Compositional Optimization of Disjunctive Abstract Interpretations

Roberto Giacobazzi⋆ Francesco Ranzato⋆⋆

⋆*Dipartimento di Informatica, Università di Pisa*
Corso Italia 40, 56125 Pisa, Italy
giaco@di.unipi.it

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

Abstract. We define the inverse operation for disjunctive completion, introducing the notion of least disjunctive basis for an abstract domain $\mathbf{D}$: this is the most abstract domain inducing the same disjunctive completion as $\mathbf{D}$. We show that the least disjunctive basis exists in most cases, and study its properties in relation with reduced product of abstract interpretations. The resulting framework is powerful enough to be applied to arbitrary abstract domains for analysis, providing advanced algebraic methods for domain manipulation and optimization. These notions are applied to abstract domains for analysis of functional and logic programming languages.

1 Introduction

It is widely acknowledged that most program properties need *relational* abstract domains to be attacked by abstract interpretation ([18, 24]). The Cousot and Cousot functional combination by *reduced power* ([8]), and Nielson's *tensor product* ([25]) were the first systematic methods to induce relational analyses by combining abstract domains. Cousot and Cousot showed in [9] that a relational analysis can be induced by combining *reduced product* (denoted $\sqcap$) and *disjunctive completion* (denoted $\mho$) of abstract domains. If $\mathbf{D}_1$ and $\mathbf{D}_2$ are abstract domains, a corresponding domain for relational analysis can always be defined as $\mho(\mathbf{D}_1 \sqcap \mathbf{D}_2)$. In this construction, reduced product is *attribute independent* (viz. the information obtainable from the combination of analyses is essentially the same as the one obtainable by performing the analyses separately), while disjunctive completion introduces relational information by exploiting sets of attribute independent abstract properties. Disjunctive completion is therefore fundamental to implement relational analyses.

Disjunctive completion was originally introduced to exploit disjunctive program properties, notably to prove that *merge-over-all-paths* (MOP) data-flow analysis can be always expressed in fixpoint form ([8]). This notion was also considered in Nielson's approach to abstract interpretation using domain theory ([24]), and applied in data-flow analysis of functional and logic languages, e.g., to express disjunctive information in Jensen's *strictness logic* ([17]), in Cousot and Cousot *comportment analysis* ([10]), and in analysis of *ground-dependencies* ([12]).

A natural question is: can we invert a process of "domain refinement"? Namely, can we reconstruct the "least basis" which induces a given domain by composition or completion? Recently, [5] attacked the problem of inverting reduced product, introducing the notion of *complementation* in abstract interpretation. Complementation provides an important tool for abstract domain decomposition into attribute independent factors. In this paper, we consider the inverse for the remaining fundamental operation of disjunctive completion, denoted Ω. We introduce the notion of *least disjunctive basis* for an abstract domain, and study its properties in relation with reduced product. The interest in this operation is twofold: (1) theoretically, least disjunctive bases contain the least amount of information which characterizes a given disjunctive property; and (2) practically, least disjunctive bases are minimal (viz. non-redundant), providing useful space saving techniques to implement disjunctive completions and relational analyses. In particular, the disjunctive completion of the least disjunctive basis involves the least number of reduction tests in domain implementation (e.g. by powerset construction), as most redundant information has been removed from the source. This operation can be combined with complementation, in order to characterize optimal (viz. most abstract) decompositions for complex relational abstract domains. The resulting framework is powerful enough to be applied to arbitrary abstract domains for analysis, providing advanced algebraic methods for domain manipulation and optimization.

The main achievements of the paper can be summarized as follows.

- Under weak hypotheses, an abstract domain $\mathbf{D}$ can be associated with a unique *least disjunctive basis*, which is the most abstract domain inducing the same disjunctive completion as $\mathbf{D}$.
- Least disjunctive bases distribute compositionally with respect to the reduced product, and enjoy remarkable algebraic properties.
- We apply the above results to domains for analysis of functional and logic programming languages. In particular, we show that:
  - The Cousot and Cousot lattice of *basic comportments* ([10]) is not the least disjunctive basis of the lattice for disjunctive *comportment analysis* ([10]).
  - The Marriott and Søndergaard domain **Def** ([19]) is the least disjunctive basis inducing the domain for disjunctive ground-dependency analysis of logic programs. This shows that **Def**, which is strictly less expensive than **Pos** ([6, 19]), always induces the same disjunctive ground-dependency analysis, i.e., $\Omega(\mathbf{Pos}) = \mathbf{Def}$.

Throughout the paper, we assume familiarity with lattice theory (e.g. see [3, 14]), in particular closure operators (see [20, 28]), and abstract interpretation ([7, 8]).

2 Abstract Interpretation and Closure Operators

The standard Cousot and Cousot theory of abstract interpretation is based on the notion of Galois connection ([7, 8]). In this section, we briefly introduce some notation and recall some well known notions.

If $\mathbf{C}$ and $\mathbf{D}$ are posets and $\alpha : \mathbf{C} \to \mathbf{D}$, $\gamma : \mathbf{D} \to \mathbf{C}$ are monotonic functions such that $\forall \mathbf{c} \in \mathbf{C}.\ \mathbf{c} \leq_{\mathbf{C}} \gamma(\alpha(\mathbf{c}))$ and $\forall \mathbf{d} \in \mathbf{D}.\ \alpha(\gamma(\mathbf{d})) \leq_{\mathbf{D}} \mathbf{d}$, then we call the quadruple $(\gamma, \mathbf{D}, \mathbf{C}, \alpha)$ a *Galois connection* (G.c.) between $\mathbf{C}$ and $\mathbf{D}$. If in addition $\forall \mathbf{d} \in \mathbf{D}.\ \alpha(\gamma(\mathbf{d})) = \mathbf{d}$, then $(\gamma, \mathbf{D}, \mathbf{C}, \alpha)$ is a *Galois insertion* (G.i.) of $\mathbf{D}$ in $\mathbf{C}$. In the setting of abstract interpretation, $\mathbf{C}$ and $\mathbf{D}$ are called, respectively, the *concrete* and the *abstract domain*, and they are assumed to be complete lattices, whereas α and γ are called the *abstraction* and the *concretization* maps, respectively. $\mathbf{D}$ is called an *abstraction* (or *abstract interpretation*) of $\mathbf{C}$, and $\mathbf{C}$ a *concretization* of $\mathbf{D}$. Further, $\mathbf{D}$ is a *proper* abstraction of $\mathbf{C}$ if $\gamma \circ \alpha \neq \lambda \mathbf{x}.\mathbf{x}$. Galois insertions characterize "ideal" abstractions, as any abstract object is the abstraction of a concrete one. In this case, the concretization and abstraction mappings are 1-1 and onto, respectively. Any G.c. can be lifted to a G.i. identifying in an equivalence class those values of the abstract domain with the same concrete meaning. This process is known as *reduction*.

