The PER Model of Abstract Non-Interference

Sebastian Hunt[1] and Isabella Mastroeni[2]

[1] Department of Computing, School of Informatics, City University, London, UK
`seb@soi.city.ac.uk`

[2] Department of Computing and Information Sciences, Kansas State University
Manhattan, Kansas, USA
`isabellm@cis.ksu.edu`

Abstract. In this paper, we study the relationship between two models of secure information flow: the PER model (which uses equivalence relations) and the abstract non-interference model (which uses upper closure operators). We embed the lattice of equivalence relations into the lattice of closures, re-interpreting abstract non-interference over the lattice of equivalence relations. For narrow abstract non-interference, we show that the new definition is equivalent to the original, whereas for abstract non-interference it is strictly less general. The relational presentation of abstract non-interference leads to a simplified construction of the most concrete harmless attacker. Moreover, the PER model of abstract non-interference allows us to derive unconstrained attacker models, which do not necessarily either observe all public information or ignore all private information. Finally, we show how abstract domain completeness can be used for enforcing the PER model of abstract non-interference.

Keywords: information flow, non-interference, abstract interpretation, language-based security.

1 Introduction

An important task of language based security is to protect confidentiality of data manipulated by computational systems. Namely, it is important to guarantee that no information, about confidential/private data, can be caught by an external viewer. In the standard approach to the confidentiality problem, called *non-interference*, the characterization of attackers does not impose any observational or complexity restriction on the attackers' power. This means that the attackers are *all powerful*: they are modeled without any limitation in their quest to obtain confidential information. For this reason non-interference is an extremely restrictive policy. The problem of refining these security policies is considered as a major challenge in language-based information flow security [17]. Refining security policies means weakening standard non-interference, in such a way that it can be used in practice. Specifically, we need a weaker notion of non-interference where the power of the attacker (or external viewer) is bounded, and where intentional leakage of information is allowed.

Abstract non-interference is introduced [9] for modeling the *secrecy degree* of programs by means of abstract interpretation. In particular, it is possible to characterize the observational capability of the most powerful *harmless* attacker, that is, the most powerful attacker that cannot disclose any confidential information. Moreover, this model also allows one to characterize which aspects of private information can flow during the execution of a given program, when non-interference fails. These two complementary aspects of non-interference have been proved to be adjoint transformers of semantics in [10], where non-interference has been modeled as an abstract domain completeness problem.

In the PER model of secure information flow [18], a generalised notion of non-interference is obtained by using equivalence relations to model attackers. In this paper we show that, since equivalence relations can be viewed as particular types of closures called *partitioning* closures [16], the definitions of narrow and abstract non-interference from [9] can be re-interpreted by using equivalence relations only in place of arbitrary closures. For narrow abstract non-interference, we show that the new definition is equivalent to the original, whereas for abstract non-interference it is strictly less general. The difference lies in the fact that abstract non-interference depends on being able to distinguish properties of *sets* of values, such as intervals, congruences, etc, and this cannot be done with equivalence relations on the underlying set. We then show how the relational presentation of narrow abstract non-interference leads to a simplified construction of the most powerful harmless attacker. Moreover, the generalization of the PER model of secure information flow allows us to derive *unconstrained* attacker models, which do not necessarily either observe all public information or ignore all private information. Finally, we show how abstract domain completeness can be used for enforcing the PER model of abstract non-interference, proving that abstract non-interference corresponds to abstract domain completeness of the corresponding partitioning closures.

2 Mathematical Background

In this paper we use the standard framework of abstract interpretation [5, 7] for modeling the observational capability of attackers. The idea is that, instead of observing the concrete semantics of programs, namely the values of public data, attackers can only observe *properties* of public data, namely an *abstract semantics* of the program. For this reason we model attackers by means of abstract domains. Abstract domains are used for denoting properties of concrete domains, since their mathematical structure guarantees, for each concrete element, the existence of the *best correct approximation* in the abstract domain. This is due to the fact that abstract domains are closed under the concrete greatest lower bound. The relation between abstract and concrete domains is formalized by Galois connections (GC). In GC-based abstract interpretation the concrete domain C and abstract domain A are often assumed to be complete lattices and are related by an abstraction map $\alpha : C \to A$ and concretization map $\gamma : A \to C$ forming a GC $\langle C, \alpha, \gamma, A\rangle$ [5], i.e., for any $x \in C$ and $y \in A$: $\alpha(x) \leq_A y \Leftrightarrow x \leq_C \gamma(y)$. When α is surjective then the GC is said to be a Galois insertion (GI) and uniquely determines an abstract domain. Formally, the *lattice of abstract interpretations of* C is isomorphic to the lattice $uco(C)$ of all the upper closure operators on C [7]. An *upper closure operator* $\rho : C \to C$ on a poset C is monotone, idempotent, and extensive[3]. The dual notion of *lower closure operator* (lco) is a monotone, idempotent and reductive[4] map. Any closure operator is uniquely determined by the set of its fix points $\rho(C)$, which forms an abstract domain. If C is a complete lattice then $\langle uco(C), \sqsubseteq, \sqcup, \sqcap, \top, id\rangle$ is the lattice of upper closures, where $\top \stackrel{\text{def}}{=} \lambda x.\ \top$, $id \stackrel{\text{def}}{=} \lambda x.\ x$, and for every $\rho, \eta \in uco(C)$, $\{\rho_i\}_{i\in I} \subseteq uco(C)$ and $x \in C$: $\rho \sqsubseteq \eta$ iff $\eta(C) \subseteq \rho(C)$; $\sqcup_{i\in I}\rho_i = \bigcap_{i\in I} \rho_i$; and $\sqcap_{i\in I}\rho_i = \mathcal{M}(\bigcup_{i\in I} \rho_i)$, where $\mathcal{M}$ is the operation of closing a domain by concrete greatest lower bound, e.g., intersection on power domains. The *disjunctive completion* of an abstract domain $\rho \in uco(C)$ is the most abstract domain able to represent the concrete disjunction of its objects: $\curlyvee(\rho) = \sqcup\{\eta \in uco(C) | \eta \sqsubseteq \rho \text{ and } \eta \text{ is additive}\}$. ρ is disjunctive (or additive) iff $\curlyvee(\rho) = \rho$ (cf. [7]).

2.1 Equivalence Relations vs Closure Operators

In this section we review the relationships between equivalence relations and upper closures which are key to the development in the rest of the paper.

