D, op_D \rangle$ is *optimal* if $op_D = \alpha \circ op_C \circ \gamma$. If $\langle D, op_D \rangle$ is optimal, then op_D is the best approximation of op_C relative to D. $\langle D, op_D \rangle$ is *complete* if $\alpha \circ op_C = op_D \circ \alpha$. Completeness is a stronger property than optimality. Indeed, whenever $\langle D, op_D \rangle$ is complete, it can be shown that $op_D = \alpha \circ op_C \circ \gamma$ [10, 12]. The completeness of op_C depends on D and is a property of the abstract domain.

If (C, α, D, γ) is a Galois insertion, each value of the abstract domain D is useful in the presentation of the concrete domain as all the elements of D represent distinct members of C. Moreover, any Galois connection may be lifted to a Galois insertion. This is done by identifying those values of the abstract domain with the same concrete meaning into an equivalence class. This process is known as *reduction* of the abstract domain. Each Galois insertion (C, α, D, γ)

can equivalently be considered as an upper closure operator on C, $\rho = \gamma \circ \alpha$. For every Galois connection (C, α, D, γ), let $(C, \alpha_{\equiv}, D_{\equiv}, \gamma_{\equiv})$ be the Galois insertion obtained by reducing (C, α, D, γ). We associate the (upper) closure operator $\rho = \gamma_{\equiv} \circ \alpha_{\equiv}$ with (C, α, D, γ). The set of closure operators on C is partially ordered such that $\rho_1 \sqsubseteq \rho_2$ if $\forall c \in C.\ \rho_1(c) \sqsubseteq_C \rho_2(c)$. In this approach, the order relation on the set of closure operators on C corresponds to the order by means of which abstract domains are compared with regard to precision. More formally, if $(C, \alpha_1, D_1, \gamma_1)$ and $(C, \alpha_2, D_2, \gamma_2)$ are Galois connections with the associated closure operators ρ_1 and ρ_2, respectively, then we say D_1 is more precise than D_2 if $\rho_1 \sqsubseteq \rho_2$.

3 Properties of Programs

In abstract interpretation, Galois connections are used to specify properties of programs. To define a Galois connection (C, α, D, γ) between a concrete domain C and an abstract domain D, all we need to do is define an additive function $\alpha : C \to D$. It is well known that in the case where the concrete lattice C is join-generated by $JI(C)$, additive functions mapping C to an abstract domain D are completely determined by their values for join irreducible elements. More specifically, if $\alpha : C \to D$ is additive then

$$\alpha(c) = \bigsqcup_D \{\alpha(x) \mid x \in JI(C) \wedge x \sqsubseteq_C c\}.$$

Example 2. For logic programs, a standard choice of concrete lattice is the atomistic lattice $C_L = \langle \wp(Sub), \subseteq, \cap, \cup, \emptyset, Sub \rangle$, where Sub denotes the set of idempotent substitutions.

A program variable is ground if it is bound to a unique value. Groundness can be thought of as a property over Sub, i.e. as a property over $JI(C_L)$. Let X be the set of variables of interest. Then the set of variables ground under $\theta \in Sub$ is given by $gr : JI(C_L) \to \wp(X)$ defined

$$gr(\theta) = \{x \in X \mid var(\theta(x)) = \emptyset\}.$$

Let $\Theta \subseteq Sub$. The set of variables that are definitely ground under all $\theta \in \Theta$ is given by $Gr^{\forall} : C_L \to \wp(X)$ where

$$Gr^{\forall}(\Theta) = \{x \in X \mid \forall \theta \in \Theta . var(\theta(x)) = \emptyset\} = \bigcap \{gr(\theta) \mid \theta \in \Theta\}.$$

Alternatively, the set of variables that are possibly ground under all $\theta \in \Theta$ is given by $Gr^{\exists} : C_L \to \wp(X)$ where

$$Gr^{\exists}(\Theta) = \{x \in X \mid \exists \theta \in \Theta . var(\theta(x)) = \emptyset\} = \bigcup \{gr(\theta) \mid \theta \in \Theta\}. \ \square$$

Definition 1. Let C be a lattice. Then p is an *JI property* for C if there exists a set D such that p maps $JI(C)$ to D (denoted $p : JI(C) \to D$). $\square$

Definition 2. Suppose C is join-generated by $JI(C)$ and let $p : JI(C) \to D$ be a JI property for C. Suppose D forms a complete lattice under the partial ordering $\sqsubseteq_D$. Then the *D-lattice* property of p, $P : C \to D$, is defined such that for every $c \in C$,

$$P(c) = \bigsqcup_D \{p(x) \mid x \in JI(C) \land x \sqsubseteq_C c\}.$$

Let D^d be the dual lattice of D. If P is the D-lattice property of p then we define the *mirror* property of P to be the D^d-lattice property of p. □

Note that the mirror of the mirror of P is P.

Example 3. Let D_{gr} be the complete lattice $(\wp(X), \subseteq, \cap, \cup, \emptyset, X)$. In Example 2, $Gr^{\exists}$ is the D_{gr}-lattice property of gr, and $Gr^{\forall}$ is the D^d_{gr}-lattice property of gr. Hence $Gr^{\forall}$ and $Gr^{\exists}$ are mirror properties. □

In the case where C is also a completely distributive lattice, we have the following theorem.