Let $\langle \mathbf{L}, \leq, \wedge, \vee, \top, \bot \rangle$ be a complete lattice. An (*upper*) *closure operator* on $\mathbf{L}$ is an operator $\rho : \mathbf{L} \to \mathbf{L}$ monotonic, idempotent and extensive (viz. $\forall \mathbf{x} \in \mathbf{L}.\ \mathbf{x} \leq \rho(\mathbf{x})$). Each closure operator ρ is uniquely determined by the set of its fixpoints, which is its image $\rho(\mathbf{L})$. A set $\mathbf{X} \subseteq \mathbf{L}$ is the set of fixpoints of a closure operator iff $\mathbf{X}$ is a *Moore-family* of $\mathbf{L}$, i.e. $\top \in \mathbf{X}$ and $\mathbf{X}$ is meet-closed (viz. for any non-empty $\mathbf{Y} \subseteq \mathbf{X}$, $\wedge \mathbf{Y} \in \mathbf{X}$). For any $\mathbf{X} \subseteq \mathbf{L}$, we denote by $\mathcal{M}(\mathbf{X})$ the *Moore-closure* of $\mathbf{X}$, i.e. the least subset of $\mathbf{L}$ containing $\mathbf{X}$ which is a Moore-family of $\mathbf{L}$. $\rho(\mathbf{L})$ is a complete lattice with respect to the order of $\mathbf{L}$, but, in general, it is not a complete sublattice of $\mathbf{L}$, since the *lub* in $\rho(\mathbf{L})$ might be different from that in $\mathbf{L}$. Indeed, $\rho(\mathbf{L})$ is a complete sublattice of $\mathbf{L}$ iff ρ is additive, i.e. for all $\mathbf{X} \subseteq \mathbf{L}$, $\rho(\vee \mathbf{X}) = \vee \rho(\mathbf{X})$. In the following, we will often denote a closure operator by the set of its fixpoints. We denote by $\langle \mathbf{uco}(\mathbf{L}), \sqsubseteq, \sqcap, \sqcup, \lambda \mathbf{x}.\top, \lambda \mathbf{x}.\mathbf{x} \rangle$ the complete lattice of all upper closure operators on the complete lattice $\mathbf{L}$, with top element $\lambda \mathbf{x}.\top$ and bottom element $\lambda \mathbf{x}.\mathbf{x}$, where for every $\rho, \eta \in \mathbf{uco}(\mathbf{L})$, $\{\rho_\mathbf{i}\}_{\mathbf{i} \in \mathbf{I}} \subseteq \mathbf{uco}(\mathbf{L})$ and $\mathbf{x} \in \mathbf{L}$: $\rho \sqsubseteq \eta$ iff $\forall \mathbf{x} \in \mathbf{L}.\ \rho(\mathbf{x}) \leq \eta(\mathbf{x})$, or equivalently $\rho \sqsubseteq \eta$ iff $\eta(\mathbf{L}) \subseteq \rho(\mathbf{L})$; $(\sqcap_{\mathbf{i} \in \mathbf{I}} \rho_\mathbf{i})(\mathbf{x}) = \wedge_{\mathbf{i} \in \mathbf{I}} \rho_\mathbf{i}(\mathbf{x})$; $(\sqcup_{\mathbf{i} \in \mathbf{I}} \rho_\mathbf{i})(\mathbf{x}) = \mathbf{x} \ \Leftrightarrow\ \forall \mathbf{i} \in \mathbf{I}.\ \rho_\mathbf{i}(\mathbf{x}) = \mathbf{x}$. A *lower closure operator* $\varphi : \mathbf{L} \to \mathbf{L}$ is monotonic, idempotent and reductive (viz. $\forall \mathbf{x} \in \mathbf{L}.\ \varphi(\mathbf{x}) \leq \mathbf{x}$). The complete lattice of all lower closure operators on the complete lattice $\mathbf{L}$ is denoted by $\mathbf{lco}(\mathbf{L})$. Its lattice-theoretic properties can all be derived by duality from those above for $\mathbf{uco}(\mathbf{L})$.

The lattice of abstract interpretations. A key point in Cousot and Cousot abstract interpretation theory is the equivalence between the Galois insertion and closure operator approach to the design of abstract domains. Actually, an abstract domain is just a "computer representation" of its logical meaning, namely its image in the concrete domain. In fact, using a different but lattice-theoretic isomorphic domain changes nothing in the abstract reasoning. The logical meaning of an abstract domain is exactly captured by the associated closure operator on the concrete domain. More formally, on one hand, if $(\gamma, \mathbf{D}, \mathbf{C}, \alpha)$ is a G.i. then the closure associated with $\mathbf{D}$ is the operator $\rho_\mathbf{D} = \gamma \circ \alpha$ on $\mathbf{C}$. On the other hand, if ρ is a closure on $\mathbf{C}$ and $\iota : \rho(\mathbf{C}) \to \mathbf{D}$ is an isomorphism of complete lattices (with inverse ι^{-1}) then $(\iota^{-1}, \mathbf{D}, \mathbf{C}, \iota \circ \rho)$ is a G.i.. The complete lattice of all abstract interpretations (identified up to isomorphism) of a domain $\mathbf{C}$ is therefore isomorphic to $\mathbf{uco}(\mathbf{C})$. By the above equivalence, it is not restrictive to use the closure operator approach to reason about abstract properties up to isomorphic representations of abstract domains. Thus, in the rest of the paper, we will feel free to use most of the times this approach, and whenever we will say that $\mathbf{D}$ is an abstraction of $\mathbf{C}$, we will mean that $\mathbf{D}$ is isomorphic to $\rho_\mathbf{D}(\mathbf{C})$ (denoted by $\mathbf{D} \cong \rho_\mathbf{D}(\mathbf{C})$), for some closure $\rho_\mathbf{D} \in \mathbf{uco}(\mathbf{C})$.

In this approach, the order relation on $\mathbf{uco}(\mathbf{C})$ corresponds to the order by means of which abstract domains are compared with regard to their precision. More formally, if $\rho_{\mathbf{i}} \in \mathbf{uco}(\mathbf{C})$ and $\mathbf{D}_{\mathbf{i}} \cong \rho_{\mathbf{i}}(\mathbf{C})$ $(\mathbf{i} = 1, 2)$, $\mathbf{D}_1$ is *more precise* than $\mathbf{D}_2$ iff $\rho_1 \sqsubseteq \rho_2$ (i.e. $\rho_2(\mathbf{C}) \subseteq \rho_1(\mathbf{C})$). Therefore, to compare domains with regard to their precision, we will only speak about abstractions between them, and use $\sqsubseteq$ to relate both closure operators and domains ($\sqsubset$ denotes strict ordering). Further, we will often use the equality symbol $=$ instead of $\cong$. In view of this equivalence, the *lub* and *glb* on $\mathbf{uco}(\mathbf{C})$ get a clear meaning. Suppose $\{\rho_{\mathbf{i}}\}_{\mathbf{i} \in \mathbf{I}} \subseteq \mathbf{uco}(\mathbf{C})$ and $\mathbf{D}_{\mathbf{i}} \cong \rho_{\mathbf{i}}(\mathbf{C})$ for each $\mathbf{i} \in \mathbf{I}$. Any domain $\mathbf{D}$ isomorphic to the *lub* $(\sqcup_{\mathbf{i} \in \mathbf{I}} \rho_{\mathbf{i}})(\mathbf{C})$ is the most concrete among the domains which are abstractions of all the $\mathbf{D}_{\mathbf{i}}$'s. The interpretation of the *glb* operation on $\mathbf{uco}(\mathbf{C})$ is twofold. Firstly, any domain $\mathbf{D}$ isomorphic to the *glb* $(\sqcap_{\mathbf{i} \in \mathbf{I}} \rho_{\mathbf{i}})(\mathbf{C})$ is (isomorphic to) the well known *reduced product* ([8]) of all the domains $\mathbf{D}_{\mathbf{i}}$. Also, the *glb* $\mathbf{D}$, and hence the reduced product, is the most abstract among the domains (abstracting $\mathbf{C}$) which are more concrete than every $\mathbf{D}_{\mathbf{i}}$. Thus, we will denote the reduced product of abstract domains by the *glb* symbol $\sqcap$.

Complementation in abstract interpretation. *Complementation* ([5]) corresponds to the *inverse operation for reduced product*, namely an operation which starting from any two domains $\mathbf{C} \sqsubseteq \mathbf{D}$, gives as result the (unique) most abstract domain $\mathbf{C} \sim \mathbf{D}$, whose reduced product with $\mathbf{D}$ is exactly $\mathbf{C}$ (i.e., $(\mathbf{C} \sim \mathbf{D}) \sqcap \mathbf{D} = \mathbf{C}$). If $\mathbf{C}$ is a *meet-continuous* complete lattice (i.e., for any chain $\mathbf{Y} \subseteq \mathbf{C}$ and $\mathbf{x} \in \mathbf{C}$, $\mathbf{x} \wedge (\vee \mathbf{Y}) = \vee_{\mathbf{y} \in \mathbf{Y}} (\mathbf{x} \wedge \mathbf{y})$) and $\mathbf{C} \sqsubseteq \mathbf{D}$ then $\mathbf{C} \sim \mathbf{D}$ always exists, and can be defined as $\mathbf{C} \sim \mathbf{D} = \sqcup \{\rho \in \mathbf{uco}(\mathbf{C}) \mid (\rho_{\mathbf{D}} \sqcap \rho)(\mathbf{C}) = \mathbf{C}\}$ (cf. [5]).

3 Disjunctive Completions by Closures

In this section, we formulate by closure operators the standard Cousot and Cousot definition of disjunctive completion of an abstract domain ([8]), and introduce some basic properties. Let $\mathbf{C}$ be any complete lattice, and consider its lattice of abstract interpretations $\mathbf{uco}(\mathbf{C})$.

