Classifications et treillis

Olivier Brunet

Classifications et Treillis

Olivier Brunet

September 29, 2000

1 Introduction

Dans leur livre *Information Flow : The Logic of Distributed Systems*, Jon Barwise et Jerry Seligman développent une théorie formalisant les échanges d'information entre systèmes complexes. Cette théorie est basée sur un formalisme comprenant deux entitées fondamentales : les classifications et les infomorphismes. Les *classifications* sont la donnée de deux ensembles, les *objets* (en anglais *tokens*) qui seront les éléments étudiés et les *types* qui seront utilisés pour représenter l'information disponible sur les objets. Le fait d'avoir de l'information est formalisée par une relation binaire entre les deux ensembles.

Les autres entitées de base de la théories sont les infomorphismes qui sont des fonctions entre classifications. De façon algébriques, les classifications et les infomorphismes forment une catégorie.

A partir de ce formalisme, ils développent des méthodes permettant de caractériser des logiques à partir du comportement des objets et des types correspondant. De plus, par le biais des infomorphismes, ils étendent leur étude à des systèmes complexes composés de la mise en relation de plusieurs sous-parties (ce qui se traduit par la composition de classifications).

Nous présentons dans cet article une traduction de cette théorie sous forme de treillis et de correspondance de Galois. La principale motivation de cette traduction vient du fait que le treillis représente une structure naturelle pour inclure dans la représentation une notion d'approximation, qui est formalisée par un ordre sur les éléments. Comme nous le verrons dans la dernière partie, cela enrichit la théorie, puisque cela nous permet de considérer un plus grand nombre de relations entre classifications.

De plus, ce formalisme est déjà utilisé dans de nombreux domaines comme l'analyse de programmes [CC77] [CC79] et l'analyse de concepts [GW99]. Enfin, il possède une base mathématique très développée (voir par exemple [Bir67]). Cela permet de bénéficier d'outils et de méthodes existant par ailleurs.

Les trois parties de cet articles sont chacune dédiées à l'étude de la traduction d'un élément du formalisme de l'*information flow theory* sous forme de treillis. Ainsi, dans la première partie, nous traiterons des théories régulières et de leur traduction en *ensembles de parties abstraits*. Ce sera aussi l'occasion de présenter les conventions utilisées. Dans la deuxième partie, nous nous intéresserons aux classifications et à leurs équivalents, les *approximations*. La troisième partie sera consacrée à l'étude de la possibilité de traduction entre les infomorphismes et les *morphismes d'approximations*. Nous verrons en particulier que cette traduction n'est pas toujours possible, les premiers étant moins expressifs que les seconds.

2 Représentation de théories régulières

Une manière efficace pour formaliser le raisonnement sur un ensemble Σ donné est de lui adjoindre un logique exprimée sous la forme d'une relation binaire $\vdash$ entre les sous-ensembles de Σ. On a alors $\Gamma \vdash \Delta$ qui signifie que si chaque élément de Γ est vrai, alors l'un des éléments de Δ l'est aussi.

Définition 2.1 (Théorie) *Une* théorie *est une paire* $\langle \Sigma, \vdash \rangle$ *où* Σ *est un ensemble et où :*

$$\vdash \in \wp(\Sigma) \times \wp(\Sigma)$$

Par la suite, nous nous limiterons aux théories régulières, où $\vdash$ vérifie les axiomes supplémentaires suivants :

Définition 2.2 (Théorie régulière) *Une théorie* $\langle \Sigma, \vdash \rangle$ *est dite* régulière *si et seulement si :*

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

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

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

respectivement appelés Identité, Affaiblissement *et* Coupure globale.

Le résultat principal de cette partie sera que pour un ensemble Σ donné, les théories régulières sur cet ensemble seront en bijection avec certains ensembles de sous-parties de Σ. Nous allons exprimer cela de façon purement algébrique en utilisant un formalisme de treillis. La prochaine sous-section va être l'occasion de développer ce formalisme et de préciser les notations.

2.1 Correspondances de Galois

Tous les ensembles que nous allons considérer par la suite seront munis d'une structure de *treillis complet*. Pour un ensemble quelconque E, on le transforme en treillis complet en considérant le n-uplet $\langle \wp(E), \subseteq, \cup, \cap, E, \emptyset \rangle$. Dans le cas général, on notera un treillis complet $\langle E, \leq_E, \vee_E, \wedge_E, \top_E, \bot_E \rangle$ ou simplement E si il n'y a pas de confusion possible.

