Lattice Approach to Classifications

Olivier Brunet

December 4, 2002

Abstract

We present a translation of J. Barwise and J. Seligman's *Information Flow Theory* into a lattice and Galois connection based formalism. We show how to transform the different structures of the theory into this formalism and show that this translation extends the expressivity of the theory.

1 Introduction

Information is now present everywhere in science. Should it be the way it is represented, it evolves or it is shared, it arises in a wide range of fields such as biology, quantum physics and, of course, computer science. Our present work is based on a theory of information flow developped by Jon Barwise and Jerry Seligman and presented in a comprehensive way in [BS97]. Their formalism is based on two basic entities : classifications and infomorphisms. Classifications can be viewed as a pair of two sets, one for *tokens* (entities which are studied) and one for *types* (pieces of information), together with a relation between those two sets, which expresses the information one has about the tokens. The other basic entities, informorphisms, are functions which relate different classifications. Using this formalism, they present a way to build logical theories out of observations, and they explain how simple objects can be put together using what is called an information channel, and what it implies on the respective logics. In the following, we will present a translation of this formalism into lattices and Galois connections. This kind of formalism has strong mathematical foundations (see [Bir67]), and is widely used in fields such as program analysis [Cou96] and concept analysis [GW99], in both of which partial information representation issues are present.

The present article is divided into several parts, each devoted to one component of the theory. In section 1, we deal with regular theories and their equivalent translation into what we call abstract power-sets, and give a few basic definitions. In section 2, we present the main part of the transformation, namely the one of classifications into approximations. In section 3, we transform informorphisms into approximation morphisms, and show that the latter formalism is strictly more expressive than the former. We close with a section on further developments of this translation, and a conclusion.

2 Lattice representation of regular theories

A very simple way to provide a set Σ with some structure is to give it a deduction relation which gives rise to a logic on Σ. This deduction relation is a relation between subsets of Σ : If we denote $\vdash$ this relation, $\Gamma \vdash \Delta$ means that if all the elements in Γ are true, then at least one element in Δ is true either. In the following, we will consider *regular* theories, which verify some properties concerning monotonicity.

Definition 1 (Theory)
A theory *is a pair* $\langle \Sigma, \vdash \rangle$ *where* Σ *is a set and* $\vdash$ *is a binary relation between subsets of* Σ.

Definition 2 (Regular theory)
A theory $\langle \Sigma, \vdash \rangle$ *is* regular *if it satisfies the following three properties :*

$$\forall \alpha \in \Sigma, \alpha \vdash \alpha \tag{1}$$

$$\forall \Gamma, \Delta, \Sigma_0, \Sigma_1 \subseteq \Sigma, \Gamma \vdash \Delta \Rightarrow \Gamma, \Sigma_0 \vdash \Delta, \Sigma_1 \tag{2}$$

$$\forall \Gamma, \Delta, \Sigma' \subseteq \Sigma, \left(\forall \Sigma_0, \Sigma_1, \left\{ \begin{array}{c} \Sigma_0 \cup \Sigma_1 = \Sigma' \\ \Sigma_0 \cap \Sigma_1 = \emptyset \end{array} \right\} \Rightarrow \Gamma, \Sigma_0 \vdash \Delta, \Sigma_1 \right) \Rightarrow \Gamma \vdash \Delta \tag{3}$$

which are respectively called Identity, Weakening *and* Global Cut.

We are now going to define a similar structure based on complete lattices which will be proved to be equivalent to regular theories. But we first need to setup our notations. Following abstract interpretations conventions, we will only consider additive morphismes in the following, for which we can always define an adjoint.

Definition 3 (Additive morphism)
A poset-morphism $f : E \rightarrow F$ *between two complete lattices is said to be* additive *if :*

$$\forall X \subseteq E, \bigvee_{x \in X} f(x) = f(\bigvee_{x \in X} x)$$

Those morphisms are ordered using the pointwise order :

$$f \leq g \Leftrightarrow \forall x, f(x) \leq g(x)$$

Definition 4 (Adjoint)
Given an additive morphism $f : E \rightarrow F$, *we define its* adjoint $f^{\#} : F \rightarrow E$ *by :*

$$\forall y \in F, f^{\#}(y) = \bigvee \{ x \in E \mid f(x) \leq_F y \}$$

Proposition 1 (Galois connection)
Given an additive morphism $f : E \rightarrow F$, *we have :*

$$\forall x \in E, \forall y \in F, f(x) \leq_F y \Leftrightarrow x \leq_E f^{\#}(y)$$

With this, we can easily define what we call *abstract power-sets* (a.p.s.). An abstract power-set can be seen as a collection of subsets of a set Σ, with equivalent operations for union and intersection.