Theorem 1. Suppose C is a completely distributive lattice join generated by $JI(C)$ and D is a complete lattice. Let (C, α, D, γ) be a Galois connection. Then there exists α^m, γ^m such that

1. α^m is the mirror property of α.
2. $(C, \alpha^m, D^d, \gamma^m)$ is a Galois connection.

Proof. To prove 1, observe that as C is join-generated by $JI(C)$, for each $c \in C$,

$$\alpha(c) = \bigsqcup_D \{\alpha(x) \mid x \in JI(C) \land x \sqsubseteq_C c\}.$$

Hence by Definition 2,

$$\alpha^m(c) = \bigsqcap_D \{\alpha(x) \mid x \in JI(C) \land x \sqsubseteq_C c\}.$$

To prove 2, it is sufficient to show that α^m is additive. But

$$\begin{aligned}\alpha^m(\textstyle\bigsqcup_C S) &= \textstyle\bigsqcap_D \{\alpha(x) \mid x \in JI(C) \land x \sqsubseteq_C \bigsqcup_C S\} && \text{(by Definition 2)}\\ &= \textstyle\bigsqcap_D \{\alpha(x) \mid x \in JI(C) \land x \sqsubseteq_C s \land s \in S\} && \text{(by Lemma 1)}\\ &= \textstyle\bigsqcap_D \{\alpha^m(s) \mid s \in S\}.\end{aligned}$$

Hence α^m is additive. □

The compositional design of Galois connections is a method for specifying program properties by successive refinements. The following lemma gives a sufficient condition for the preservation of compositions of Galois connections between mirror properties.

Lemma 2. Suppose C is a completely distributive lattice join-generated by $JI(C)$, and A, D are complete lattices. Suppose $(C, \alpha_p, D, \gamma_p)$, $(C, \alpha_p^m, D^d, \gamma_p^m)$, $(C, \alpha_A, A, \gamma_A)$ and $(C, \alpha_A^m, A^d, \gamma_A^m)$ are Galois connections such that α_p, α_p^m and α_A, α_A^m are mirror properties. Also suppose $(A, \alpha_D, D, \gamma_D)$ is a Galois connection such that $(C, \alpha_p, D, \gamma_p)$ is the composition of $(C, \alpha_A, A, \gamma_A)$ and $(A, \alpha_D, D, \gamma_D)$. Then if α_D is co-additive, there exists $\gamma_D : D^d \to A^d$ such that $(A^d, \alpha_D, D^d, \gamma_D)$ forms a Galois connection and $(C, \alpha_p^m, D^d, \gamma_p^m)$ is the composition of $(C, \alpha_A^m, A^d, \gamma_A^m)$ and $(A^d, \alpha_D, D^d, \gamma_D)$.

Proof. First note that $\alpha_D : A \to D$ is co-additive implies that $\alpha_D : A^d \to D^d$ is additive, and so there exists $\gamma_D : D^d \to A^d$ such that $(A^d, \alpha_D, D^d, \gamma_D)$ forms a Galois connection.

To show that $(C, \alpha_p^m, D^d, \gamma_p^m)$ is the composition of $(C, \alpha_A^m, A^d, \gamma_A^m)$ and $(A^d, \alpha_D, D^d, \gamma_D)$, it is sufficient to show that $\alpha_p^m = \alpha_D \circ \alpha_A^m$. Suppose $c \in C$. By Definition 2,

$$\alpha_p^m(c) = \bigsqcap_D \{\alpha_p(x) \mid x \in JI(C) \wedge x \sqsubseteq_C c\}.$$

Now $\alpha_p(x) = \alpha_D(\alpha_A(x))$ and so

$$\alpha_p^m(c) = \bigsqcap_D \{\alpha_D(\alpha_A(x)) \mid x \in JI(C) \wedge x \sqsubseteq_C c\}.$$

But α_D is co-additive and so

$$\alpha_p^m(c) = \alpha_D(\bigsqcap_A \{\alpha_A(x) \mid x \in JI(C) \wedge x \sqsubseteq_C c\}) = \alpha_D(\alpha_A^m(c)).\square$$

Let ρ_p, ρ_A be the associated closure operators of $(C, \alpha_p, D, \gamma_p)$ and $(C, \alpha_A, A, \gamma_A)$, respectively. Note that whenever $(C, \alpha_p, D, \gamma_p)$ is the composition of $(C, \alpha_A, A, \gamma_A)$ and $(A, \alpha_D, D, \gamma_D)$, then $\rho_A \sqsubseteq \rho_p$. Thus Lemma 2 can be interpreted as giving a sufficient condition for the preservation of the relative precision between mirror properties, that is, when $\rho_A \sqsubseteq \rho_p$ implies $\rho_A^m \sqsubseteq \rho_p^m$ (where ρ_p^m, ρ_A^m are the associated closure operators of $(C, \alpha_A^m, A^d, \gamma_A^m)$ and $(C, \alpha_p^m, D^d, \gamma_p^m)$, respectively).