Définition 2.3 (Morphisme additif) *Un morphisme additif entre deux treillis complets E et F est une fonction $f : E \to F$ vérifiant :*

$$\forall\, x \leq_E y,\ f(x) \leq_F f(y)$$
$$\forall\, X \subseteq E,\ f\left(\bigvee \{x \,/\, x \in X\}\right) = \bigvee \{f(x) \,/\, x \in X\}$$

Par la suite, les morphismes considérés seront toujours supposés additifs. L'ensemble des morphismes entre deux treillis E et F donnés est ordonné par :

$$f \leq g \Leftrightarrow \forall\, x \in E,\ f(x) \leq_F f(y)$$

Définition 2.4 (Adjoint) *Étant donné un morphisme $f : E \to F$, on définit son* adjoint *$f^{\#} : F \to E$ par :*

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

Un morphisme et son adjoint forment une correspondance de Galois entre les deux treillis, ce qui se traduit ainsi :

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

On rappelle la propriété d'unicité suivante : deux fonctions f et g forment une correspondance de Galois si et seulement si on a $g = f^{\#}$.

2.2 Ensemble de parties abstrait

Nous allons maintenant introduire les *ensembles de parties abstraits* qui sont la traduction algébrique de familles de sous-ensembles stables par intersection.

Définition 2.5 (Ensemble de parties abstrait) *Étant donné un ensemble Σ, un* ensemble de parties abstrait *(e.p.a.) sur Σ est une paire $\langle P, e\rangle$ avec P un treillis complet et e un morphisme surjective de $\wp(\Sigma)$ vers P vérifiant $e(\Sigma) = \top_P$.*

Le premier résultat comme nous allons montrer est que pour un ensemble Σ donné, on peut associer de façon bijective les théories régulières et les e.p.a. Pour le prouver, nous exhiberons les fonctions réalisant cette bijection.

Définition 2.6 *Étant donné un e.p.a. $\langle P, e\rangle$ sur Σ, on définit la logique $\langle \Sigma, \vdash_P\rangle$ avec :*

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

ce qui est équivalent à :

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

Proposition 2.1 *$\langle \Sigma, \vdash_P\rangle$ est une théorie régulière.*

Preuve Les axiomes *Identité* et *Affaiblissement* sont vérifiés de façon évidente. Il reste à prouver la *Coupure Globale*. Pour cela, soit $\Gamma, \Delta, \Sigma' \in \Sigma$. On va définir la partition $\langle \Sigma_0, \Sigma_1 \rangle$ en choisissant $\Sigma_0 = \{\sigma \in \Sigma \,/\, \Gamma \vdash_P \sigma\}$. On suppose que :

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

Or, on a $e(\Gamma \cup \Sigma_0) = e(\Gamma) \vee \bigvee \{e(\{\sigma\}) \,/\, \sigma \in \Sigma_0\}$. Et comme par définition, $\forall\, \sigma \in \Sigma_0$, $e(\{\sigma\}) \leq e(\Gamma)$, on a $e(\Gamma \cup \Sigma_0) = e(\Gamma)$. Ainsi,

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

Maintenant, on a $\Gamma \not\vdash \Sigma_1$, donc :

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

Ceci termine la preuve.

Réciproquement, on peut définir facilement un e.p.a. à partir d'une logique régulière en prenant comme treillis l'ensemble des sous-ensembles de Σ clos par déduction, ordonnés par l'inclusion. On note pour cela $\Sigma_\vdash$ l'ensemble des parties de Σ closes par déduction, i.e. les Γ tel que $\forall\, \delta,\, \Gamma \vdash \delta \Rightarrow \delta \in \Gamma$.

Définition 2.7 *Étant donné une théorie régulière $\langle \Sigma, \vdash \rangle$, on définit l'e.p.a. $\langle \Sigma_\vdash, e \rangle$ où :*

- *$\Sigma_\vdash$ est ordonné par la restriction de $\subseteq$ à $\Sigma_\vdash$ et*
- *$e = \lambda X. \bigcap \{Y \in \Sigma_\vdash \,/\, X \subseteq Y\}$*

Proposition 2.2 *Pour un ensemble Σ donné, il existe une bijection entre les théories régulières et les e.p.a. . De plus, cette bijection est réalisée par les deux transformations précédentes.*

Justification Cela vient du fait qu'une logique régulière est uniquement déterminée par les sous-parties qu'elle stabilise :

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

Pour revenir à la présentation des e.p.a., ils correspondent bien à des ensembles de sous-parties clos par intersection, puisque c'est le cas de l'ensembles de sous-ensembles clos par déduction pour une théorie régulière donnée.