Definition 5 (Abstract power-set)
Given a set Σ, *an* abstract power-set *over* Σ *is a pair* $\langle P, e\rangle$ *where* P *is a lattice and where* $e : \langle \wp(\Sigma), \subseteq\rangle \to P$ *is an additive onto poset-morphism verifying* $e(\Sigma) = \top_P$.

To prove that abstract power-sets are equivalent to regular theories, we will define the functions which will realize the one-to-one transformation. We first show how to define a regular theory generated by an abstract power-set.

Definition 6
Given an abstract-power set $\langle P, e\rangle$ *over a set* Σ, *we define the logic* $\langle \Sigma, \vdash_P\rangle$ *where :*

$$\Gamma \vdash_P \Delta \Leftrightarrow \exists\, \delta \in \Delta : e(\{\delta\}) \leq_P e(\Gamma)$$

This is equivalent to :

$$\exists\, \delta \in \Delta : \forall\, x \in P,\ (\forall\, \gamma \in \Gamma, e(\{\gamma\}) \leq x) \Rightarrow e(\{\delta\}) \leq x$$

Proposition 2
$\langle \Sigma, \vdash_P\rangle$ *is a regular theory.*

Proof The only point to prove is the *Global Cut*. Now, let Γ, Δ, Σ' verify the left part of the axiom. We split Σ' into the partition $\{\Sigma_0, \Sigma_1\}$ where $\Sigma_0 = \{\sigma \in \Sigma' \,/\, \Gamma \vdash \sigma\}$. We have :

$$\Gamma, \Sigma_0 \vdash \Delta, \Sigma_1$$

Now, we have $\bigvee\{e(\{\gamma\}) \,/\, \gamma \in \Gamma \cup \Sigma_0\} = \bigvee\{e(\{\gamma\}) \,/\, \gamma \in \Gamma\}$, so that :

$$\Gamma, \Sigma_0 \vdash \Delta, \Sigma_1 \Leftrightarrow \Gamma \vdash \Delta, \Sigma_1$$

By definition, $\Gamma \not\vdash \Sigma_1$ so that we have :

$$\Gamma \vdash \Delta, \Sigma_1 \Rightarrow \Gamma \vdash \Delta$$

This ends the proof.

Conversely, regular theories can easily be turned into abstract power-sets. For convenience of notation, $\Sigma_\vdash$ will denote the set of subsets of Σ closed under $\vdash$. The fact that this contains Σ and is closed under intersection leads us to the following definition :

Definition 7
Given a regular theory $\langle \Sigma, \vdash\rangle$, *we define an abstract power-set* $\langle \Sigma_\vdash, e\rangle$ *where :*

– $\Sigma_\vdash$ *is ordered by* $\subseteq$

– $e = \lambda X. \bigcap\{Y \in \Sigma_\vdash \,/\, X \subseteq Y\}$

Proposition 3
There is an isomorphism between the regular theories of a set Σ *and its abstract power-sets, this isomorphism being realized by the functions* $P \rightarrow \langle \Sigma, \vdash_P \rangle$ *and* $\langle \Sigma, \vdash \rangle \rightarrow \langle \Sigma_\vdash, e \rangle$ *as defined above.*

Justification It comes from the fact that a regular theory is uniquely determined by its closed subsets :

$$\Gamma \vdash \delta \Leftrightarrow \forall E \in \Sigma_\vdash, \Gamma \subseteq E \Rightarrow \delta \in E$$

3 Classifications and Approximations

Now that we have introduced the basic ideas of our formalism, we can present the lattice version of classifications. But contrary to what was shown in the previous section, there is no one-to-one translation between classifications and what we call approximations, since approximations can embed some of the regular logic generated by the typing relation. In [BS97], classifications are defined as follows :

Definition 8 (Classification)
A classification $\mathbf{C} = \langle \mathrm{tok}_C, \mathrm{typ}_C, \vdash \rangle$ *consists of :*

1. *a set* tok_C *of objects to be classified, called the* tokens *of* **C**,
2. *a set* typ_C *of objects used to classify the tokens, called the* types *of* **C**, *and*
3. *a binary relation* $\vdash$ *between* tok_C *and* typ_C.

