

# Méthode de construction d'ontologie de termes à partir du treillis de l'iceberg de Galois

Chiraz Latiri\*, Mehdi Mtir\*\*  
Sadok Ben Yahia\*

\*Faculté des Sciences de Tunis  
Département Informatique  
Unité de recherche URPAH  
Campus Universitaire El Manar, Tunis  
Tunisie

chiraz.latiri@gnet.tn, sadok.benyahia@fst.rnu.tn

\*\*Ecole Supérieure de Commerce de Tunis  
Unité MIAGE  
Campus Universitaire La Manouba, Tunis  
Tunisie  
mehdi\_mtir@voila.fr

**Résumé.** L'approche présentée dans cet article a pour objectif la construction d'une ontologie à partir du treillis de l'iceberg de Galois. Nous entendons par *ontologie* un ensemble de termes structurés entre eux par un ensemble de liens de divers types. Dans notre cas d'étude, cette ontologie constitue un support de connaissances "documentaires". En effet, elle peut être utilisée dans diverses applications en Recherche d'Information (RI), telles que l'indexation automatique et l'expansion de requêtes ainsi qu'en text-mining. La méthode de construction que nous proposons est fondée sur l'analyse formelle de concepts (AFC) et plus précisément, la structure du treillis de l'iceberg de Galois. En utilisant cette structure hiérarchique partiellement ordonnée, nous présentons une translation directe des relations laticielles vers celles ontologiques. Nous proposons ainsi d'enrichir l'ontologie dérivée par des règles associatives génériques entre termes, découvertes dans le cadre d'un processus de text-mining.

**Mots-clés :** Ontologie, analyse formelle de concepts, treillis de Galois, règles associatives entre termes.

## 1 Introduction et contexte général

Les ontologies se sont imposées comme ressources essentielles pour la construction de Systèmes d'Information (SI) et pour la définition d'un vocabulaire commun pouvant être partagé par tous les intervenants dans un domaine spécifique. Toutefois, l'expansion de ce nouveau concept est éfrénée à cause des difficultés que nous éprouvons encore lors du processus de conception et de modélisation [Haav, 2004].

Dans la littérature, plusieurs approches ont été proposées pour essayer d'automatiser la construction des ontologies [Stumme, 2002], [Lame et al., 2000], [Haav, 2004]. Cependant, la plupart de ces recherches se sont limitées à découvrir les relations taxo-

nomiques entre les concepts et ont ainsi négligé plusieurs types de relations qui pourraient être très importants et sémantiquement valides dans certains contextes. Dans cet article, nous nous intéressons particulièrement à la construction automatique d'une ontologie de termes à partir de données textuelles, *i.e.* corpus de textes. Nous entendons par *ontologie* un ensemble de termes structurés entre eux par un ensemble de liens de divers types telles que des relations linguistiques. En effet, la méthode de construction d'ontologie proposée dans cet article comporte plusieurs phases décrites dans la figure 1.

Étant donné que nous manipulons des données textuelles brutes, il s'avère indispensable de commencer cette méthode par une phase de sélection et de prétraitement linguistique, dans le but d'extraire les termes les plus représentatifs dans les corpus analysés. Nous proposons de ne considérer que deux catégories grammaticales, à savoir les substantifs communs (SUBC) et les substantifs propres (SUBP). Ce choix se base sur le fait que la sémantique est souvent portée par les substantifs.

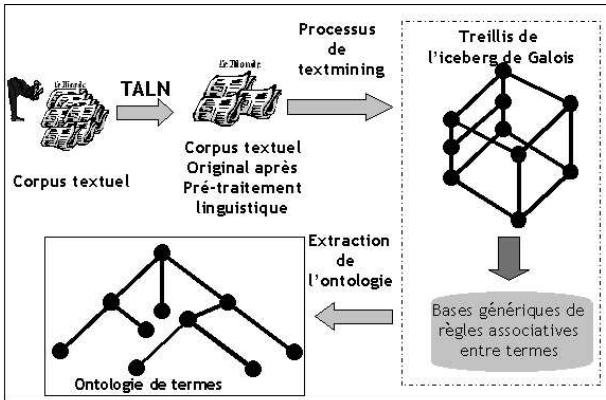


FIG. 1 – Contexte général de la méthode de construction d'ontologie

À partir de la deuxième phase de la méthode que nous proposons, nous considérons l'analyse formelle de concepts (AFC) comme fondement mathématique pour la construction de l'ontologie. Dans un premier temps, les données textuelles filtrées sont représentées sous forme d'un contexte d'extraction formel. Dans un deuxième temps, le treillis de l'iceberg de Galois ainsi que la base générique des règles associatives entre termes sont générés par les algorithmes appropriés décrits dans [Latiri et al., 2003], [Latiri, 2004].