3 Classifications et Approximations

Nous allons maintenant nous intéresser aux structures principales de notre formalisation, en l'occurence les classifications introduites par Barwise et Seligman, et les approximations qui leur correspondent dans l'approche par treillis. Contrairement à la partie précédente, le résultat que nous allons montrer n'est pas un résultat d'équivalence entre les deux formalismes. En fait, une classification peut être représentée par plusieurs approximations, lesquelles pouvant comme nous le verrons être ordonnées suivant un critère d'internalisation de la structure.

Nous commençons par rappeler la définition d'une classification telle qu'elle est donnée dans [BS97] :

Définition 3.1 (Classification) *Une* classification $\mathbf{C} = \langle \mathrm{tok}_C, \mathrm{typ}_C, \models \rangle$ *est la donnée de :*

1. *un ensemble* tok_C *d'objets à étudier,*
2. *un ensemble* typ_C *de types utilisés pour classifier les objects, et*
3. *une relation binaire de classification* $\models$ *entre* tok_C *et* typ_C.

Lorsque $a \models \alpha$, nous dirons que a est de type α. Pour la suite, nous définissons $\mathrm{Typ}(x) = \{\alpha \,/\, x \models \alpha\}$ et $\mathrm{Typ}(X) = \bigcup \{\mathrm{Typ}(x) \,/\, x \in X\}$. De même, soit $\mathrm{Tok}(\alpha) = \{x \,/\, x \models \alpha\}$ et $\mathrm{Tok}(A) = \bigcup \{\mathrm{Tok}(\alpha) \,/\, \alpha \in A\}$.

Exemple 1 *Considérons un circuit éléctrique composé d'une pile, de deux interrupteurs et d'une lampe. A chaque instant $\tau \in T$, on peut décrire son état par un triplet $S(\tau) = \langle I_1(\tau), I_2(\tau), L(\tau)\rangle$ où $I_i(\tau) \in \{\mathtt{On}, \mathtt{Off}\}$ indique la position de l'interrupteur i au temps τ, et $L(\tau) \in \{\mathtt{Lit}, \mathtt{Unlit}\}$ celui de la lampe. Par la suite, on écrira l'état d'un circuit comme une suite de trois bits. Par exemple,* `101` *signifie que le premier interrupteur est un position* `On`*, le second est* `Off` *et la lampe est allumée.*

On peut donc considérer la classification **Cir** *où* $\mathrm{tok}_{\mathbf{Cir}} = T$ *et* $\mathrm{typ}_{\mathbf{Cir}} = \mathbf{2}^3$ *l'ensemble des suites de 3 bits. La relation de typage* $\models_{\mathbf{Cir}}$ *associera alors à un moment* τ *donné l'état du circuit à cet instant :*

$$\tau \models_{\mathbf{Cir}} S \Leftrightarrow S = S(\tau)$$

Exemple 2 *Dans l'exemple précédent, la relation de typage peut être vue comme une fonction puisqu'elle associe à chaque objet exactement un type. Avec les mêmes notations, considérons une classification où les objets sont les états de la lampe et les types ceux des interrupteurs. Cela donne* $\mathbf{Lamp} = \langle \mathbf{2}^1, \mathbf{2}^2, \models_{\mathrm{L}} \rangle$ *où :*

$$\alpha \models_{\mathrm{L}} \langle \beta_1, \beta_2 \rangle \Leftrightarrow \exists\, \tau \in T\colon\, S(\tau) = \langle \beta_1, \beta_2, \alpha \rangle$$

Dans cet exemple, on voit qu'en général, un objet est associé à plusieurs types.

Définition 3.2 (Approximation) *Une* approximation *de E par F est un triplet* $\mathbf{A} = \langle \mathrm{tok}_A, \mathrm{typ}_A, p_A \rangle$ *où :*

1. tok_A *(resp.* typ_A*) est un e.p.a. de E (resp. of F)*
2. $p_A : \mathrm{tok}_A \to \mathrm{typ}_A$ *est la relation d'approximation entre les classes d'éléments de E (les élements de* tok_A*) et les classes d'éléments de F.*

A chaque approximation correspond une classification que nous pouvons définir de la façon suivante :