If $a \vdash \alpha$, we say that a is of type α. Let us define $\mathrm{Typ}(x) = \{\alpha \,/\, x \vdash \alpha\}$ and $\mathrm{Typ}(X) = \bigcup\{\mathrm{Typ}(x) \,/\, x \in X\}$. Conversely, we define $\mathrm{Tok}(\alpha)$ and $\mathrm{Tok}(A)$.

Example

> Suppose we have an electrical circuit composed by a battery, two switches S_1 and S_2 and a light bulb L. At various moments $\tau \in T$, we note the state $St(\tau) = \langle S_1(\tau), S_2(\tau), L(\tau) \rangle$ of the circuit. To shorten the notations, we use bits to denote the parts state. A `1` (resp. a `0`) means for a switch that it is *on* (resp. *off*) and for the bulb that it is *lit* (resp. *unlit*). Thus, the state of the circuit can be denoted by a list of 3 bits. We get a classification $\mathbf{Cir} = \left\langle T, \mathbf{2}^3, \vdash_C \right\rangle$ where :
>
> $$\tau \vdash_C S \Leftrightarrow S = St(\tau)$$

Example

With the same circuit, we define another classification, where the tokens are the state of the light bulb and the types are the states of the switches. Thus, we have $\mathbf{Bulb} = \left\langle \mathbf{2}^1, \mathbf{2}^2, \vdash_B \right\rangle$ where :

$$\alpha \vdash_B \langle \beta_1, \beta_2 \rangle \Leftrightarrow \exists\, \tau \in T : St(\tau) = \langle \beta_1, \beta_2, \alpha \rangle$$

It is to be noted that contrary to $\mathbf{Cir}$, the typing relation of $\mathbf{Bulb}$ is not function-like.

Definition 9 (Approximation)
An approximation *of E by F is a tuple $\mathbf{A} = \langle \mathrm{tok}_A, \mathrm{typ}_A, p_A \rangle$ where :*

1. *tok_A (resp. typ_A) is an abstract power-set of E (resp. of F)*
2. *$p_A : \mathrm{tok}_A \to \mathrm{typ}_A$ is the relation of approximation between the tokens (representing elements of E) and the types (for F).*

We now give the first part of the relation between classifications and approximations.

Definition 10
Given an approximation $\mathbf{A}$ from E to F, we define a classification $\mathrm{cla}(\mathbf{A}) = \langle E, F, \vdash_{\mathbf{A}} \rangle$ where :

$$a \vdash_{\mathbf{A}} \alpha \Leftrightarrow p_A \circ e_E(\{a\}) \leq_F e_F(\{\alpha\})$$

Conversely, we would be tempted to perform the reverse transformation. But as said above, given a classification $\mathbf{C}$, we can associate several approximations to it. We shall then consider the approximations which agree with a classification $\mathbf{C}$ on any subset of $\mathrm{tok}_{\mathbf{C}}$ (due to additivity) :

$$A_{\mathbf{C}} = \{\mathbf{A} \,/\, \forall\, X \subseteq \mathrm{tok}_C, \, \mathrm{Typ}(X) = e^{\#}_{\mathrm{typ}_A} \circ p_A \circ e_{\mathrm{tok}_A}(X)\}$$

In order to specify the structure of this $A_{\mathbf{C}}$, we make the set of approximation a poset with the order $\leq_{\mathrm{appr}}$ defined by :

Definition 11
Given two approximations $\langle A, B, p \rangle$ and $\langle A', B', p' \rangle$ of a classification $\mathbf{C}$, we say that :

$$\langle A, B, p \rangle \leq_{\mathrm{appr}} \langle A', B', p' \rangle \Leftrightarrow \begin{cases} A \sqsubseteq A' \\ B \sqsubseteq B' \\ e^{\#}_B \circ p \circ e_A = e^{\#}_{B'} \circ p' \circ e_{A'} \end{cases}$$

where $E \sqsubseteq E'$ means that $\{e^{\#}_E(x) \,/\, x \in E\} \subseteq \{e^{\#}_{E'}(x) \,/\, x \in E'\}$.

With this order, an approximation which is above another is more general, since it can distinguish more tokens and types. Conversely, the one below is more abstract.