L'idée de cet article est donc de proposer une méthode automatique pour la construction d'ontologies à partir du treillis de l'iceberg de Galois. En utilisant cette structure hiérarchique partiellement ordonnée, nous présentons une translation directe des relations laticielles vers celles ontologiques. Nous proposons ainsi d'enrichir l'ontologie dérivée, par des règles associatives génériques entre termes découvertes dans le cadre d'un processus de text-mining [Latiri, 2004].

L'article est organisé comme suit : la deuxième section donne un aperçu rapide

sur les ontologies. Dans la troisième section, nous introduisons les concepts de base de l'analyse formelle de concepts. Nous passons ensuite au niveau de la quatrième section à la description de la méthode automatique de construction d'une ontologie de termes à partir du treillis de l'iceberg de Galois, illustrée par un exemple. Les travaux en cours ainsi que quelques perspectives de l'approche sont décrits dans la dernière section.

## 2 Aperçu sur les ontologies

Plusieurs définitions ont été proposées pour donner un sens à ce qu'est une ontologie. Gruber donne les deux définitions suivantes qui sont les plus citées dans la littérature [Gruber, 1993a], [Gruber, 1993b] :

**Définition 1** *Une ontologie est une spécification formelle et explicite des termes d'un domaine ainsi que des relations que ces termes entretiennent entre-eux.*

**Définition 2** *Une ontologie est une formalisation d'une conceptualisation. C'est une description (comparable à une spécification formelle d'un programme) des concepts et des relations qui peuvent exister pour un agent ou pour une communauté d'agents.*

Nous pouvons en conclure qu'il s'agit d'un vocabulaire formalisé de termes et de relations les liant, destiné à être partagé par une communauté d'hommes ou de machines. Les ontologies ne sont donc pas des bases de connaissances, mais elles représentent un outil pour expliciter les notions utiles à la formulation des connaissances.

L'intérêt éprouvé à l'égard des ontologies croit de jour en jour. Elles ont été adoptées dans divers domaines tels que la recherche d'information (RI), le web sémantique, le commerce électronique, etc. Cette diversité est expliquée par les avantages qu'elles offrent. À titre d'exemple, nous citons [Lame et al., 2000], [Stumme, 2002] :

- Elles peuvent constituer un cadre de pensée pour modéliser un problème. C'est ainsi qu'elles sont mobilisées dans le cadre de l'ingénierie des connaissances.
- Elles constituent un cadre partagé que différents acteurs peuvent mobiliser et dans lequel ils peuvent se reconnaître, ce qui leur procure un rôle de mutualisation et de fédération de connaissances. C'est ainsi qu'elles sont fort utilisées dans la gestion des connaissances.
- Elles peuvent servir de ressources pour représenter le sens de différents contenus échangés dans des SI. D'où, leur intérêt pour le web sémantique [Bachimont, 2001].

Nous passons dans ce qui suit à l'introduction du fondement mathématique retenu pour le développement de notre méthode, à savoir l'analyse formelle de concepts (AFC) [Ganter et al., 1999].

## 3 Notions de treillis de l'iceberg de Galois et de concept formel réduit fermé

L'utilisation des treillis de Galois en tant qu'outil de hiérarchisation a démarré au début des années 80 [Godin et al., 1994], [Ganter et al., 1999]. En effet, la notion de *treillis de Galois* ou *treillis de concepts*, est à la base d'une famille de méthodes de

classification conceptuelle qui propose de considérer chaque élément du treillis comme un *concept formel*, et le graphe associé comme une relation de généralisation/spécialisation.

