Building Complete Abstract Interpretations in a Linear Logic-based Setting

Roberto Giacobazzi[†] Francesco Ranzato[‡,⋆] Francesca Scozzari[†]

[†]*Dipartimento di Informatica, Università di Pisa*
Corso Italia 40, 56125 Pisa, Italy
{giaco,scozzari}@di.unipi.it

[‡]*Dipartimento di Matematica Pura ed Applicata, Università di Padova*
Via Belzoni 7, 35131 Padova, Italy
franz@math.unipd.it

Abstract. Completeness is an important, but rather uncommon, property of abstract interpretations, ensuring that abstract computations are as precise as possible w.r.t. concrete ones. It turns out that completeness for an abstract interpretation depends only on its underlying abstract domains, and therefore it is an abstract domain property. Recently, the first two authors proved that for a given abstract domain A, in all significant cases, there exists the most abstract domain, called least complete extension of A, which includes A and induces a complete abstract interpretation. In addition to the standard formulation, we introduce and study a novel and particularly interesting type of completeness, called observation completeness. Standard and observation completeness are here considered in the context of quantales, i.e. models of linear logic, as concrete interpretations. In this setting, we prove that various kinds of least complete and observationally complete extensions exist and, more importantly, we show that such complete extensions can all be explicitly characterized by elegant linear logic-based formulations. As an application, we determine the least complete extension of a generic abstract domain w.r.t. a standard bottom-up semantics for logic programs observing computed answer substitutions. This general result is then instantiated to the relevant case of groundness analysis.

1 Introduction

It is widely held that the ideal goal of any semantics design method is to find sound and complete representations for some properties of concrete (actual) computations. Abstract interpretation is one such methodology, where soundness is always required, while completeness more rarely holds. Completeness issues in abstract interpretation have been studied since the Cousot and Cousot seminal paper [5]. The intuition is that a complete abstract interpretation induces an

⋆ The work of Francesco Ranzato has been supported by an individual post-doctoral grant from Università di Padova, Italy.

abstract semantics which is as precise as possible relatively to its underlying abstract domains and to the concrete interpretation of reference. The paradigmatic rule of signs is a typical and simple example of an abstract interpretation which is (sound and) complete for integer multiplication but merely sound for integer addition [13].

Although in static program analysis decidability issues often force to sacrifice completeness for achieving termination and/or efficiency, examples of complete abstract interpretations are common in other fields of application. For instance, several complete abstractions of algebraic polynomial systems have been studied by Cousot and Cousot in [7] and many complete abstract interpretations can be found in comparative program semantics [3,6,9]. Moreover, being completeness a notion relative to the concrete semantics of reference, complete abstract interpretations which are more concrete than a certain, possibly approximated and decidable, property of interest, yield an absolute upper bound for the precision that one can achieve in computing that property. Thus, complete abstract interpretations may play a useful rôle in static program analysis as well. These argumentations probably stimulated the recent trend of research on completeness in abstract interpretation [2,10,11,13,15,17,18].

One key feature of completeness in abstract interpretation is that this property uniquely depends upon the abstraction function. Let us denote by $\mathcal{L}_C$ the so-called lattice of abstract interpretations of a concrete domain C (cf. [4,5]), where, for all $A, B \in \mathcal{L}_C$, $A \sqsubseteq B$ means that A is more precise (i.e. concrete) than B. Let us consider the simple case of an abstract interpretation $f^\sharp : A \to A$ of a concrete semantic function $f : C \to C$, where the abstract domain $A \in \mathcal{L}_C$ is related to C by an adjoint pair of abstraction and concretization maps $\alpha : C \to A$ and $\gamma : A \to C$. Then, $f^\sharp$ is (sound and) complete if $\alpha \circ f = f^\sharp \circ \alpha$. It is easily seen that if $f^\sharp$ is complete then the best correct approximation f^b of f in A, i.e. $f^b \stackrel{\text{def}}{=} \alpha \circ f \circ \gamma : A \to A$, is complete as well, and, in this case, $f^\sharp$ indeed coincides with f^b (cf. [10]). Thus, given an abstract domain $A \in \mathcal{L}_C$, one can define a complete abstract semantic function $f^\sharp : A \to A$ over A if and only if $f^b : A \to A$ is complete. This simple observation makes completeness an abstract domain property, namely a characteristic of the abstract domain. It is then clear that a key problem consists in devising systematic and constructive methodologies for transforming abstract domains in such a way that completeness is achieved and the resulting complete abstract domains are as close as possible to the initial (noncomplete) ones. This problem has been first raised in a predicate-based approach to abstract interpretation by Mycroft [13, Section 3.2], who gave a methodology for deriving the most concrete domain, called "canonical abstract interpretation", which is complete and included in a given domain of properties. More recently, the first two authors proved in [10] that when the concrete semantic function f is continuous, any domain A can always be extended into the most abstract domain which includes A and is complete for f — the so-called *least complete extension* of A. Analogously, [10] solved the aforementioned problem raised by Mycroft, by showing that, for any given domain A, the most concrete domain which is complete and included in A — the so-called *complete kernel* of A — exists for any monotone semantic function. Very recently, we improved considerably such results, by providing explicit constructive characterizations for least complete extensions and complete kernels of abstract domains [11].

In this paper, we are concerned with completeness problems arising when concrete semantic binary[1] operations of type $C_1 \times C_2 \to C$ are assumed to give rise to a generalized form of quantales, called *typed quantales*. Quantales are well-known algebraic structures which turn out to be models of intuitionistic linear logic [16,19]. In this logical setting, we provide elegant linear logic-based solutions to a number of interesting completeness problems for abstract interpretations. Such solutions find relevant applications in static program analysis and comparative semantics, for instance in logic programming, where unification — the prime computational step of any logic program semantics — turns out to be a binary operation in a quantale of substitutions, or in data structure analysis, considering binary data constructors such as *cons* for lists. More in detail, a typed quantale $\langle C, C_1, C_2, \otimes\rangle$ consists of three complete lattices C, C_1 and C_2, and of an operation $\otimes : C_1 \times C_2 \to C$ which is additive (i.e., preserves lub's) on both arguments. When $C = C_1 = C_2$, typed quantales boil down to standard quantales $\langle C, \otimes\rangle$. The main feature of (typed) quantales is that they support a notion of left and right linear implication between domain's objects: Given $a \in C_1$ and $b \in C$, there exists a unique greatest object $a \multimap b \in C_2$ which, when combined by $\otimes$ with a, gives a result less than or equal to b. In other terms, the following right modus ponens law $a \otimes x \leq b \;\Leftrightarrow\; x \leq a \multimap b$ holds. Analogous left implicational objects exist for a correspoding left modus ponens law.