Définition 3.3 *Etant donné une approximation* $\mathbf{A}$ *entre* E *et* F*, on définit la classification* $\mathrm{cla}(\mathbf{A}) = \langle E, F, \models_{\mathbf{A}} \rangle$ *où :*

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

Dans le sens contraire, on se convaincra aisément qu'une classification peut être traduite sous forme d'approximation. En effet, il suffit de se rappeler que l'ensemble des parties d'un ensemble peut être vu comme un e.p.a. de cet ensemble. Nous définissons donc :

Définition 3.4 *Etant donné une classification* $\mathbf{C}$*, on définit l'approximation* $\mathrm{appr}_\top(\mathbf{C}) = \langle \wp(\mathrm{tok}_{\mathbf{C}}), \wp(\mathrm{typ}_{\mathbf{C}}), \lambda X. \mathrm{Typ}_{\mathbf{C}}(X) \rangle$.

Exemple 3 *En reprenant la classification de l'exemple précédent, on a :*

$$\mathrm{appr}_\top(\mathbf{Cir}) = \left\langle \wp(T), \wp(\mathbf{2}^3), \lambda X. \bigcup \{\mathrm{Typ}(x) \,/\, x \in X\} \right\rangle$$

Cette transformation est correcte, en ce sens qu'elle permet de retrouver la classification de départ, ce que l'on exprime par :

Proposition 3.1 *Pour tout classification* $\mathbf{C}$*, on a :*

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

Le principal résultat de cette partie va être de définir l'ensemble des approximations représentant une classification donnée, et exhiber un aspect de sa structure. Nous allons cependant nous restreindre aux approximations qui représentent une classification $\mathbf{C}$ non seulement sur les atomes de $\mathrm{tok}_{\mathbf{C}}$ mais qui préservent aussi l'union, ce qui assure l'additivité des morphismes. Nous définissons donc :

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

Nous allons définir sur l'ensemble des approximations l'ordre suivant :

Définition 3.5 *Etant donné deux approximations* $\mathbf{A} = \langle A, B, p \rangle$ *et* $\mathbf{A}' = \langle A', B', p' \rangle$*, on définit :*

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

où $E \sqsubseteq E'$ *signifie* $\{e^{\#}_E(x) \,/\, x \in E\} \subseteq \{e^{\#}_{E'}(x) \,/\, x \in E'\}$.

Dans cette définition, la dernière condition revient à dire que $\mathrm{cla}(\mathbf{A}) = \mathrm{cla}(\mathbf{A}')$. Selon cet ordre, si une approximation est sous une autre, cela signifie qu'elle distingue moins d'objets et de types. En particulier, pour cette ordre, $\mathrm{appr}_\top(\mathbf{C})$ est la plus grande approximation de $A_{\mathbf{C}}$.

Proposition 3.2 *On a pour toute classification* $\mathbf{C}$ *:*

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

Nous allons maintenant définir une autre approximation, et nous montrerons qu'elle est la plus petite.

Définition 3.6 *Etant donné une classification* $\mathbf{C}$*, nous définissons* $\text{appr}_{\perp}(\mathbf{C}) = \langle \text{tok}_{\perp},\ \text{typ}_{\perp}, p_{\perp}\rangle$ *où :*

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

Proposition 3.3 *Pour toute classification* $\mathbf{C}$*, on a :*

$$\begin{aligned}
\text{typ}_{\perp} &= \{ \text{Typ}_{\mathbf{C}}(X) \,/\, X \subseteq \text{tok}_{\mathbf{C}} \} \\
\text{tok}_{\perp} &= \left\{ \bigcup \{ X \,/\, \text{Typ}(X) \subseteq Y \} \,/\, Y \in \text{typ}_{\perp} \right\}
\end{aligned}$$

En particulier, les deux ensembles sont isomorphes.

Exemple 4 *Continuons d'examiner la classification* $\mathbf{Cir}$*. On va supposer que dans le circuit considéré, les interrupteurs sont montés en série. On a alors :*

$$\text{typ}_{\perp}(\mathbf{Cir}) = \wp\left(\{\mathtt{000}, \mathtt{100}, \mathtt{011}, \mathtt{111}\}\right)$$

Si les interrupteurs avaient étés montés en va-et-vient, on aurait pu avoir :

$$\text{typ}'_{\perp}(\mathbf{Cir}) = \wp\left(\{\mathtt{000}, \mathtt{101}, \mathtt{011}, \mathtt{110}\}\right)$$