4 Applications

We consider the abstract domains *Pos* and *Sharing* from logic programming. In the following, let *Vars* denote a countable set of variables, and X denote a non-empty finite subset of *Vars* containing the variables of interest.

4.1 *Pos*

We briefly recall the definition of *Pos*. The domain *Pos* consists of the set of positive propositional formulae on X, where a propositional formula is positive if it is satisfied when every variable is assigned the value true. *Pos* is a lattice whose ordering is given by logical consequence, and the join and meet by logical disjunction and conjunction, respectively. Adding the bottom propositional formula *false* to *Pos*, makes *Pos* a complete lattice. Letting C_L be the concrete domain defined in Example 2, the Galois insertion $(C_L, \alpha_{pos}, Pos, \gamma_{pos})$ is such that $\alpha_{pos} : C_L \to Pos$ where for all $\Theta \in C_L$,

$$\alpha_{pos}(\Theta) = \bigvee_{\theta \in \Theta} \bigwedge_{x \in X} \{x \leftrightarrow \bigwedge var(\theta(x))\}.$$

Note that α_{pos} is the *Pos*-lattice property of the *JI* property $p_{pos} : Sub \to Pos$ defined such that

$$p_{pos}(\theta) = \bigwedge_{x \in X} \{x \leftrightarrow \bigwedge var(\theta(x))\}.$$

The abstract unification function for *Pos*, $Unif^{pos} : Pos \times Pos \to Pos$, is given by logical conjunction, that is, the meet operation of *Pos*.

Recall that in Examples 2 and 3, definite groundness is specified by $Gr^{\forall}$. In fact $Gr^{\forall}$ maps C_L onto D^d_{gr} and so there exists $\gamma^{\forall}$ such that $(C_L, Gr^{\forall}, D^d_{gr}, \gamma^{\forall})$ forms a Galois insertion. This domain is originally due to Jones and Søndergaard [16]. In [18], when considering the concrete domain to be sets of substitutions closed by instantiation, it is shown that *Pos* can be constructed by using only the definition of groundness. More specifically, [18] shows that *Pos* is exactly the least abstract domain which contains all the (double) intuitionistic implications between elements of D^d_{gr}.

Let $\alpha_D : Pos \to D^d_{gr}$ be defined such that for all $\phi \in Pos$,

$$\alpha_D(\phi) = \{x \in X \mid \phi \models x\}.$$

Now α_D is additive since $\alpha_D(\phi_1 \vee \phi_2) = \alpha_D(\phi_1) \cap \alpha_D(\phi_2)$. Hence there exists γ_D such that $(Pos, \alpha_D, D^d_{gr}, \gamma_D)$ forms a Galois connection. Also $Gr^{\forall}(\Theta) = \alpha_D(\alpha_{pos}(\Theta))$ for all $\Theta \in C_L$, therefore $(C_L, Gr^{\forall}, D^d_{gr}, \gamma^{\forall})$ is the composition of $(C_L, \alpha_{pos}, Pos, \gamma_{pos})$ and $(Pos, \alpha_D, D^d_{gr}, \gamma_D)$.

The mirror property of $Gr^{\forall}$ is $Gr^{\exists}$. Now $Gr^{\exists}$ maps C_L onto D_{gr} and so there exists $\gamma^{\exists}$ such that $(C_L, Gr^{\exists}, D^d_{gr}, \gamma^{\exists})$ forms a Galois insertion.

The mirror property of α_{pos} is $\alpha^m_{pos} : C_L \to Pos^d$ where

$$\alpha^m_{pos}(\Theta) = \bigwedge_{\theta \in \Theta} \bigwedge_{x \in X} \{x \leftrightarrow \bigwedge var(\theta(x))\}.$$

Lemma 3. There exists γ^m_{pos} such that $(C_L, \alpha^m_{pos}, Pos^d, \gamma^m_{pos})$ forms a Galois connection. Also $(C_L, Gr^{\exists}, D_{gr}, \gamma^{\exists})$ is the composition of $(C_L, \alpha^m_{pos}, Pos^d, \gamma^m_{pos})$ and $(Pos^d, \alpha_D, D_{gr}, \gamma_D)$.

Proof. By Theorem 1 there exists γ^m_{pos} such that $(C_L, \alpha^m_{pos}, Pos^d, \gamma^m_{pos})$ forms a Galois connection. Now $\alpha_D(\phi \wedge \psi) = \alpha_D(\phi) \cup \alpha_D(\psi)$, and so $\alpha_D : Pos \to D^d_{gr}$ is co-additive. Therefore by Lemma 2, $(C_L, Gr^{\exists}, D_{gr}, \gamma^{\exists})$ is the composition of $(C_L, \alpha^m_{pos}, Pos^d, \gamma^m_{pos})$ and $(Pos^d, \alpha_D, D_{gr}, \gamma_D)$. □

Lemma 4. If $Card(X) \geq 2$, α^m_{pos} is not onto, thus $(C_L, \alpha^m_{pos}, Pos^d, \gamma^m_{pos})$ is not a Galois insertion.

