Correction et complétude d'un algorithme de recherche d'information par treillis de concepts

Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, Malika Smaïl-Tabbone

UMR 7503 LORIA, BP 239, 54506 Vandœuvre-lès-Nancy, FRANCE
{messai,devignes,napoli,malika}@loria.fr
http://www.loria.fr/∼messai

Résumé. Dans cet article nous présentons BR-Explorer, un algorithme de recherche d'information correct et complet qui s'appuie sur la classification par treillis de concepts. L'algorithme BR-Explorer a pour objectif la recherche d'objets pertinents pour une requête donnée. Initialement, nous disposons d'un contexte formel représentant la relation entre un ensemble d'objets décrits par un ensemble d'attributs et du treillis de concepts correspondant à ce contexte. Étant donné une requête, l'algorithme BR-Explorer commence par générer un concept formel représentant la requête puis l'insère dans le treillis de concepts. Ensuite, BR-Explorer localise, dans le treillis résultant, un "concept pivot" à partir duquel il construit la réponse à la requête étape par étape en effectuant un parcours en largeur des concepts subsumants le concept pivot jusqu'au concept le plus général du treillis. Finalement BR-Explorer retourne en réponse l'ensemble des objets pertinents ordonnés selon leur degré de pertinence pour la requête.

1 Introduction

La notion de treillis de Galois (ou treillis de concepts) d'une relation est à la base d'une famille de méthodes de classification conceptuelle (Godin et al., 1995b). Introduite par Barbut et Monjardet (1970), cette approche a été à la base de l'analyse formelle de concepts (Ganter et Wille, 1999) où on propose de considérer chaque élément du treillis comme un concept formel et le graphe (diagramme de Hasse) comme une relation de généralisation/spécialisation entre les concepts. Cette forme de classification constitue l'une des motivations de l'application des treillis de concepts pour la recherche d'information dès l'apparition de l'analyse formelle de concepts. La recherche d'information a par la suite été explicitement mentionnée comme étant l'une des applications possibles des treillis de concepts (Godin et al., 1995b). Plus tard, une approche pour la recherche d'information en utilisant les treillis de concepts, la recherche d'information par treillis, a été proposée dans (Carpineto et Romano, 2000, 2004). La motivation principale qui a conduit à l'utilisation des treillis de concepts pour la recherche d'information vient du fait que la classification en treillis permet de combiner la recherche par requête et la recherche par navigation (Godin et al., 1995a). En effet, les concepts formels qui s'organisent en un treillis de concepts (voir les définition dans la section 2) sont vus comme des classes de documents pertinents pour un ensemble de contraintes données (représenté par une requête).

La relation de subsomption (la relation d'ordre partiel entre les concepts du treillis) entre les concepts permet le passage d'un concept à un autre concept plus général ou plus spécifique offrant ainsi la possibilité de naviguer dans la hiérarchie considérée.

L'utilisation des treillis de concepts dans les systèmes de recherche d'information spécialisés favorise un meilleur rappel et une meilleure précision. En effet, le domaine de recherche (corpus) est souvent plus restreint, ce qui permet d'une part de réduire les problèmes de passage à l'échelle notamment pour l'aspect navigation et d'autre part d'améliorer les performances du processus de recherche du fait de la non nécessité d'indexation automatique et de l'assise formelle de la FCA

Une méthode de recherche d'information par treillis de concepts a été introduite dans (Messai et al., 2005) et appliquée pour la recherche de sources de données biologiques dans l'annuaire BioRegistry (Smail-Tabbone et al., 2005; Messai et al., 2006). Dans cet article nous développons la présentation formelle de l'algorithme BR-Explorer qui constitue la base de la méthode de recherche proposée dans (Messai et al., 2005) et nous donnons les preuves de correction de complétude de cet algorithme.

Le reste de cet article est organisé comme suit. Nous commençons en section 2 par un bref rappel des notions et définitions de base relatives à l'analyse formelle de concepts. Ensuite nous détaillons, dans la section 3, l'algorithme de recherche d'information par treillis, BR-Explorer. Dans la section 4, nous illustrons l'application de l'algorithme BR-Explorer sur un exemple simple. Dans la section 5, nous donnons les preuves de correction et complétude de BR-Explorer par rapport au critère de pertinence défini. Nous concluons et donnons les perspectives de ce travail en section 6.

2 Rappel des notions de base de l'analyse formelle de concepts

Dans cette section nous rappelons brièvement les définitions et les résultats concernant l'analyse formelle de concepts nécessaires pour la suite de cet article (voir aussi (Ganter et Wille, 1999)).

Définition 1 (Contexte formel) *Un contexte formel est un triplet $\mathbb{K} = (G, M, I)$ où G est un ensemble d'objets, M est un ensemble d'attributs et I est une relation binaire entre G et M appelée relation d'incidence de $\mathbb{K}$ et vérifiant :*

$I \subseteq G \times M$ et $(g, m) \in I$ (notée aussi gIm) signifie que l'objet $g \in G$ possède l'attribut $m \in M$.