Definition 3.1 The *disjunctive completion operator* is the map $\mho_{\mathbf{C}} : \mathbf{uco}(\mathbf{C}) \to \mathbf{uco}(\mathbf{C})$ defined as: $\mho_{\mathbf{C}}(\rho) = \sqcup \{\eta \in \mathbf{uco}(\mathbf{C}) \mid \eta \sqsubseteq \rho,\ \eta \text{ is additive}\}$, for any $\rho \in \mathbf{uco}(\mathbf{C})$. □

Lemma 3.2 *For any $\rho \in \mathbf{uco}(\mathbf{C})$, $\mho_{\mathbf{C}}(\rho)$ is additive.*

$\mho_{\mathbf{C}}(\rho)$ is called the *disjunctive completion of $\rho \in \mathbf{uco}(\mathbf{C})$ in* $\mathbf{C}$. In other terms, for a domain $\mathbf{D}$ abstracting $\mathbf{C}$, $\mho_{\mathbf{C}}(\mathbf{D})$ is the most abstract domain which is a concretization of $\mathbf{D}$ and (isomorphic to) a complete sublattice of $\mathbf{C}$ (or, equivalently, join-closed). It is worth noting that for any domain $\mathbf{D}$ its disjunctive completion $\mho_{\mathbf{C}}(\mathbf{D})$ contains a denotation $\bot_{\mathbf{C}}$ for the bottom element of $\mathbf{C}$, since $\emptyset \subseteq \mathbf{D}$ and $\bot_{\mathbf{C}} = \vee_{\mathbf{C}} \emptyset$. It is evident that the above definition corresponds exactly to the standard definition of disjunctive completion given by means of Galois connections (cf. [8, 9, 10, 12]). This notion is also comprehensive for other forms of disjunctive completions: e.g., disjunctive and order-ideal completions (e.g. [17]) are equivalent (see [10]). Moreover, whenever $\mathbf{D}$ satisfies the ascending chain condition, disjunctive, Scott-closed ideal and anti-chain completions are equivalent (see [10]).

Proposition 3.3 $\mho_{\mathbf{C}} \in \mathbf{lco}(\mathbf{uco}(\mathbf{C}))$.

The meaning of the above proposition is clear: the disjunctive completion is a *domain refinement* (viz., a monotonic and reductive mapping in $\mathbf{uco}(\mathbf{C})$). Moreover, no refinement can be obtained by disjunctive completion of a domain which is already disjunctively completed (viz., $\mho_{\mathbf{C}}$ is idempotent). Being a lower closure operator, $\mho_{\mathbf{C}}$ is uniquely determined by its set of fixpoints, namely its image $\mho_{\mathbf{C}}(\mathbf{uco}(\mathbf{C})) = \{\rho \in \mathbf{uco}(\mathbf{C}) \mid \rho \text{ is additive}\}$, which is precisely the set of all *disjunctive abstract interpretations* of $\mathbf{C}$. The following result is an immediate consequence of Proposition 3.3, and characterizes the compositionality of the disjunctive completion with respect to the reduced product of abstract domains.

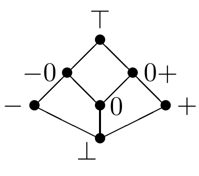
Proposition 3.4 *If* $\mathbf{C} \sqsubseteq \mathbf{D}, \mathbf{E}$ *then* $\mho_{\mathbf{C}}(\mathbf{D} \sqcap \mathbf{E}) = \mho_{\mathbf{C}}(\mho_{\mathbf{C}}(\mathbf{D}) \sqcap \mho_{\mathbf{C}}(\mathbf{E}))$.

It is worth noting that the disjunctive completion of an abstract domain depends on the fixed concrete domain. If $\mathbf{C} \sqsubseteq \mathbf{D} \sqsubseteq \mathbf{E}$, then $\mho_{\mathbf{C}}(\mathbf{E})$ is in general different from $\mho_{\mathbf{D}}(\mathbf{E})$. Indeed, they coincide when $\mathbf{D}$ is disjunctive.

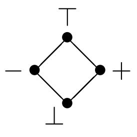
Proposition 3.5 *If* $\mathbf{C} \sqsubseteq \mathbf{D} \sqsubseteq \mathbf{E}$ *then* $\mho_{\mathbf{C}}(\mathbf{E}) \sqsubseteq \mho_{\mathbf{D}}(\mathbf{E})$*, and if, in addition,* $\mho_{\mathbf{C}}(\mathbf{D}) = \mathbf{D}$ *then* $\mho_{\mathbf{C}}(\mathbf{E}) = \mho_{\mathbf{D}}(\mathbf{E})$.

Next example shows this phenomenon.[1]

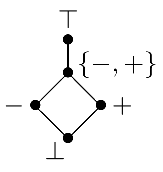
Example 3.6 Consider the usual lattices $\mathbf{D}$ and $\mathbf{E}$ for sign analysis of an integer variable ([8]), depicted below. The concrete domain is $\langle \wp(\mathbb{Z}), \subseteq \rangle$, and concretization and abstraction maps are the most natural. Evidently, $\mathbf{E}$ is an abstraction of $\mathbf{D}$.



$\mathbf{D}$



$\mathbf{E} = \mho_{\mathbf{D}}(\mathbf{E})$



$\mho_{\mathbf{C}}(\mathbf{E})$

Clearly, $\mathbf{D}$ is not a disjunctive abstract interpretation of the concrete domain $\mathbf{C} = \wp(\mathbb{Z})$, since $\gamma(-) \cup \gamma(+) \subset \gamma(- \vee +) = \gamma(\top)$. In this case, $\mho_{\mathbf{C}}(\mathbf{E})$ does not coincide with $\mho_{\mathbf{D}}(\mathbf{E})$. In fact, the disjunctive completion of $\mathbf{E}$ with respect to $\mathbf{D}$, viz. $\mho_{\mathbf{D}}(\mathbf{E})$, is $\mathbf{E}$ itself, while the disjunctive completion with respect to $\mathbf{C}$, viz. $\mho_{\mathbf{C}}(\mathbf{E})$, is the lattice depicted above. □

4 Optimizing Disjunctive Completions

Our goal is to answer to the following question:

> *Given a domain* $\mathbf{D}$ *abstracting* $\mathbf{C}$*, under what hypotheses does exist the least abstraction of* $\mathbf{C}$ *having the same disjunctive completion of* $\mathbf{D}$ *in* $\mathbf{C}$*?*

[1] Throughout the paper, if $\mathbf{X}$ and $\mathbf{Y}$ are sets then we write $\mathbf{X} \subset \mathbf{Y}$ to denote that $\mathbf{X}$ is a proper subset of $\mathbf{Y}$, and $\mathbf{X} \setminus \mathbf{Y}$ to denote their set-difference.

In the following, we positively answer the question above. Firstly, we need two preliminary definitions formally stating by closure operators this question.

Definition 4.1 Given a complete lattice $\mathbf{C}$, $\rho \in \mathbf{uco}(\mathbf{C})$ is *disjunctively optimizable* if $\mho_{\mathbf{C}}(\sqcup\{\eta \in \mathbf{uco}(\mathbf{C}) \mid \mho_{\mathbf{C}}(\eta) = \mho_{\mathbf{C}}(\rho)\}) = \mho_{\mathbf{C}}(\rho)$. □

In the following, for any $\rho \in \mathbf{uco}(\mathbf{C})$, the closure $\sqcup\{\eta \in \mathbf{uco}(\mathbf{C}) \mid \mho_{\mathbf{C}}(\eta) = \mho_{\mathbf{C}}(\rho)\}$ is denoted by $\Omega_{\mathbf{C}}(\rho)$.

Definition 4.2 Assume that $\mathbf{C} \sqsubseteq \mathbf{D}$, and the corresponding $\rho_{\mathbf{D}} \in \mathbf{uco}(\mathbf{C})$ is disjunctively optimizable. The *least disjunctive basis* for $\mathbf{D}$ in $\mathbf{C}$ is the complete lattice $\Omega_{\mathbf{C}}(\mathbf{D})$ given by the set of fixpoints of $\Omega_{\mathbf{C}}(\rho_{\mathbf{D}})$ in $\mathbf{C}$. □

$\Omega_{\mathbf{C}}(\mathbf{D})$ is therefore the most abstract domain such that $\mho_{\mathbf{C}}(\Omega_{\mathbf{C}}(\mathbf{D})) = \mho_{\mathbf{C}}(\mathbf{D})$. Being $\Omega_{\mathbf{C}}(\mathbf{D}) = (\Omega_{\mathbf{C}}(\rho_{\mathbf{D}}))(\mathbf{C})$, the above definition implies that $\Omega_{\mathbf{C}}(\mathbf{D})$ is a subset of $\mathbf{C}$. Obviously, any other lattice isomorphic to $\Omega_{\mathbf{C}}(\mathbf{D})$ can be considered in all respects as the least disjunctive basis.

The following proposition provides an alternative characterization for the operator $\Omega_{\mathbf{C}}$.