Proof. By inspecting the definition of α^m_{pos}, it can be seen that $\alpha^m_{pos}(\Theta) \neq \bigvee X$ when $Card(X) \geq 2$, for any $\Theta \in C_L$. Hence α^m_{pos} is not onto. □

In order to obtain a Galois insertion, we apply the reduction process to Pos^d. $(C_L, \alpha^m_{pos}, Pos^d, \gamma^m_{pos})$ reduces to $(C_L, \alpha^m_{pos/\equiv}, Pos^d/\equiv, \gamma^m_{pos/\equiv})$ where for $\phi, \psi \in Pos^d$,

$$\phi \equiv \psi \Leftrightarrow \gamma^m_{pos}(\phi) = \gamma^m_{pos}(\psi), \quad \alpha^m_{pos/\equiv}(c) = \{\phi \mid \phi \equiv \alpha^m_{pos}(c)\}.$$

Let $\Gamma \subseteq Pos^d$ be defined such that

$$\Gamma = \{x \leftrightarrow \bigwedge\{y_1, \ldots, y_n\} \mid \forall 1 \leq i \leq n. x \neq y_i\}.$$

By inspecting the definition of α^m_{pos} (and noting that Sub is the set of idempotent substitutions, i.e. $\theta \in Sub$ implies $x \notin var(\theta(x))$ for all x), it can be seen that $Pos^d/\equiv$ is the lattice $\Lambda \subseteq Pos^d$ where Λ is the closure of Γ under conjunction. From Lemma 3 we obtain:

Theorem 2. $Pos^d/\equiv$ is more precise than D_{gr}.

Thus the precision ordering has been preserved for the mirror properties.

4.2 *Sharing*

We define *Sharing* as in [1]. We define the *set sharing* domain $SH = \wp(SG)$ where $SG = \{S \subseteq \wp(X) \mid \emptyset \notin S\}$. SH is partially ordered by set inclusion such that the join is given by set union and the meet by set intersection.

Let C_L be the concrete domain defined in Example 2. The set of variables occurring in a substitution θ through the variable v is given by the mapping $occs : Sub \times X \rightarrow \wp(X)$ defined such that

$$occs(\theta, x) = \{y \in X \mid x \in var(\theta(y))\}.$$

Given this, the Galois insertion $(C_L, \alpha_{sh}, SH, \gamma_{sh})$ specifying SH can be defined such that

$$\alpha_{sh}(\Theta) = \bigcup_{\theta \in \Theta} \{occs(\theta, x) \mid x \in Vars, occs(\theta, x) \neq \emptyset\}.$$

Note that α_{sh} is the SH-lattice property of the *JI* property $p_{sh} : Sub \rightarrow SH$ defined such that

$$p_{sh}(\theta) = \{occs(\theta, x) \mid x \in Vars, occs(\theta, x) \neq \emptyset\}.$$

For *Sharing*, the abstract unification function is defined as a mapping which captures the effects of a binding $x \rightarrow t$ on an element of SH. The definition uses the following three operations defined over SH.

The function $bin : SH \times SH \to SH$, called *binary union* is given by

$$bin(S_1, S_2) = \{s_1 \cup s_2 \mid s_1 \in S_1, s_2 \in S_2\}.$$

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

$$S^* = \{s \in SG \mid \exists S' \subseteq S. s = \bigcup S'\}.$$

The *relevant component* function $rel : \wp(X) \times SH \to SH$ is given by

$$rel(V, S) = \{s \in S \mid s \cap V \neq \emptyset\}.$$

Let $v_x = \{x\}$, $v_t = var(t)$ and $v_{xt} = v_x \cup v_t$. Then

$$Unif^{sh}(S, x \to t) = (S \setminus (rel(v_{xt}, S)) \cup bin(rel(v_x, S)^*, rel(v_t, S)^*).$$

A domain for pair sharing is $PS = \wp(Pairs(X))$ where $Pairs(X) = \{\{x, y\} \mid x, y \in X,\ x \neq y\}$. PS is specified by the Galois insertion $(C_L, \alpha_{ps}, PS, \gamma_{ps})$, where

$$\alpha_{ps}(\Theta) = \bigcup_{\theta \in \Theta} \{\{x, y\} \in Pairs(X) \mid var(\theta(x)) \cap var(\theta(y)) \neq \emptyset\}.$$

Note that α_{ps} is the PS-lattice property of the *JI* property $p_{ps} : Sub \to PS$ defined such that

$$p_{ps}(\theta) = \{\{x, y\} \in Pairs(X) \mid var(\theta(x)) \cap var(\theta(y)) \neq \emptyset\}.$$

Defining $\alpha_{sp} : SH \to PS$ such that

$$\alpha_{sp}(S) = \bigcup \{Pairs(s) \mid s \in S\},$$

it follows that $\alpha_{ps}(\Theta) = \alpha_{sp}(\alpha_{sh}(\Theta))$ for all $\Theta \in C_L$. Also $\alpha_{sp}(S_1 \cup S_2) = \bigcup \{Pairs(s) \mid s \in S_1 \cup S_2\} = \alpha_{sp}(S_1) \cup \alpha_{sp}(S_2)$. Therefore α_{sp} is additive and so there exists γ_{sp} such that $(SH, \alpha_{sp}, PS, \gamma_{sp})$ forms a Galois connection. It follows that $(C_L, \alpha_{ps}, PS, \gamma_{ps})$ is the composition of $(C_L, \alpha_{sh}, SH, \gamma_{sh})$ and $(SH, \alpha_{sp}, PS, \gamma_{sp})$, and so PS is more abstract than SH.