Proposition 3.4 *Pour toute classification* $\mathbf{C}$*, on a :*

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

Preuve Soit $x \in \text{tok}_{\mathbf{C}}$. On définit $[x] = \{ y \,/\, \text{Typ}(y) \subseteq \text{Typ}(x)\}$, l'ensemble des y dont le type est inclus dans celui de x. On a en particulier :

$$\text{Typ}(Y) \subseteq \text{Typ}(x) \Leftrightarrow Y \subseteq [x]$$

Nous allons montrer que $[x] \in \text{typ}_{\perp}$. Il suffit pour cela de montrer que $[x] = e^{\#}_{\text{tok}_A} \circ e_{\text{tok}_A}([x])$ pour tout $\mathbf{A} \in A_{\mathbf{C}}$. En utilisant des propriétés des correspondances de Galois, on obtient :

$$\begin{aligned}
[x] &\subseteq e^{\#}_{\text{tok}_A} \circ e_{\text{tok}_A}([x]) \\
e_{\text{tok}_A}([x]) &= e_{\text{tok}_A} \circ e^{\#}_{\text{tok}_A} \circ e_{\text{tok}_A}([x])
\end{aligned}$$

Or, si $\mathbf{A}$ est dans $A_{\mathbf{C}}$, on en déduit :

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

Ainsi, par définition de $[x]$, cela signifie que $e^{\#}_{\mathrm{tok}_A} \circ e_{\mathrm{tok}_A}([x]) \subseteq [x]$. L'inégalité inverse venant des propriétés de $e^{\#}_{\mathrm{tok}_A} \circ e_{\mathrm{tok}_A}$, on a donc :

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

On a donc montré que $[x] \in \mathrm{tok}_\perp$. Cela implique de plus que $\mathrm{Typ}(x) = \mathrm{Typ}([x]) \in \mathrm{typ}_\perp$. Il reste à montrer que $e^{\#}_{\mathrm{typ}_\perp} \circ p_\perp \circ e_{\mathrm{tok}_\perp}(x) = \mathrm{Typ}(x)$.

Mais on a $\mathrm{Typ}(x) \subseteq e^{\#}_{\mathrm{typ}_\perp} \circ p_\perp \circ e_{\mathrm{tok}_\perp}(x) \subseteq e^{\#}_{\mathrm{typ}_\perp} \circ p_\perp \circ e_{\mathrm{tok}_\perp}([x]) = \mathrm{Typ}(x)$ ce qui permet de conclure.

Proposition 3.5 *Etant donné une classification* $\mathbf{C}$*, on a :*

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

Justification Comme :

$$\forall\, X \in \mathrm{tok}_\perp,\, e^{\#}_{\mathrm{tok}_\perp}(X) = X = e^{\#}_{\mathrm{tok}_A} \circ e_{\mathrm{tok}_A}(X)$$

on a $\mathrm{tok}_\perp \sqsubseteq \mathrm{tok}_A$. De même, $\mathrm{typ}_\perp \sqsubseteq \mathrm{typ}_A$.

Proposition 3.6 $\mathrm{tok}_{\mathrm{appr}_\perp(\mathbf{C})}$ *est isomorphe à* $\mathrm{Sep}(\mathbf{C})$ *tel qu'elle est définie dans [BS97] page 85.* $\mathrm{typ}_{\mathrm{appr}_\perp(\mathbf{C})}$ *est isomorphe à la théorie régulière* $\mathrm{Th}(\mathbf{C})$ *engendrée par* $\mathbf{C}$*.*

Cette proposition exprime le fait que dans le cas de $\mathrm{appr}_\perp$, l'ensemble des objets étudiés est réduit de telle sorte que l'on ne puisse plus distinguer des objets de même type. Ainsi, l'information obtenue par observation du type est incluse dans l'ensemble des objets.

Nous arrivons maintenant au dernier résultat concernant les ensembles $A_{\mathbf{C}}$:

Proposition 3.7 *Etant donné une classification* $\mathbf{C}$*, on a :*

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

Preuve Cela vient simplement du fait que si deux approximations sont comparables, cela implique qu'elles représentent une même classification et qu'elles donnent les bons types sur toutes les sous-parties. Ainsi, si $\mathbf{A} \leq_{\mathrm{appr}} \mathbf{A'}$, on a $\mathbf{A} \in A_{\mathbf{C}} \Leftrightarrow \mathbf{A'} \in A_{\mathbf{C}}$.