Proposition 4.3 *$\rho \in \mathbf{uco}(\mathbf{C})$ is disjunctively optimizable iff there exists a unique element $\Omega_{\mathbf{C}}(\rho) \in \mathbf{uco}(\mathbf{C})$ such that:*

(i) $\mho_{\mathbf{C}}(\Omega_{\mathbf{C}}(\rho)) = \mho_{\mathbf{C}}(\rho)$;
(ii) $\forall \eta \in \mathbf{uco}(\mathbf{C}).\ \mho_{\mathbf{C}}(\eta) = \mho_{\mathbf{C}}(\rho) \Rightarrow \eta \sqsubseteq \Omega_{\mathbf{C}}(\rho)$.

Below, we state two theorems, one orthogonal to the other, for the existence of the least disjunctive basis. The first guarantees the existence of the least disjunctive basis for finite abstract domains.

Theorem 4.4 *If $\mathbf{D} \in \mathbf{uco}(\mathbf{C})$ is finite then it is disjunctively optimizable.*

It is important to note that most of the abstract domains used as basis of a static analysis are finite, and hence, by the above result, disjunctively optimizable. For functional languages, these comprise the abstract domains for standard strictness analysis ([4, 21, 22]), and for its generalization of comportment analysis ([10]). For logic languages, ground-dependency analysis, supposedly the most known analysis, involves traditionally finite abstract domains ([1, 6, 16, 19]); in Section 7, we will determine the least disjunctive basis for one of these abstract domains.

Next theorem provides a condition on the concrete domain in order that every of its abstractions is disjunctively optimizable.

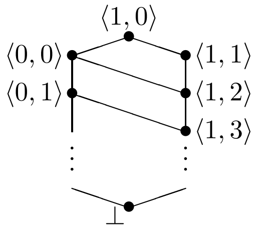
Theorem 4.5 *If $\mathbf{C}$ is dual-algebraic*[2] *then each $\mathbf{D} \in \mathbf{uco}(\mathbf{C})$ is disjunctively optimizable.*

[2] If $\mathbf{C}$ is a complete lattice then the subset of the *dual-compacts* of $\mathbf{C}$ is defined as $\mathbf{dK}(\mathbf{C}) = \{\mathbf{x} \in \mathbf{C} \mid \forall \mathbf{S} \subseteq \mathbf{C}.\ (\mathbf{x} \geq \wedge\mathbf{S}) \Rightarrow (\exists \mathbf{T} \subseteq \mathbf{S}.\ \mathbf{T}\text{ finite } \&\ \mathbf{x} \geq \wedge\mathbf{T})\}$. $\mathbf{C}$ is *dual-algebraic* if for any $\mathbf{x} \in \mathbf{C}$, $\mathbf{x} = \wedge\{\mathbf{z} \in \mathbf{dK}(\mathbf{C}) \mid \mathbf{z} \geq \mathbf{x}\}$.

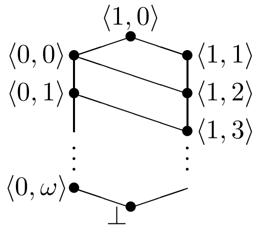
The class of (dual-)algebraic lattices is well known from denotational semantics. It is worth noting that this class is wide enough for practical purposes: in fact, any *well-founded* domain, i.e. any lattice satisfying the descending chain condition, is dual-algebraic, as well as any *collecting* domain, i.e. any powerset $\wp(\mathbf{X})$, for some set $\mathbf{X}$, ordered with the subset or supset relation. The latter case includes the standard concrete domains for collecting semantics in functional and logic programming (e.g. [2, 23]). Complete lattices which are *join-continuous* and that satisfy the *ascending chain condition* are also dual-algebraic.

Dual-algebraicity plays a fundamental rôle in Theorem 4.5. In general, if $\mathbf{C}$ is not dual-algebraic, then it might exist $\rho \in \mathbf{uco}(\mathbf{C})$ non-disjunctively optimizable.

Example 4.6 Let $\mathbf{L}$ be the complete lattice $\{\langle \mathbf{m}, \mathbf{n}\rangle \mid \mathbf{m} \in \{0,1\},\ \mathbf{n} \in \mathbb{N}\} \cup \{\bot\}$, where the ordering relation is determined by the Hasse diagram below.



$\mathbf{L}$



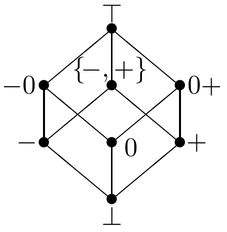
$\mathbf{L}_\omega$

$\mathbf{L}$ is not dual-algebraic: in fact, it is simple to verify that $\mathbf{dK}(\mathbf{L}) = \{\langle 1,0\rangle\} \cup \{\langle 0,\mathbf{n}\rangle\}_{\mathbf{n}\in\mathbb{N}}$, and if $\mathbf{n} > 0$ then $\langle 1,\mathbf{n}\rangle < \wedge\{\mathbf{z} \in \mathbf{dK}(\mathbf{L}) \mid \mathbf{z} \geq \langle 1,\mathbf{n}\rangle\}$. For any $\mathbf{k} \in \mathbb{N}$, consider the closure $\rho_\mathbf{k} = \{\langle 1,\mathbf{n}\rangle\}_{\mathbf{n}\in\mathbb{N}} \cup \{\langle 0,\mathbf{n}\rangle\}_{\mathbf{k}\leq\mathbf{n}} \cup \{\bot\}$. It is clear that for any $\mathbf{k} \in \mathbb{N}$, $\mho_\mathbf{L}(\rho_\mathbf{k}) = \mathbf{L}$, and $(\sqcup_{\mathbf{k}\in\mathbb{N}}\rho_\mathbf{k})(\mathbf{L}) = \{\langle 1,\mathbf{n}\rangle\}_{\mathbf{n}\in\mathbb{N}} \cup \{\bot\} = (\mho_\mathbf{L}(\sqcup_{\mathbf{k}\in\mathbb{N}}\rho_\mathbf{k}))(\mathbf{L})$. If, by contradiction, $\Omega_\mathbf{L}(\mathbf{L})$ exists, then $\sqcup_{\mathbf{k}\in\mathbb{N}}\rho_\mathbf{k} \sqsubseteq \Omega_\mathbf{L}(\mathbf{L})$. But, since $\mho_\mathbf{L}$ is monotonic, we should have $\mho_\mathbf{L}(\sqcup_{\mathbf{k}\in\mathbb{N}}\rho_\mathbf{k}) \sqsubseteq \mho_\mathbf{L}(\Omega_\mathbf{L}(\mathbf{L})) = \mathbf{L}$, i.e. we should obtain the contradiction $\mathbf{L} \subseteq \{\langle 1,\mathbf{n}\rangle\}_{\mathbf{n}\in\mathbb{N}} \cup \{\bot\}$.

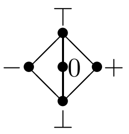
If we consider the lattice $\mathbf{L}_\omega$ depicted above, and obtained from $\mathbf{L}$ by adding the element $\langle 0,\omega\rangle$, it is possible to check that $\mathbf{L}_\omega$ is dual-algebraic (although it does not satisfy the descending chain condition). In this case, $\Omega_{\mathbf{L}_\omega}(\mathbf{L}_\omega)$ exists, and it is $\{\langle 1,\mathbf{n}\rangle\}_{\mathbf{n}\in\mathbb{N}} \cup \{\bot\} \cup \{\langle 0,\omega\rangle\}$. Indeed, for any $\mathbf{n} \in \mathbb{N}$, the lacking element $\langle 0,\mathbf{n}\rangle$ is obtained by disjunctive completion of $\mathbf{L}_\omega$ as $\langle 0,\omega\rangle \vee \langle 1,\mathbf{n}+2\rangle$. Evidently, this is the least closure whose disjunctive completion is $\mathbf{L}_\omega$. □

From now on, whenever we will speak about least disjunctive bases we will suppose that the conditions for their existence hold.

Example 4.7 Consider the domain $\mathbf{D}$ of Example 3.6. It is immediate to check that its disjunctive completion $\mho_\mathbf{C}(\mathbf{D})$ (with respect to $\mathbf{C} = \wp(\mathbb{Z})$) is the lattice depicted below.



$\mho_\mathbf{C}(\mathbf{D})$



$\Omega_\mathbf{C}(\mathbf{D})$

By Theorem 4.4 or Theorem 4.5, its least disjunctive basis $\Omega_{\mathbf{C}}(\mathbf{D})$ (in $\mathbf{C}$) exists. Indeed, it is easy to verify that $\Omega_{\mathbf{C}}(\mathbf{D})$ is the lattice depicted above, which is a proper abstraction of $\mathbf{D}$. □

It is worth noting that the least disjunctive basis operator depends on the fixed concrete domain of reference (an example will be given at the end of Section 7), unless disjunctive abstract interpretations are considered.