Définition 2 *Soit $\mathbb{K} = (G, M, I)$ un contexte formel. Pour tout $A \subseteq G$ et $B \subseteq M$, on définit :*

- $A' = \{m \in M \mid \forall\, g \in A,\ gIm\}$
- $B' = \{g \in G \mid \forall\, m \in B,\ gIm\}$

Intuitivement, A' est l'ensemble des attributs communs à tous les objets de A et B' est l'ensemble des objets possédant tous les attributs de B. L'opérateur $'$ est appelé opérateur de dérivation et s'applique aussi bien aux sous-ensembles de G qu'aux sous-ensembles de M. Cet opérateur peut se composer avec lui même, pour partir d'un sous-ensemble d'objets A, produire A' et à partir de A' produire le sous-ensemble d'objets A'' (la notation $''$ est utilisée pour marquer la composition). Les opérateurs $'$ et $''$ vérifient les propriétés suivantes pour A, $A1$, $A2$ des sous-ensembles de G et B, $B1$, $B2$ des sous-ensembles de M :

– $A1 \subseteq A2 \Rightarrow A2' \subseteq A1'$, $B1 \subseteq B2 \Rightarrow B2' \subseteq B1'$
– $A \subseteq A''$ et $A' = A'''$, $B \subseteq B''$ et $B' = B'''$ ($'''$ est la composition de $'$ et $''$)
– $A \subseteq B' \Leftrightarrow B \subseteq A'$

L'opérateur composé $''$ définit une fermeture sur l'ensemble des parties de G, noté $\wp(G)$, ou sur l'ensemble des parties de M, noté $\wp(M)$. Rappelons qu'une fermeture h sur un ensemble partiellement ordonné $(E, \leq)$ est extensive ($\forall\ x \in E$, $h(x) \leq x$), monotone croissante ($\forall\ x$, $y \in E$, $x \leq y$ alors $h(x) \leq h(y)$) et idempotente ($\forall\ x \in E$, $h(h(x)) \leq h(x)$). Un élément $x \in E$ est dit fermé pour h si et seulement si $x = h(x)$.

Définition 3 (Concept formel) *Soit $\mathbb{K} = (G, M, I)$ un contexte formel. Un concept formel est un couple (A, B) tel que $A \subseteq G$, $B \subseteq M$, $A' = B$ et $B' = A$. A et B sont respectivement appelés $extension$ (extent) et $intension$ (intent) du concept formel (A, B). On note $\mathfrak{B}(G, M, I)$ l'ensemble des concepts formels associés au contexte formel $\mathbb{K} = (G, M, I)$.*

Un sous-ensemble B de M est l'intension d'un concept formel dans $\mathfrak{B}(G, M, I)$ si et seulement si $B'' = B$ (B est fermé pour l'opérateur $''$) et, de façon duale, un sous-ensemble A de G est l'extension d'un concept formel dans $\mathfrak{B}(G, M, I)$ si et seulement si $A'' = A$ (A est fermé pour l'opérateur $''$). Les concepts de $\mathfrak{B}(G, M, I)$ sont ordonnés par une relation de $subsomption$ entre concepts (notée $\sqsubseteq$) qui se définit par : $(A1, B1) \sqsubseteq (A2, B2)$ si et seulement si $A1 \subseteq A2$ (ou de façon duale $B2 \subseteq B1$), $(A1, B1)$ et $(A2, B2)$ étant deux concepts formels de $\mathfrak{B}(G, M, I)$. $(A2, B2)$ est dit $subsumant$ de $(A1, B1)$. Cette relation de subsomption permet d'organiser les concepts formels en un treillis complet, $(\mathfrak{B}(G, M, I), \sqsubseteq)$, appelé *treillis de concepts* ou encore *treillis de Galois* (Barbut et Monjardet, 1970), qui est noté $\underline{\mathfrak{B}}(G, M, I)$. Rappelons pour finir qu'un *treillis* est un ensemble partiellement ordonné $(E, \leq)$ tel que tout couple d'éléments (x, y) dans $E \times E$ admet une *borne inférieure* (ou *infimum*) notée $x \vee y$ et une *borne supérieure* (ou *supremum*) notée $x \wedge y$. Le treillis est complet si toute partie S de E admet une borne inférieure notée $\bigwedge S$ et une borne supérieure notée $\bigvee S$. En particulier, un treillis complet admet un *élément minimal* (*bottom*) noté $\perp$ et un *élément maximal* (*top*) noté $\top$.

3 Formalisation de l'algorithme BR-Explorer

Dans cette section nous formalisons l'algorithme BR-Explorer et nous définissons les notions que nous avons utilisé lors de cette formalisation.

3.1 Définitions

Définition 4 (Requête) *Une requête Q est un couple ($\{x\}$,$\{x\}'$) où x est un objet virtuel et $\{x\}'$ est un ensemble d'attributs décrivant les objets à chercher.*

Dans la définition 4, nous supposons que l'objet virtuel x possède tous les attributs dans $\{x\}'$ ($\{x\}'$ est l'image de $\{x\}$ par l'opérateur de dérivation $'$ défini dans la section 2).