Nous voyons donc que les différentes approximations d'une même classification viennent de la quantité d'information de typage que l'on inclut dans l'ensemble des objets observés, ainsi que de la structuration des types en fonction des observations. Cette hiérarchie peut aussi être interprétée différemment. Ainsi, dans le cas $\mathrm{appr}_\perp$, les objets sont uniquement définis par leur type. A l'inverse, pour $\mathrm{appr}_\top$, les types obtenus ne sont qu'un résultat mais ne sont pas considérés comme définitifs pour classer les objets, comme c'est le cas pour $\mathrm{appr}_\perp$.

4 Infomorphismes et Correspondances de Galois

Il reste dans cette partie à présenter la dernière composante de notre formalisme en introduisant les morphismes, appelés informorphismes d'une part et morphismes d'approximations d'autre part. Nous donnerons ensuite des méthodes de traductions entre les deux formalismes, et montrerons que pour les morphismes, le cadre utilisant les approximations est plus général que celui utilisant les classifications.

Lorsque ce ne sera pas précisé, étant donné une classification $\mathbf{C}$, l'approximation correspondante sera $\mathrm{appr}_\top(\mathbf{C})$. En effet, l'utilisation d'une approximation moins générale représentant la classification donnée affaiblirait la validité des propositions à venir.

Définition 4.1 (Infomorphisme) *Un* infomorphisme *entre deux classifications C_1 et C_2 est une paire de fonctions $\langle f^\wedge, f^\vee \rangle$ verifiant :*

$$\forall\, \alpha \in \mathrm{typ}_1,\ \forall\, b \in \mathrm{tok}_2,\ \ f^\vee(b) \models_1 \alpha \Leftrightarrow b \models_2 f^\wedge(\alpha)$$

Exemple 5 *Nous allons considérer que pour la classification* **Cir** *des exemples précédents, le montage des interrupteurs est effectué en série. Soit de plus la classification* $\mathbf{And} = \left\langle \mathbf{2}^1, \mathbf{2}^2, \models_\wedge \right\rangle$ *où :*

$$\alpha \models_\wedge \langle \beta_1, \beta_2 \rangle \Leftrightarrow \beta_1 \wedge \beta_2 = \alpha$$

Alors la paire de fonction :

$$\langle \tau \mapsto L(\tau), \langle \alpha, \beta \rangle \mapsto \langle \alpha, \beta, \alpha \wedge \beta \rangle \rangle$$

forme un infomorphisme de **And** *vers* **Cir***.*

Définition 4.2 (Morphisme d'approximation) *Etant donné deux approximations* **A** *et* **B***, un* morphisme d'approximation *de* **A** *vers* **B** *est une paire* $\left\langle g^\triangle, g^\triangledown \right\rangle$ *vérifiant :*

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

Les deux fonctions apparaissant dans un morphisme d'approximation $\left\langle g^\triangle, g^\triangledown \right\rangle$ sont liées par la relation suivante :

Proposition 4.1 *Les morphismes $p_A \circ g^\triangledown$ et $p_B^\# \circ g^\triangle$ forment une correspondance de Galois.*

Preuve En utilisant la définition précédente, on obtient :

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

Une conséquence de cette proposition, en utilisant les résultats d'unicité dans les correspondance de Galois, est la suivante :

Proposition 4.2 *Etant donné un morphisme d'approximation g de $\mathbf{A}$ vers $\mathbf{B}$, les fonctions $g^{\triangle}$ et $g^{\triangledown}$ verifient :*

$$g^{\triangledown} \leq p_A^{\#} \circ g^{\triangle\#} \circ p_B \qquad g^{\triangle} \geq p_B \circ g^{\triangledown\#} \circ p_A^{\#}$$

On en déduit en particulier qu'à partir d'une fonction entre deux approximations, on peut construire un morphisme d'approximations : si $g^{\triangle} : \mathrm{typ}_A \to \mathrm{typ}_B$ est un morphisme de treillis, la paire $\langle g^{\triangle}, p_A^{\#} \circ g^{\triangle\#} \circ p_B \rangle$ forme une approximation de $\mathbf{A}$ to $\mathbf{B}$. Une construction similaire peut se faire à partir de $g^{\triangledown}$.

Proposition 4.3 *Etant donné un morphisme $g^{\triangle} : \mathrm{typ}_A \to \mathrm{typ}_B$, il existe $g^{\triangledown} : \mathrm{tok}_B \to \mathrm{tok}_A$ tel que $\langle g^{\triangle}, g^{\triangledown} \rangle$ est un morphisme d'approximation de A vers B.*

Nous allons maintenant voir que si l'on peut toujours transformer un infomorphisme en morphisme d'approximation, l'inverse n'est pas toujours vrai.