Proposition 4.8 *If* $\mathbf{C} \sqsubseteq \mathbf{D} \sqsubseteq \mathbf{E}$ *and* $\mho_{\mathbf{C}}(\mathbf{D}) = \mathbf{D}$ *then* $\Omega_{\mathbf{C}}(\mathbf{E}) = \Omega_{\mathbf{D}}(\mathbf{E})$.

Join-irreducible elements[3] play an important rôle in the computation of the least disjunctive basis. In fact, the least disjunctive basis of an abstract domain $\mathbf{D}$ which is disjunctive, is precisely the Moore-closure of the set $\mathbf{JI_D}$ of the join-irreducible elements of $\mathbf{D}$.

Theorem 4.9 *If* $\mathbf{C} \sqsubseteq \mathbf{D}$ *and* $\mho_{\mathbf{C}}(\mathbf{D}) = \mathbf{D}$ *then* $\Omega_{\mathbf{C}}(\mathbf{D}) = \mathcal{M}(\mathbf{JI_D})$.[4]

Obviously, since $\mho_{\mathbf{C}}(\mathbf{C}) = \mathbf{C}$, the least disjunctive basis of every concrete domain is just the Moore-closure of its join-irreducible elements. Theorem 4.9 has another interesting consequence.

Corollary 4.10 *If* $\mathbf{C} \sqsubseteq \mathbf{D}$ *then* $\Omega_{\mathbf{C}}(\mathbf{D})(= \Omega_{\mathbf{C}}(\mho_{\mathbf{C}}(\mathbf{D}))) = \mathcal{M}(\mathbf{JI}_{\mho_{\mathbf{C}}(\mathbf{D})})$.

In other terms, the least disjunctive basis of an abstract domain $\mathbf{D}$ can always be computed by means of the method of the join-irreducible elements, computing the Moore-closure of the join-irreducible elements of the disjunctive completion of $\mathbf{D}$. Obviously, this is always theoretically possible, but hardly feasible, because of the (usually exponential) size of $\mho_{\mathbf{C}}(\mathbf{D})$. Indeed, when $\mho_{\mathbf{C}}(\mathbf{D})$ is finite and isomorphic to a powerset, $|\Omega_{\mathbf{C}}(\mathbf{D})| = \mathbf{log}(|\mho_{\mathbf{C}}(\mathbf{D})|) + \mathbf{k}$, where $\mathbf{k}$ is a constant.

5 Algebraic Properties and Compositionality

In this section, we study the algebraic properties of the least disjunctive basis with respect to disjunctive completion and reduced product of abstract interpretations.

Proposition 5.1 *Assume that* $\mathbf{C} \sqsubseteq \mathbf{D}, \mathbf{E}$*,* $\top$ *is the most abstract interpretation of* $\mathbf{C}$*, and* $\Omega_{\mathbf{C}}(\mathbf{D}), \Omega_{\mathbf{C}}(\mathbf{E})$ *exist. Then,*

(a) $\mathbf{D} \sqsubseteq \Omega_{\mathbf{C}}(\mathbf{D})$*;*
(b) $\Omega_{\mathbf{C}}(\Omega_{\mathbf{C}}(\mathbf{D})) = \Omega_{\mathbf{C}}(\mathbf{D})$*;*
(c) $\Omega_{\mathbf{C}}(\top) = \top$*;*
(d) $\Omega_{\mathbf{C}}(\mho_{\mathbf{C}}(\mathbf{D})) = \Omega_{\mathbf{C}}(\mathbf{D})$*;*

[3] An element $\mathbf{x}$ of a complete lattice $\mathbf{L}$ is (*completely*) *join-irreducible* if $\forall \mathbf{Y} \subseteq \mathbf{L}$. $(\mathbf{x} = \vee \mathbf{Y} \Rightarrow \mathbf{x} \in \mathbf{Y})$. $\mathbf{JI_L}$ denotes the set of join-irreducible elements of $\mathbf{L}$. Note that $\bot_{\mathbf{L}} \notin \mathbf{JI_L}$.

[4] Note that this result does not hold if the least disjunctive basis $\Omega_{\mathbf{C}}(\mathbf{D})$ does not exist. For instance, in Example 4.6, $\mathbf{JI_L} = \{\langle 1, \mathbf{n}\rangle\}_{\mathbf{n} \in \mathbb{N} \setminus \{0\}}$, but, $\mathcal{M}(\mathbf{JI_L}) = \{\langle 1, \mathbf{n}\rangle\}_{\mathbf{n} \in \mathbb{N}} \cup \{\bot\}$ is not the least disjunctive basis for $\mathbf{L}$, as shown in that example.

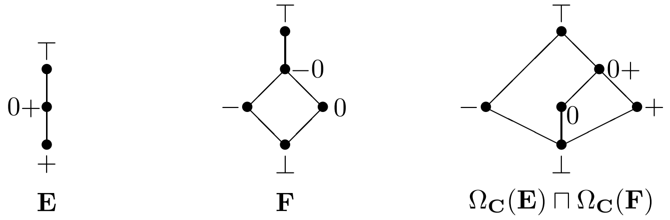
(e) $\Omega_{\mathbf{C}}(\mathbf{D}) = \Omega_{\mathbf{C}}(\mathbf{E}) \Rightarrow \Omega_{\mathbf{C}}(\mathbf{D} \sqcup \mathbf{E}) = \Omega_{\mathbf{C}}(\mathbf{D})$*;*
(f) $\mho_{\mathbf{C}}(\mathbf{D}) = \mho_{\mathbf{C}}(\mathbf{E}) \Leftrightarrow \Omega_{\mathbf{C}}(\mathbf{D}) = \Omega_{\mathbf{C}}(\mathbf{E})$*;*
(g) $\mho_{\mathbf{C}}(\mathbf{D} \sqcap \mathbf{E}) = \mho_{\mathbf{C}}(\Omega_{\mathbf{C}}(\mathbf{D}) \sqcap \Omega_{\mathbf{C}}(\mathbf{E}))$.

Combining points *(f)* and *(g)* above, we get an interesting form of *compositionality* of the least disjunctive basis operator with respect to the reduced product of abstract domains.

Corollary 5.2 $\Omega_{\mathbf{C}}(\mathbf{D} \sqcap \mathbf{E}) = \Omega_{\mathbf{C}}(\Omega_{\mathbf{C}}(\mathbf{D}) \sqcap \Omega_{\mathbf{C}}(\mathbf{E}))$.

The following is a simple example exploiting the above result on compositionality.

Example 5.3 Suppose that the domain $\mathbf{D}$ of Example 3.6 has been incrementally designed by reduced product of the domains $\mathbf{E}$ and $\mathbf{F}$ given below (the concrete domain is $\mathbf{C} = \wp(\mathbb{Z})$).



$\mathbf{E}$ $\mathbf{F}$ $\Omega_{\mathbf{C}}(\mathbf{E}) \sqcap \Omega_{\mathbf{C}}(\mathbf{F})$

By Corollary 5.2, we can compositionally compute the least disjunctive basis $\Omega_{\mathbf{C}}(\mathbf{D})$ (in Example 4.7) of $\mathbf{D}$ from the least disjunctive bases $\Omega_{\mathbf{C}}(\mathbf{E})$ and $\Omega_{\mathbf{C}}(\mathbf{F})$ of its factors. Indeed, the domain $\Omega_{\mathbf{C}}(\mathbf{E}) \sqcap \Omega_{\mathbf{C}}(\mathbf{F})$, which is depicted above, is a proper abstraction of the starting domain $\mathbf{D} = \mathbf{E} \sqcap \mathbf{F}$, and therefore the task of computing the least disjunctive basis of $\Omega_{\mathbf{C}}(\mathbf{E}) \sqcap \Omega_{\mathbf{C}}(\mathbf{F})$ is more simple. □

Domain decomposition by complementation ([5]) and least disjunctive bases can be combined to exploit this form of compositionality of the least disjunctive basis operator. Indeed, complementation provides binary decompositions of abstract domains, and therefore least disjunctive bases can be computed compositionally.

It is important to remark that the least disjunctive basis operator is neither monotonic nor anti-monotonic (hence it is not a closure), as shown below.