Comme dans la majorité des algorithmes de recherche d'information par treillis (Godin et al., 1995a; Carpineto et Romano, 2000), BR-Explorer effectue la recherche d'objets pertinents en insérant la requête Q = ($\{x\}$,$\{x\}'$) dans le treillis de concept du contexte formel considéré. Une telle insertion peut être considérée comme l'ajout d'une nouvelle entrée (un

nouvel objet et ses attributs) dans le contexte formel considéré. De cette façon, deux alternatives sont possibles pour la construction du nouveau treillis de concepts : soit la reconstruction à partir de zéro soit l'extension de façon incrémentale à partir du treillis déjà construit en utilisant des algorithmes de construction incrémentale de treillis de concept comme l'algorithme de Godin et al. (1995b) et l'algorithme *AddIntent* de Merwe et al. (2004). Dans ce travail nous avons choisi d'utiliser la deuxième solution qui est plus efficace que la première.

Définition 5 ($\oplus$) *Pour un contexte formel $\mathbb{K} = (G, M, I)$ et une requête $Q = (\{x\}, \{x\}')$, nous définissons l'opérateur d'addition $\oplus$ comme suit :*

$$\begin{aligned} \mathbb{K}_Q &= \mathbb{K} \oplus Q \\ &= (G, M, I) \oplus (\{x\}, \{x\}') \\ &= (G \cup \{x\}, M \cup \{x\}', I_Q) \\ &= (G_Q, M_Q, I_Q) \end{aligned}$$

Définition 6 (Pivot) *Considérons $\mathbb{K} = (G, M, I)$ un contexte formel et $Q = (\{x\}, \{x\}')$ une requête. Le concept pivot dans le treillis de concept $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ du contexte formel $\mathbb{K}_Q$ est le concept formel $P = (\{x\}'', \{x\}')$.*

Considérons l'ensemble de concepts formels $\mathfrak{B}(G, M, I)$ du contexte formel $\mathbb{K} = (G, M, I)$ et considérons la requête Q= ($\{x\}$,$\{x\}'$). Selon la relation entre l'ensemble $\{x\}'$ et les intensions des concepts formels dans $\mathfrak{B}(G, M, I)$, nous distinguons les cas suivants pour le concept pivot P = ($\{x\}''$,$\{x\}'$) dans le treillis de concepts $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$:

- S'il n'existe pas de concept $C = (A, B) \in \mathfrak{B}(G, M, I)$ tel que $\{x\}' \subseteq B$ alors $\{x\}$ et $\{x\}'$ sont fermés dans G_Q et M_Q respectivement et $\{x\}''$ = $\{x\}$. Cela signifie que l'ajout de la nouvelle entrée dans le contexte formel $\mathbb{K} \oplus Q$ produit au moins un nouveau concept formel ($\{x\}$,$\{x\}'$) = ($\{x\}''$,$\{x\}'$). De plus, chaque concept formel $C_1 = (A_1, B_1)$ vérifiant $B_1 \subseteq \{x\}'$ est transformé en $C_2 = (A_1 \cup \{x\}, B_1)$ dans $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$.
- S'il existe un concept $C = (A, B) \in \mathfrak{B}(G, M, I)$ tel que $\{x\}' \subseteq B$ alors nous distinguons deux cas :
  - Si $\{x\}' \subset B$ alors $\{x\}'' = A \cup \{x\}$ et le concept pivot est P = ($A \cup \{x\}$,$\{x\}'$). Cela signifie que l'objet virtuel x (la nouvelle entrée dans $\mathbb{K} \oplus Q$) est fusionnée avec d'autres objets possédant les mêmes attributs. Comme dans le cas précédent, l'opération $\mathbb{K} \oplus Q$ produit la création d'au moins un nouveau concept formel dans $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ à savoir le concept pivot P = ($A \cup \{x\}$,$\{x\}'$). De plus, chaque concept formel $C_1 = (A_1, B_1)$ vérifiant $B_1 \subseteq \{x\}'$ est transformé en $C_2 = (A_1 \cup \{x\}, B_1)$ dans $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$.
  - Si $\{x\}'$ = B alors $\{x\}'' = A \cup \{x\}$ et le concept pivot est P = ($A \cup \{x\}$,$\{x\}'$) = ($A \cup \{x\}$,B). Toutefois, contrairement aux deux cas précédents, l'opération $\mathbb{K} \oplus Q$ ne produit aucun nouveau concept formel dans $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$. Elle transforme simplement les concepts formels de la forme $C_1 = (A_1, B_1)$ tels que $B_1 \subseteq \{x\}'$ en C_2 = ($A_1 \cup \{x\}$,B_1).