Définition 4.3 *Etant donné un informorphisme $f : C_1 \to C_2$, on définit $a(f)$ par :*

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

Proposition 4.4 *Avec la définition précédente, $a(f)$ est un morphisme d'approximations*

Preuve En effet, nous avons :

$$\begin{aligned} p_1 \circ a(f)^{\triangledown}(X) \subseteq Y &\Leftrightarrow (\forall x \in X, \forall \alpha \in \mathrm{typ}_1, \, f^{\vee}(x) \models \alpha \Rightarrow \alpha \in Y) \\ &\Leftrightarrow (\forall x \in X, \forall \alpha \in \mathrm{typ}_1, \, x \models f^{\wedge}(\alpha) \Rightarrow \alpha \in Y) \\ &\Leftrightarrow p_2(X) \subseteq a(f)^{\triangle}(Y) \end{aligned}$$

Nous allons maintenant explorer les propositions inverses des propositions 4.3 et 4.4.

Soit un morphisme $f^{\wedge} : \mathrm{typ}_1 \to \mathrm{typ}_2$. On définit $g^{\triangle} = \lambda X.\{f^{\wedge}(x) \,/\, x \in X\}$. En utilisant la proposition 4.3, soit $g^{\triangledown} = p_1^{\#} \circ g^{\triangle} \circ p_2$. Plus précisement, on a :

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

Maintenant, si il existe $f^{\vee}$ tel que $\lambda Y.\{f^{\vee}(y) \,/\, y \in Y\} \leq g^{\triangledown}$, on a en particulier :

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

Une condition necessaire pour que ce soit le cas est donc $\forall y,\, g^{\triangledown}(y) \neq \emptyset$, soit exprimé en fonction de $f^{\wedge}$:

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

Cette condition est suffisante pour que $\langle f^{\wedge}, f^{\vee}\rangle$ forme un infomorphisme, ce que l'on exprime ainsi :

Proposition 4.5 *Etant donné* $f^{\wedge} : \mathrm{typ}_1 \to \mathrm{typ}_2$*, il existe un morphisme* $f^{\vee} : \mathrm{tok}_2 \to \mathrm{tok}_1$ *tel que* $\langle f^{\wedge}, f^{\vee}\rangle$ *est un informorphisme si et seulement si :*

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

Cette propriété est l'équivalent pour les informorphismes de la propriété 4.3 pour les morphismes d'approximations. On s'aperçoit en particulier que ces derniers sont en un sens plus généraux, puisqu'un morphisme de *poset* additif peut toujours être inclus dans un morphisme d'approximations.

Exemple 6 *En plus de la classification* **And** *vue précédemment, considéront* $\mathbf{Or} = \left\langle \mathbf{2}^1, \mathbf{2}^2, \models_{\vee} \right\rangle$ *avec* $\alpha \models_{\vee} \langle \beta_1, \beta_2\rangle \Leftrightarrow \alpha = \beta_1 \vee \beta_2$*. Posons* $f^{\wedge} : \mathrm{typ}_{\mathbf{And}} \to \mathrm{typ}_{\mathbf{Or}}$ *comme étant l'identité, ce qui signifie que si l'on s'imagine deux circuits, l'un monté en série et l'autre en parallèle, on cherche à identifier les deux circuits suivant leurs positions d'interrupteurs. D'après la proposition précédente, on peut include* $f^{\wedge}$ *dans un infomorphisme si et seulement si :*

$$\forall \beta,\, \exists \alpha : \forall x_1, x_2,\, \alpha \models_{\vee} \langle x_1, x_2\rangle \Rightarrow \beta \models_{\wedge} \langle x_1, x_2\rangle$$

soit exprimé autrement,

$$\forall \beta,\, \exists \alpha : x_1 \vee x_2 = \alpha \Rightarrow x_1 \wedge x_2 = \beta$$

Cela est faux, il suffit de considérer $\alpha = \beta = \mathtt{1}$*.*

La situation est différent pour les morphismes d'approximation. En considérant $g^{\triangle} : X \mapsto X$ *et en posant* $g^{\triangledown} = p_{\wedge}^{\#} \circ g^{\triangle^{\#}} \circ p_{\vee}$*, on a pour les singletons* $g^{\triangledown}(\{\alpha\}) = \{x \wedge y \,/\, x \vee y = \alpha\}$*, soit :*

$$g^{\triangledown}(\{\mathtt{0}\}) = \{\mathtt{0}\}$$
$$g^{\triangledown}(\{\mathtt{1}\}) = \{\mathtt{0}, \mathtt{1}\}$$

Ainsi, en utilisant les morphismes d'approximation, on arrive à exprimer des relations entre classifications, même si celles-ci ne sont pas fonctionnelles, ce qui doit être le cas pour avoir un infomorphisme.