Example 5.4 Consider the abstract domains $\mathbf{D}$ of Example 3.6 and $\mathbf{E}$ of Example 5.3, where $\mathbf{D} \sqsubseteq \mathbf{E}$ (the concrete domain is $\mathbf{C} = \wp(\mathbb{Z})$). The least disjunctive basis $\Omega_{\mathbf{C}}(\mathbf{D})$ is in Example 4.7, while it is simple to check that $\Omega_{\mathbf{C}}(\mathbf{E}) = \mathbf{E}$. This proves that the least disjunctive basis operator is neither monotonic nor anti-monotonic, since $\Omega_{\mathbf{C}}(\mathbf{D})$ and $\Omega_{\mathbf{C}}(\mathbf{E})$ are incomparable abstractions of $\mathbf{C}$. □

6 Functional Programming: Optimizing Comportment Analysis

In this section, we apply the theory of the least disjunctive basis to the *comportment* analysis, designed by Cousot and Cousot in [10] to generalize Mycroft's *strictness*

truth	$\gamma^{\beta\to\beta}(\mathbf{top})$	$= \mathbf{D}^{\beta\to\beta}$
strictness	$\gamma^{\beta\to\beta}(\mathbf{str})$	$= \{\mathbf{f} \mid \mathbf{f}(\bot) = \bot\}$
totality	$\gamma^{\beta\to\beta}(\mathbf{tot})$	$= \{\mathbf{f} \mid \forall \mathbf{x} \in \mathbf{D}^{\beta} \setminus \{\bot\}.\ \mathbf{f}(\mathbf{x}) \neq \bot\}$
identity	$\gamma^{\beta\to\beta}(\mathbf{ide})$	$= \{\mathbf{f} \mid \forall \mathbf{x} \in \mathbf{D}^{\beta}.\ \mathbf{f}(\mathbf{x}) = \bot \Leftrightarrow \mathbf{x} = \bot\}$
divergence	$\gamma^{\beta\to\beta}(\mathbf{div})$	$= \{\mathbf{f} \mid \forall \mathbf{x} \in \mathbf{D}^{\beta}.\ \mathbf{f}(\mathbf{x}) = \bot\}$
convergence	$\gamma^{\beta\to\beta}(\mathbf{con})$	$= \{\mathbf{f} \mid \forall \mathbf{x} \in \mathbf{D}^{\beta}.\ \mathbf{f}(\mathbf{x}) \neq \bot\}$
falsity	$\gamma^{\beta\to\beta}(\emptyset)$	$= \emptyset$

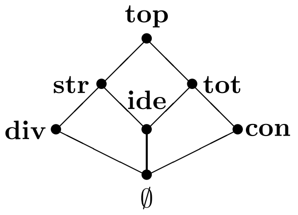
Table 1: Basic comportment analysis $\mathcal{B}_{\mathcal{C}}$.

and *termination* analysis ([21, 22]), Wadler and Hughes' *projection* analysis ([27]), and Hunt's *PER* analysis ([15]). The comportment analysis applies to higher order monomorphically typed lazy functional programming languages.

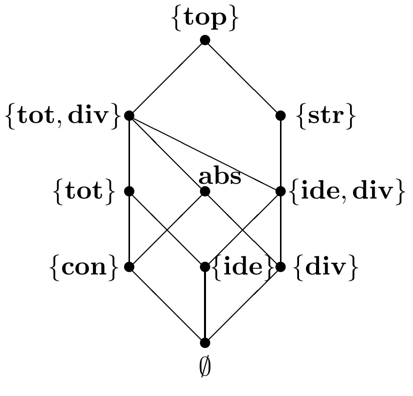
To illustrate Cousot and Cousot's comportment analysis, we consider abstract interpretation of a simply typed lambda calculus with basic types β. Denote $\mathbf{D}^{\tau}$ the domain of values of a type τ, and by $\bot$ its bottom element. For simplicity, we will consider abstractions of function basic types $\beta \to \beta$ (i.e., elements in $\mathbf{D}^{\beta\to\beta} = \mathbf{D}^{\beta} \to \mathbf{D}^{\beta}$, the lattice of monotonic functions from $\mathbf{D}^{\beta}$ to $\mathbf{D}^{\beta}$ ordered pointwise). The abstract domain $\mathcal{B}_{\mathcal{C}}$ below represents the lattice of *basic comportment* analysis, ordered with respect to the approximation order, for function basic types $\beta \to \beta$.

The meaning of basic comportments in $\mathcal{B}_{\mathcal{C}}$ is given in Table 1, in terms of a concretization function $\gamma^{\beta\to\beta}$ mapping basic comportments into $\langle \wp(\mathbf{D}^{\beta\to\beta}), \subseteq\rangle$, which is the concrete domain of the collecting semantics.

As proved by Cousot and Cousot in [10], more precise comportment properties for higher-order functional languages can be characterized by disjunctive completion of the lattice $\mathcal{B}_{\mathcal{C}}$ of basic comportment analysis. In this case, the meaning of sets Ψ of basic comportments is given by a concretization $\gamma^{\wp}$ such that $\gamma^{\wp}(\Psi) = \cup\{\gamma^{\beta\to\beta}(\psi) \mid \psi \in \Psi\}$. The lattice $\mathcal{C}$ below, ordered by the approximation order, corresponds precisely to this (extended) comportment analysis for function basic types $\beta \to \beta$. It is obtained by a powerset completion (e.g. anti-chain) and reduction. The new element **abs** corresponds here to the set of basic comportments $\{\mathbf{con}, \mathbf{div}\}$ and represents *absence.*



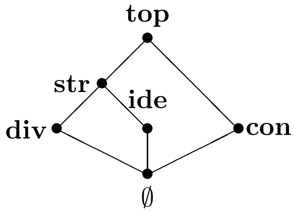
Basic comportment $\mathcal{B}_{\mathcal{C}}$



Comportment $\mathcal{C}$

The disjunctive completion of $\mathcal{B}_\mathcal{C}$ is the lattice $\mathcal{C}$ above since: (i) $\gamma^{\wp}(\{\mathbf{con}, \mathbf{div}\}) \subset \gamma^{\beta\to\beta}(\mathbf{con} \vee \mathbf{div})$, (ii) $\gamma^{\wp}(\{\mathbf{ide}, \mathbf{div}\}) \subset \gamma^{\beta\to\beta}(\mathbf{ide} \vee \mathbf{div})$, and (iii) $\gamma^{\wp}(\{\mathbf{tot}, \mathbf{div}\}) \subset \gamma^{\beta\to\beta}(\mathbf{tot} \vee \mathbf{div})$, while for any other $\Psi \subseteq \mathcal{B}_\mathcal{C}$, $\gamma^{\wp}(\Psi) = \gamma^{\beta\to\beta}(\vee\Psi)$. For instance, for any basic type β, the identity map $\lambda \mathbf{x}^\beta.\mathbf{x}^\beta$ is such that $\lambda \mathbf{x}^\beta.\mathbf{x}^\beta \in \gamma^{\beta\to\beta}(\mathbf{con} \vee \mathbf{div}) = \gamma^{\beta\to\beta}(\mathbf{top}) = \mathbf{D}^{\beta\to\beta}$, whilst $\lambda \mathbf{x}^\beta.\mathbf{x}^\beta \notin \gamma^{\wp}(\{\mathbf{con}, \mathbf{div}\}) = \gamma^{\beta\to\beta}(\mathbf{con}) \cup \gamma^{\beta\to\beta}(\mathbf{div})$.

To find out the least disjunctive basis of $\mathcal{B}_\mathcal{C}$ in $\wp(\mathbf{D}^{\beta\to\beta})$, viz. the least disjunctive basis for the lattice of basic comportment analysis (which, by Theorem 4.4 or Theorem 4.5, exists since $\mathcal{B}_\mathcal{C}$ is a finite lattice or $\langle\wp(\mathbf{D}^{\beta\to\beta}), \subseteq\rangle$ is dual-algebraic), we simply apply Corollary 4.10, that is we compute the Moore-closure of the join-irreducible elements of the disjunctive completion $\mathcal{C}$ of $\mathcal{B}_\mathcal{C}$. It is straightforward to verify that the least disjunctive basis $\Omega_{\wp(\mathbf{D}^{\beta\to\beta})}(\mathcal{B}_\mathcal{C})$ is the lattice depicted below.



The least disjunctive basis $\Omega_{\wp(\mathbf{D}^{\beta\to\beta})}(\mathcal{B}_\mathcal{C})$

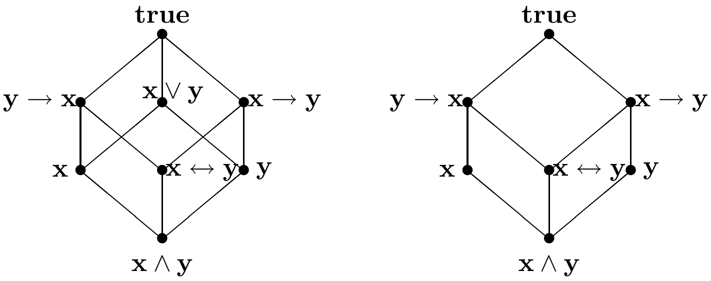
The least disjunctive basis $\Omega_{\wp(\mathbf{D}^{\beta\to\beta})}(\mathcal{B}_\mathcal{C})$ is therefore a proper abstraction of the original basis $\mathcal{B}_\mathcal{C}$ of basic comportments. In fact, the element **tot** denoting totality does not belong to the least disjunctive basis, since it can be recovered as *lub* of the elements **ide** and **con** representing identity and convergence, respectively.