Définition 7 (couverture supérieure) *(1) Considérons un contexte formel $\mathbb{K} = (G, M, I)$, l'ensemble des ses concepts formels $\mathfrak{B}(G, M, I)$ et le treillis de concepts $\underline{\mathfrak{B}}(G, M, I)$. La couverture supérieure (abréviée en couverture-sup) d'un concept formel $D \in \mathfrak{B}(G, M, I)$ est*

constituée des subsumants directs (appelés $upperneighbors$ dans (Ganter et Wille, 1999)) de D dans $\underline{\mathfrak{B}}(G, M, I)$:

$$couverture-sup(D) = \{C \in \mathfrak{B}(G, M, I) \mid D \sqsubseteq C \; and \nexists \, Z \in \mathfrak{B}(G, M, I) \mid D \sqsubseteq Z \sqsubseteq C\}$$

(2) Étant donné un ensemble $\{C_j\}_{j \in J}$ (J étant un entier positif inférieur au nombre de concepts formels dans $\mathfrak{B}(G, M, I)$) de concepts formels dans $\mathfrak{B}(G, M, I)$, la couverture supérieure de l'ensemble $\{C_j\}_{j \in J}$ est définie par l'union des couvertures supérieures des concepts formels :

$$couverture-sup(\{C_j\}_{j \in J}) = \bigcup_{j \in J} couverture-sup(C_j)$$

Définition 8 (pertinence) *(1) Considérons une entrée (un couple formé par un objet et l'ensemble de ses attributs) ($\{a\}$,$\{a\}'$) dans un contexte formel $\mathbb{K} = (G, M, I)$ et une requête Q = ($\{x\}$,$\{x\}'$). l'objet a est pertinent pour la requête Q si et seulement si $\{a\}' \cap \{x\}' \neq \emptyset$, i.e. il existe au moins un attribut dans $\{x\}'$ qui est possédé par l'objet a.*

(2) Le degré de pertinence de l'objet a par rapport à la requête Q est donné par le nombre d'attributs communs entre $\{a\}'$ et $\{x\}'$ i.e. $|\{a\}' \cap \{x\}'|$.

Proposition 1 *Considérons un contexte formel $\mathbb{K} = (G, M, I)$ et une requête Q = ($\{x\}$,$\{x\}'$). Tous les objets dans G pertinents pour Q sont dans l'extension du concept pivot P = ($\{x\}''$,$\{x\}'$) et dans les extensions de ses subsumants dans $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$.*

Preuve 1 *Considérons d'abord les objets dans $\{x\}''$, l'extension du concept pivot. Selon la définition de P = ($\{x\}''$,$\{x\}'$) (définition 6) et la définition de la pertinence (définition 8), tous les objets dans $\{x\}''$ sont pertinents pour Q = ($\{x\}$,$\{x\}'$) puisqu'ils partagent tous les attributs de la requête (l'ensemble $\{x\}'$).*

Considérons maintenant le cas des subsumants du concept pivot P. Soit $C = (A, B)$ un subsumant de P dans $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$, i.e. P = ($\{x\}''$,$\{x\}'$) $\sqsubseteq C = (A, B)$. Par définition de la relation de subsomption "$\sqsubseteq$" dans le treillis de concepts, $B \subseteq \{x\}'$. Ce qui signifie que tout objet dans A possède au moins un attribut dans $\{x\}'$, d'où la pertinence pour Q.

Lorsque son intension est vide, le concept le plus général dans le treillis de concepts (top) n'est pas considéré lors de l'identification des objets pertinents pour une requête. En effet dans un tel cas, il peut exister des objets dans l'extension de top qui ne possèdent aucun attribut de $\{x\}'$ et ne vérifient donc pas le critère de pertinence (définition 8).

3.2 L'algorithme BR-Explorer

Dans cette section nous détaillons l'algorithme de recherche d'information par treillis BR-Explorer. Pour ce faire considérons un contexte formel $\mathbb{K} = (G, M, I)$, le treillis de concepts qui lui correspond $\underline{\mathfrak{B}}(G, M, I)$ et une requête Q = ($\{x\}$,$\{x\}'$).

Intuitivement BR-Explorer procède comme suit. D'abord la requête Q est insérée dans le treillis $\underline{\mathfrak{B}}(G, M, I)$ (voir algorithme 1 ligne 1). Cette insertion produit un nouveau treillis de concepts $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ contenant le concept pivot P = ($\{x\}''$,$\{x\}'$) (ligne 2 de l'algorithme). Comme mentionné précédemment, les objets pertinents pour la requête sont dans

l'extension du concept pivot et des subsumants du concept pivot. Donc pour identifier ces objets, il suffit de commencer la recherche dans le treillis $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ à partir du concept pivot P et effectuer un parcours en largeur (vers le haut) du treillis jusqu'au concept le plus général, $\top$.

La localisation de P dans $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ est détaillée dans la procédure $Localiser\_Pivot$. L'idée de base de la procédure de recherche du concept pivot est l'élagage. La recherche de P commence à partir du concept $bottom$, $\perp$, et à chaque fois que l'intension B d'un concept $C = (A, B)$ ne vérifie pas $\{x\}' \subseteq B$ tous ses subsumants sont ignorés puisque eux aussi ne vérifieront pas cette condition (car leurs intensions sont des parties de B).