The mirror property of α_{sh} is $\alpha^m_{sh} : C_L \to SH^d$ defined such that

$$\alpha^m_{sh}(\Theta) = \bigcap_{\theta \in \Theta} \{occs(\theta, x) \mid x \in Vars, occs(\theta, x) \neq \emptyset\}.$$

Lemma 5. There exists γ^m_{sh} such that $(C_L, \alpha^m_{sh}, SH^d, \gamma^m_{sh})$ forms a Galois insertion.

Proof. By Theorem 1, there exists γ^m_{sh} such that $(C_L, \alpha^m_{sh}, SH^d, \gamma^m_{sh})$ forms a Galois connection. To prove α^m_{sh} is onto, we show $\forall a \in SH^d. \exists \theta \in Sub. \alpha^m_{sh}(\{\theta\}) = a$ by induction on $Card(a)$.

The base case is when $a = \emptyset$. Let $\theta = \{x \to t \mid x \in X\}$ where t is a ground term. Then $\alpha^m_{sh}(\{\theta\}) = \emptyset$.

Suppose $\exists s \in a$ and let $a' = a \setminus \{s\}$. Using the induction hypothesis, $\exists \theta' \in Sub.\alpha^m_{sh}(\{\theta'\}) = a'$. Let $u \in Vars \setminus X$ be a variable such that $u \notin var(\theta'(x))$ for any $x \in X$. For every $y \in s$, suppose $\theta'(y) = t'_y$. Let t_y be a term such that $var(t_y) = var(t'_y) \cup \{u\}$. Then defining θ such that $\theta(x) = t_x$ for all $x \in s$ and $\theta(x) = \theta'(x)$ otherwise, $\alpha^m_{sh}(\{\theta\}) = a$. □

The mirror property of α_{ps} is $\alpha^m_{ps} : C_L \to PS^d$ defined such that

$$\alpha^m_{ps}(\Theta) = \bigcap_{\theta \in \Theta} \{\{x, y\} \in Pairs(X) \mid var(\theta(x)) \cap var(\theta(y)) \neq \emptyset\}.$$

Lemma 6. There exists γ^m_{ps} such that $(C_L, \alpha^m_{ps}, PS^d, \gamma^m_{ps})$ forms a Galois insertion.

Proof. By Theorem 1, there exists γ^m_{ps} such that $(C_L, \alpha^m_{ps}, PS^d, \gamma^m_{ps})$ forms a Galois connection. We show that α^m_{ps} is onto.

First suppose $a = \top_{ps} = Pairs(X)$. Let $u \in Vars \setminus X$. Then if $\theta(x) = u$ for every $x \in X$, $\alpha^m_{ps}(\{\theta\}) = Pairs(X)$ as required.

Suppose $a \neq \top_{ps}$. PS is dual-atomistic with $atom_{ps^d} = \{Pairs(X) \setminus \{\{x, y\}\} \mid \{x, y\} \in PS\}$. Therefore for every $a \neq \top_{ps}$, $a = \bigcap\{x \mid x \in atom_{ps^d} \wedge a \subseteq x\}$. But $\alpha^m_{ps}(\Theta) = \bigcap\{p_{ps}(\theta) \mid \theta \in \Theta\}$, and so it is sufficient to show that $\forall a \in atom_{ps^d}.\exists \theta \in Sub.p_{ps}(\theta) = a$.

Suppose $a = Pairs(X) \setminus \{\{x, y\}\}$ and let $u, v \in Vars \setminus X$. Defining θ such that $\theta(x) = u, \theta(y) = v$ and $\theta(z) = f(u, v)$ for every $z \in X \setminus \{x, y\}$, $p_{ps}(\theta) = a$. □

Theorem 3. If $Card(X) \geq 3$ then SH^m is not more precise than PS^m.

Proof. We need to show there exists $\Theta \in C_L$ such that $\gamma^m_{sh}(\alpha^m_{sh}(\Theta)) \not\subseteq \gamma^m_{ps}(\alpha^m_{ps}(\Theta))$.

Suppose $X = \{x, y, z\}$ (it is easy to generalise the proof for $Card(X) > 3$). Let $\Theta = \{\theta_1, \theta_2\}$ where $\theta_1 = \{x \to y, z \to y\}$ and $\theta_2 = \{x \to y\}$. It follows that $\gamma^m_{sh}(\alpha^m_{sh}(\{\theta_1, \theta_2\})) = \gamma^m_{sh}(\{\{x, y, x\}\} \cap \{\{x, y\}\}) = \gamma^m_{sh}(\emptyset) = Sub$. But $\gamma^m_{ps}(\alpha^m_{ps}(\{\theta_1, \theta_2\})) = \gamma^m_{ps}(\{\{x, y\}\}) \subset Sub$. Therefore $\gamma^m_{sh}(\alpha^m_{sh}(\Theta)) \not\subseteq \gamma^m_{ps}(\alpha^m_{ps}(\Theta))$. □

Thus in general the precision ordering is not preserved for mirror properties.