7 Logic Programming: Optimizing Disjunctive Ground-Dependency Analysis

In this section, we apply the theory of the least disjunctive basis to **Pos**, a well known relational domain of propositional formulae for ground-dependency analysis of (constraint) logic programs ([1, 6, 19]). The disjunctive completion of **Pos** has been recently studied by Filé and Ranzato in [12], where it has been shown that it is strictly more precise than **Pos** itself. In a sense, this was a surprising result, since the fact that **Pos** is closed under logical disjunction should lead to an opposite conclusion. Therefore, static analyses based on the disjunctive completion of **Pos** are more precise. We show that the least disjunctive basis for the disjunctive completion of **Pos** is the domain **Def**, which is a proper abstraction of **Pos**. **Def** is a domain of propositional formulae already existing in literature, introduced by Dart in [11] for groundness analysis in deductive databases, and used by Marriott and Søndergaard in [19] for ground-dependency analysis of logic programs. Recently, Armstrong *et al.* in [1] investigated various representations for the formulae in **Pos** and **Def**, and they experimentally compared the resulting precision and efficiency of these different static analyses. They showed that analyses using **Pos** achieve a higher precision than those using **Def**, although there is an additional cost relatively small. However, this additional cost becomes relevant when lifting **Pos** and **Def** to the powerset, due to the combinatorial explosion of the disjunctive completion. In view of the work in [1], the results of this section gain an important and significative practical impact: the disjunctive ground-dependency analysis of logic programs can be always obtained by disjunctive completion of **Def**, without losing precision and at a lower cost with respect to the disjunctive completion of **Pos**. Moreover, this completes the understanding of the problem, since it is the best that one can do in this direction.

The domains *Pos* and *Def*. Let *Var* be a countable set of variables, and let *VI* be any (non-empty) finite subset of *Var* containing the variables of interest. As usual, variables are denoted by $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \ldots$. We assume that the concrete domain of computation of a given logic program is the powerset $\wp(\mathbf{Sub})$ of idempotent substitutions, ordered with set-theoretic inclusion. Every substitution $\sigma \in \mathbf{Sub}$ is an idempotent function mapping each $\mathbf{x} \in \mathit{Var}$ to a term $\sigma(\mathbf{x})$ built on the variables in *Var*, such that $\sigma(\mathbf{x}) \neq \mathbf{x}$ for a finite number of variables $\mathbf{x}$. A substitution σ is typically specified by listing its non-trivial bindings, viz. $\sigma = \{\mathbf{x}/\sigma(\mathbf{x}) \mid \sigma(\mathbf{x}) \neq \mathbf{x}\}$.

Pos is the finite lattice (indeed Boolean lattice) of *positive* Boolean functions on *VI*, where a Boolean function $\mathbf{f}$ is positive if $\mathbf{f}(\mathbf{true}, \ldots, \mathbf{true}) = \mathbf{true}$. Obviously, the order of **Pos** is given by the logical consequence $\models$, and, *lub* and *glb* on **Pos** are given by logical disjunction and conjunction, respectively. **Def** is the finite lattice of positive Boolean functions on *VI* whose models are closed under intersection. Formulae in **Def** are called *definite*. For more details about **Pos** and **Def** see [1, 19]. It is well known that Boolean functions can be represented by means of propositional formulae. Thus, in the following, we will use propositional formulae over *VI* to represent Boolean functions in **Pos** and **Def**. Below, **Pos** and **Def** are depicted for $\mathit{VI} = \{\mathbf{x}, \mathbf{y}\}$.



The domains **Pos** and **Def** for $\mathit{VI} = \{\mathbf{x}, \mathbf{y}\}$

As observed in [1], **Def** is a meet-sublattice of **Pos**. Further, the top Boolean function **true** is in **Def**. Hence, **Def** is a Moore-family of **Pos**, namely, being (the set of fixpoints of) a closure operator on **Pos**, it is an abstract interpretation of **Pos**. The abstraction and concretization maps between **Pos**, **Def** and $\wp(\mathbf{Sub})$ are well known, and can be found, e.g., in [6, 19]. For instance, assuming $\mathit{VI} = \{\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}\}$, the formula $\mathbf{x} \wedge (\mathbf{y} \leftrightarrow \mathbf{z})$ is an element of **Pos** (and **Def**) that represents the substitutions σ such that for any instance σ' of σ: (i) the term $\sigma'(\mathbf{x})$ is ground; (ii) $\sigma'(\mathbf{y})$ is ground iff also $\sigma'(\mathbf{z})$ is ground. In particular,[5] $\sigma_1 = \{\mathbf{x}/\mathbf{a}, \mathbf{y}/\mathbf{b}, \mathbf{z}/\mathbf{c}\}$ and $\sigma_2 = \{\mathbf{x}/\mathbf{a}, \mathbf{y}/\mathbf{w}, \mathbf{z}/\mathbf{w}, \mathbf{v}/\mathbf{u}\}$ satisfy this property. Thus, $\{\sigma_1, \sigma_2\} \subseteq \gamma(\mathbf{x} \wedge (\mathbf{y} \leftrightarrow \mathbf{z}))$.

[5] By $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$, we denote ground terms.

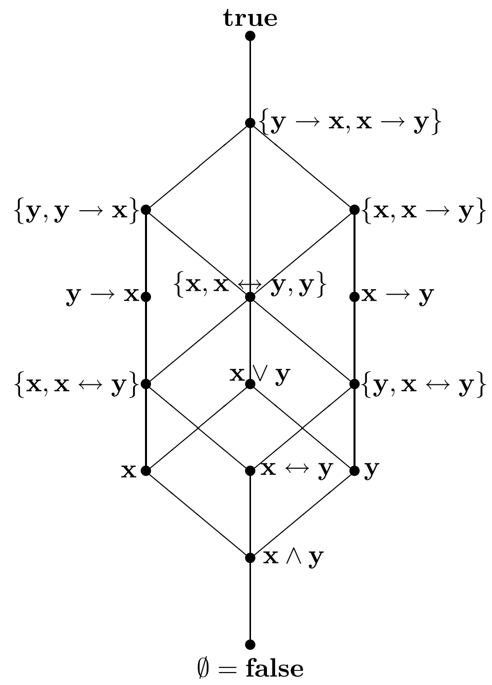
The least disjunctive basis of *Pos* is *Def*. Filé and Ranzato showed in [12] that $\mho_{\wp(\mathbf{Sub})}(\mathbf{Pos}) \sqsubset \mathbf{Pos}$, i.e. there is a strict improvement in precision lifting **Pos** to its disjunctive completion. For example, by considering as variables of interest *VI*= $\{\mathbf{x}, \mathbf{y}\}$, the logical disjunction $(\mathbf{x} \rightarrow \mathbf{y}) \vee (\mathbf{y} \rightarrow \mathbf{x})$ in **Pos** does not represent the concrete disjunction of the two formulae $\mathbf{x} \rightarrow \mathbf{y}$ and $\mathbf{y} \rightarrow \mathbf{x}$, i.e. the union of their concretizations. In fact, $\gamma(\mathbf{x} \rightarrow \mathbf{y}) \cup \gamma(\mathbf{y} \rightarrow \mathbf{x}) \subset \gamma((\mathbf{x} \rightarrow \mathbf{y}) \vee (\mathbf{y} \rightarrow \mathbf{x})) = \gamma(\mathbf{true})$: $\sigma = \{\mathbf{x}/\mathbf{v}, \mathbf{y}/\mathbf{w}\}$ is such that $\sigma \in \gamma((\mathbf{x} \rightarrow \mathbf{y}) \vee (\mathbf{y} \rightarrow \mathbf{x})) \setminus \gamma(\mathbf{x} \rightarrow \mathbf{y}) \cup \gamma(\mathbf{y} \rightarrow \mathbf{x})$.

Sets of positive formulae for which logical (i.e., in **Pos**) and concrete (i.e., in $\wp(\mathbf{Sub})$) disjunctions coincide have been characterized as follows (cf. [13]).