Une fois le concept pivot localisé, nous procédons à la construction du résultat, noté $\mathcal{R}_{objets}$ (ligne 8 et 18 dans l'algorithme BR-Explorer). On note par $\mathcal{SUBS}_i$ l'ensemble de concepts considérés à l'itération i de BR-Explorer. La première étape consiste à considérer l'extension de P, le seul concept dans $\mathcal{SUBS}_0$, et à ajouter à $\mathcal{R}_{objets}$ les éventuels objets autres que x dans cette extension avec un rang égal à 1 (lignes 3 à 10 de BR-Explorer). À l'étape suivante nous considérons $SUBS_1 = couverture\text{-}sup(P)$. L'ensemble des objets dans les extension des concepts de $\mathcal{SUBS}_1$ qui ne figurent pas encore dans le résultat (objets émergeants) sont ajoutés à $\mathcal{R}_{objets}$ avec leur rang correspondant (soit 1 si aucun objet n'a été ajouté au résultat à l'étape précédente, soit 2 sinon) (lignes 11 à 21 de BR-Explorer). Le reste des étapes est constitué des itérations sur $\mathcal{SUBS}_i$ en effectuant le même traitement que sur $\mathcal{SUBS}_1$ jusqu'à atteindre un ensemble $\mathcal{SUBS}_i$ qui est vide, auquel cas BR-Explorer s'arrête et retourne le résultat $\mathcal{R}_{objets}$ obtenu. À chaque étape i, si le concept $\top$ appartient à $SUBS_i$ et si son intension est vide alors les objets figurant dans son extension sont ignorés.

4 Déroulement de l'algorithme sur un exemple

Considérons le contexte formel $\mathbb{K} = (G, M, I)$ donné par le tableau 1. Le treillis de

$G \setminus M$	m_1	m_2	m_3	m_4	m_5	m_6
g_1		×	×	×	×	×
g_2		×	×	×	×	
g_3		×	×	×	×	
g_4	×	×	×	×		
g_5				×	×	
g_6	×	×		×		
g_7	×	×				

TAB. 1 – *Exemple de contexte formel* $\mathbb{K} = (G, M, I)$.

concepts $\underline{\mathfrak{B}}(G, M, I)$ correspondant au contexte formel $\mathbb{K} = (G, M, I)$ est donné par la figure 1 (faite par le logiciel ConExp [1] (Yevtushenko, 2000)). Cette représentation, dite *réduite*, des treillis de concepts est la plus utilisée dans la communauté de l'analyse formelle de concepts. Elle s'appuie sur l'héritage à la fois des attributs et des objets entre les nœuds représentant les concepts du treillis. Les attributs sont placés au plus haut dans le treillis : à chaque fois

[1] http ://sourceforge.net/projects/conexp

Algorithm 1 BR-Explorer

Require: $\mathbb{K} = (G, M, I)$, $\underline{\mathfrak{B}}(G, M, I)$ et Q = ($\{x\}$,$\{x\}'$)
Ensure: R_{objets}
1: Insérer Q dans $\underline{\mathfrak{B}}(G, M, I)$
2: P = ($\{x\}''$,$\{x\}'$):= $Localiser\_Pivot(\underline{\mathfrak{B}}(G_Q, M_Q, I_Q), Q)$
3: $n := 1 \setminus\setminus$ n est le niveau dans le treillis de concepts $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ à partir de P
4: $\mathcal{SUBS}_{n-1} := \{P\}$
5: $rang := 1$
6: **if** $\{x\}'' \neq \{x\}$ **then**
7: $\quad \mathcal{R}_{rang} := X''\backslash X$
8: $\quad \mathcal{R}_{objets} := (rang, \mathcal{R}_{rang})$
9: $\quad rang := rang + 1$
10: **end if**
11: **while** $\mathcal{SUBS}_{n-1} \neq$ Ø **do**
12: $\quad \mathcal{SUBS}_n := couverture - sup(\mathcal{SUBS}_{n-1})$
13: $\quad \mathcal{R}_{rang} := \emptyset$
14: $\quad$ **for all** $C = (A, B) \in \mathcal{SUBS}_n$ tel que $B \neq$ Ø **do**
15: $\quad\quad \mathcal{R}_{rang} := \mathcal{R}_{rang} \cup A$
16: $\quad$ **end for**
17: $\quad ObjetsEmergeants := \mathcal{R}_{rang}\backslash(X \cup \mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_{rang-1})$
18: $\quad \mathcal{R}_{objets} := \mathcal{R}_{objets} \cup (rang, ObjetsEmergeants)$
19: $\quad n := n + 1$
20: $\quad rang := rang + 1$
21: **end while**

Algorithm 2 Localiser_Pivot

Require: $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ et Q = ($\{x\}$,$\{x\}'$)
Ensure: P = ($\{x\}''$,$\{x\}'$)
1: trouvé := faux
2: $\mathcal{SUBS} := \perp \setminus\setminus \perp$ est le concept bottom dans $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$
3: **while** trouvé = faux **do**
4: $\quad$ **for** tout $C = (A, B) \in \mathcal{SUBS}$ **do**
5: $\quad\quad$ **if** $\{x\}' = B$ **then**
6: $\quad\quad\quad P := C$
7: $\quad\quad\quad$ trouvé := vrai
8: $\quad\quad\quad$ break
9: $\quad\quad$ **else if** $\{x\}' \subset B$ **then**
10: $\quad\quad\quad \mathcal{SUBS} := couverture - sup(\mathcal{SUBS})$
11: $\quad\quad\quad$ break
12: $\quad\quad$ **end if**
13: $\quad$ **end for**
14: **end while**