### 3.1 Contexte et concept formels

Un contexte formel est défini comme suit [Ganter et al., 1999] :

**Définition 3** *Un contexte formel  $K = (O, I, \mathcal{R})$  est un triplet, tels que  $O$  est un ensemble d'objets,  $I$  un ensemble de propriétés et  $\mathcal{R}$  une relation binaire de  $O$  dans  $I$ . Le couple  $(o, i) \in \mathcal{R}$  représente le fait que l'objet  $o$  possède la propriété  $i$ .*

En considérant un contexte formel  $K = (O, I, \mathcal{R})$ , le concept formel est défini comme suit [Ganter et al., 1999] :

**Définition 4** *Étant donné deux ensembles  $A$  et  $B$  tels que  $A \subset O$  et  $B \subset I$ , un concept formel  $(A, B)$  est une paire, tel que si  $A \times B \subseteq A' \times B' \subseteq \mathcal{R}$  alors  $A = A'$  et  $B = B'$ . Les ensembles  $A$  et  $B$  sont appelés, respectivement, l'extension (domaine) et l'intention (co-domaine) du concept  $(A, B)$ .*

L'ensemble de tous les concepts formels du contexte formel  $K = (O, I, \mathcal{R})$  est noté par  $E\mathcal{CF}_K$ . Une relation d'ordre, notée par  $\leq$ , est définie sur cet ensemble comme suit :  $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow B_2 \subseteq B_1$

### 3.2 Connexion de Galois

Étant donné les applications  $f$  et  $h$  suivantes :

$$\begin{aligned} f & : P(O) \longrightarrow P(I) \\ f(A) & = \{i \in I \mid \forall o, o \in A \implies (o, i) \in \mathcal{R}\} \end{aligned}$$

$$\begin{aligned} h & : P(I) \longrightarrow P(O) \\ h(B) & = \{o \in O \mid \forall i, i \in B \implies (o, i) \in \mathcal{R}\} \end{aligned}$$

Le couple d'applications  $(f, h)$  définit une *connexion de Galois* entre l'ensemble des parties de  $O$  et l'ensemble des parties de  $I$  [Ganter et al., 1999]. Pour un ensemble  $A \subseteq O$  et un ensemble  $B \subseteq I$ , les opérateurs  $f \circ h$  et  $h \circ f^{-1}$  sont appelés les *opérateurs de fermeture de la connexion de Galois* [Ganter et al., 1999].

---

<sup>1</sup> $h \circ f(A) = h(f(A))$  et  $f \circ h(B) = f(h(B))$

### 3.3 Treillis de Galois

Soit  $E\mathcal{CF}_K$  l'ensemble de concepts formels dérivés à partir du contexte formel  $\mathcal{K} = (O, I, \mathcal{R})$ , en utilisant l'opérateur de fermeture de la connexion de Galois  $f \circ h$ . La paire  $\mathcal{L}_c = (E\mathcal{CF}_K, \leq)$  est appelée *treillis de Galois* ou *treillis de concepts*, et vérifie les deux propriétés suivantes [Ganter et al., 1999] :

**Propriété 1** *Un ordre partiel est défini sur les éléments du treillis de Galois  $\mathcal{L}_c$ , tel que pour tout concept formel  $c_1, c_2 \in \mathcal{L}_c$ ,  $c_1 \leq c_2$  si et seulement si  $c_1 \subseteq c_2$ . Il existe une relation duale entre les concepts  $c_i$  et leurs extensions  $h(c_i)$  dans le treillis de Galois, i.e.  $c_1 \subseteq c_2 \Leftrightarrow h(c_2) \subseteq h(c_1)$ , et par conséquent,  $c_1 \leq c_2 \Leftrightarrow h(c_2) \subseteq h(c_1)$ .*

Cet ordre partiel est utilisé pour générer le graphe du treillis de la manière suivante : il existe un arc  $(c_1, c_2)$  noté par  $(c_1 \prec c_2)$ , si  $c_1 \leq c_2$ , et il n'existe pas d'élément  $c_3$ , tel que  $c_1 \leq c_3 \leq c_2$ . Cette relation de *couverture* désigne que  $c_1$  est un *parent* de  $c_2$  et que  $c_2$  est un *fil* de  $c_1$ .