When solving completeness problems in a setting where concrete interpretations are typed quantales, implicational domain objects allow to elegantly characterize complete abstract domains in a variety of situations. Our first result provides a characterization based on linear implications between domain's objects of the least complete extension of any abstract domain of any quantale. Then, we consider the following completeness problem over typed quantales: Given a typed quantale $\langle C, C_1, C_2, \otimes\rangle$, a fixed abstraction $A \in \mathcal{L}_C$, with corresponding abstraction map $\alpha_A : C \to A$, and a pair of abstract domains $\langle A_1, A_2\rangle \in \mathcal{L}_{C_1} \times \mathcal{L}_{C_2}$, does there exist the most abstract pair of domains $\langle A'_1, A'_2\rangle \in \mathcal{L}_{C_1} \times \mathcal{L}_{C_2}$, with corresponding abstraction maps $\alpha_{A_i} : C_i \to A_i$ $(i = 1, 2)$, such that $\langle A'_1, A'_2\rangle \sqsubseteq_{\mathcal{L}_{C_1} \times \mathcal{L}_{C_2}} \langle A_1, A_2\rangle$ and $\alpha_A(\_ \otimes \_) = \alpha_A(\alpha_{A'_1}(\_) \otimes \alpha_{A'_2}(\_))$? Here, the *observation* domain A is fixed, and we are thus looking for the most abstract pair of domains in $\mathcal{L}_{C_1} \times \mathcal{L}_{C_2}$ which is more concrete than an initial pair $\langle A_1, A_2\rangle$ and simultaneously induces a complete abstract interpretation w.r.t. $\otimes$. This is termed an *observation completeness* problem. Again, solutions to this observation completeness problem are built in terms of linear implications between domains.

To illustrate the practical scope of our results, we first consider a simple example in data structure analysis involving abstract domains for lists. In particular, our results are applied in order to solve various observation completeness problems concerning abstract domains useful for detecting irredundant lists of

[1] Clearly, a generalization to n-ary operations would be straightforward.

objects. Then, in the context of logic program semantics, we consider an immediate consequences operator T_P defined in terms of unification and union of sets of idempotent substitutions, and characterizing computed answer substitutions in a s-semantics style (cf. [1,8]). As usual, unification turns out to be the key operation to take into account in order to build least complete extensions of abstract domains. Sets of idempotent substitutions and unification give rise to a unital commutative quantale: Given an abstract domain A, we show how the least complete extension of A w.r.t. this quantale naturally induces the least complete extension of A w.r.t. T_P functions. This permits to give explicitly, in terms of linear implications, the least complete extension, for any T_P, of a generic domain abstracting sets of substitutions. As a remarkable instance of our construction, we characterize the least complete extension of the plain groundness domain w.r.t. computed answer substitutions s-semantics.

2 Basic notions

The lattice of abstract interpretations. In standard Cousot and Cousot's abstract interpretation theory, abstract domains can be equivalently specified either by Galois connections (GCs), i.e. adjunctions, or by upper closure operators (uco's) [5]. In the first case, the concrete and abstract domains C and A (both assumed to be complete lattices) are related by a pair of adjoint functions of a GC (α, C, A, γ). Also, it is generally assumed that (α, C, A, γ) is a Galois insertion (GI), i.e. α is onto or, equivalently, γ is 1-1. In the second case instead, an abstract domain is specified as a uco on the concrete domain C, i.e. a monotone, idempotent and extensive operator on C. These two approaches are equivalent, modulo isomorphic representation of domain's objects. Given a complete lattice C, it is well known that the set $uco(C)$ of all uco's on C, endowed with the pointwise ordering $\sqsubseteq$, is a complete lattice $\langle uco(C), \sqsubseteq, \sqcup, \sqcap, \lambda x.\top_C, id\rangle$ (id denotes the identity function). Let us also recall that each $\rho \in uco(C)$ is uniquely determined by the set of its fixpoints, which is its image, i.e. $\rho(C) = \{x \in C \mid \rho(x) = x\}$, and that $\rho \sqsubseteq \eta$ iff $\eta(C) \subseteq \rho(C)$. Moreover, a subset $X \subseteq C$ is the set of fixpoints of a uco on C iff X is meet-closed, i.e. $X = \curlywedge(X) \stackrel{\text{def}}{=} \{\bigwedge_C Y \mid Y \subseteq X\}$ (note that $\top_C = \bigwedge_C \varnothing \in \curlywedge(X)$). Often, we will identify closures with their sets of fixpoints. This does not give rise to ambiguity, since one can distinguish their use as functions or sets according to the context. In view of the equivalence above, throughout the paper, $\langle uco(C), \sqsubseteq\rangle$ will play the rôle of the lattice $\mathcal{L}_C$ of abstract interpretations of C [4,5], i.e. the complete lattice of all possible abstract domains of the concrete domain C. For an abstract domain $A \in \mathcal{L}_C$, $\rho_A \in uco(C)$ will denote the corresponding uco on C, and if A is specified by a GI (α, C, A, γ) then $\rho_A \stackrel{\text{def}}{=} \gamma \circ \alpha$. The ordering on $uco(C)$ corresponds to the standard order used to compare abstract domains with regard to their precision: A_1 is more precise than A_2 (i.e., A_1 is more concrete than A_2 or A_2 is more abstract than A_1) iff $A_1 \sqsubseteq A_2$ in $uco(C)$. Lub and glb on $uco(C)$ have therefore the following reading as operators on domains. Suppose $\{A_i\}_{i \in I} \subseteq uco(C)$: (i) $\sqcup_{i \in I} A_i$ is the most concrete among the domains which are abstractions of all the A_i's, i.e. it is their least (w.r.t. $\sqsubseteq$) common abstraction; (ii) $\sqcap_{i\in I}A_i$ is the most abstract among the domains (abstracting C) which are more concrete than every A_i; this domain is also known as reduced product of all the A_i's.