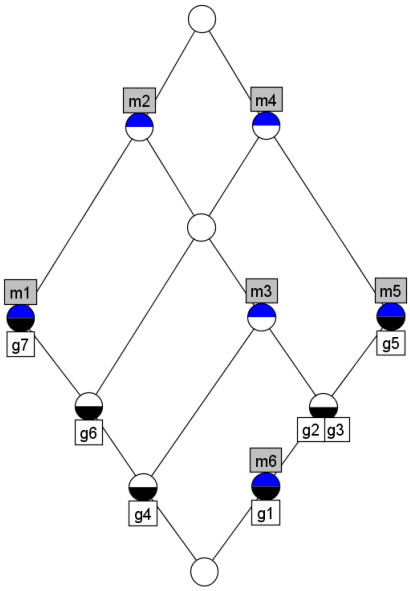

FIG. 1 – *Le treillis de concepts* $\underline{\mathfrak{B}}(G, M, I)$ *correspondant au contexte formel* $\mathbb{K} = (G, M, I)$.

qu'un nœud N est étiqueté par un attribut m, tous les descendants de N dans le treillis héritent l'attribut m. De façon duale, les objets sont placés au plus bas dans le treillis : à chaque fois qu'un nœud N est étiqueté par un objet g tous ses ancêtres héritent vers le haut l'objet g. Ainsi l'extension d'un concept C (correspondant à un nœud N dans le treillis) est obtenue en considérant tous les objets qui apparaissent sur les descendants du nœud N dans le treillis et l'intension de C est obtenue en considérant tous les attributs qui apparaissent sur les ancêtres du nœud N dans le treillis.

Considérons la requête $Q = (\{x\},\{x\}')$ avec $\{x\}' = \{m_3,m_5,m_6\}$. L'ajout de $Q = (\{x\},\{x\}')$ au contexte formel $\mathbb{K} = (G, M, I)$ à l'aide de l'opérateur " $\oplus$ " produit le contexte formel $\mathbb{K}_{\mathbb{Q}} = (G_Q, M_Q, I_Q) = \mathbb{K} \oplus Q$ donné par le tableau 2. Le treillis de concept résultant de

$G \setminus M$	m_1	m_2	m_3	m_4	m_5	m_6
g_1		×	×	×	×	×
g_2		×	×	×	×	
g_3		×	×	×	×	
g_4	×	×	×	×		
g_5				×	×	
g_6	×	×		×		
g_7	×	×				
x			×		×	×

TAB. 2 – *Le contexte formel* $\mathbb{K}_{\mathbb{Q}} = (G_Q, M_Q, I_Q) = \mathbb{K} \oplus Q$.

l'insertion de Q dans le treillis $\mathfrak{B}(G, M, I)$), noté $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$, est donné par la figure 2. Sur cette figure nous avons schématisé les étapes du déroulement de l'algorithme BR-Explorer

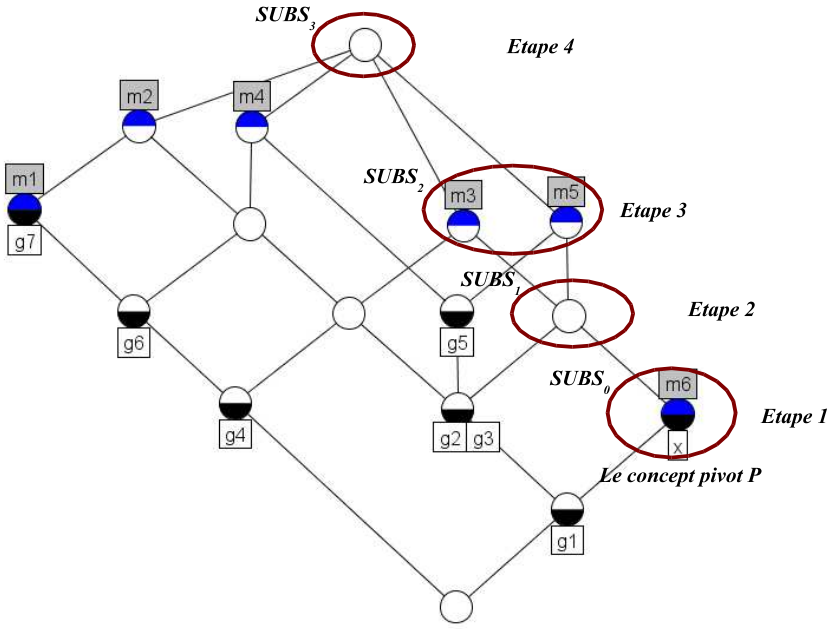

FIG. 2 – *L'application de l'algorithme BR-Explorer sur le treillis* $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$.

pour cet exemple. Les traitements effectués dans chacune de ces étapes sont respectivement les suivantes :