**Propriété 2** *Tous les sous-ensembles de  $\mathcal{L}_c$  possèdent une borne inférieure appelée l'élément Meet, i.e. le plus grand minorant (GLB) de  $\mathcal{L}_c$ , et une borne supérieure appelée l'élément Join, i.e. le plus petit majorant (LUB) de  $\mathcal{L}_c$ . Pour tout  $S \subseteq \mathcal{L}_c$ , nous avons :*

$$\begin{aligned} \text{Join}(S) &= f \circ h\left(\bigcup_{c \in S} c\right) \\ \text{Meet}(S) &= \bigcap_{c \in S} c \end{aligned}$$

Étant donné que nous nous intéressons à la construction d'ontologie initialement à partir d'un corpus textuel, nous donnons la définition suivante [Latiri et al., 2003] :

**Définition 5** *Un contexte d'extraction textuel est un triplet  $\mathcal{T} = (D, T, \mathcal{R})$  décrivant un ensemble fini  $D$  de documents, un ensemble fini  $T$  de termes et une relation binaire  $\mathcal{R}$  ( $\mathcal{R} \subseteq D \times T$ ). Chaque couple  $(d, t) \in \mathcal{R}$  signifie que le document  $d \in D$  contient le terme  $t \in T$ .*

### 3.4 Treillis de l'iceberg de Galois

Dans des travaux antérieurs relatifs au text-mining, nous avons défini de nouvelles approches d'extraction de concepts réduits fréquents et de dérivation de règles associatives entre termes. Nous rappelons ainsi la définition de *concept réduit* [Latiri et al., 2003].

**Définition 6** *Soit  $T_k$  un  $k$ -termset<sup>2</sup>. Il est appelé *Concept Réduit (CR)*, si et seulement si il est égal à sa fermeture, i.e.  $T_k = f \circ h(T_k)$ . L'extension de  $T_k$  est donnée par  $h(T_k)$  [Latiri et al., 2003].*

<sup>2</sup>Un ensemble de termes  $T_k \subseteq T$  ( $k$  est le nombre de termes de  $T_k$ ) est appelé un  $k$ -termset.

	Informatique	Réseau	Langages	Matériel	Logiciels	Bureautique
D1	1	1				
D2	1	1	1	1		
D3	1		1			
D4	1				1	1
D5	1				1	
D6	1			1		

TAB. 1 – Contexte formel d’extraction textuel relatif au domaine *Informatique*.

L’ensemble de tous les concepts réduits fréquents, noté par  $ECRF$ , constitue un semi-treillis de Galois, appelé *treillis de l’iceberg de Galois* et défini comme suit :

**Définition 7** *En considérant que les concepts réduits fréquents ( $ECRF$ ) et l’inclusion ensembliste pour organiser les termsets fermés, nous obtenons une structure de treillis, notée par  $\hat{\mathcal{L}} = (ECRF, \subseteq)$ , qui préserve uniquement les plus grands minorants (GLBs), i.e. l’opérateur Meet (voir section 3.3). Cette structure définit un semi-treillis, appelé treillis de l’iceberg de Galois [Stumme et al., 2002].*

En utilisant le fondement mathématique présenté ci-dessus, nous introduisons dans la section qui suit une nouvelle méthode de construction d’une ontologie de termes à partir d’un contexte d’extraction textuel et du treillis de l’iceberg de Galois qui lui correspond.

## 4 Méthode automatique de construction d’une ontologie à partir de l’iceberg de Galois

### 4.1 Principe de la méthode

La méthode proposée commence par nettoyer le corpus textuel par une phase de pré-traitement linguistique. Nous appliquons par la suite un processus de text-mining [Latiri et al., 2003], [Latiri, 2004] qui nous fournit comme résultat une structure de concepts ordonnés, à savoir le treillis de l’iceberg de Galois, noté par  $(\hat{\mathcal{L}})$ , et enrichi par les générateurs minimaux.

À partir de cette structure, les concepts réduits fermés sont extraits en associant à chaque concept son générateur et l’ensemble de ses successeurs. Ensuite, nous exprimons les relations ontologiques existantes entre ces concepts pour pourvoir, dans l’étape suivante construire l’ontologie de termes. Enfin, nous enrichissons cette ontologie par des règles associatives génériques entre termes découvertes dans le cadre d’un processus de text-mining.

Ces différentes étapes seront développées dans les sections suivantes et illustrées par un exemple extrait du domaine informatique.

**Exemple 1** *Nous considérons le contexte d’extraction formel figurant dans le tableau 1. Le treillis de l’iceberg de Galois relatif au contexte formel d’extraction décrit dans le*

tableau 1 et dérivé par l'algorithme ICE-HASSE [Latiri et al., 2003], [Latiri, 2004] est représenté dans la figure 2.