La dernière question que nous allons étudier est la caractérisation des morphismes d'approximations g pour lesquels il existe un informorphisme f tel que $a(f) = g$.

Pour cela, par additivité, il suffit de vérifier que pour chaque élement $y \in \mathrm{Tok}_B$, on a $\mathrm{Typ}_A(f^{\vee}(y)) = \mathrm{Typ}_A(g^{\triangledown}(\{y\}))$ et de même pour les types de **A**.

Cela mène à la propriété suivante :

Proposition 4.6 *Etant donné un morphisme d'approximations g, il existe un infomorphisme f tel que $a(f) = g$ si et seulement si :*

$$\forall y, \exists x \in \mathrm{Tok}_A : \mathrm{Typ}_A(x) = \mathrm{Typ}_A\left(g^{\triangledown}(\{y\})\right)$$
$$\forall \alpha, \exists \beta \in \mathrm{Typ}_B : \mathrm{Tok}_B(\beta) = \mathrm{Tok}_B\left(g^{\vartriangle}(\{\alpha\})\right)$$

Une cas où cette condition est respectée est lorsque le cardinal des $g^{\vartriangle}(\{\alpha\})$ et des $g^{\triangledown}(\{y\})$ est égal à 1. On a enfin la proposition suivante :

Proposition 4.7 *Si deux fonctions $f^{\wedge}$ et $f^{\vee}$ vérifient :*

$$\forall y, \mathrm{Typ}_A(f^{\vee}(y)) = \mathrm{Typ}_A(g^{\triangledown}(\{y\}))$$
$$\forall \alpha, \mathrm{Tok}_B(f^{\wedge}(\alpha)) = \mathrm{Tok}_B(g^{\vartriangle}(\{\alpha\}))$$

alors $\langle f^{\wedge}, f^{\vee} \rangle$ est un infomorphisme.

5 Conclusion

Cet article n'a présenté que les outils de base de traduction entre les deux formalismes. La construction centrale de l'*information flow theory*, appelé *canal d'information* n'a pas été étudié, mais il n'y a besoin d'aucun outil supplémentaire pour le faire. De plus, cette traduction n'implique aucune nouvelle construction dans le domaine des correspondances de Galois. En fait, cela montre au contraire que l'*information flow theory* se plonge parfaitement dans ce formalisme.

Une direction de recherche pour approfondir les propriétés de la traduction présentée est une étude des implications venant de l'introduction de la notion d'approximation dans les classifications. Un premier résultat, présenté dans cet article, est le fait que cette introduction permet d'exprimer strictement plus de relations que précédemment. Un sujet d'intéret pourrait être l'étude de la modification des logiques régulières et de leurs relations, compositions, etc... du fait des approximations.

Une autre piste pourrait être l'étude des relations entre les types de deux classifications $\mathbf{A}_1$ et $\mathbf{A}_2$ ayant le même ensemble d'objets d'étude, comme cela est esquissé dans l'exemple 6. Moyennant quelques manipulations sur les approximations, on peut construire une nouvelle classification où les objets sont les types de $\mathbf{A}_1$ et les nouveaux types ceux de $\mathbf{A}_2$.

Enfin, en utilisant des techniques issues de l'interprétation abstraite, notamment les travaux de R. Giacobazzi, F. Ranzato et F. Scozzari [BRS00] sur les manipulations de domaines abstraits, on peut étudier les propriétés des modifications de classications.

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.

[CC77] P. Cousot and R. Cousot. *Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints.* In *Conference Record of the Fourth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 238–252, Los Angeles, California, 1977. ACM Press, New York, NY, USA.

[CC79] P. Cousot and R. Cousot. *Systematic design of program analysis frameworks.* In *Conference Record of the Sixth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 269–282, San Antonio, Texas, 1979. ACM Press, New York, NY, U.S.A.

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

[BRS00] R. Giacobazzi, F. Ranzato and F. Scozzari. *Making abstract interpretations complete.* A paraître dans Journal of the ACM, May 2000 issue.