Given a classification $\mathbf{C}$, there is a most general approximation generating it (since $\wp(X)$ can be seen as an a.p.s of X) :

Definition 12
Given a classification $\mathbf{C} = \langle E, F, \vdash \rangle$*, we define the most general approximation* $\mathrm{appr}_{\top}(\mathbf{C}) = \langle \wp(E), \wp(F), \lambda X.\mathrm{Typ}(X) \rangle$.

This is the exact translation of a classification in terms of lattices. It is the most general approximation, since there is no loss of information, so any property on the original classification can be transfered to its most general approximation. We have the correctness proposition :

Proposition 4
For any classification $\mathbf{C}$*, we have :*

$$\mathrm{cla}(\mathrm{appr}_{\top}(\mathbf{C})) = \mathbf{C}$$

The fact that $\mathrm{appr}_{\top}$ is the most general approximation comes from the following proposition :

Proposition 5
Given a classification $\mathbf{C}$*, we have :*

$$\forall\, \mathbf{A} \in A_{\mathbf{C}},\, \mathbf{A} \leq_{\mathrm{appr}} \mathrm{appr}_{\top}(\mathbf{C})$$

Conversely, we define a most abstract approximation :

Definition 13
Given a classification $\mathbf{C}$*, we define the most abstract approximation* $\mathrm{appr}_{\perp}(\mathbf{C}) = \langle \mathrm{tok}_{\perp},\ \mathrm{typ}_{\perp}, p_{\perp} \rangle$ *where :*

$$\begin{aligned}
\mathrm{tok}_{\perp} &= \left\{ X \subseteq \mathrm{tok}_{\mathbf{C}} \,/\, \forall\, \mathbf{A} \in A_{\mathbf{C}}, X = e^{\#}_{\mathrm{tok}_A} \circ e_{\mathrm{tok}_A}(X) \right\} \\
\mathrm{typ}_{\perp} &= \left\{ Y \subseteq \mathrm{typ}_{\mathbf{C}} \,/\, \forall\, \mathbf{A} \in A_{\mathbf{C}}, Y = e^{\#}_{\mathrm{typ}_A} \circ e_{\mathrm{typ}_A}(Y) \right\} \\
p_{\perp} &= e_{\mathrm{typ}_{\perp}} \circ \mathrm{Typ} \circ e^{\#}_{\mathrm{tok}_{\perp}}
\end{aligned}$$

Proposition 6
The following equalities do hold :

$$\begin{aligned}
\mathrm{typ}_{\perp} &= \{ \mathrm{Typ}(X) \mid X \subseteq \mathrm{tok}_{\mathbf{C}} \} \\
\mathrm{tok}_{\perp} &= \left\{ \bigcup \{ X \mid \mathrm{Typ}(X) \subseteq Y \} \;\middle|\; Y \in \mathrm{typ}_{\perp} \right\}
\end{aligned}$$

This proposition shows that there is a one-to-one correspondance between $\mathrm{typ}_{\perp}$ and $\mathrm{tok}_{\perp}$.

Proposition 7
For any classification $\mathbf{C}$, *we have :*

$$\mathrm{cla}(\mathrm{appr}_\perp(\mathbf{C})) = \mathbf{C}$$

Proof Let $x \in \mathrm{tok}_\mathbf{C}$. We define $[x] = \{y \,/\, \mathrm{Typ}(y) \subseteq \mathrm{Typ}(x)\}$. We have $\mathrm{Typ}(Y) \subseteq \mathrm{Typ}(x) \Leftrightarrow Y \subseteq [x]$. We prove that $[x] \in \mathrm{typ}_\perp$. Let $\mathbf{A} \in A_\mathbf{C}$. Galois-connection properties tell us that :

$$[x] \subseteq e^{\#}_{\mathrm{tok}_A} \circ e_{\mathrm{tok}_A}([x])$$
$$e_{\mathrm{tok}_A}([x]) = e_{\mathrm{tok}_A} \circ e^{\#}_{\mathrm{tok}_A} \circ e_{\mathrm{tok}_A}([x])$$

But since $\mathbf{A}$ is in $A_\mathbf{C}$, we have :

$$\begin{aligned}\mathrm{Typ}\left(e^{\#}_{\mathrm{tok}_A} \circ e_{\mathrm{tok}_A}([x])\right) &= e^{\#}_{\mathrm{typ}_A} \circ p_A \circ e_{\mathrm{tok}_A}(e^{\#}_{\mathrm{tok}_A} \circ e_{\mathrm{tok}_A}([x])) \\ &= e^{\#}_{\mathrm{typ}_A} \circ p_A \circ e_{\mathrm{tok}_A}([x]) \\ &= \mathrm{Typ}([x])\end{aligned}$$