Theorem 4. PS^m is not more precise than SH^m.

Proof. We need to show there exists $\Theta \in C_L$ such that $\gamma^m_{ps}(\alpha^m_{ps}(\Theta)) \not\subseteq \gamma^m_{sh}(\alpha^m_{sh}(\Theta))$.

Let $\Theta = \{\epsilon\}$ where ϵ is the identity substitution. Now $\gamma^m_{sh}(\alpha^m_{sh}(\{\epsilon\})) = \gamma^m_{sh}(\{\{x\} \mid x \in X\}) \subset Sub$ and $\gamma^m_{ps}(\alpha^m_{ps}(\{\epsilon\})) = \gamma^m_{ps}(\emptyset) = Sub$. Therefore $\gamma^m_{ps}(\alpha^m_{ps}(\Theta')) \not\subseteq \gamma^m_{sh}(\alpha^m_{sh}(\Theta'))$. □

Hence the precision of SH^m and PS^m is not comparable in general.

5 Operations on Concrete Domains

When the concrete lattice C is join-generated by $JI(C)$, many operations on C can be defined in terms of operations on $JI(C)$.

Definition 3. Suppose C is join-generated by $JI(C)$. Then op is a JI operation if $op : JI(C) \times JI(C) \to JI(C)$[1]. For each concrete operation $Op : C \times C \to C$, we say Op is uniformly defined from a JI operation op if for all $c_1, c_2 \in C$,

$$Op(c_1, c_2) = \bigsqcup_C \{op(x_1, x_2) \mid x_1, x_2 \in JI(C) \wedge x_1 \sqsubseteq_C c_1 \wedge x_2 \sqsubseteq_C c_2\}.$$

Example 4. In logic programming, unification and projection can both be defined as JI operations $unif : Sub \times Sub \to Sub$, $proj_V : Sub \to Sub$ (for $V \subseteq Vars$) as follows:

$$unif(\theta_1, \theta_2) = mgu(eqn(\theta_1), eqn(\theta_2)),$$

$$proj_V(\theta) = \theta' \text{ where for each } x \in Vars,\ \theta'(x) = \begin{cases} \theta(x) & \text{if } x \in V \\ x & \text{otherwise} \end{cases}$$

where $eqn(\theta) = \{x = t \mid x \to t \in \theta\}$.

The concrete operations $Unif : C_L \times C_L \to C_L$ and $Proj_V : C_L \to C_L$ can be uniformly defined from $unif$ and $proj$ as follows:

$$Unif(\Theta_1, \Theta_2) = \bigcup \{unif(\theta_1, \theta_2) \mid \theta_1 \in \Theta_1 \wedge \theta_2 \in \Theta_2\},$$

$$Proj_V(\Theta) = \bigcup \{proj_V(\theta) \mid \theta \in \Theta\}. \ \square$$

Given an abstract operation Op_D, we show that if $\langle D, Op_D \rangle$ is a complete (and therefore also correct) abstract interpretation of $\langle C, Op \rangle$, then $\langle D, Op_D^d \rangle$ is a correct abstract interpretation of $\langle C, Op \rangle$.

Lemma 7. Suppose C, D are complete lattices and C is join-generated by $JI(C)$. Let $Op : C \times C \to C$ be a concrete operation uniformly defined from the JI operation $op : JI(C) \times JI(C) \to JI(C)$. Let $\langle D, Op_D \rangle$ be a complete abstract interpretation of Op specified by (C, α, D, γ). Then $\langle D^d, Op_D \rangle$ is a correct abstract interpretation of $\langle C, Op \rangle$ specified by $(C, \alpha^m, D^d, \gamma^m)$.

Proof. We need to show that $Op(\gamma^m(d_1), \gamma^m(d_2)) \sqsubseteq_C \gamma^m(Op_D(d_1, d_2))$ for all $d_1, d_2 \in D$.

Note that from Definition 3 it follows that Op is monotonic, i.e. if $c_1 \sqsubseteq_C c_1'$ and $c_2 \sqsubseteq_C c_2'$ then $Op(c_1, c_2) \sqsubseteq_C Op(c_1', c_2')$. Since $\langle D, Op_D \rangle$ is complete, $Op_D = \alpha \circ Op \circ \gamma$. Hence since Op, α, γ are all monotonic, Op_D is also monotonic. Now

[1] Note that to simplify the notation we assume that a JI operation has at most two input arguments. The results presented can easily be extended to operations with any number of arguments.

$Op(\gamma^m(d_1), \gamma^m(d_2)) =$
$\bigsqcup_C \{op(x_1, x_2) \mid x_1, x_2 \in JI(C) \wedge x_1 \sqsubseteq_C \gamma^m(d_1) \wedge x_2 \sqsubseteq_C \gamma^m(d_2)\}$.