Quantales and linear logic. Quantales originated as algebraic foundations of the so-called quantum logic. They have been successively considered for the lattice-theoretic semantics of Girard's linear logic (see [16] for an exhaustive treatment of quantales). We introduce a mild generalization of the notion of quantale, which, up to knowledge, appears to be new. A *typed quantale* is a multisorted algebra $\langle C, C_1, C_2, \otimes\rangle$, where C, C_1, C_2 are complete lattices and $\otimes : C_1 \times C_2 \to C$ is a function such that $(\bigvee_i x_i) \otimes c_2 = \bigvee_i (x_i \otimes c_2)$ and $c_1 \otimes (\bigvee_i x_i) = \bigvee_i (c_1 \otimes x_i)$. In other terms, a typed quantale is a 3-sorted algebra endowed with a "product" $\otimes$ which distributes over arbitrary lub's on both sides. Thus, for any $c_1 \in C_1$ and $c_2 \in C_2$, both functions $c_1 \otimes \_$ and $\_ \otimes c_2$ have right adjoints denoted, resp., by $c_1 \multimap \_$ and $\_ \multimapinv c_2$. Hence, for all $c \in C$, $c_1 \otimes c_2 \leq c \Leftrightarrow c_2 \leq c_1 \multimap c$, and, dually, $c_1 \otimes c_2 \leq c \Leftrightarrow c_1 \leq c \multimapinv c_2$. Two functions $\multimap : C_1 \times C \to C_2$ and $\multimapinv : C \times C_2 \to C_1$ can be therefore defined as follows:

$$c_1 \multimap c \stackrel{\text{def}}{=} \bigvee \{z \in C_2 \mid c_1 \otimes z \leq c\}; \qquad c \multimapinv c_2 \stackrel{\text{def}}{=} \bigvee \{y \in C_1 \mid y \otimes c_2 \leq c\}.$$

Any typed quantale $\langle C, C_1, C_2, \otimes\rangle$ enjoys the following main properties: For all $c \in C$, $\{x_i\}_{i\in I} \subseteq C$, $c_1 \in C_1$, $\{y_i\}_{i\in I} \subseteq C_1$, $c_2 \in C_2$ and $\{z_i\}_{i\in I} \subseteq C_2$:

$$\begin{array}{llll}
\text{(i)} & c_1 \otimes (c_1 \multimap c) \leq c & \text{(ii)} & (c \multimapinv c_2) \otimes c_2 \leq c \\
\text{(iii)} & c_1 \multimap (\bigwedge_{i\in I} x_i) = \bigwedge_{i\in I} (c_1 \multimap x_i) & \text{(iv)} & (\bigwedge_{i\in I} x_i) \multimapinv c_2 = \bigwedge_{i\in I} (x_i \multimapinv c_2) \\
\text{(v)} & (\bigvee_{i\in I} y_i) \multimap c = \bigwedge_{i\in I} (y_i \multimap c) & \text{(vi)} & c \multimapinv (\bigvee_{i\in I} z_i) = \bigwedge_{i\in I} (c \multimapinv z_i)
\end{array}$$

When $C = C_1 = C_2$ and $\otimes$ is associative, a typed quantale is called *quantale*. It is well known that quantales turn out to be models of noncommutative intuitionistic linear logic [16,19]. A quantale $\langle C, \otimes\rangle$ is called *commutative* when $\otimes$ is commutative, and this is equivalent to require that, for all $a, b \in C$, $a \multimap b = b \multimapinv a$. Also, a commutative quantale $\langle C, \otimes\rangle$ is called *unital* if there exists an object $1 \in C$ such that $1 \otimes a = a = a \otimes 1$, for all $a \in C$. For a quantale $\langle C, \otimes\rangle$, the following additional properties hold for all $a, b, c \in C$:

$$\begin{array}{rl}
\text{(vii)} & a \multimap (c \multimapinv b) = (a \multimap c) \multimapinv b \\
\text{(viii)} & b \multimap (a \multimap c) = (a \otimes b) \multimap c \\
\text{(ix)} & (c \multimapinv b) \multimapinv a = c \multimapinv (a \otimes b)
\end{array}$$

3 Completeness problems in abstract interpretations

Let $\langle C, C_1, C_2, \otimes\rangle$ be a concrete interpretation, i.e. C, C_1 and C_2 are concrete semantic domains provided with a semantic operation $\otimes : C_1 \times C_2 \to C$. When

$C = C_1 = C_2$, we adopt the simpler notation $\langle C, \otimes \rangle$. Given the abstractions $A_1 \in \mathcal{L}_{C_1}$, $A_2 \in \mathcal{L}_{C_2}$ and $A \in \mathcal{L}_C$, let us recall [5] that the best correct approximation $\otimes^b : A_1 \times A_2 \to A$ of $\otimes$ is defined as $\otimes^b \stackrel{\text{def}}{=} \alpha_{C,A} \circ \otimes \circ \langle \gamma_{A_1,C_1}, \gamma_{A_2,C_2} \rangle$. It has been shown in [10] that completeness for an abstract interpretation is a property depending only on the underlying abstract domains. In our setting, this means that an abstract interpretation $\langle A, A_1, A_2, \otimes^\sharp \rangle$, with $\otimes^\sharp : A_1 \times A_2 \to A$, is complete for $\langle C, C_1, C_2, \otimes \rangle$ iff $\langle A, A_1, A_2, \otimes^b \rangle$ is complete and $\otimes^\sharp = \otimes^b$. In other terms, the best correct approximation induced by the underlying abstract domains determines the property of being complete. Hence, we find more convenient, elegant and, of course, completely equivalent, to reason on completeness by using abstract domains specified by the closure operator approach.

Full completeness problems. Within the closure operator approach, given a concrete interpretation $\langle C, \otimes \rangle$, an abstraction $A \in \mathcal{L}_C$ is complete when the equation $\rho_A \circ \otimes \circ \langle \rho_A, \rho_A \rangle = \rho_A \circ \otimes$ holds. Giacobazzi and Ranzato [10] stated the following *full completeness problem*, here specialized to binary semantic operations: Given an abstract domain $A \in \mathcal{L}_C$, does the following system with variable ρ admit a most abstract solution?

$$\begin{cases} \rho \sqsubseteq A \\ \rho \circ \otimes \circ \langle \rho, \rho \rangle = \rho \circ \otimes \end{cases} \qquad (1)$$

Hence, a solution to the above full completeness problem is the necessarily unique (up to domain isomorphism) most abstract domain which includes A and induces a complete abstract interpretation. Following [10], such most abstract solution to System (1) is called the *least complete extension* of A w.r.t. $\otimes$. It is shown in [10] that if $\otimes$ is continuous in both arguments, then least complete extensions of any A exist.

Full completeness problems clearly make sense only for concrete interpretations of type $\langle C, \otimes \rangle$. When generic concrete interpretations of type $\langle C, C_1, C_2, \otimes \rangle$ are considered, different abstractions are involved in a completeness equation, and therefore various completeness problems arise by fixing some of these abstractions. In the following, we introduce three such completeness problems, which turn out to be of particular interest. Such completeness problems still depend only on best correct approximations of the concrete operation, and therefore the corresponding completeness notions are again abstract domain properties.