The lattice of equivalence relations. The equivalence relations on a set C form a lattice $\langle Eq(C), \sqsubseteq, \sqcap, \sqcup, Id_C, All_C\rangle$, where Id_C is the relation that distinguishes all the elements in C, All_C is the relation that cannot distinguish any element in C, and:

- $\mathtt{Q} \sqsubseteq \mathtt{R}$ iff $\mathtt{Q} \subseteq \mathtt{R}$ iff $x\,\mathtt{Q}\,y \ \Rightarrow\ x\,\mathtt{R}\,y$;
- $\mathtt{Q} \sqcap \mathtt{R} = \mathtt{Q} \cap \mathtt{R}$, i.e., $x\,\mathtt{Q} \sqcap \mathtt{R}\,y$ iff $x\,\mathtt{Q}\,y \ \wedge\ x\,\mathtt{R}\,y$;
- $\mathtt{Q} \sqcup \mathtt{R} = \mathbb{T}(\mathtt{Q} \cup \mathtt{R})$, where $x\,\mathtt{Q} \cup \mathtt{R}\,y$ iff $x\,\mathtt{Q}\,y \ \vee\ x\,\mathtt{R}\,y$.

Here $\mathbb{T}(\mathtt{S})$ is the transitive closure of the relation $\mathtt{S}$ (it is easily seen that both $\cup$ and $\mathbb{T}$ preserve symmetry and reflexivity).

Relating equivalence relations and upper closures. In this paper we will generally be concerned with relationships between equivalence relations on a set C and

[3] $\forall x \in C.\ x \leq_C \rho(x)$.
[4] $\forall x \in C.\ x \geq_C \rho(x)$.

upper closure operators on the powerset $\wp(C)$. However, we start by observing the following strong correspondence between ucos (on any lattice) and their own kernels. (Recall that the kernel, $\mathtt{K}_f$, of a function $f : C \to D$, is the equivalence relation on C defined by $x \ \mathtt{K}_f \ y$ iff $f(x) = f(y)$.)

Lemma 1. *Let $\eta, \rho \in uco(C)$. Then $\eta \sqsubseteq \rho$ iff $\mathtt{K}_\eta \sqsubseteq \mathtt{K}_\rho$.*

Next, we recall that there exists an isomorphism between equivalence relations and a subclass of the upper closure operators [16]. In fact, this isomorphism arises from a Galois connection between $Eq(C)$ and $uco(\wp(C))$. For each equivalence relation on a set C, $\mathtt{R} \subseteq C \times C$, we can define an upper closure operator on $\wp(C)$, $Clo^{\mathtt{R}} \in uco(\wp(C))$, and vice versa, from each upper closure operator $\eta \in uco(\wp(C))$ we can define an equivalence relation $Rel^{\eta} \subseteq C \times C$.

Consider an upper closure operator $\eta \in uco(\wp(C))$. We define $Rel^{\eta} \subseteq C \times C$, as $\forall x, y \in C \,.\, x \ Rel^{\eta} \ y \ \Leftrightarrow \ \eta(\{x\}) = \eta(\{y\})$. Proving that Rel^{η} is an equivalence relation is immediate and doesn't depend on the fact that η is a uco, but only on the fact that it is a function.

Consider now an equivalence relation $\mathtt{R} \subseteq C \times C$. We define $Clo^{\mathtt{R}} \in uco(\wp(C))$ as follows: $\forall x \in C \,.\, Clo^{\mathtt{R}}(\{x\}) = [x]_{\mathtt{R}}$ and $\forall X \subseteq C \,.\, Clo^{\mathtt{R}}(X) = \bigcup_{x \in X} [x]_{\mathtt{R}}$. Thus $Clo^{\mathtt{R}}$ is obtained by disjunctive completion of the partition induced by $\mathtt{R}$. Proving that $Clo^{\mathtt{R}}$ is an upper closure operator is immediate. In particular idempotence derives directly from the fact that $\mathtt{R}$ is an equivalence relation.

In [16], $Clo^{\mathtt{R}}$ is identified as the most concrete uco η such that $\mathtt{R} = Rel^{\eta}$. More precisely:

Proposition 2. *Let C be any set.*

1. *The mappings defined above form a Galois connection between the lattice of equivalence relations on C and the lattice of upper closure operators on its powerset. That is, for all $\mathtt{R} \in Eq(C)$, $\eta \in \mathrm{uco}(\wp(C))$: $Clo^{\mathtt{R}} \sqsubseteq \eta \Leftrightarrow \mathtt{R} \sqsubseteq Rel^{\eta}$.*
2. *For all $\mathtt{R} \in Eq(C)$, $Rel^{Clo^{\mathtt{R}}} = \mathtt{R}$.*

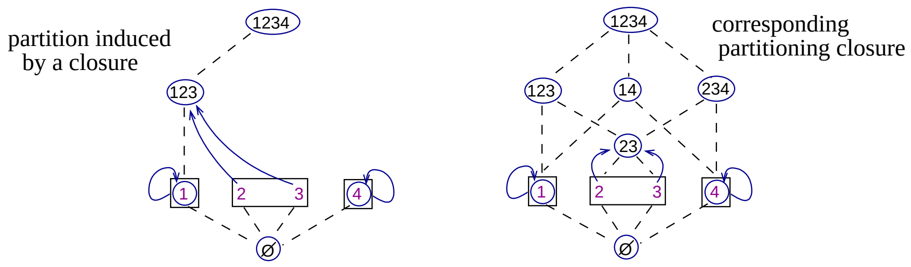
Corollary 3. *Let $\Pi(\eta)$ be defined by $\Pi(\eta) = Clo^{Rel^{\eta}}$.*

1. *$\Pi : \mathrm{uco}(\wp(C)) \to \mathrm{uco}(\wp(C))$ is a lower closure operator.*
2. *For all $\eta \in \mathrm{uco}(\wp(C))$, $\Pi(\eta)$ is the (unique) most concrete closure that induces the same equivalence relation as η ($\mathrm{Rel}^{\eta} = \mathrm{Rel}^{\Pi(\eta)}$).*

The fix points of Π are termed the *partitioning* closures [16].

Proposition 4. *An upper closure operator $\eta \in \mathrm{uco}(\wp(C))$ is partitioning, i.e., $\eta = \Pi(\eta)$, iff it is complemented, namely if $\forall X \in \eta .\ \overline{X} \overset{def}{=} C \smallsetminus X \in \eta$.*

Indeed, an upper closure operator η is always closed under glb (intersection in this context), therefore whenever it is closed also under complementation, we have that it is surely disjunctive, by De Morgan's laws. In the following we have an example of the partitioning closure associated with a partition.



partition induced by a closure
1234
123
1
2
3
4
∅
corresponding partitioning closure
1234
123
14
234
23
1
2
3
4
∅

Fig. 1. A partitioning closure.

Example 5. Consider the set $\Sigma = \{1, 2, 3, 4\}$ and one of its possible partitions $\pi = \{\{1\}, \{2, 3\}, \{4\}\}$, then the closure η with fix points $\{\varnothing, \{1\}, \{4\}, \{123\}, \Sigma\}$ induces exactly π as partition of states, but the most *concrete* closure that induces π is $\mathit{Clo}^{\pi} = \varPi(\eta) = \curlyvee(\{\varnothing, \{1\}, \{2, 3\}, \{4\}\}, \Sigma)$, which is the closure on the right in Fig. 1.