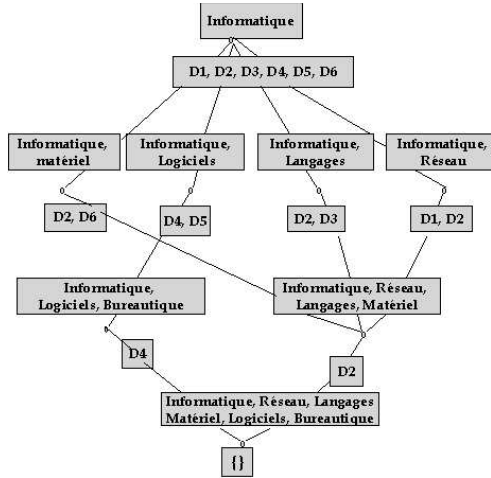


FIG. 2 – Treillis de l'iceberg de Galois

## 4.2 Extraction des concepts formels réduits fermés

Nous allons nous baser dans cette étape sur l'algorithme ICE-HASSE proposé dans [Latiri, 2004] qui permet d'aboutir, à partir d'un ensemble de concepts réduits fréquents ( $ECRF$ ), au treillis de l'iceberg de Galois enrichi par les générateurs minimaux. Cet algorithme nous permet d'avoir pour chaque concept, la liste des générateurs de ce concept ainsi que l'ensemble de ses prédécesseurs dans le treillis que nous notons par ( $cov^{inf}(c_i)$ ). Toutefois, dans notre approche nous avons aussi besoin des successeurs du concept. Pour ce, nous proposons un algorithme simpliste RECH-SUCC qui permet de trouver les successeurs d'un concept à partir de la couverture inférieure ( $cov^{inf}(c_i)$ ) (voir algorithme 1).

### Algorithme RECH-SUCC

**Entrée:** La liste des prédécesseurs de chaque concept ( $cov^{inf}(c_i)$ ).

**Sortie:** La liste des successeurs de chaque concept ( $cov^{sup}(c_i)$ ).

**Pour tout** tout concept  $c_i$  **Faire**

$cov^{sup}(c_i) \leftarrow \emptyset$

**Pour tout** tout concept  $c_i$  **Faire**

**Pour tout**  $c_j \in cov^{inf}(c_i)$  **Faire**

$cov^{sup}(c_j) \leftarrow cov^{sup}(c_j) \cup c_i$

**Algorithme 1:** L'algorithme RECH-SUCC pour trouver les successeurs d'un concept

Concept ( $c_i$ )	Générateur	prédécesseurs ( $cov^{inf}(c_i)$ )	successeurs ( $cov^{sup}(c_i)$ )
Informatique	Informatique		Réseau, Langages Matériel, Logiciels
Informatique, Réseau	Réseau	Informatique	
Informatique, Langages	Langages	Informatique	
Informatique, Matériel	Matériel	Informatique	
Informatique, Logiciels	Logiciels	Informatique	Bureautique
Informatique, Logiciels, Bureautique	Bureautique	Logiciels	

TAB. 2 – Les résultats respectifs des algorithmes ICE-HASSE et RECH-SUCC sur le contexte d'extraction illustré par le tableau 1.

Concept <i>père</i>	Concept <i>fil</i>
Informatique	Matériel, Logiciels, Langages, Réseau
Bureautique	Logiciels

TAB. 3 – Exemple de relations ontologiques Père/Fils

Les résultats respectifs des algorithmes ICE-HASSE et RECH-SUCC sur le contexte d'extraction illustré par le tableau 1, sont donnés dans le tableau 2.

### 4.3 Expression des relations ontologiques

Dans cette section, nous allons mettre l'accent sur les principales relations ontologiques que nous pouvons dériver à partir des relations laticielles [Bachimont, 2001], [Haav, 2004]. Nous retenons les relations suivantes :

#### 4.3.1 La relation Père/Fils :

En se basant sur la structure partiellement ordonnée du treillis de l'iceberg de Galois, nous énonçons la proposition suivante qui exprime la relation ontologique *Père/Fils*.