Observation completeness problems. Let $\langle C, C_1, C_2, \otimes \rangle$ be a concrete interpretation. An observation domain is any abstraction of the range C of $\otimes$. Observation completeness problems arise when in a completeness equation an observation domain is fixed. Hence, let us consider a fixed observation domain $A \in \mathcal{L}_C$. The *observation completeness problem* for a pair $\langle A_1, A_2 \rangle \in \mathcal{L}_{C_1} \times \mathcal{L}_{C_2}$ admits solution when there exists the most abstract solution in $\mathcal{L}_{C_1} \times \mathcal{L}_{C_2}$ of the following system:

$$\begin{cases} \langle \eta, \mu \rangle \sqsubseteq \langle A_1, A_2 \rangle \\ \rho_A \circ \otimes \circ \langle \eta, \mu \rangle = \rho_A \circ \otimes \end{cases} \qquad (2)$$

Let us remark that, by using adjunctions, the observation completeness equation $\rho_A \circ \otimes \circ \langle \eta, \mu \rangle = \rho_A \circ \otimes$ is equivalent to require that for all $x \in C_1$ and $y \in C_2$, $\alpha_{C_1,\eta}(x) \otimes^b \alpha_{C_2,\mu}(y) = \alpha_{C,A}(x \otimes y)$.

When in addition to the observation domain we also fix one (or more, if we would deal with n-ary operations) of the abstractions of the argument domains of $\otimes$, we obtain yet different completeness problems. In the *left observation completeness problem*, $A \in \mathcal{L}_C$ and $A_2 \in \mathcal{L}_{C_2}$ are fixed, and the solution to this problem for a given $A_1 \in \mathcal{L}_{C_1}$ exists when the following system with variable η admits a most abstract solution in $\mathcal{L}_{C_1}$:

$$\begin{cases} \eta \sqsubseteq A_1 \\ \rho_A \circ \otimes \circ \langle \eta, \rho_{A_2} \rangle = \rho_A \circ \otimes \circ \langle id, \rho_{A_2} \rangle \end{cases} \tag{3}$$

Of course, *right* observation completeness problems are analogously formulated. It turns out that a left observation completeness equation $\rho_A \circ \otimes \circ \langle \eta, \rho_{A_2} \rangle = \rho_A \circ \otimes \circ \langle id, \rho_{A_2} \rangle$ formulated in terms of GIs amounts to require that for any $x \in C_1$ and $y \in A_2$, $\alpha_{C_1,\eta}(x) \otimes^b y = \alpha_{C,A}(x \otimes \gamma_{A_2,C_2}(y))$. Hence, a left observation completeness equation $\rho_A \circ \otimes \circ \langle \eta, \rho_{A_2} \rangle = \rho_A \circ \otimes \circ \langle id, \rho_{A_2} \rangle$ states that completeness for the abstractions $A \in \mathcal{L}_C$, $\eta \in \mathcal{L}_{C_1}$ and $A_2 \in \mathcal{L}_{C_2}$ holds when the semantic operation $\otimes$ acts over $C_1 \times A_2$. Examples of observation completeness problems will be considered and solved later in Section 5.

4 Quantales and solutions to completeness problems

When concrete interpretations are quantales and typed quantales, solutions to the above completeness problems exist and can be characterized explicitly and elegantly in terms of linear implications.

Let $\langle C, C_1, C_2, \otimes \rangle$ be a typed quantale playing the rôle of concrete interpretation. In this setting, solutions to completeness problems will be characterized by exploiting two basic domain transformers $\overset{\curlywedge}{\multimap} : uco(C_1) \times uco(C) \to uco(C_2)$ and $\overset{\curlywedge}{\multimapinv} : uco(C) \times uco(C_2) \to uco(C_1)$, defined by lifting left and right linear implications $\multimap$ and $\multimapinv$ to abstract domains as follows: For any $A_1 \in \mathcal{L}_{C_1}, A_2 \in \mathcal{L}_{C_2}, A \in \mathcal{L}_C$:

$$\begin{aligned} A_1 \overset{\curlywedge}{\multimap} A &\stackrel{\text{def}}{=} \curlywedge(\{a_1 \multimap a \in C_2 \mid a_1 \in A_1, a \in A\}); \\ A \overset{\curlywedge}{\multimapinv} A_2 &\stackrel{\text{def}}{=} \curlywedge(\{a \multimapinv a_2 \in C_1 \mid a \in A, a_2 \in A_2\}). \end{aligned}$$

Hence, $A_1 \overset{\curlywedge}{\multimap} A$ is defined to be the most abstract domain in $\mathcal{L}_{C_2}$ containing all the linear implications from A_1 to A. From the logic properties of linear implication recalled in Section 2, it is not too hard to derive the following useful distributivity laws for $\overset{\curlywedge}{\multimap}$ and $\overset{\curlywedge}{\multimapinv}$ over reduced product of abstract domains.

Proposition 1. *For all* $\{B_i\}_{i \in I} \subseteq \mathcal{L}_C$,

$$A_1 \overset{\curlywedge}{\multimap} (\textstyle\bigsqcap_{i \in I} B_i) = \textstyle\bigsqcap_{i \in I}(A_1 \overset{\curlywedge}{\multimap} B_i) \quad \textit{and} \quad (\textstyle\bigsqcap_{i \in I} B_i) \overset{\curlywedge}{\multimapinv} A_2 = \textstyle\bigsqcap_{i \in I}(B_i \overset{\curlywedge}{\multimapinv} A_2).$$

Solutions to full completeness problems. Solutions to full completeness problems exist and are characterized in terms of linear implications among domain's objects as stated by the following result.

Theorem 2. $A \sqcap (C \stackrel{\curlywedge}{\multimap} A) \sqcap (A \stackrel{\curlywedge}{\multimapinv} C) \sqcap ((C \stackrel{\curlywedge}{\multimap} A) \stackrel{\curlywedge}{\multimapinv} C)$ *is the most abstract solution of System* (1)*.*

Solutions to observation completeness problems. Let us first consider left observation completeness problems. A left observation completeness equation can be characterized as follows.

Theorem 3. $\rho_A \circ \otimes \circ \langle \eta, \rho_{A_2} \rangle = \rho_A \circ \otimes \circ \langle id, \rho_{A_2} \rangle \;\Leftrightarrow\; \eta \sqsubseteq A \stackrel{\curlywedge}{\multimapinv} A_2$.