On the closures we have the following characterizations. Note that, since $\varPi$ is a lower closure operator on $uco(\wp(C))$, then $\sqcup$ in Eq coincides with $\sqcup$ in uco, whereas $\mathit{Clo}^{\mathtt{Q}\sqcap\mathtt{R}}$ can be strictly less than $\mathit{Clo}^{\mathtt{Q}} \sqcap \mathit{Clo}^{\mathtt{R}}$.

Proposition 6. *$\mathtt{Q} \sqsubseteq \mathtt{R}$ iff $\mathit{Clo}^{\mathtt{Q}} \sqsubseteq \mathit{Clo}^{\mathtt{R}}$, $\mathtt{Q} \sqcap \mathtt{R} = \mathit{Rel}^{\mathit{Clo}^{\mathtt{Q}} \sqcap \mathit{Clo}^{\mathtt{R}}}$ and $\mathtt{Q} \sqcup \mathtt{R} = \mathit{Rel}^{\mathit{Clo}^{\mathtt{Q}} \sqcup \mathit{Clo}^{\mathtt{R}}}$.*

3 Information Flows in Language-based Security

In the rest of this paper, confidential data are considered *private*, labeled with $\mathtt{H}$ (high level of secrecy), while all other data are public, labeled with $\mathtt{L}$ (low level of secrecy). Non-interference can be naturally expressed by using semantic models of program execution (this idea goes back to Cohen's work on *strong dependency* [3]). Non-interference for programs essentially means that *"a variation of confidential (high or private) input does not cause a variation of public (low) output"* [17]. When this happens, we say that the program has only *secure information flows* [1, 3, 8, 13]. This situation has been modeled by considering the denotational (input/output) semantics $[\![P]\!]$ of the program P. Program states in Σ are functions (represented as tuples) mapping variables into the set of values $\mathbb{V}$. If $\mathtt{T} \in \{\mathtt{H}, \mathtt{L}\}$, $n = |\{x \in \mathit{Var}(P) | x : \mathtt{T}\}|$, and $v \in \mathbb{V}^n$, we abuse notation by denoting $v \in \mathbb{V}^{\mathtt{T}}$ the fact that v is a possible value for the variables with security type $\mathtt{T}$. Moreover, we assume that any input s, can be seen as a pair (h, l), where $s^{\mathtt{H}} = h$ is a value for private data and $s^{\mathtt{L}} = l$ is a value for public data. In this case, *(standard) non-interference* can be formulated as follows.

$$\boxed{\text{A program } P \text{ is } \textit{secure} \text{ if } \forall \text{ input } s, t.\ s^{\mathtt{L}} = t^{\mathtt{L}} \ \Rightarrow\ ([\![P]\!](s))^{\mathtt{L}} = ([\![P]\!](t))^{\mathtt{L}}}$$

This definition has been formulated also as a *Partial Equivalence Relation* (PER) [18]. The standard methods for checking non-interference are based on security-type systems and data-flow/control-flow analysis. Type-based approaches are designed in such a way that well typed programs do not leak secrets. In a security-typed language, a type is inductively associated at compile time with program statements in such a way that any statement showing a potential flow disclosing secrets is rejected [19, 21]. Similarly, data-flow/control-flow analysis techniques are devoted to statically discover flows of secret data into public variables [2, 13, 15, 18]. All these approaches are characterized by the way they model attackers (or unauthorized users).

3.1 Abstract Non-Interference: Attack Models

The notion of abstract non-interference [9] is introduced for modeling both weaker attack models, and declassification. The idea is that an attacker can observe only some properties, modeled as abstract interpretations of program semantics, of public concrete values. The *model of an attacker*, also called *attacker*, is therefore a pair of abstractions $\langle \eta, \rho \rangle$, with $\eta, \rho \in \mathit{uco}(\wp(\mathbb{V}^{\mathtt{L}}))$, representing what an observer can see about, respectively, the input and output of a program. The notion of *narrow (abstract) non-interference* (NNI), denoted $[\eta]P(\rho)$, is given in Table 1. It says that if the attacker is able to observe the property η of public input, and the property ρ of public output, then no information flow concerning the private input is observable from the public output. The problem with this notion is that it introduces *deceptive flows* [9], generated by different public output due to different public input with the same η property. Consider, for instance, the program $l := l * h^2$ and an observer who can observe only the parity of l on input and its sign on output. Intuitively, we may say that no information flows from h, since the sign of l after the assignment does not reveal anything about the value of h. However, $[\mathit{Par}]l := l * h^2(\mathit{Sign})$ does *not* hold[5], since there is variation of the output's sign due to the existence of both negative and positive even numbers. In order to avoid deceptive flows we introduce a weaker notion of non-interference, which considers as public input the *set* of all the elements sharing the same property η. Hence, in the previous example, the observable output for l is the set of all the elements with the same parity, e.g., if $\mathit{Par}(l) = \mathit{even}$ then we check the sign of $\left\{\ l * h^2 \,\middle|\, l \text{ is even}\ \right\}$ which is always unknown, since an even number can be both positive and negative, while h^2 *does not* interfere with the final sign. Moreover, we consider also a property $\phi \in \mathit{uco}(\wp(\mathbb{V}^{\mathtt{H}}))$, modeling the private property that has not to be observed by the attacker $\langle \eta, \rho \rangle$. This notion, denoted $(\eta)P(\phi \rightsquigarrow\!\!\parallel \rho)$, is called *abstract non-interference* (ANI) and is defined in Table 1. So for example the property

[5] Here $\mathit{Par} \overset{\text{def}}{=} \{\top, ev, od, \bot\}$ and $\mathit{Sign} \overset{\text{def}}{=} \{\top, 0+, -, \bot\}$.

$[\eta]P(\rho)$ if $\forall h_1, h_2 \in \mathbb{V}^{\mathtt{H}}, \forall l_1, l_2 \in \mathbb{V}^{\mathtt{L}} \,.\; \eta(\{l_1\}) = \eta(\{l_2\}) \;\Rightarrow\; \rho(\{[\![P]\!](h_1, l_1)^{\mathtt{L}}\}) = \rho(\{[\![P]\!](h_2, l_2)^{\mathtt{L}}\})$
$(\eta)P(\phi \rightsquigarrow\!\!\!\parallel\!\rho)$ if $\forall h_1, h_2 \in \mathbb{V}^{\mathtt{H}}, \forall l \in \mathbb{V}^{\mathtt{L}} \,.\; \rho([\![P]\!](\phi(\{h_1\}), \eta(\{l\}))^{\mathtt{L}}) = \rho([\![P]\!](\phi(\{h_2\}), \eta(\{l\}))^{\mathtt{L}})$

Table 1. Narrow and Abstract Non-Interference.