So that $e^{\#}_{\mathrm{tok}_A} \circ e_{\mathrm{tok}_A}([x]) \subseteq [x]$ which leads to the equality. As this is true for any $\mathbf{A}$ in $A_\mathbf{C}$, we have $[x] \in \mathrm{tok}_\perp$. This also implies that $\mathrm{Typ}([x]) \in \mathrm{typ}_\perp$. Finally, there remains to show that $e^{\#}_{\mathrm{typ}_\perp} \circ p_\perp \circ e_{\mathrm{tok}_\perp}(x) = \mathrm{Typ}(x)$, but this comes from :

$$\mathrm{Typ}(x) \subseteq e^{\#}_{\mathrm{typ}_\perp} \circ p_\perp \circ e_{\mathrm{tok}_\perp}(x) \subseteq e^{\#}_{\mathrm{typ}_\perp} \circ p_\perp \circ e_{\mathrm{tok}_\perp}([x]) = \mathrm{Typ}(x)$$

Proposition 8
Given a classification $\mathbf{C}$, *we have :*

$$\forall\, \mathbf{A} \in A_\mathbf{C},\, \mathrm{appr}_\perp(\mathbf{C}) \leq_{\mathrm{appr}} \mathbf{A}$$

Justification It comes from the fact that $\forall\, X \in \mathrm{tok}_\perp,\, e^{\#}_{\mathrm{tok}_\perp}(X) = X = e^{\#}_{\mathrm{tok}_A} \circ e_{\mathrm{tok}_A}(X)$, so that $\mathrm{tok}_\perp \subseteq \mathrm{tok}_A$. The same argument applies to show that $\mathrm{typ}_\perp \subseteq \mathrm{typ}_A$.

Proposition 9
$\mathrm{tok}_{\mathrm{appr}_\perp(\mathbf{C})}$ *is isomorphic to* $\mathrm{Sep}(\mathbf{C})$ *as defined in [BS97] page 85.* $\mathrm{typ}_{\mathrm{appr}_\perp(\mathbf{C})}$ *is isomorphic to the regular theory* $\mathrm{Th}(\mathbf{C})$ *generated by* $\mathbf{C}$.

We can now give our main result concerning classifications and approximations :

Proposition 10
Given any classification $\mathbf{C}$, *we have :*

$$A_\mathbf{C} = \{\, \mathbf{A} \mid \mathrm{appr}_\perp(\mathbf{C}) \leq \mathbf{A} \} = \{\, \mathbf{A} \mid \mathbf{A} \leq \mathrm{appr}_\top(\mathbf{C})\}$$

Proof We only need to prove that $\mathrm{appr}_\perp(\mathbf{C}) \leq \mathbf{A} \leq \mathrm{appr}_\top(\mathbf{C}) \Rightarrow \mathbf{A} \in A_{\mathbf{C}}$. But since $A \leq \mathrm{appr}_\top(\mathbf{C})$, we have :

$$\forall\, X, e^{\#}_{\mathrm{typ}_A} \circ p_A \circ e_{\mathrm{tok}_A}(X) = e^{\#}_{\mathrm{typ}_\top} \circ \mathrm{Typ} \circ e_{\mathrm{tok}_\top}(X) = \mathrm{Typ}(X)$$

The different ways to encode a classification using approximations represent several approaches of what one can do with classifications. On the one hand, if we consider $\mathrm{appr}_\top$, none of the logic of the classification is encoded, which means that we want to keep the typing assertions as *partial* information on the tokens, and that there remains a way to sharpen the assertions. In particular, none of the tokens are aggregated to others, they are still distinct.

On the other hand, considering $\mathrm{appr}_\perp$, the situation is the same as if we looked at the tokens only through their types, i.e. considering the information we have on them. That means that different tokens having the same type are indistiguishable.