1. Localisation du concept pivot P dans le treillis $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ à l'aide de la procédure $Localiser\_Pivot$. Ici P = ($\{x, g_1\}$,$\{m_3$,m_5,$m_6\}$) = $\mathcal{SUBS}_0$. $\{x\}'' = \{x, g_1\} \neq \{x\}$ (BR-Explorer ligne 6) donc nous ajoutons au résultat $\mathcal{R}_{objets}$ (vide) l'objet g_1 avec un rang égal à 1 et nous incrémentons le rang de 1 (BR-Explorer ligne 9).
2. Comme $\mathcal{SUBS}_0 \neq \emptyset$ (BR-Explorer ligne 11) nous calculons donc $\mathcal{SUBS}_1$:
   $\mathcal{SUBS}_1$ = $couverture - sup(\mathcal{SUBS}_0)$ = {($\{g_1$,g_2,g_3,$x\}$,$\{m_3$,$m_5\}$)}. Les objets qui émergent à cette étape sont g_2 et g_3, qui sont ajoutés à $\mathcal{R}_{objets}$ avec un rang égal à 2 (BR-Explorer ligne 18) et le rang est incrémenté de 1.
3. Comme $\mathcal{SUBS}_1 \neq \emptyset$, $\mathcal{SUBS}_2$ est calculé :
   $\mathcal{SUBS}_2$ = {($\{g_1$,g_2,g_3,g_4,$x\}$,$\{m_3\}$), ($\{g_1$,g_2,g_3,g_5,$x\}$,$\{m_5\}$)}. Les objets qui émergent sont g_4 et g_5 et sont ajoutés à $\mathcal{R}_{objets}$ au rang 3.
4. À cette étape, $\mathcal{SUBS}_3$ est formé du seul concept $\top$, qui a une intension vide : il n'y a donc pas d'objets retournés.
5. Cette étape correspond à l'arrêt de l'algorithme puisque $\mathcal{SUBS}_4 = \emptyset$. Le résultat $\mathcal{R}_{objets}$ est retourné.

Finalement le résultat obtenu en réponse à la requête Q = ($\{x\}$,$\{m_3$,m_5,$m_6\}$) est :

1- g_1 pour le partage des attributs m_3, m_5 et m_6 avec Q
2- g_2 pour le partage des attributs m_3 et m_5 avec Q
g_3 pour le partage des attributs m_3 et m_5 avec Q
3- g_4 pour le partage de l'attribut m_3 avec Q
g_5 pour le partage de l'attribut m_5 avec Q

5 Correction et complétude de BR-Explorer

5.1 Correction

Définition 9 *Étant donné un contexte formel $\mathbb{K} = (G, M, I)$ et une requête $Q = (\{x\}, \{x\}')$, l'algorithme BR-Explorer est correct par rapport au critère de pertinence si tout objet retrouvé par l'algorithme est pertinent pour la requête Q.*

Proposition 2 *L'algorithme BR-Explorer est correct par rapport au critère de pertinence donné à la définition 8.*

Preuve 2 *Considérons une requête $Q = (\{x\}, \{x\}')$ et un objet g retrouvé par l'algorithme BR-Explorer ($g \in \mathcal{R}_{objets}$).*

Selon la proposition 1, si g est retrouvé par BR-Explorer alors g appartient soit à l'extension du concept pivot P soit à l'une des extensions des concepts subsumants P dans le treillis de concepts. Dans les deux cas, nous avons $\{g\}' \cap \{x\}' \neq \emptyset$, ce qui prouve la pertinence de g pour la requête Q (d'après la définition 8).

5.2 Complétude

Définition 10 *L'algorithme BR-Explorer est complet par rapport au critère de pertinence si tout objet de l'ensemble G pertinent pour Q est retrouvé par l'algorithme.*

Proposition 3 *L'algorithme BR-Explorer est complet par rapport au critère de pertinence donné à la définition 8.*

Preuve 3 *Considérons une requête $Q = (\{x\}, \{x\}')$ et un objet $g \in G$ pertinent pour Q. Selon la définition de pertinence (définition 8), nous avons $\{g\}' \cap \{x\}' \neq \emptyset$. Nous pouvons distinguer les deux cas suivants :*

- *Si $\{g\}' \subset \{x\}'$ alors il existe $C = (A, B) \in \underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ tel que $B = \{g\}'$ et $g \in A$ ($C = (\{g\}'', \{g\}') = (A, \{g\}')$). Cela signifie que C est un subsumant du concept pivot P dans le treillis de concepts $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$, et que g est dans l'extension de ce subsumant de P. Selon la proposition 1, g est retrouvé par l'algorithme BR-Explorer.*
- *Si $\{x\}' \subseteq \{g\}'$ alors $\{g\}'' \subseteq \{x\}''$. Et puisque $g \in \{g\}''$ alors $g \in \{x\}''$. Cela signifie que l'objet g appartient à l'extension du concept pivot $P = (\{x\}'', \{x\}')$. Selon la proposition 1, g est retrouvé par l'algorithme BR-Explorer.*

Les propositions 2 et 3 nous permettent d'énoncer le théorème suivant.