$(id)l := l * h^2(Sign \rightsquigarrow\!\!\!\parallel\!Sign)$ is satisfied, since the public result's sign do not depend on the private input sign, which is kept secret.

Note that $[id]P(id)$ models exactly (standard) non-interference. Moreover, we have that abstract non-interference is a weakening of both standard and narrow non-interference: $[id]P(id) \;\Rightarrow\; (\eta)P(\phi \rightsquigarrow\!\!\!\parallel\!\rho)$ and $[\eta]P(\rho) \;\Rightarrow\; (\eta)P(\phi \rightsquigarrow\!\!\!\parallel\!\rho)$, while standard non-interference is not stronger than the narrow version, due to deceptive flows. In [9], two methods are provided for deriving the most concrete output observation for a program, given the input one, for both NNI and ANI. In particular the idea is to collect in the same abstract object all the elements that, if distinguished, would generate a visible flow. These most concrete output observations, unable to get information from the program P observing η in input, are, respectively, denoted $[\eta][\![P]\!](id)$ and $(\eta)[\![P]\!](\phi \rightsquigarrow\!\!\!\parallel\!id)$, both in $uco(\wp(\mathbb{V}^{\mathtt{L}}))$. Hence, if for instance $P \overset{\text{def}}{=} l := |l| * Sign(h)$ (where $|\cdot|$ is the absolute value), we note that each value n has to be abstracted together with its opposite $-n$, in order not to generate visible flows, hence the most concrete harmless attacker can at most observe the absolute value Abs, i.e., $[Abs][\![P]\!](id) = Abs$.

3.2 PER Model

The semantic approach described above has also been equivalently formalized in [18], by using *partial equivalence relations (PER)* to model dependencies in programs. As we noted above, the problem of non-interference can be seen as absence of dependencies among data, where the meaning of dependency is given in [3]. The idea behind this characterization consists in interpreting security types as partial equivalence relations. In particular the type `H` is interpreted by using the equivalence relation *All*, and `L` by using the relation *Id*. The intuition is that *All* and *Id* model, respectively, that the user has no access to the high information and has full access to the low information. This perspective can simply be generalized to multilevel security problems.

In order to use this model in the security framework we need to combine equivalence relations on simple domains to construct new relations on more complex domains, in particular product spaces and function spaces. For the latter, it turns out to be natural to generalise slightly to consider *partial* equivalence relations, that is, relations which are symmetric and transitive but not necessarily reflexive. Let $Per(D)$ be the set of partial equivalence relations on

D. Given $\mathtt{P} \in \mathit{Per}(D)$ and $\mathtt{Q} \in \mathit{Per}(E)$ we define $(\mathtt{P} \twoheadrightarrow \mathtt{Q}) \in \mathit{Per}(D \to E)$ and $(\mathtt{P} \times \mathtt{Q}) \in \mathit{Per}(D \times E)$ as follows:

1. $f\ (\mathtt{P} \twoheadrightarrow \mathtt{Q})\ g \iff \forall x, x' \in D\ .\ x\ \mathtt{P}\ x' \Rightarrow f(x)\ \mathtt{Q}\ g(x')$
2. $\langle x, y\rangle\ \mathtt{P} \times \mathtt{Q}\ \langle x', y'\rangle \iff x\ \mathtt{P}\ x' \wedge y\ \mathtt{Q}\ y'$.

In general, for $\mathtt{P} \in \mathit{Per}(D)$ and $x \in D$, we write $x : \mathtt{P}$ to mean $x\ \mathtt{P}\ x$. In particular, if $f\ (\mathtt{P} \twoheadrightarrow \mathtt{Q})\ f$, we write $f : \mathtt{P} \twoheadrightarrow \mathtt{Q}$. Note that $\mathtt{P} \twoheadrightarrow \mathtt{Q}$ will not, in general, be reflexive, even when $\mathtt{P}$ and $\mathtt{Q}$ are (for example, $\mathit{All} \twoheadrightarrow \mathit{Id}$ relates only functions which are equal and *constant*).

At this point, we can formalize security in this model.

Definition 7. *[18] A program P is said to be* secure *iff $\forall s, t\ .\ \langle s^{\mathtt{H}}, s^{\mathtt{L}}\rangle\ \mathit{All} \times \mathit{Id}\ \langle t^{\mathtt{H}}, t^{\mathtt{L}}\rangle \Rightarrow [\![P]\!](s)\ \mathit{All} \times \mathit{Id}\ [\![P]\!](t)$, or, more concisely: $[\![P]\!] : \mathit{All} \times \mathit{Id} \twoheadrightarrow \mathit{All} \times \mathit{Id}$.*

4 PER Model vs Abstract Non-Interference

The correspondence existing between ucos and equivalence relations suggests that we can define particular notions of abstract non-interference where the closures modeling properties are all partitioning, i.e., correspond exactly to equivalence relations. As shown below, for NNI this specialisation makes essentially no difference, while for ANI it does involve a loss of generality.

First of all we introduce the natural generalization of the PER model provided in [18]. Given a program P and relations $\mathtt{Q}, \mathtt{W} \in \mathit{Eq}(\mathbb{V})$, we say that P is $\langle \mathtt{Q}, \mathtt{W}\rangle$-secure iff $[\![P]\!] : \mathtt{Q} \twoheadrightarrow \mathtt{W}$. Clearly, P is secure (Definition 7) just when it is $\langle \mathit{All} \times \mathit{Id}, \mathit{All} \times \mathit{Id}\rangle$-secure.

4.1 PER Model vs NNI

Proposition 8. *Let P be a deterministic program. Let $\eta, \rho \in uco(\wp(\mathbb{V}^{\mathtt{L}}))$. Then:*

1. *$[\eta]P(\rho)$ iff $[\![P]\!] : \mathit{All} \times \mathit{Rel}^{\eta} \twoheadrightarrow \mathit{All} \times \mathit{Rel}^{\rho}$*
2. *$[\eta]P(\rho)$ iff $[\Pi(\eta)]P(\Pi(\rho))$*

Proof. Part 1 is immediate from the definitions. Part 2 follows from part 1 by part 2 of Corollary 3. □

Since every equivalence relation $\mathtt{R}$ is represented exactly by the uco $\mathit{Clo}^{\mathtt{R}}$, this result shows that precisely the same class of NNI properties can be expressed using equivalence relations or partitioning closures as using arbitrary ucos. In particular, we may define NNI directly in terms of equivalence relations:

Definition 9. *Let P be a program. Let $\mathtt{R}, \mathtt{S} \in \mathit{Eq}(\mathbb{V}^{\mathtt{L}})$. Then P is said to be $\langle \mathtt{R}, \mathtt{S}\rangle$-NSecret, written $[\mathtt{R}]P(\mathtt{S})$, iff $[\![P]\!] : \mathit{All} \times \mathtt{R} \twoheadrightarrow \mathit{All} \times \mathtt{S}$.*

By Proposition 8, all NNI properties may be written in this form.