Thus, as an immediate consequence, left observation completeness problems admit the following solutions.

Corollary 4. $A_1 \sqcap (A \stackrel{\curlywedge}{\multimapinv} A_2)$ *is the most abstract solution of System* (3)*.*

Of course, dual results can be given for right observation completeness problems. In this case, the most abstract solution therefore is $A_2 \sqcap (A_1 \stackrel{\curlywedge}{\multimap} A)$.

The above results for left and right observation completeness turn out to be useful for solving observation completeness problems. In fact, an observation completeness equation is characterized as an independent combination of left and right observation completeness equations as follows.

Theorem 5. $\rho_A \circ \otimes \circ \langle \eta, \mu \rangle = \rho_A \circ \otimes \;\Leftrightarrow\; \langle \eta, \mu \rangle \sqsubseteq \langle A \stackrel{\curlywedge}{\multimapinv} C_2, C_1 \stackrel{\curlywedge}{\multimap} A \rangle$.

As a straight consequence, we get the following result.

Corollary 6. $\langle A_1 \sqcap (A \stackrel{\curlywedge}{\multimapinv} C_2), A_2 \sqcap (C_1 \stackrel{\curlywedge}{\multimap} A) \rangle$ *is the most abstract solution of System* (2)*.*

4.1 The case of unital commutative quantales

When we deal with unital and commutative quantales — i.e. models of intuitionistic linear logic [16,19] — the above solutions to full completeness problems given by Theorem 2 can be significantly simplified by exploiting the logical properties of linear implication. In a unital commutative quantale $\langle C, \otimes \rangle$, the following additional properties hold: For any $a, b, c \in C$:

- $a \multimap (b \multimap c) = b \multimap (a \multimap c)$
- $1 \multimap a = a$
- $c \leq (c \multimap a) \multimap a$
- $((c \multimap a) \multimap a) \multimap a = c \multimap a$

Therefore, from these properties it is not hard to check that for all $a \in C$, $\lambda c.(c \multimap a) \multimap a \in uco(C)$. This turns out to be the key observation in order to give a more compact form to solutions of full completeness problems on unital commutative quantales. Moreover, the objects of such solutions enjoy a clean logical characterization in terms of linear implications as specified by the third point of the next result.

Theorem 7. *Let $\langle C, \otimes\rangle$ be a unital commutative quantale.*

1. $C \stackrel{\curlywedge}{\multimap} A$ *is the most abstract solution of System* (1)*;*
2. *If* $B \in \mathcal{L}_C$ *is such that* $B \sqsubseteq C \stackrel{\curlywedge}{\multimap} A$*, then* $C \stackrel{\curlywedge}{\multimap} A = B \stackrel{\curlywedge}{\multimap} A$*;*
3. *For all* $c \in C$*,* $c \in C \stackrel{\curlywedge}{\multimap} A \;\Leftrightarrow\; c = \bigwedge_{a\in A}(c \multimap a) \multimap a$*.*

5 An application in data structure completeness

In this section, we consider some examples of completeness problems in abstract interpretation of list data structures. For the sake of practicality, we consider lists of natural numbers, even if our discussion holds more in general for lists of objects of arbitrary type.

Consider the structure $\langle \wp(list(\mathbb{N})), \wp(\mathbb{N}), \wp(list(\mathbb{N})), ::\rangle$, where $list(\mathbb{N})$ is the set of all finite lists of natural numbers, $\wp(list(\mathbb{N}))$ and $\wp(\mathbb{N})$ are complete lattices w.r.t. set-inclusion, and $::\, : \wp(\mathbb{N}) \times \wp(list(\mathbb{N})) \to \wp(list(\mathbb{N}))$ is a "collecting" version of concatenation defined as follows:

$$N :: L \stackrel{\text{def}}{=} \{[n|l] \mid n \in N,\, l \in L\},$$

where $\varnothing :: L = N :: \varnothing = \varnothing$. It is clear that this structure is a typed quantale.

We say that a list is *irredundant* if it does not contain two occurrences of the same object. An abstract domain $\rho \in uco(\wp(list(\mathbb{N})))$ for detecting irredundant lists can be defined by $\rho \stackrel{\text{def}}{=} \{list(\mathbb{N}), Irr\}$, where $Irr \subseteq list(\mathbb{N})$ is the set of irredundant lists over $\mathbb{N}$. We consider ρ as an observation domain and we look for the most abstract solution $\langle X, Y\rangle \in uco(\wp(\mathbb{N})) \times uco(\wp(list(\mathbb{N})))$ to the following observation completeness problem:

$$\rho \circ :: \circ \langle \rho_X, \rho_Y\rangle = \rho \circ ::$$

Here, we dropped the first constraint of the generic System (2), since it amounts to the trivial constraint $\langle \rho_X, \rho_Y\rangle \sqsubseteq \langle\{\mathbb{N}\}, \{list(\mathbb{N})\}\rangle$ which is always satisfied. By Corollary 6, the most abstract solution of this observation completeness equation exists. This will be the most abstract pair of domains $\langle X, Y\rangle$ for which abstract concatenation in X and Y results to be complete when observing irredundancy as represented by ρ. By Corollary 6, the solution $\langle X, Y\rangle$ is as follows.

$$\begin{aligned}
X &= \{\mathbb{N}\} \sqcap (\rho \stackrel{\curlywedge}{\multimapinv} \wp(list(\mathbb{N})))\\
&= \rho \stackrel{\curlywedge}{\multimapinv} \wp(list(\mathbb{N}))\\
&= \curlywedge(\{L \multimapinv M \mid L \in \rho,\; M \in \wp(list(\mathbb{N}))\})\\
&\qquad\qquad (\text{since, for all } M,\, list(\mathbb{N}) \multimapinv M = list(\mathbb{N}))\\
&= \curlywedge(\{Irr \multimapinv M \mid M \in \wp(list(\mathbb{N}))\})\\
&\qquad\qquad (\text{since, for all } n \in \mathbb{N},\, Irr \multimapinv \{[n]\} = \mathbb{N} \smallsetminus \{n\})\\
&= \wp(\mathbb{N})
\end{aligned}$$

$$
\begin{aligned}
Y &= \{\, list(\mathbb{N}) \,\} \sqcap (\wp(\mathbb{N}) \overset{\curlywedge}{\multimap} \rho) \\
&= \wp(\mathbb{N}) \overset{\curlywedge}{\multimap} \rho \\
&= \curlywedge(\{N \multimap L \mid N \in \wp(\mathbb{N}),\ L \in \rho\}) \\
&\qquad\qquad \text{(since, for all } N,\ N \multimap list(\mathbb{N}) = list(\mathbb{N})) \\
&= \curlywedge(\{N \multimap Irr \mid N \in \wp(\mathbb{N})\}) \\
&\qquad\qquad \text{(since, by (v) in Section 2, } \textstyle\bigcap_i (N_i \multimap Irr) = (\bigcup_i N_i) \multimap Irr) \\
&= \{N \multimap Irr \mid N \in \wp(\mathbb{N})\} \\
&= \textstyle\bigcup_{N \subseteq \mathbb{N}} \{L \in \wp(list(\mathbb{N})) \mid l \in L \ \Leftrightarrow\ (l \in Irr \text{ and } \forall n \in N.\ n \text{ is not in } l)\}
\end{aligned}
$$