Therefore it is sufficient to show that $op(x_1, x_2) \sqsubseteq_C \gamma^m(Op_D(d_1, d_2))$ for all $x_1, x_2 \in JI(C)$ such that $x_1 \sqsubseteq_C \gamma^m(d_1)$ and $x_2 \sqsubseteq_C \gamma^m(d_2)$. Now $x_1 \sqsubseteq_C \gamma^m(d_1)$ implies $\alpha^m(x_1) \sqsubseteq_D^d d_1$ and $x_2 \sqsubseteq_C \gamma^m(d_2)$ implies $\alpha^m(x_2) \sqsubseteq_D^d d_2$. Hence since Op_D is monotonic,

$$Op_D(\alpha^m(x_1), \alpha^m(x_2)) \sqsubseteq_D^d Op_D(d_1, d_2).$$

But $x_1, x_2 \in JI(C)$, thus $Op_D(\alpha^m(x_1), \alpha^m(x_2)) = Op_D(\alpha(x_1), \alpha(x_2))$. Since Op_D is complete,

$$Op_D(\alpha(x_1), \alpha(x_2)) = \alpha(Op(x_1, x_2)) = \alpha(op(x_1, x_2)).$$

By Definition 3, $op(x_1, x_2) \in JI(C)$ and so $\alpha(op(x_1, x_2)) = \alpha^m(op(x_1, x_2))$. Thus $\alpha^m(op(x_1, x_2)) \sqsubseteq_D^d Op_D(d_1, d_2)$ and so $op(x_1, x_2) \sqsubseteq_C \gamma^m(Op_D(d_1, d_2))$. □

Example 5. The abstract projection function for *Pos*, $Proj_V^{pos} : Pos \rightarrow Pos$, amounts to existentially quantifying a formula (see [6] for details). It is shown that $\langle Pos, Proj_V^{pos} \rangle$ is complete in Lemma 36 [6][2]. Therefore by Lemma 7, $\langle Pos^d, Proj_V^{pos} \rangle$ is a correct abstract interpretation of $\langle C_L, Proj_V \rangle$.

The abstract projection function for *Sharing*, $Proj_V^{sh} : SH \rightarrow SH$, is defined such that

$$Proj_V^{sh}(S) = \{s \cap V \mid s \in S\}$$

Theorem 5.2 [5] shows that $\langle SH, Proj_V^{sh} \rangle$ is complete. Therefore by Lemma 7, $\langle SH^d, Proj_V^{sh} \rangle$ is a correct abstract interpretation of $\langle C_L, Proj_V \rangle$.

On the other hand, [6] shows that $\langle Pos, Unif^{pos} \rangle$ is not complete and [5] shows that $\langle SH, Unif^{sh} \rangle$ is not complete. □

In fact, it can be shown that both $\langle Pos^d, Unif^{pos} \rangle$ and $\langle SH^d, Unif^{sh} \rangle$ are not correct abstract interpretations of $\langle C_L, Unif \rangle$.

Lemma 8. $\langle Pos^d, Unif^{pos} \rangle$ is not a correct abstract interpretation of $\langle C_L, Unif \rangle$.

Proof. It is sufficient to find $\phi \in Pos^d$ such that

$$Unif^{pos}(\phi, \phi) \not\models \alpha_{pos}^m(Unif(\gamma_{pos}^m(\phi), \gamma_{pos}^m(\phi))).$$

Let ϕ be the formula $x \leftrightarrow y$ and $\theta_1 = \{x \rightarrow f(1, y)\}$ and $\theta_2 = \{x \rightarrow f(y, 1)\}$. Note that $\theta_1, \theta_2 \in \gamma_{pos}^m(\phi)$. Now $unif(\theta_1, \theta_2) = \{x \rightarrow f(1, 1), y \rightarrow 1\}$ and so it follows that

$$\alpha_{pos}^m(Unif(\gamma_{pos}^m(\phi), \gamma_{pos}^m(\phi))) \models x \wedge y.$$

But $Unif^{pos}(\phi, \phi) = \phi$ and so $Unif^{pos}(\phi, \phi) \not\models \alpha_{pos}^m(Unif(\gamma_{pos}^m(\phi), \gamma_{pos}^m(\phi)))$, as required. □

[2] Note that in [6] and [5], *Pos* and *Sharing* are formulated differently from our presentation. In [6] and [5], however, it is evident that the proofs can be adapted.

Lemma 9. $\langle SH^d, Unif^{sh}\rangle$ is not a correct abstract interpretation of $\langle C_L, Unif\rangle$.

Proof. It is sufficient to find $S \in SH^d$ and a binding $x \to t$ such that

$$Unif^{sh}(S, x \to t) \not\subseteq \alpha^m_{sh}(Unif(\gamma^m_{sh}(S), \{\{x \to t\}\})).$$

Let $S = \{\{x, y\}\}$, $t = f(1, y)$ and $\theta = \{x \to f(y, 1)\}$. Note that $\theta \in \gamma^m_{sh}(S)$. Now $unif(\theta, \{x \to t\}) = \{x \to f(1, 1), y \to 1\}$ and so it follows that

$$\alpha^m_{sh}(Unif(\gamma^m_{sh}(S), \{\{x \to t\}\}) = \emptyset.$$

But $Unif^{sh}(S, x \to t) = \{\{x, y\}, \{x\}, \{y\}\}$ and so the result follows. □

Hence new abstract unification operations need to be devised for both Pos^d and SH^d.