4.2 PER Model vs ANI

To compare the relative expressive power of the PER model and the general notion of abstract non-interference using arbitrary ucos, it is helpful to consider the extension of a relation on C to a relation on subsets of C. The basic construction is that used in defining Plotkin's powerdomain.

Definition 10. *Let* $\mathtt{R}$ *be a binary relation on a set* C*. Then the extension of* $\mathtt{R}$ *to* $\wp(C)$ *is the relation* $\mathcal{P}[\mathtt{R}] \subseteq \wp(C) \times \wp(C)$ *such that* $X \; \mathcal{P}[\mathtt{R}] \; Y$ *iff*

$$\forall x \in X. \, \exists y \in Y \, . \, x \; \mathtt{R} \; y \text{ and } \forall y \in Y. \, \exists x \in X \, . \, x \; \mathtt{R} \; y$$

For a partitioning closure, the extension of its corresponding equivalence relation from C to $\wp(C)$ has a particularly simple characterisation:

Proposition 11. *Let* C *be any set and let* $\eta \in uco(\wp(C))$ *be partitioning. Then* $\mathcal{P}[Rel^{\eta}] = \mathtt{K}_{\eta}$*, that is:* $X \; \mathcal{P}[Rel^{\eta}] \; Y \;\; \Leftrightarrow \;\; \eta(X) = \eta(Y)$.

Corollary 12. *Let* $\eta, \phi \in uco(\wp(\mathbb{V}^{\mathtt{L}}))$ *and let* $\rho \in uco(\wp(\mathbb{V}^{\mathtt{H}}))$*. If* ρ *is partitioning, then* $(\eta)P(\phi \rightsquigarrow\!\!\| \rho)$ *iff*

$$\forall X_1, X_2 \in \mathbb{V}^{\mathtt{H}}/Rel^{\phi}, \forall Y \in \mathbb{V}^{\mathtt{L}}/Rel^{\eta} \, . \, [\![P]\!](X_1, Y) \; \mathcal{P}[All \times Rel^{\rho}] \; [\![P]\!](X_2, Y)$$

The following proposition shows that, in contrast to NNI, there are ANI properties which cannot be expressed using the partitioning closures alone.

Proposition 13. *Let* P *be a program, let* $\eta, \phi \in uco(\wp(\mathbb{V}^{\mathtt{L}}))$ *and* $\rho \in uco(\wp(\mathbb{V}^{\mathtt{H}}))$*. Then* $(\Pi(\eta))P(\Pi(\phi) \rightsquigarrow\!\!\| \Pi(\rho)) \Rightarrow (\eta)P(\phi \rightsquigarrow\!\!\| \rho)$ *but, in general, the reverse implication does not hold.*

The following example shows where the difference between the two notions lies.

Example 14. Consider the following program fragment:

$$P \stackrel{\text{def}}{=} \textbf{if } h = 0 \textbf{ then } l := l \; mod \; 6 + 2; \textbf{else if } l < 0 \textbf{ then } l := 2 \textbf{ else } l := 7;$$

with security typing $h : \mathtt{H}$, $l : \mathtt{L}$. Consider $\eta \stackrel{\text{def}}{=} \{\top, 2\mathbb{Z}, 2\mathbb{Z}+1, \bot\}$ for parity, $\phi = \{\top, 0+, -, \bot\}$ for sign, and $\rho \stackrel{\text{def}}{=} Int$ of intervals [5], in $uco(\wp(\mathbb{Z}))$. Note that, since each integer number is in particular an interval, we have that $\Pi(Int) = id$, distinguishing all the integer values, while $\Pi(\eta) = \eta$ and $\Pi(\phi) = \phi$. Let us see what happens in abstract non-interference. Consider $\eta(l) = 2\mathbb{Z}$, then if $\phi(h) = 0+$ we have that $\rho([\![P]\!](\phi(h), \eta(l))^{\mathtt{L}}) = \rho(\{2, 4, 6, 7\}) = [2, 7]$. While, if $\phi(h) = -$, then we have $\rho([\![P]\!](\phi(h), \eta(l))^{\mathtt{L}}) = \rho(\{2, 7\}) = [2, 7]$. On the other hand, if $\eta(l) = 2\mathbb{Z}+1$ and $\phi(h) = 0+$, then $\rho([\![P]\!](\phi(h), \eta(l))^{\mathtt{L}}) = \rho(\{2, 3, 5, 7\}) = [2, 7]$, and when $\phi(h) = -$ we have $\rho([\![P]\!](\phi(h), \eta(l))^{\mathtt{L}}) = \rho(\{2, 7\}) = [2, 7]$. So $(\eta)P(\phi \rightsquigarrow\!\!\| \rho)$ holds. Consider now $\Pi(\rho) = id$. It is clear that if we substitute above ρ with id, then we have that $(\Pi(\eta))P(\Pi(\phi) \rightsquigarrow\!\!\| \Pi(\rho))$ does not hold. □

Hence, ANI with ucos is a more precise notion whenever we have to deal with sets of values, instead of with singletons. This may be particularly useful, for example, for non-deterministic systems, where the denotational semantics returns a *set* of states as output.

5 Deriving Attacker Models by Abstract Interpretation

In this section we consider the PER model of NNI and use it to derive simple, constructive characterisations of various classes of attacker considered in [9]. For example, suppose given a class of attackers whose power to observe low security inputs is given by $\mathtt{R}$: for a given program P, what is the most powerful attacker in the class (with respect to observation of low security outputs), for which P is secure? There are two cases of principal interest:

1. *Most powerful attacker*: given $\mathtt{R} \in Eq(\mathbb{V}^{\mathtt{L}})$, is there a smallest $\mathtt{S} \in Eq(\mathbb{V}^{\mathtt{L}})$ such that $[\mathtt{R}]P(\mathtt{S})$? Or, given $\mathtt{S} \in Eq(\mathbb{V}^{\mathtt{L}})$, is there a greatest $\mathtt{R} \in Eq(\mathbb{V}^{\mathtt{L}})$ such that $[\mathtt{R}]P(\mathtt{S})$?
2. *Fix point (canonical) attacker*: is there a smallest $\mathtt{R}$ such that $[\mathtt{R}]P(\mathtt{R})$?

The particular interest of fix point attackers is that, in many situations, the power of the attacker to observe low security data may be independent of the data's rôle as input or output.