Thus, in order to be complete for concatenation when observing irredundancy, X must coincide with the concrete domain $\wp(\mathbb{N})$, while it suffices that Y contains all the sets of irreduntant lists which do not contain some set of numbers. Note that Y coincides with the set of all the sets of irredundant lists closed by permutation of their objects, and this is a strict abstraction of the concrete domain $\wp(list(\mathbb{N}))$.

Let us now consider the standard abstract domain $\eta \in uco(\wp(\mathbb{N}))$ for parity analysis given by $\eta \overset{\text{def}}{=} \{\,\mathbb{N}, even, odd, \varnothing\,\}$. We consider the following left and right observation completeness problems: We look respectively for the the most abstract domains $X \in uco(\wp(\mathbb{N}))$ and $Y \in uco(\wp(list(\mathbb{N})))$ such that:

$$
\text{(i)}\ \ \rho \circ :: \circ \langle \rho_X, \rho\rangle = \rho \circ :: \langle id, \rho\rangle \qquad \text{(ii)}\ \ \rho \circ :: \circ \langle \eta, \rho_Y\rangle = \rho \circ :: \langle \eta, id\rangle
$$

Here again, there are no upper bound constraints for X and Y. By Corollary 4 (and its dual), we get the following solutions:

$$
\begin{aligned}
X &= \{\,\mathbb{N}\,\} \sqcap (\rho \overset{\curlywedge}{\multimapinv} \rho) \\
&= \rho \overset{\curlywedge}{\multimapinv} \rho \\
&= \curlywedge(\{\, list(\mathbb{N}) \multimapinv list(\mathbb{N}), list(\mathbb{N}) \multimapinv Irr, Irr \multimapinv list(\mathbb{N}), Irr \multimapinv Irr \,\}) \\
&= \{\,\mathbb{N}, \varnothing\,\},
\end{aligned}
$$

$$
\begin{aligned}
Y &= \{\, list(\mathbb{N}) \,\} \sqcap (\eta \overset{\curlywedge}{\multimap} \rho) \\
&= \eta \overset{\curlywedge}{\multimap} \rho \\
&= \curlywedge(\{\,\mathbb{N} \multimap Irr, even \multimap Irr, odd \multimap Irr, \varnothing \multimap Irr\,\}) \\
&= \{\,\{\,[]\,\}, Irr_{even}, Irr_{odd}, list(\mathbb{N})\,\},
\end{aligned}
$$

where $Irr_{even} \overset{\text{def}}{=} \{l \in list(\mathbb{N}) \mid l \in Irr$ and l does not contain even numbers$\}$ and $Irr_{odd} \overset{\text{def}}{=} \{l \in list(\mathbb{N}) \mid l \in Irr$ and l does not contain odd numbers$\}$. Thus, for problem (i), in order to get completeness, it is enough to check whether a given set of numbers is empty or not, while, for problem (ii), we only need to consider sets of irreduntant lists which do not contain either even or odd numbers.

6 Complete semantics for logic program analysis

In this section, we determine the least complete extension of any logic program property w.r.t. a bottom-up semantics characterizing computed answer substitutions, which turns out to be equivalent to the well-known s-semantics [8]. This complete semantics can be thought of as a logic program semantics which is "optimal" (i.e. neither too concrete nor too abstract) for characterizing a given property of interest. The problem of determining optimal semantics for program analysis was raised in [9]. We show that such optimal semantics can be obtained by solving a full completeness problem relatively to the operation of unification on sets of substitutions.

Completeness for an abstract domain, in general, depends both on the considered concrete domains and on the semantic operation defined on them. However, the following important result shows that, under certain conditions, completeness is instead independent from the choice of the concrete semantics.

Theorem 8. *Let $\alpha \in \mathcal{L}_C$ and $f, g : C \to C$ such that $f \sqsubseteq g \sqsubseteq \alpha \circ f$. Then, α is complete for f iff α is complete for g.*

We will exploit this result for computing our least complete extension of abstract domains w.r.t. s-semantics. The idea is that of considering a "simplified" and more concrete version, denoted by T_P, of the immediate consequences operator T_P^s of s-semantics, which does not take into account variable renaming. This will simplify a lot the technical development of this section. Hence, s-semantics results to be an abstract interpretation of our T_P semantics: If r denotes the closure under variable renaming of sets of substitutions, then we have that $T_P^s = r \circ T_P$. Then, leaving out the details, if α denotes the least complete extension, relatively to T_P, of any domain A abstracting computed answer substitutions, since $T_P \sqsubseteq r \circ T_P = T_P^s \sqsubseteq \alpha \circ T_P$ holds, we can apply Theorem 8, from which we get that α not only is complete for T_P^s, but actually α turns out to be the least complete extension of A relatively to T_P^s.

6.1 Notation

Let $\mathcal{V}$ be an infinite, recursively enumerable (r.e.) set of variables, Σ be a set of function symbols and Π be a set of predicate symbols, defining a r.e. first-order language $\mathcal{L}$. *Term* denotes the set of terms of $\mathcal{L}$. If s is any syntactic object and σ and θ are substitutions, then $vars(s)$ denotes the set of variables occurring in s, $dom(\sigma) \stackrel{\text{def}}{=} \{v \in \mathcal{V} \mid \sigma(v) \neq v\}$, $rng(\sigma) \stackrel{\text{def}}{=} \cup\{vars(\sigma(x)) \mid x \in dom(\sigma)\}$, $s\sigma$ denotes the application of σ to s, and $\sigma \circ \theta$ denotes the standard composition of θ and σ (i.e., $\sigma \circ \theta = \lambda x.(\theta(x))\sigma$). The set of idempotent substitutions modulo renaming (i.e., $\theta \sim \sigma$ iff there exists β and δ such that $\theta = \sigma \circ \beta$ and $\sigma = \theta \circ \delta$) on $\mathcal{L}$ is denoted by Sub. Objects in Sub are partially ordered by instantiation, denoted by $\preceq$. By adding to Sub an extra object τ as least element, one gets a complete lattice $\langle Sub^\tau, \preceq, \vee, \wedge, \epsilon, \tau\rangle$, where $\vee$ is least general anti-instance, $\wedge$ is standard unification and ϵ is the empty substitution (see [14] for more details). The set of most general atoms is given by $GAtom \stackrel{\text{def}}{=} \{p(\bar{X}) \mid p \in \Pi\}$, where $\bar{X}$ is a tuple of distinct variables. We consider logic programs in normalized form, that is, a generic Horn clause is $p(\bar{X}) : -c, q_1(\bar{X}_1), \ldots, q_n(\bar{X}_n)$, where all the tuples of variables are distinct and $c \in Sub$ is an idempotent substitution binding variables to terms.