6 Conclusion

We have shown how, given an abstract domain D specifying a lattice property α_p, an abstract domain D^d specifying the mirror property α^m_p can be constructed. We have also shown that if $\langle D, Op_D\rangle$ is a complete abstract interpretation of $\langle C, Op_C\rangle$, then $\langle D^d, Op_D\rangle$ is a correct abstract interpretation of $\langle C, Op_C\rangle$.

There are instances when non-complete abstract operations computing a property can be used to improve the precision of operations computing the mirror property. For example, formulae of the form $x \to y$ in Pos are interpreted as meaning "x ground implies y ground". The contrapositive of this is "y non-ground implies x non-ground". Thus this information could be used to improve the precision of a Pos^d analysis. In fact, since non-groundness information is approximated by freeness information, it would seem reasonable to implement Pos^d as a reduced product construction with Pos and a domain expressing freeness information. It would be interesting to see if generalisations of this method could be meaningfully applied to other domains. Another direction for future work is to see how our approach relates to lower/upper approximations used in concurrency [17].

Acknowledgments

We thank the anonymous referees for their useful comments. This work was supported by EPSRC Grant GR/M05645.

References

1. R. Bagnara, P. M. Hill, and E. Zaffanella. Set-sharing is redundant for pair-sharing. In P. Van Hentenryck, editor, *Static Analysis: Proceedings of the 4th International Symposium*, volume 1302, pages 53–67, Paris, France, 1997.

2. R. Balbes and P .Dwinger. *Distributive Lattices.* University of Missouri Press, Columbia, Missouri, 1974.
3. G. Birkhoff. *Lattice Theory.* AMS Colloquium Publication, Providence, RI, 3rd edition, 1967.
4. M. Codish, A. Mulkers, M. Bruynooghe, M. García de la Banda, and M. Hermenegildo. Improving abstract interpretations by combining domains. *ACM Transactions on Programming Languages and Systems*, 17(1):28–44, 1995.
5. A. Cortesi and G. Filè. Sharing is Optimal. *Journal of Logic Programming*, 38(3):371–386, 1999.
6. A. Cortesi, G. Filè, and W. Winsborough. Optimal Groundness Analysis Using Propositional Logic. *Journal of Logic Programming*, 27(2):137–167, 1996.
7. P. Cousot and R. Cousot. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In *Proc. Fourth ACM Symp. Principles of Programming Languages*, pages 238–252, 1977.
8. P. Cousot and R. Cousot. Systematic Design of Program Analysis Frameworks. In *Proc. Sixth ACM Symp. Principles of Programming Languages*, pages 269–282, 1979.
9. P. Cousot and R. Cousot. Abstract Interpretation and Applications to Logic Programs. *Journal of Logic Programming*, 13(2 & 3):103–179, 1992.
10. R. Giacobazzi and F. Ranzato. Refining and Compressing Abstract Domains. In *Proceedings of the 24th International Colloquium on Automata, Languages and Programming ICALP 97*, volume 1256 of *Lecture Notes in Computer Science*, pages 771–781. Springer-Verlag, 1997.
11. R. Giacobazzi and .F Ranzato. Optimal Domains for Disjunctive Abstract Interpretation. *Science of Computer Programming*, 32:177–210, 1998.
12. R. Giacobazzi, F. Ranzato, and F. Scozzari. Making Abstract Interpretations Complete. *Journal of the ACM.* (to appear).
13. R. Giacobazzi, F. Ranzato, and F. Scozzari. Building Complete Abstract Interpretations in a Linear Logic-based Setting. In G. Levi, editor, *Static Analysis, Proceedings of the Fifth International Static Analysis Symposium SAS 98*, volume 1503 of *Lecture Notes in Computer Science*, pages 215–229. Springer-Verlag, 1998.
14. R. Giacobazzi and F. Scozzari. A Logical Model for Relational Abstract Domains. *ACM Transactions on Programming Languages and Systems*, 20(5):1067–1109, 1998.
15. P. Hill and F. Spoto. Freeness Analysis through Linear Refinement. In *Static Analysis: Proceedings of the 6th International Symposium*, volume 1694, pages 85–100, 1999.
16. N.D. Jones and H. Søndergaard. A Semantics-based Framework for the Abstract Interpretation of Prolog. In S. Abramsky and C. Hankin, editors, *Abstract Interpretation of Declarative Languages*, pages 123–142. Ellis Horwood Ltd, 1987.
17. F. Levi. A Symbolic Semantics for Abstract Model Checking. In *Static Analysis: Proceedings of the 5th International Symposium*, volume 1503, pages 134–151, 1998.
18. F. Scozzari. Logical Optimality of Groundness Analysis. In P. Van Hentenryck, editor, *Proceedings of International Static Analysis Symposium, SAS'97*, volume 1302 of *Lecture Notes in Computer Science*, pages 83–97. Springer-Verlag, 1997.