**Proposition 1** *Un concept  $C_1$  est le père d'un concept  $C_2$  (fil) si et seulement si  $C_2$  est un successeur immédiat de  $C_1$  ou encore si  $C_1$  est un prédécesseur direct de  $C_2$ .*

L'idée est d'associer à chaque concept  $c_i$  l'ensemble de ses successeurs, i.e.  $cov^{sup}(c_i)$  (voir tableau 2). Ainsi, nous pouvons dire que tout concept  $c_j \in cov^{sup}(c_i)$  est un concept fils du concept  $c_i$ . De même, tout concept  $c_k \in cov^{inf}(c_i)$  est un concept père du concept  $c_i$ .

Par rapport à l'exemple illustratif que nous avons choisi (voir l'exemple 1), nous retrouvons les relations *Père/Fils* décrites dans le tableau 3.

Par rapport à l'exemple illustratif que nous avons choisi (voir l'exemple 1), nous retrouvons les relations *Père/Fils* décrites dans le tableau 3.



### 4.3.2 La relation d'héritage :

La relation ontologique *d'héritage* est une conséquence directe de la relation *Père/Fils* puisqu'un concept hérite de tous ses pères. Ces derniers héritent également chacun de ses pères et ainsi de suite. Nous énonçons ainsi la proposition suivante :

**Proposition 2** *Étant donné que la relation d'héritage est transitive, i.e. si  $c_1$  hérite de  $c_2$  et  $c_2$  hérite de  $c_3$  alors  $c_1$  hérite de  $c_3$ , alors tout concept héritera implicitement de tous ses ancêtres.*

Par rapport à l'exemple 1, nous retrouvons les relations *d'héritage* suivantes :

- *Bureautique* est fils de *Logiciels* alors *Bureautique* hérite de *Logiciels*.
- *Logiciels* est fils d'*Informatique* alors *Logiciels* hérite d'*Informatique*.

Des deux relations précédentes, nous pouvons déduire que *Bureautique* hérite d'*Informatique*.

**Remarque 1** *La relation ontologique d'héritage multiple est possible dans le cas où un concept fermé admet plus qu'un seul père, i.e.  $\|cov^{inf}(c_i)\| > 1$ . Dans notre exemple (voir exemple 2), nous n'avons pas de cas d'héritage multiple car chaque concept a au plus un seul père, i.e. pour tout  $c_i$ , nous avons  $\|cov^{inf}(c_i)\| \leq 1$ .*

### 4.3.3 La relation Frère :

La relation *Frère* est aussi une conséquence directe de la relation *Père/Fils*. Nous énonçons la proposition suivante :

**Proposition 3** *Un concept  $c_1$  est dit frère d'un concept  $c_2$  si et seulement si  $c_1$  et  $c_2$  ont au moins un père commun. Autrement dit, il faut tester si  $cov^{inf}(c_1) \cap cov^{inf}(c_2) \neq \emptyset$ .*

Par rapport à l'exemple 1, nous pouvons constater que les concepts *Réseau*, *Langages*, *Matériel* et *Logiciels* sont des frères car le concept *Informatique* est leur père commun.

**Remarque 2** *Entre deux concepts frères, il n'existe pas de relation d'ordre.*

## 4.4 Construction de l'ontologie de termes

Une fois les relations ontologiques sont dérivées à partir des relations laticielles, il devient possible de construire l'ontologie de termes. Nous proposons l'algorithme CMS-ONTOLOGY qui est basé sur la relation *Père/Fils* définie précédemment.

L'algorithme CMS-ONTOLOGY commence par trier l'ensemble des concepts réduits fermés ( $E_{\mathcal{CRF}}$ ) dans un ordre décroissant, puis insère le générateur minimal du premier concept dans l'ontologie en cours de construction. La complétude est par la suite effectuée d'une manière descendante, en partant du nœud relatif à la borne supérieure pour arriver aux nœuds les plus bas. Dans chaque itération, nous traitons tous les concepts  *fils* relatifs au concept courant, s'ils existent.

Pour chaque fils, nous vérifions s'il n'est pas déjà inséré dans la structure de l'ontologie (cas d'un héritage multiple). Dans ce cas de figure, nous commençons par insérer son générateur minimal. Ensuite, nous établissons un lien entre le concept père et son fils. Les différentes itérations de la construction de l'ontologie sont décrites dans l'algorithme 2.