5.1 Deriving unconstrained attackers

In this section, given a semantics f and an input [output] equivalence relation $\mathtt{R}$ [$\mathtt{S}$], we show how we can derive the most concrete [abstract] output [input] relation $\mathtt{S}$ [$\mathtt{R}$] that makes the program satisfy $f : \mathtt{R} \twoheadrightarrow \mathtt{S}$. Consider an arbitrary function $f : A \to B$ between sets. As is well known, any such f lifts to an adjunction between $\wp(A)$ and $\wp(B)$, in the form of f's direct and inverse image mappings. It turns out that f can be lifted to an adjunction $\langle Eq(A), \widehat{f}, \widehat{f}^{-1}, Eq(B)\rangle$ between lattices of equivalence relations in a similar way. In this section we detail the construction of $\widehat{f}$ and $\widehat{f}^{-1}$, and we go onto show how they are used to derive attackers.

Given an output relation $\mathtt{S}$ it is always possible to find a good candidate for input relation $\mathtt{R}$, essentially by simply imposing the condition $f : \mathtt{R} \twoheadrightarrow \mathtt{S}$. In other words we can always define the equivalence relation $\widehat{f}^{-1}(\mathtt{S})$ in the following way:

$$x \; \widehat{f}^{-1}(\mathtt{S}) \; y \text{ iff } f(x) \; \mathtt{S} \; f(y) \tag{1}$$

This is the key definition in [14] and is also exactly the idea used in [22] on the trace semantics, namely we collect together all the elements whose semantics are equivalent in the output observation[6].

[6] This transformation corresponds to the *quotient* of the concrete semantic domain with respect to the property $Clo^{\mathtt{S}}$ [4]

Lemma 15. *$\widehat{f}^{-1}(\mathtt{S})$ is an equivalence relation and $f : \mathtt{R} \twoheadrightarrow \mathtt{S} \Leftrightarrow \mathtt{R} \sqsubseteq \widehat{f}^{-1}(\mathtt{S})$.*

Note that for each $\mathtt{S}$ we have $\widehat{f}^{-1}(\mathtt{S}) \sqsupseteq \mathtt{K}_f$. This means that the input relation has, at least, to identify all the elements with the same image under f. This observation makes the definition of $\widehat{f}$ a bit more complicated. Indeed, given $\mathtt{R}$, we would like to find the best relation $\mathtt{S}$ which satisfies $f : \mathtt{R} \twoheadrightarrow \mathtt{S}$. A naive construction leads to the function $\widetilde{f} : \mathit{Rel}(C) \to \mathit{Rel}(C)$, as follows:

$$y\ \widetilde{f}(\mathtt{R})\ y' \text{ iff } (\exists x, x' \,.\, x\ \mathtt{R}\ x' \text{ and } f(x) = y,\ f(x') = y' \ \vee\ y = y')$$

Note that the disjunct $y = y'$ guarantees that the relation is reflexive. However, $\widetilde{f}(\mathtt{R})$ may fail to be transitive, as we can see in the following example.

Example 16. Consider a domain $C = \{1, 2, 3, 4, 5, 6\}$ and a function f such that $f(1) = 1$, $f(3) = f(4) = 2$, $f(2) = f(6) = 5$ and $f(5) = 3$, and suppose that $\mathtt{R} = \{[1, 3], [2, 4], [5, 6]\}$, then we would have $1\widetilde{f}(\mathtt{R})2$, $2\widetilde{f}(\mathtt{R})5$ and $5\widetilde{f}(\mathtt{R})3$, but for example $1\neg\widetilde{f}(\mathtt{R})3$.

The problem is that f is not injective ($\mathtt{K}_f \neq \mathit{Id}$) and therefore, in the example the fact that $f(3) = f(4)$ while $\mathtt{R}$ distinguishes 3 from 4, creates the problems.

Proposition 17. *Consider $f : C \to C$ and $\mathtt{R} \in \mathit{Eq}(C)$. If $\mathtt{K}_f \sqsubseteq \mathtt{R}$, then $\widetilde{f}(\mathtt{R})$ is an equivalence relation, if $\mathtt{K}_f = \mathtt{R}$, then $\widetilde{f}(\mathtt{R}) = \mathit{Id}$.*

We would like to modify $\widetilde{f}$ in order to guarantee that $\widetilde{f}(\mathtt{R})$ is always an equivalence relation. For this reason we prove the following result.

Proposition 18. *Let $f : A \to B$. Then $\widehat{f}^{-1} : \mathit{Eq}(B) \to \mathit{Eq}(A)$ is co-additive.*

This means that $\widehat{f}^{-1}$ is the right adjoint of a Galois connection. Thus we can define the following function, which is its left adjoint [5]:

$$\widehat{f}(\mathtt{R}) \overset{\text{def}}{=} \bigsqcap \left\{\ \mathtt{Q} \ \middle|\ \mathtt{R} \sqsubseteq \widehat{f}^{-1}(\mathtt{Q})\ \right\} \tag{2}$$

The co-additivity of $\widehat{f}^{-1}$ guarantees that the element uniquely exists. We manipulate this set obtaining that $\widehat{f}(\mathtt{R}) = \bigsqcap \left\{\ \mathtt{Q} \ \middle|\ x\ \mathtt{R}\ y \ \Rightarrow\ f(x)\ \mathtt{Q}\ f(y)\ \right\}$.

Theorem 19. *$\widehat{f}(\mathtt{R}) = \widetilde{f}(\mathtt{R} \sqcup \mathtt{K}_f) = \mathbb{T}(\widetilde{f}(\mathtt{R}))$.*

This means that, when $\mathtt{R} \sqsupseteq \mathtt{K}_f$, then $\widetilde{f}(\mathtt{R}) = \widehat{f}(\mathtt{R})$.

By construction, the following result is straightforward:

Proposition 20. *$\langle \mathit{Eq}(A), \widehat{f}, \widehat{f}^{-1}, \mathit{Eq}(B)\rangle$ is a Galois connection. That is, for all $\mathtt{R} \in \mathit{Eq}(A), \mathtt{S} \in \mathit{Eq}(B)$: $\widehat{f}(\mathtt{R}) \sqsubseteq \mathtt{S} \Leftrightarrow \mathtt{R} \sqsubseteq \widehat{f}^{-1}(\mathtt{S})$.*

Combining Proposition 20 with Lemma 15, gives:

Theorem 21. *$f : \mathtt{R} \twoheadrightarrow \mathtt{S}\ \Leftrightarrow\ \widehat{f}(\mathtt{R}) \sqsubseteq \mathtt{S}\ \Leftrightarrow\ \mathtt{R} \sqsubseteq \widehat{f}^{-1}(\mathtt{S})$.*

This result shows which is the rôle of the two operators $\widehat{f}$ and $\widehat{f}^{-1}$ in the whole construction. Indeed, by Theorem 21 we have that f satisfies non-interference, namely $f : \mathtt{R} \twoheadrightarrow \mathtt{S}$, iff $\widehat{f}(\mathtt{R}) \sqsubseteq \mathtt{S}$. This means that $\widehat{f}$ characterizes exactly the *most concrete output relation* that guarantees non-interference for f, fixed the input relation. By the adjunction relation we can also say that $f : \mathtt{R} \twoheadrightarrow \mathtt{S}$ iff $\mathtt{R} \sqsubseteq \widehat{f}^{-1}(\mathtt{S})$. Thus $\widehat{f}^{-1}$ characterizes the *most abstract input relation* that guarantees non-interference for f, fixed the output relation. Indeed, as expected, we can always abstract the output observation and we can always concretize the input one. Note that [9] *misses* exactly a construction of the input observation that makes a program secure, given the output one, while this is possible in this context since we are considering equivalence relations. An example is provided in Fig. 2.