4 Infomorphisms and Galois connections

After having presented a way to formalize classifications using lattices, we now need to setup our formalism concerning informorphisms, i.e. to have the possibility to connect classifications (and their lattice counterparts) together. The basic idea of informorphisms is to connect classifications in a Galois connection-like way between tokens and types :

Definition 14 (Infomorphism)
An infomorphism *between two classifications C_1 and C_2 is a tuple $\langle f^\wedge, f^\vee \rangle$ verifying :*

$$\forall\, \alpha \in \mathrm{typ}_1, \forall\, b \in \mathrm{tok}_2,\, f^\vee(b) \vdash_1 \alpha \Leftrightarrow b \vdash_2 f^\wedge(\alpha)$$

It is natural to consider an equivalent for approximations which verifies a similar relation. This leads to the following definition :

Definition 15 (Approximation morphism)
Given two approximations $\mathbf{A}$ and $\mathbf{B}$, an approximation morphism *from $\mathbf{A}$ to $\mathbf{B}$ is a pair $\langle g^\triangle, g^\triangledown \rangle$ verifying the property :*

$$\forall\, \alpha \in \mathrm{typ}_A, \forall\, b \in \mathrm{tok}_B, p_A \circ g^\triangledown(b) \leq \alpha \Leftrightarrow p_B(b) \leq g^\triangle(\alpha)$$

Given an approximation morphism $\langle g^\triangle, g^\triangledown \rangle$, there is a strong relation between the two functions of the pair. This is expressed in the next propositions :

Proposition 11
The two morphisms $p_A \circ g^\triangledown$ and $p^{\#}_B \circ g^\triangle$ form a Galois connection.

Proof Using the former definition, we can write

$$p_A \circ g^{\bigtriangledown}(b) \leq \alpha \Leftrightarrow p_B(b) \leq g^{\bigtriangleup}(\alpha) \Leftrightarrow b \leq p_B^{\#} \circ g^{\bigtriangleup}(\alpha)$$

As a consequence, using the uniqueness of functions in a Galois connection, we have the following proposition :

Proposition 12
Given an approximation morphism g from $\mathbf{A}$ to $\mathbf{B}$, the two functions $g^{\bigtriangleup}$ and $g^{\bigtriangledown}$ verify :

$$g^{\bigtriangledown} \leq p_A^{\#} \circ g^{\bigtriangleup\#} \circ p_B \qquad g^{\bigtriangleup} \geq p_B \circ g^{\bigtriangledown\#} \circ p_A^{\#}$$

A consequence of this is that given a morphism $g^{\bigtriangleup} : \mathrm{typ}_A \to \mathrm{typ}_B$, the pair $\langle g^{\bigtriangleup}, p_A^{\#} \circ g^{\bigtriangleup\#} \circ p_B \rangle$ forms an approximation from $\mathbf{A}$ to $\mathbf{B}$. The same can also be done given $g^{\bigtriangledown}$.

Proposition 13
Given a morphism $g^{\bigtriangleup} : \mathrm{typ}_A \to \mathrm{typ}_B$, there exists a morphism $g^{\bigtriangledown} : \mathrm{tok}_B \to \mathrm{tok}_A$ such that $\langle g^{\bigtriangleup}, g^{\bigtriangledown} \rangle$ is an approximation morphism from A to B.

We now turn to the translation between informorphisms and approximation morphisms. In the following, given a classification $\mathbf{C}$, the corresponding approximation considered will be $\mathrm{appr}_{\top}(\mathbf{C})$. This way, the results are expressed using subsets, which increases their readability. Another reason is that we consider the most general approximation of a given classification, in which all the following results apply, while it might not be the case for more abstract approximations.

Definition 16
Given an informorphism $f : C_1 \to C_2$, we define the approximation morphism $a(f)$ by :

$$a(f) = \big\langle \lambda X.\{f^{\wedge}(x) \,/\, x \in X\}, \lambda Y.\{f^{\vee}(y) \,/\, y \in Y\}\big\rangle$$

$a(f)$ is indeed an approximation morphism since :

$$\begin{aligned} p_1 \circ a(f)^{\bigtriangledown}(X) \subseteq Y &\Leftrightarrow \big(\forall\, x \in \mathrm{tok}_2,\ \forall\, \alpha \in \mathrm{typ}_1,\ f^{\vee}(x) \vdash \alpha \Rightarrow \alpha \in Y\big) \\ &\Leftrightarrow \big(\forall\, x \in \mathrm{tok}_2,\ \forall\, \alpha \in \mathrm{typ}_1,\ x \vdash f^{\wedge}(\alpha) \Rightarrow \alpha \in Y\big) \\ &\Leftrightarrow p_2(X) \subseteq a(f)^{\bigtriangleup}(Y) \end{aligned}$$

There is no reverse transformation in general, since informorphisms are defined on elements, whereas approximation morphisms are defined on sets.