Théorème 1 *Étant donné un contexte formel $\mathbb{K} = (G, M, I)$ et une requête $Q = (\{x\}, \{x\}')$. L'algorithme de recherche d'information par treillis BR-Explorer (algorithme 1) est correct et complet par rapport au critère de pertinence donné à la définition 8.*

Le théorème précédent précise que, pour un contexte formel $\mathbb{K} = (G, M, I)$ et une requête Q = ($\{x\}$,$\{x\}'$), si un objet $g \in G$ possède un ensemble de propriétés $\{g\}' \subseteq M$ qui vérifie $\{g\}' \cap X' \neq \emptyset$, alors g est pertinent pour la requête Q et g est retrouvé par l'algorithme BR-Explorer.

6 Conclusion et perspectives

Nous avons détaillé, dans cet article, l'algorithme BR-Explorer de recherche d'information par treillis BR-Explorer. Nous avons prouvé la correction et la complétude de cet algorithme par rapport au critère de pertinence que nous avons défini. Ces preuves garantissent, étant donnée une requête, de trouver tous les objets pertinents et uniquement les objets pertinents pour cette requête dans le contexte considéré.

Nous avons appliqué l'algorithme BR-Explorer pour la recherche de sources de données biologiques dans l'annuaire BioRegistry (Messai et al., 2005; Smail-Tabbone et al., 2005; Messai et al., 2006). Les objets du contexte formel considéré sont les sources de données biologiques et les attributs sont l'ensemble des propriétés associées à ces sources. Ces propriétés sont extraites,des experts biologistes, à partir des métadonnées qui annotent les sources et appartiennent souvent à des ontologies de domaines ou à des vocabulaires contrôlés. BR-Explorer peut aussi être réutilisé pour tout corpus décrit sous forme d'un contexte formel.

En perspectives à ce travail, nous proposons d'étendre la méthode pour traiter des contextes plus complexes tels que les contextes flous (pondération des attributs), les contextes avec des attributs complexes (à valeur dans des intervalles ou des domaines discrets) et de façon générale les contextes multivalués. Une telle extension permettra d'améliorer la définition de la pertinence en considérant par exemple les préférences des utilisateurs et les relation sémantiques qui peuvent exister entre les attributs.

Références

Barbut, M. et B. Monjardet (1970). *Ordre et classification : Algèbre et combinatoire, Tome II*. Paris : Hachette.

Carpineto, C. et G. Romano (2000). Order-theoretical ranking. *Journal of the American Society for Information Science 51*(7), 587–601.

Carpineto, C. et G. Romano (2004). *Concept Data Analysis : Theory and Applications*. John Wiley & Sons.

Ganter, B. et R. Wille (1999). *Formal Concept Analysis* (Mathematical Foundations ed.). Springer.

Godin, R., G. W. Mineau, R. Missaoui, et H. Mili (1995a). Méthodes de classification conceptuelle basées sur les treillis de Galois et applications. *Revue d'intelligence artificielle 9*(2), 105–137.

Godin, R., R. Missaoui, et H. Alaoui (1995b). Incremental Concept Formation Algorithms Based on Galois (Concept) Lattices. *Computational Intelligence 11*, 246–267.

Merwe, D. V. D., S. A. Obiedkov, et D. G. Kourie (2004). AddIntent : A New Incremental Algorithm for Constructing Concept Lattices. In P. W. Eklund (Ed.), *ICFCA Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings*, Volume 2961, pp. 372–385. Springer.

Messai, N., M.-D. Devignes, A. Napoli, et M. Smail-Tabbone (2005). Querying a bioinformatic data sources registry with concept lattices. In G. S. F. Dau, M.-L. Mugnier (Ed.), *ICCS'05*, LNAI, pp. 323–336. Springer-Verlag Berlin Heidelberg.

Messai, N., M.-D. Devignes, A. Napoli, et M. Smail-Tabbone (2006). Treillis de concepts et ontologies pour interroger l'annuaire de sources de données biologiques BioRegistry. *Ingénierie des Systèmes d'Information : Systèmes d'information spécialisés 11*(1), 39–60.

Smail-Tabbone, M., S. Osman, N. Messai, A. Napoli, et M.-D. Devignes (2005). Bioregistry : a structured metadata repository for bioinformatic databases. In *The 1st International Symposium on Computational Life Science, CompLife'05, Konstanz, Germany, September 25-27, 2005*, Konstanz, Germany.

Yevtushenko, S. A. (2000). System of data analysis "Concept Explorer". In *Proceedings of the 7th national conference on Artificial Intelligence KII-2000*, Russia, pp. 127–134.

Summary

In this paper we present the BR-Explorer algorithm, a sound and complete information retrieval algorithm, based on Formal Concept Analysis. The BR-Explorer algorithm addresses the problem of retrieving the relevant objects for a given query. Initially, a formal context representing the relation between a set of objects and the corresponding set of their attributes is given, and the associated concept lattice is built. The BR-Explorer algorithm starts by generating a formal concept representing the considered query, and classifies this query concept in the concept lattice. Then, the BR-Explorer tries to locate the so-called “pivot” concept in the concept lattice, for building step by step the query result (considering the pivot superconcepts in the concept lattice). Finally, the BR-Explorer algorithm returns a set of relevant objects ranked with respect to their relevance with the query.