Theorem 7.1 ([13]) *If* $\emptyset \neq \Phi \subseteq \mathbf{Pos}$ *then*

$$\cup\{\gamma(\mathbf{f}) \mid \mathbf{f} \in \Phi\} = \gamma(\vee\Phi) \Leftrightarrow \forall \mathbf{g} \in \mathbf{Def}.\ ((\mathbf{g} \models \vee\Phi) \Rightarrow (\exists \mathbf{f} \in \Phi.\ \mathbf{g} \models \mathbf{f})).$$

By Theorem 7.1, the disjunctive completion of **Pos** (after reduction), for *VI*= $\{\mathbf{x}, \mathbf{y}\}$, is the lattice depicted below.



$\mho_{\wp(\mathbf{Sub})}(\mathbf{Pos})$ for *VI*= $\{\mathbf{x}, \mathbf{y}\}$

By Theorem 4.4 or Theorem 4.5, $\Omega_{\wp(\mathbf{Sub})}(\mathbf{Pos})$ exists, since **Pos** is a finite lattice or $\langle \wp(\mathbf{Sub}), \subseteq \rangle$ is dual-algebraic. The complexity of the simple case of two variables shows how difficult is the application of the method of join irreducible elements of Corollary 4.10 to compute the least disjunctive basis of **Pos**. The main result of this section can however be proved directly on the definition of **Def** and **Pos**.

Theorem 7.2 $\Omega_{\wp(\mathbf{Sub})}(\mathbf{Pos}) = \mathbf{Def}$.

Indeed, it is simple to verify on the previous diagrams for the case of two variables *VI*$= \{\mathbf{x}, \mathbf{y}\}$, that **Def** is the least abstraction of **Pos** having the same disjunctive completion. This particular case is also verifiable by applying Corollary 4.10: in fact, **Def** is precisely the Moore-closure of the join-irreducible elements of $\mho_{\wp(\mathbf{Sub})}(\mathbf{Pos})$. Moreover, $\Omega_{\wp(\mathbf{Sub})}(\mathbf{Pos}) = \mathbf{Def}$, while $\Omega_{\mathbf{Pos}}(\mathbf{Pos}) = \mathcal{M}(\mathbf{JI}_{\mathbf{Pos}})$, and for the case *VI*$= \{\mathbf{x}, \mathbf{y}\}$, the Moore-closure of the join-irreducible elements of **Pos** clearly does not coincide with **Def** (for instance, $\mathbf{y} \rightarrow \mathbf{x} \in \mathbf{Def} \setminus \mathcal{M}(\mathbf{JI}_{\mathbf{Pos}})$). This proves the dependency of the least disjunctive basis operator on the concrete domain of reference, as postulated in Section 4.

8 Related Work

To the best of our knowledge, this is the first systematic characterization of least disjunctive bases for disjunctive completion in abstract interpretation. However, the use of join-irreducible elements to represent disjunctive properties is definitely not new, in particular in relation with the work of Nielson. Join-irreducible elements were firstly investigated in the context of abstract interpretation in [24], with the aim of giving an alternative (more concise) representation for the relational (Hoare) powerdomain in analysis of typed functional languages. We extend Nielson's idea in the definition of our notion of least disjunctive basis. Least disjunctive bases are more general in this sense, since join-irreducible elements can only represent domains which are already disjunctive. The least disjunctive basis operator is applicable to arbitrary abstract interpretations, provided that the hypotheses of Theorem 4.4 or Theorem 4.5 are satisfied. In [24], Nielson investigates also the situation where the abstraction function maps join-irreducibile elements to join-irreducibile elements, defining the notion of *expected form* for an abstract interpretation, further studied in [26]. This is a related topic, and provides an interesting application of least disjunctive bases. Nielson suggests the use of expected forms in order to simplify the implementation of functionals induced in abstract interpretation of denotational semantics. The aim of expected forms is therefore similar to that of least disjunctive bases, both providing sensible simplifications in abstract interpretation design. In particular, some expected forms defined on collecting semantics, i.e. on some powerset domain, (e.g. for **cond** in [26]) can be viewed as functionals on the least disjunctive basis of the abstract domain.

Acknowledgments. We would like to thank Patrick Cousot, Gilberto Filé and Harald Søndergaard for many stimulating discussions on the subject of this paper.

References

1. T. Armstrong, K. Marriott, P. Schachte, and H. Søndergaard. Boolean functions for dependency analysis: algebraic properties and efficient representation. In *Proc. of SAS '94*, LNCS 864, pp. 266–280, 1994. To appear in *Sci. of Comp. Programming.*
2. R. Barbuti, R. Giacobazzi, and G. Levi. A general framework for semantics-based bottom-up abstract interpretation of logic programs. *ACM TOPLAS*, 15(1):133–181, 1993.
3. G. Birkhoff. *Lattice Theory.* AMS Colloq. Publ., vol. XXV, 3rd ed., 1967.

4. G.L. Burn, C.L. Hankin, and S. Abramsky. Strictness analysis of higher-order functions. *Sci. of Comp. Programming*, 7:249–278, 1986.
5. A. Cortesi, G. Filé, R. Giacobazzi, C. Palamidessi, and F. Ranzato. Complementation in abstract interpretation. In *Proc. of SAS '95*, LNCS 983, pp. 100–117, 1995.
6. A. Cortesi, G. Filé, and W. Winsborough. Prop revisited: propositional formula as abstract domain for groundness analysis. In *Proc. of LICS '91*, pp. 322–327, 1991.
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. of ACM POPL '77*, pp. 238–252. 1977.
8. P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Proc. of ACM POPL '79*, pp. 269–282, 1979.
9. P. Cousot and R. Cousot. Abstract interpretation and application to logic programs. *J. of Logic Programming*, 13(2,3):103–179, 1992.
10. P. Cousot and R. Cousot. Higher-order abstract interpretation (and application to comportment analysis generalizing strictness, termination, projection and PER analysis of functional languages). In *Proc. of IEEE ICCL '94*, pp. 95–112, 1994.
11. P. Dart. On derived dependencies and connected databases. *J. of Logic Programming*, 11(2):163–188, 1991.
12. G. Filé and F. Ranzato. Improving abstract interpretations by systematic lifting to the powerset. In *Proc. of ILPS '94*, pp. 655–669, 1994.
13. G. Filé and F. Ranzato. The powerset operator on abstract interpretations. Tech. Rep., Dip. di Matematica Pura ed Appl., U. di Padova, 1995.
14. G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, and D.S. Scott. *A Compendium of Continuous Lattices*. Springer-Verlag, 1980.
15. S. Hunt. PERs generalize projections for strictness analysis. In *Proc. of the 1990 Glasgow Funct. Progr. Workshop*, pp. 156–168. Springer Workshops in Comp., 1990.
16. D. Jacobs and A. Langen. Static analysis of logic programs for independent AND-parallelism. *J. of Logic Programming*, 13(2,3):154–165, 1992.
17. T.P. Jensen. Disjunctive strictness analysis. In *Proc. of LICS '92*, pp. 174–185. 1992.
18. N.D. Jones and S.S. Muchnick. Complexity of flow analysis, inductive assertion synthesis and a language due to Dijkstra. In *Program Flow Analysis: Theory and Applications*, pp. 380–393. Prentice-Hall, 1981.
19. K. Marriott and H. Søndergaard. Precise and efficient groundness analysis for logic programs. *ACM LOPLAS*, 2(1–4):181–196, 1993.
20. J. Morgado. Some results on the closure operators of partially ordered sets. *Port. Math.*, 19(2):101–139, 1960.
21. A. Mycroft. The theory and practice of transforming call-by-need into call-by-value. In *Proc. of the 4th Int. Symp. on Programming*, LNCS 83, pp. 270–281, 1980.
22. A. Mycroft. *Abstract interpretation and optimizing transformations for applicative programs*. Ph.D. Thesis, Dept. of Computer Science, U. of Edinburgh, CST-15-81, 1981.
23. A. Mycroft and F. Nielson. Strong abstract interpretation using power domains. In *Proc. of ICALP '83*, LNCS 154, pp. 536–547, 1983.
24. F. Nielson. *Abstract interpretation using domain theory*. Ph.D. Thesis, Dept. of Computer Science, U. of Edinburgh, CST-31-84, 1984.
25. F. Nielson. Tensor products generalize the relational data flow analysis method. In *Proc. of the 4th Hung. Computer Science Conf.*, pp. 211–225, 1985.
26. F. Nielson. Expected forms of data flow analysis. In *Programs as Data Objects*, LNCS 217, pp. 192–205, 1986.
27. P.L. Wadler and R.J.M. Hughes. Projections for strictness analysis. In *Proc. of FPCA '87*, LNCS 274, pp. 385–407, 1987.
28. M. Ward. The closure operators of a lattice. *Ann. of Math.*, 43(2):191–196, 1942.