Now, there remains to explore the ways a morphism $f^{\wedge}$ can be embedded in an informorphism. Suppose we have a morphism $f^{\wedge} : \mathrm{typ}_1 \to \mathrm{typ}_2$. We define :

$$g^{\bigtriangleup} = \lambda X.\{f^{\wedge}(x) \,/\, x \in X\}$$

Using the former proposition, let's define $g^{\triangledown} = p_1^{\#} \circ g^{\triangle} \circ p_2$. We have :

$$g^{\triangledown}(\{y\}) = \{x \,/\, g^{\triangle}(\mathrm{typ}_1(x)) \subseteq \mathrm{typ}_2(y)\} = \{x \,/\, \forall\, \alpha,\ x \vdash \alpha \Rightarrow y \vdash f^{\wedge}(\alpha)\}$$

Now suppose we have $f^{\vee}$ such that $\lambda Y.\{f^{\vee}(y) \,/\, y \in Y\} \leq g^{\triangledown}$. Then,

$$\forall\, y,\ \{f^{\vee}(y)\} \subseteq g^{\triangledown}(\{y\})$$

so that we need to have $\forall\, y,\ g^{\triangledown}(\{y\}) \neq \emptyset$ which is equivalent to :

$$\forall\, y, \exists\, x : \forall\, \alpha, x \vdash \alpha \Rightarrow y \vdash f^{\wedge}(\alpha)$$

This is precisely the condition we need to verify for every informorphism, and any $f^{\vee}$ such that $\lambda Y.\{f^{\vee}(y) \,/\, y \in Y\} \leq g^{\triangledown}$ do form an infomorphism with $f^{\wedge}$.

Proposition 14
Given a morphism $f^{\wedge} : \mathrm{typ}_1 \to \mathrm{typ}_2$*, there exists* $f^{\vee} : \mathrm{tok}_2 \to \mathrm{tok}_1$ *such that* $\langle f^{\wedge}, f^{\vee} \rangle$ *is an infomorphism if and only if :*

$$\forall\, y, \exists\, x : \forall\, \alpha, x \vdash \alpha \Rightarrow y \vdash f^{\wedge}(\alpha)$$

It should be noted that this result can be found without using any lattice formalism. But what is important is that approximation morphisms can exactly represent infomorphisms (which comes from the previous proposition) but are strictly more expressive, since any morphism can be turned into an approximation morphism.

5 Conclusion

The work presented here is just a sketch of what would be the whole information flow theory expressed in a lattice formalism. The central structure of the theory, namely the information channel, has not been studied. Yet, all the necessery machinery is provided in this paper.

For further study, more investigation might be done concerning the effects of adding imprecision (and so approximations) to classifications. As we have seen, a first result is that it allows to consider more morphismes between classification structures.

Another direction can be presented this way : consider two classifications $\mathbf{C}_1$ and $\mathbf{C}_2$ with the same token set. Then, to what extend can one build a new classification $\mathbf{D}$ where $\mathrm{tok}_{\mathbf{D}} = \mathrm{typ}_{\mathbf{C}_1}$ and $\mathrm{typ}_{\mathbf{D}} = \mathrm{typ}_{\mathbf{C}_2}$ and which would be the properties of this classification.

Finally, a last direction would be to explore the equivalent to classification manipulation to approximations. This is closely related to abstract domain manipulation as developped for instance by R. Giacobazzi, F. Ranzato and F. Scozzari [GRS00].

References

[Bir67] G. Birkhoff. *Lattice Theory*. Amer. Math. Soc., 1967. 3rd ed., Colloquium Publications.

[BS97] J. Barwise and J. Seligman. *Information Flow, the logic of distributed systems.* Cambridge University Press, 1997.

[Cou96] P. Cousot. *Abstract Interpretation.* Symposium on Models of Programming Languages and Computation, ACM Computing Surveys, 28(2):324-328, June 1996.

[GW99] B. Ganter and G. Wille. *Formal Concept Analysis.* Springer Verlag, 1999.

[GRS00] R. Giacobazzi, F. Ranzato and F. Scozzari. *Making Abstract Interpretations Complete.* To appear in Journal of the ACM, may 2000 issue.