6.2 T_P-completeness

Our basic semantic structure is the unital commutative quantale $\langle\wp(Sub),\otimes\rangle$, where $\langle\wp(Sub),\subseteq\rangle$ is a complete lattice, $\otimes : \wp(Sub)\times\wp(Sub)\to\wp(Sub)$ is the obvious lifting of unification $\wedge$ to sets of substitutions, i.e. it is defined by: $X\otimes Y \stackrel{\text{def}}{=} \{x\wedge y \mid x\in X,\ y\in Y\}$, and $\{\epsilon\}\in\wp(Sub)$ is the unit of $\otimes$. It is immediate to check that $\langle\wp(Sub),\otimes\rangle$ actually is a unital commutative quantale [16, Example 10, p. 18]. In the following, we will slightly abuse notation by applying the operation $\otimes$ also to substitutions.

As mentioned above, we consider a bottom-up semantics based on an immediate consequences operator T_P which is more concrete than the standard operator T_P^s of s-semantics. In fact, T_P is defined using only the operations of unification $\otimes$ and union of sets of idempotent substitutions. The s-semantics operator T_P^s can be recovered from T_P by a simple step of abstract interpretation considering the closure of sets of substitutions over variables renaming.

We consider a concrete semantic domain $CInt$ of functions — as usual, called interpretations — which map most general atoms to sets of substitutions: $CInt \stackrel{\text{def}}{=} GAtom\to\wp(Sub)$, which ordered pointwise is trivially a complete lattice. Often, we will find convenient to denote an interpretation $I\in CInt$ by the set $\{\langle p(\bar{X}), I(p(\bar{X}))\rangle \mid p\in\Pi\}$. Then, for any program P, the immediate consequences operator $T_P : CInt\to CInt$ is defined as follows: For any $I\in CInt$,

$$T_P(I)(p(\bar{Y})) = \bigcup_{C\lll P}\Big(c\otimes\big(\bigotimes_{i=1..n} I(q_i(\bar{X}_i))\big)\otimes\{\bar{Y}=\bar{X}\}\Big),$$

where $C = p(\bar{X}) :- c, q_1(\bar{X}_1),\ldots,q_n(\bar{X}_n)$. Here, $C\lll P$ denotes that the clause C of P is renamed apart with fresh variables, and $I(q_i(\bar{X}_i))$ is intended modulo renaming.

If $\rho\in uco(\wp(Sub))$ is any abstraction on sets of substitutions, a corresponding abstraction on interpretations $\wr\rho\wr\in uco(CInt)$ which acts accordingly to ρ can be defined as follows: For any $I\in CInt$, $\wr\rho\wr(I)\stackrel{\text{def}}{=}\{\langle p(\bar{X}),\rho(c)\rangle \mid \langle p(\bar{X}),c\rangle\in I\}$. Note that $\wr\cdot\wr$ is monotone, i.e., for all $\rho,\eta\in uco(\wp(Sub))$, if $\rho\sqsubseteq\eta$ then $\wr\rho\wr\sqsubseteq\wr\eta\wr$.

Given a basic abstract domain of properties of interest $\pi\in uco(\langle\wp(Sub),\subseteq\rangle)$, our goal is therefore to find the most abstract domain which contains π (more precisely, $\wr\pi\wr$) and is complete for any T_P operator. Our strategy consists in characterizing the least complete extension of π for the basic operations involved in the definition of T_P, and then to show that, under reasonable hypotheses, this domain turns out to be the right one. Since every abstract domain is trivially always complete for lub's, it turns out that union of sets of substitutions is troubleless, and therefore it is enough to concentrate on unification $\otimes$. Indeed, the following result shows that completeness for $\otimes$ implies completeness for any T_P.

Theorem 9. *Let $\rho\in uco(\wp(Sub))$. If ρ is complete for $\otimes$ then $\wr\rho\wr$ is complete for any T_P.*

By Theorem 7, given $\pi \in uco(\wp(Sub))$, the least complete extension of π w.r.t. the quantale operation $\otimes$ exists and is the domain $\wp(Sub) \stackrel{\curlywedge}{\multimap} \pi$ of linear implications from $\wp(Sub)$ to π. This complete domain $\wp(Sub) \stackrel{\curlywedge}{\multimap} \pi$ results to be the right one whenever the abstract domain π satisfies the following weak decidability property.

Definition 10. $\pi \in uco(\wp(Sub))$ is *decidable* if any $S \in \pi$ is a r.e. set. □

It should be clear that decidability is a reasonable requirement for most abstract domains used in program analysis: In fact, for such a decidable abstraction (of sets) of substitutions, an effective procedure for checking whether a substitution belongs to (is approximated by) an abstract object is available. As announced, the following key result shows that least complete extensions of decidable abstract domains w.r.t. unification $\otimes$ actually are least complete extensions for T_P operators as well.

Theorem 11. *Let $\pi \in uco(\wp(Sub))$ be decidable. Then, $\lbag \wp(Sub) \stackrel{\curlywedge}{\multimap} \pi \rbag$ is the least complete extension of $\lbag \pi \rbag$ for any T_P.*

As we discussed just after Theorem 8, it turns out that, for any decidable $\pi \in uco(\wp(Sub))$, $\lbag \wp(Sub) \stackrel{\curlywedge}{\multimap} \pi \rbag$ actually is the least complete extension of $\lbag \pi \rbag$ for any immediate consequences operator T_P^s of s-semantics. In fact, if $r \in uco(\wp(Sub))$ is the closure under renaming of sets of substitions, namely variables in the range of substitutions are renamed in each possible way, then $T_P^s = r \circ T_P$; moreover, $\lbag \wp(Sub) \stackrel{\curlywedge}{\multimap} \pi \rbag$ clearly induces a semantic transformer less precise than T_P^s, i.e. $T_P^s \sqsubseteq \lbag \wp(Sub) \stackrel{\curlywedge}{\multimap} \pi \rbag \circ T_P$. Hence, by Theorems 8 and 11, we get the following desired consequence.