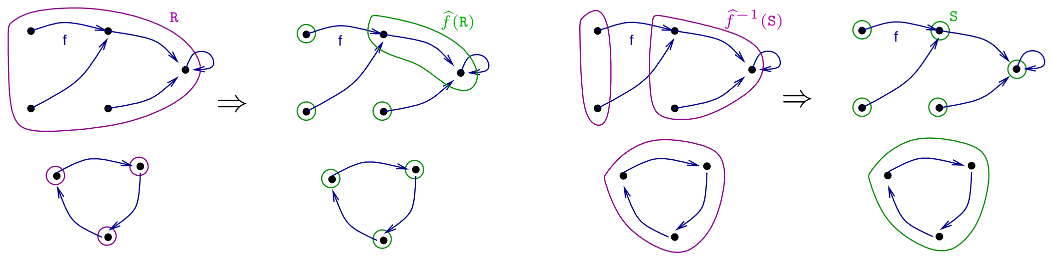


Fig. 2. Example of application of $\widehat{f}$ and of $\widehat{f}^{-1}$.

5.2 Fix point attackers

In this section we look for the characterisation of attackers that observe the same property both in input and in output. The idea is to consider the fix points of the unconstrained attackers derived above. Unfortunately, the most concrete and the most abstract non trivial (different from top and identity) attacker models do not exist as can be also verified in Fig. 3, therefore we can use the fix point iteration simply as a possible systematic construction of canonical attackers.

Fix point of $\widehat{f}^{-1}$. Note that $\widehat{f}^{-1}(\top) = \top$, this means that the interesting case, if it exists, is the least fix point of $\widehat{f}^{-1}$ starting from *Id*. We know that $\widehat{f}^{-1}$ is monotone (Prop. 20), therefore the least fix point exists and can be obtained as the limit of the iterative application of $\widehat{f}^{-1}$ starting from *Id*, the bottom of the lattice of relations [6, 20].

Fix point of $\widehat{f}$. Note that $\widehat{f}(Id) = Id$, this means that we can find, if it exists, only the greatest fix point of $\widehat{f}$ starting from $\top$. We know that $\widehat{f}$ is monotone

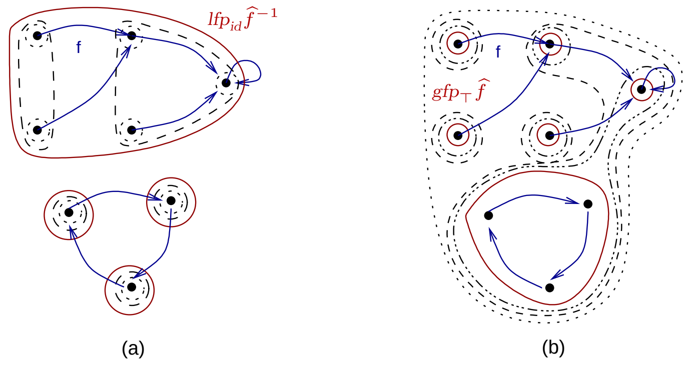


Fig. 3. Examples of fix points.

(Prop. 20), therefore the greatest fix point exists and can be obtained as the limit of the iterative application of $\widehat{f}$ starting from the element $\top$ of the lattice of relations [6, 20].

5.3 Deriving contrained attackers

In this section, we consider attackers which are unable to observe private data, and which can only observe properties of public data. In this way we derive attackers for abstract non-interference [9], where the attackers are modeled by equivalence relations instead of by closure operators.

Most Powerful Attackers. We can use $\widehat{f}$ to construct the most powerful attacker. Firstly, note that it follows directly from the definitions that $[\mathtt{R}]P(\mathtt{S})$ iff $\pi_2 \circ [\![P]\!] : \mathit{All} \times \mathtt{R} \twoheadrightarrow \mathtt{S}$[7]. The following result is then a straightforward consequence of Theorem 21:

Proposition 22. *Let P be a program and let $\mathtt{R} \in \mathit{Eq}(\mathbb{V}^{\mathtt{L}})$. Then the smallest $\mathtt{S}$ such that $[\mathtt{R}]P(\mathtt{S})$ is $\widehat{f}(\mathit{All} \times \mathtt{R})$, where $f = \pi_2 \circ [\![P]\!]$.*

Fix Point Attackers. We wish to construct the smallest $\mathtt{R}$ such that:

$$[\![P]\!] : \mathit{All} \times \mathtt{R} \twoheadrightarrow \mathit{All} \times \mathtt{R} \tag{3}$$

Let $F_P(\mathtt{R}) \overset{\text{def}}{=} \widehat{f}(\mathit{All} \times \mathtt{R})$, where $f = \pi_2 \circ [\![P]\!]$. Then, using Theorem 21, it is easily verified that (3) holds iff $F_P(\mathtt{R}) \sqsubseteq \mathtt{R}$. Thus the solutions to (3) are just the post-fix points of F_P. Since F_P is clearly monotone on $\mathit{Eq}(\mathbb{V}^{\mathtt{L}})$, Tarski's fix point theorem gives:

[7] Here $\pi_2(\langle a, b\rangle) = b$ is the projection on the second component of a pair.

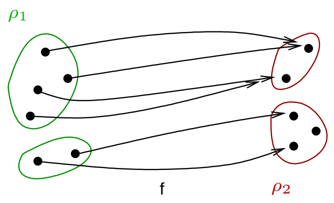
Proposition 23. *Let P be a program and let $F_P : Eq(\mathbb{V}^{\mathtt{L}}) \rightarrow Eq(\mathbb{V}^{\mathtt{L}})$ be defined as above. Then the smallest* $\mathtt{R}$ *such that* $[\mathtt{R}]P(\mathtt{R})$ *is lfp* F_P.

Note that this construction corresponds exactly to the characterization given in [9] for arbitrary closures. Indeed here we collect elements, in the new relation $\widehat{f}(\mathtt{R})$, iff they are images by f of elements that are in the input relation. In [9] the elements are collected, for obtaining the resulting closure, when they are images, under f of inputs that differ only in the private information (which is the input relation in ANI).

5.4 Non-interference and Completeness

In [10] it is proved that (abstract) non-interference can be modeled as a problem of completeness in the standard framework of abstract interpretation. Since partitions are particular closure operators, we can use completeness also for the PER model of abstract non-interference. We would like to understand how completeness can be helpful in order to obtain non-interference. First of all let us consider a new characterization of completeness.