**Algorithme** CMS-ONTOLOGY  
**Entrée:**  $E\_CRF = \{c_1, c_2, \dots, c_l\}$ ,  $(cov^{sup}(c_i); i = 1..l)$  et  $(c_i.gen; i = 1..l)$ .  
**Sortie:** L'ontologie( $\hat{O}$ )  
 Trier-Dec ( $E\_CRF$ )  
 $\hat{O} \leftarrow \{c_1.gen\}$   
**Pour**  $i$  **Allant de** 1 **à**  $l$  **Faire**  
   **Si**  $cov^{sup}(c_i) \neq \emptyset$  **alors**  
     **Pour tout**  $c_j \in cov^{sup}(c_i)$  **Faire**  
       **Si**  $(c_j.gen) \notin \hat{O}$  **alors**  
          $\hat{O} \leftarrow \hat{O} \cup \{c_j.gen\}$   
         Etabir - Lien - Père/Fils( $c_i, c_j$ )

**Algorithme 2:** L'algorithme CMS-ONTOLOGY pour la construction de l'ontologie de termes

À titre d'exemple, l'ontologie construite à partir du treillis de l'iceberg de Galois (voir figure 2) et relative à notre exemple (voir exemple 1) est illustrée dans la figure 3.

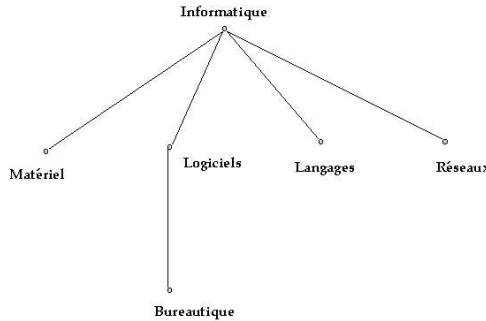


FIG. 3 – L'ontologie de termes dérivée à partir du treillis de l'iceberg de Galois

### 4.5 Extension de l'ontologie par les règles associatives entre termes

Afin d'enrichir les relations ontologiques, nous proposons d'utiliser une approche d'extraction de règles associatives non redondantes, qui explore le treillis de l'iceberg de Galois enrichi par les générateurs minimaux [Latiri et al., 2003], [Latiri, 2004]. Nous

distinguons deux types de règles associatives, à savoir les règles associatives *exactes*, dont la confiance est égale à 1, et les règles associatives *approximatives*, dont la confiance est inférieure à 1 [Latiri et al., 2003]. Nous faisons une distinction entre les règles associatives exactes et approximatives car elles possèdent des propriétés différentes par rapport aux concepts réduits fréquents. Ces propriétés permettent d'identifier les règles redondantes ainsi que celles qui sont non significatives, i.e. dont la confiance est faible, parmi toutes les règles associatives valides.

Nous avons défini les règles associatives approximatives et exactes en considérant les générateurs minimaux des concepts réduits fréquents comme suit [Latiri et al., 2003] :

**Définition 8** Une règle associative approximative est une implication entre deux concepts réduits fréquents  $c_i$  et  $c_j$  du treillis de l'iceberg de Galois enrichi, de la forme  $g_i \Rightarrow (f \circ h(c_j.gen) - g_i)$ , avec  $c_i \leq c_j$  et  $g_i \in c_i.gen$ .

Étant donné que les règles approximatives entre termes représentent les implications des sous-générateurs vers les super-générateurs dans le treillis de l'iceberg de Galois, nous pouvons donc exploiter ces associations pour illustrer les relations ontologiques de type *Related-to*.

**Exemple 2** Par rapport au contexte d'extraction illustré dans l'exemple 1, nous donnons dans cet exemple un extrait de quelques règles approximatives qui traduisent la relation linguistique *Related-to* dans l'ontologie pour un *minconf* de 2/6 et un *minsupp* de 1/6.

- Informatique, Matériel  $\longrightarrow$  Réseau, Langages.
- Informatique, Logiciels  $\longrightarrow$  Bureautique.
- Informatique, Langage  $\longrightarrow$  Réseau, Matériel.
- Informatique, Réseau  $\longrightarrow$  Langages, Matériel.

**Définition 9** Étant donné un générateur minimal  $g_i$  associé au concept réduit fréquent  $c_i$  ayant une couverture inférieure vide, i.e. un concept réduit fréquent  $c_j$  tel que  $c_i \prec c_j$ , alors une règle associative exacte est une implication de la forme :  $c_i.gen \Rightarrow (f \circ h(c_i.gen) - c_i.gen)$ .