Corollary 12. *Let $\pi \in uco(\wp(Sub))$ be decidable, such that $T_P^s \sqsubseteq \pi \circ T_P$ for any P. Then, $\lbag \wp(Sub) \stackrel{\curlywedge}{\multimap} \pi \rbag$ is the least complete extension of $\lbag \pi \rbag$ for any T_P^s.*

6.3 Complete semantics for groundness analysis

Groundness analysis is arguably one of the most important analysis for logic-based programming languages. Groundness analysis aims to statically detect whether variables will be bound to ground terms in successful derivations. By instantiating the results above, we are able to characterize the least complete extension of the basic abstract domain representing plain groundness information w.r.t. any immediate consequences operator of s-semantics. The resulting semantics can be therefore interpreted as the "optimal" semantics for groundness analysis.

If $V \subseteq \mathcal{V}$ is a finite set of variables of interest, the simplest abstract domain for representing plain groundness information of variables in V is $\mathcal{G}_V \stackrel{\text{def}}{=} \langle \wp(V), \supseteq \rangle$, as first put forward by Jones and Søndergaard [12]. The intuition is that each $W \in \mathcal{G}_V$ represents the set of substitutions which ground every variable in W. $\mathcal{G}_V$ is related to the concrete domain $\wp(Sub)$ by the following concretization map: For each $W \in \mathcal{G}$, $\gamma_{\mathcal{G}_V}(W) \stackrel{\text{def}}{=} \{\theta \in Sub \mid \forall v \in W.\, vars(\theta(v)) = \varnothing\}$. As usual, we shall abuse notation by denoting with $\mathcal{G}_V \in uco(\wp(Sub))$ the corresponding isomorphic image $\gamma_{\mathcal{G}_V}(\mathcal{G}_V)$ in $\wp(Sub)$.

A variable independent abstract domain $\mathcal{G} \in uco(CInt)$ for representing plain groundness information can be therefore defined as follows:

$$\mathcal{G}(I) \stackrel{\text{def}}{=} \{\langle p(\bar{X}), \mathcal{G}_{\bar{X}}(c)\rangle \mid \langle p(\bar{X}), c\rangle \in I\}.$$

It is easy to check that $\mathcal{G}$ actually is a uco on the concrete semantic domain $CInt$. Furthermore, $\mathcal{G}$ is clearly decidable and $T_P^s \sqsubseteq \mathcal{G} \circ T_P$. Thus, as an easy consequence of Corollary 12, we can prove the following result, where $V \subset_f \mathcal{V}$ means that V is a finite subset of $\mathcal{V}$.

Theorem 13. $\bigsqcap_{V \subset_f \mathcal{V}} \wr \wp(Sub) \stackrel{\curlywedge}{\rightarrow} \mathcal{G}_V \wr$ *is the least complete extension of* $\mathcal{G}$ *for any* T_P^s.

References

1. A. Bossi, M. Gabbrielli, G. Levi, and M. Martelli. The s-semantics approach: theory and applications. *J. Logic Program.*, 19-20:149–197, 1994.
2. P. Cousot. Completeness in abstract interpretation (Invited Talk). In *Proc. 1995 Joint Italian-Spanish Conference on Declarative Programming*, pages 37–38, 1995.
3. P. Cousot. Constructive design of a hierarchy of semantics of a transition system by abstract interpretation (Invited Paper). In *Proc. of the 13th Int. Symp. on Math. Found. of Programming Semantics (MFPS'97)*, vol. 6 of Electronic Notes in Theor. Comput. Sci., 1997.
4. P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proc. 4th ACM POPL*, pages 238–252, 1977.
5. P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Proc. 6th ACM POPL*, pages 269–282, 1979.
6. P. Cousot and R. Cousot. Inductive definitions, semantics and abstract interpretation. In *Proc. 19th ACM POPL*, pages 83–94, 1992.
7. P. Cousot and R. Cousot. Abstract interpretation of algebraic polynomial systems. In *Proc. 6th Int. Conf. on Algebraic Methodology and Software Technology (AMAST'97)*, LNCS 1349, pages 138–154, 1997.
8. M. Falaschi, G. Levi, M. Martelli, and C. Palamidessi. Declarative modeling of the operational behavior of logic languages. *Theor. Comput. Sci.*, 69(3):289–318, 1989.
9. R. Giacobazzi. "Optimal" collecting semantics for analysis in a hierarchy of logic program semantics. In *Proc. 13th Int. Symp. on Theor. Aspects of Comput. Sci. (STACS'96)*, LNCS 1046, pages 503–514, 1996.
10. R. Giacobazzi and F. Ranzato. Completeness in abstract interpretation: a domain perspective. In *Proc. 6th Int. Conf. on Algebraic Methodology and Software Technology (AMAST'97)*, LNCS 1349, pages 231–245, 1997.
11. R. Giacobazzi, F. Ranzato and F. Scozzari. Complete abstract interpretations made constructive. In *Proc. 23rd Int. Symp. on Math. Found. of Comp. Sci. (MFCS'98)*, LNCS, 1998.
12. 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.

13. A. Mycroft. Completeness and predicate-based abstract interpretation. In *Proc. ACM Conf. on Partial Evaluation and Program Manipulation (PEPM'93)*, pages 179–185, 1993.
14. C. Palamidessi. Algebraic properties of idempotent substitutions. In *Proc. 17th Int. Colloq. on Automata, Languages and Programming (ICALP'90)*, LNCS 443, pages 386–399, 1990.
15. U.S. Reddy and S.N. Kamin. On the power of abstract interpretation. *Computer Languages*, 19(2):79–89, 1993.
16. K.I Rosenthal. *Quantales and their Applications.* Longman Scientific & Technical, 1990.
17. R.C. Sekar, P. Mishra, and I.V. Ramakrishnan. On the power and limitation of strictness analysis. *J. ACM*, 44(3):505–525, 1997.
18. B. Steffen. Optimal data flow analysis via observational equivalence. In *Proc. 14th Int. Symp. on Math. Found. of Comp. Sci. (MFCS'89)*, LNCS 379, pages 492–502, 1989.
19. D. Yetter. Quantales and (noncommutative) linear logic. *J. Symbolic Logic*, 55(1):41–64, 1990.