Theorem 24. *Given* $\rho_1, \rho_2 \in uco(\wp(C))$, *and* $f : C \rightarrow C$, *then* $\langle \rho_1, \rho_2 \rangle$ *is complete for* f, *i.e.,* $\rho_2 \circ f \circ \rho_1 = \rho_2 \circ f$ *iff* $\forall X \in \rho_1 . \exists Y \in \rho_2$ *such that* $\forall z . (\rho_1(z) = X \Rightarrow \rho_2(f(z)) = Y)$.



At this point let us define completeness of equivalence relations in terms of completeness of the corresponding closure operators. Let $\mathtt{R}, \mathtt{S} \in Eq(C)$, and f a map on C: $\mathtt{S} \circ f \circ \mathtt{R} = \mathtt{S} \circ f$ iff $Clo^{\mathtt{S}} \circ f \circ Clo^{\mathtt{R}} = Clo^{\mathtt{S}} \circ f$.

Corollary 25. $f : \mathtt{R} \twoheadrightarrow \mathtt{S}$ *iff* $\mathtt{S} \circ f \circ \mathtt{R} = \mathtt{S} \circ f$.

(It is interesting to note that precisely this relationship was used in [12] to establish a correspondence between PER-based and projection-based program analyses. It holds generally for idempotent maps and their kernels.)

This means that we can use the constructive method given in [11] for making abstract domains complete. Clearly the result of this transformation need not be a partitioning closure, hence we have then to derive the partition associated with the complete domain. In this way we obtain a method for making equivalence relations complete.

6 Conclusion

In this paper we define abstract non-interference in terms of the PER model. In particular, we consider equivalence relations instead of arbitrary abstract domains. We show that the notion does not change for narrow non-interference, while it becomes less general when we consider abstract non-interference. And it is possible to show that, even if we lift PERs to sets then we cannot reach the generality of uco since lifted PERs correspond only to additive closures. The use of equivalence relations allows us to simplify the characterization of the most powerful harmless attacker. Moreover we can also derive distinguished attackers for the generic PER model of security ($\langle \mathtt{Q}, \mathtt{W} \rangle$-security, Sect. 4).

Finally, we show that the PER model of abstract non-interference can be rewritten as an abstract domain completeness problem. This result is interesting for us since it suggests how we may approach the problem of making partitioning closures complete, similarly to what is done in [11]. Such a result could be useful also in other fields of computer science, such as completeness in model checking [16]. In this paper we only provide the relation-based construction of the most powerful harmless attacker for the narrow case, which is the straightforward generalization of the PER model [18]. It could be interesting to investigate if the restriction to partitioning closures simplifies also the characterization of the harmless attacker for abstract non-interference.

Acknowledgments

We would like to thank Roberto Giacobazzi for his insightful comments and contribution of ideas, and the anonymous referees for their helpful suggestions for improvement.

References

1. D. E. Bell and L. J. LaPadula. Secure computer systems: Mathematical foundations and model. Technical Report M74-244, MITRE Corp. Badford, MA, 1973.
2. D. Clark, C. Hankin, and S. Hunt. Information flow for algol-like languages. *Computer Languages*, 28(1):3–28, 2002.
3. E. S. Cohen. Information transmission in sequential programs. *Foundations of Secure Computation*, pages 297–335, 1978.
4. A. Cortesi, G. Filé, and W. Winsborough. The quotient of an abstract interpretation. *Theor. Comput. Sci.*, 202(1-2):163–192, 1998.
5. 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 Conf. Record of the 4th ACM Symp. on Principles of Programming Languages* (*POPL '77*), pages 238–252. ACM Press, New York, 1977.
6. P. Cousot and R. Cousot. Constructive versions of Tarski's fixed point theorems. *Pacific J. Math.*, 82(1):43–57, 1979.
7. P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Proc. of Conf. Record of the 6th ACM Symp. on Principles of Programming Languages* (*POPL '79*), pages 269–282. ACM Press, New York, 1979.

8. D. E. Denning and P. Denning. Certification of programs for secure information flow. *Communications of the ACM*, 20(7):504–513, 1977.
9. R. Giacobazzi and I. Mastroeni. Abstract non-interference: Parameterizing non-interference by abstract interpretation. In *Proc. of the 31st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '04)*, pages 186–197. ACM-Press, NY, 2004.
10. R. Giacobazzi and I. Mastroeni. Adjoining declassification and attack models by abstract interpretation. In *Proc. of the European Symposium on Programming (ESOP'05)*, volume 3444 of *Lecture Notes in Computer Science*, pages 295–310. Springer-Verlag, 2005.
11. R. Giacobazzi, F. Ranzato, and F. Scozzari. Making abstract interpretations complete. *J. of the ACM.*, 47(2):361–416, 2000.
12. Sebastian Hunt. Pers generalise projections for strictness analysis (extended abstract). In *Proc. 1990 Glasgow Workshop on Functional Programming*, Workshops in Computing, Ullapool, 1991. Springer-Verlag.
13. R. Joshi and K. R. M. Leino. A semantic approach to secure information flow. *Science of Computer Programming*, 37:113–138, 2000.
14. J. Landauer and T. Redmond. A lattice of information. In *Proc. of the IEEE Computer Security Foundations Workshop*, pages 65–70. IEEE Computer Society Press, 1993.
15. P. Laud. Semantics and program analysis of computationally secure information flow. In *In Programming Languages and Systems, 10th European Symp. On Programming, ESOP*, volume 2028 of *Lecture Notes in Computer Science*, pages 77–91. Springer-Verlag, 2001.
16. F. Ranzato and F. Tapparo. Strong preservation as completeness in abstract interpretation. In D. Schmidt, editor, *Proc. of the 13th European Symposium on Programming (ESOP'04)*, volume 2986 of *Lecture Notes in Computer Science*, pages 18–32. Springer-Verlag, 2004.
17. A. Sabelfeld and A.C. Myers. Language-based information-flow security. *IEEE J. on selected ares in communications*, 21(1):5–19, 2003.
18. A. Sabelfeld and D. Sands. A PER model of secure information flow in sequential programs. *Higher-Order and Symbolic Computation*, 14(1):59–91, 2001.
19. C. Skalka and S. Smith. Static enforcement of security with types. In *ICFP'00*, pages 254–267. ACM Press, New York, 2000.
20. A. Tarski. A lattice theoretical fixpoint theorem and its applications. *Pacific J. Math.*, 5:285–310, 1955.
21. D. Volpano, G. Smith, and C. Irvine. A sound type system for secure flow analysis. *Journal of Computer Security*, 4(2,3):167–187, 1996.
22. S. Zdancewic and A. C. Myers. Robust declassification. In *Proc. of the IEEE Computer Security Foundations Workshop*, pages 15–23. IEEE Computer Society Press, 2001.