Nous considérons ainsi que les règles exactes illustrent les relations ontologiques de type *Part-of*. Ce constat est validé par une série d'expérimentations détaillées dans [Latiri, 2004] et qui a montré que les règles associatives exactes traduisent souvent les termes composés ou les multi-termes.

**Exemple 3** Par rapport au contexte d'extraction illustré dans l'exemple 1, nous donnons dans cet exemple un extrait de quelques règles exactes qui traduisent la relation linguistique *Part-of* dans l'ontologie.

- Logiciels  $\longrightarrow$  Informatique.
- Réseau, Langages, Matériel  $\longrightarrow$  Informatique.
- Bureautique  $\longrightarrow$  Informatique, Logiciels.

Nous remarquons que les termes "logiciels informatiques" et "langages informatiques" forment bien des termes composés qu'il est utile de trouver dans une ontologie de termes.

## 5 Conclusion

Dans cet article, nous avons proposé une nouvelle méthode automatique de construction d'ontologie de termes à partir de la structure partiellement ordonnée du treillis de l'iceberg de Galois. Nous avons également enrichi cette ontologie par les règles associatives entre termes exactes et approximatives qui traduisent les relations linguistiques de type part-of et related-to. L'implémentation de la méthode et les expérimentations sur les corpus OFIL et INIST du projet Amaryllis sont en cours de développement. Comme perspective de ce travail, nous envisageons d'exploiter le poids des termes dans les corpus qui permettent d'avoir un contexte d'extraction textuel flou et une ontologie pondérée de termes.

## Références

- [Bachimont, 2001] Bachimont, B. 2001, *Modélisation linguistique et modélisation logique des ontologies : l'apport de l'ontologie formelle*. Pages 349–368 of : *Proceedings IC 2001, Plateforme AFIA, Grenoble* .
- [Ganter et al., 1999] Ganter, B., & Wille, R. 1999, *Formal Concept Analysis*. Edition Springer-Verlag, Heidelberg.
- [Godin et al., 1994] Godin, R., & Missaoui, R. 1994. An incremental concept formation approach for learning from databases. *Theoretical Computer Science*, (133), 387–419.
- [Gruber, 1993a] Gruber, T. R. 1993, A translation approach to portable ontologies. *Journal of Knowledge Acquisition*, 2(5), 199–220.
- [Gruber, 1993b] Gruber, T. R. 1993, *Toward principles for the design of ontologies used for knowledge sharing*. *Proceedings of the Padua workshop on Formal Ontology, March 1993*.
- [Haav, 2004] Haav, H. 2004, A semi-automatic method to ontology design by using FCA. Pages 13–24 of : *Proceedings of the second International Workshop Concept Lattices and Their Applications (CLA 2004)*, VSB-TU, September, Ostrava).
- [Lame et al., 2000] Lame, G., & Bourigault, D. 2000, Analyse distributionnelle et structuration de terminologie. Application à la construction d'une ontologie documentaire du droit. *Journal TAL*.
- [Latiri, 2004] Latiri, C. Ch. 2004, *Approche de Découverte de Règles Associatives Classiques et Floues à partir de Textes : Application à la Recherche d'Information*. Thèse de Doctorat en Informatique. Ecole Nationale des Sciences de l'Informatique, Université de la Manouba.
- [Latiri et al., 2003] Latiri, C. Ch., Benyahia, S., Mineau, G., & Jaoua, A. 2003, Découverte des règles associatives non redondantes : Application aux corpus textuels. Pages 131–144 des : *Actes des Journées francophones EGC'2003, Lyon*. Publié dans la revue *d'Intelligence Artificielle*, Vol 17, No. 1, 2003.
- [Stumme, 2002] Stumme, G. 2002, Using Ontologies and Formal Concept Analysis for Organizing Business Knowledge. *Editors : J. Becker, R. Knackstedt. Wissensmanagement mit Referenzmodellen* Physica, Heidelberg.
- [Stumme et al., 2002] Stumme, G. Taouil, R., Bastide, Y., Pasquier, N., & Lakhal, L. 2002, Computing iceberg concept lattices with Titanic. *IEEE Transactions on Knowledge and Data Engineering*, 2(42), 189–222.