Traitement d'attributs inter-dépendants pour la recherche d'information par treillis

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

Traitement d'attributs inter-dépendants pour la recherche d'information par treillis

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

LORIA UMR 7503, BP 239, 54506 Vandœuvre-Lès-Nancy, FRANCE
Nizar.Messai@loria.fr
http://www.loria.fr/∼messai

Résumé :

Dans cet article nous étudions la prise en compte des relations entre attributs dans le cadre de la recherche d'information par treillis de concepts. Il s'agit en premier d'étendre l'analyse de concepts formels (FCA) pour gérer les contextes formels où il existe des relations entre les attributs et par la suite d'appliquer cette extension à la recherche d'information par treillis. Les relations entre attributs permettent de distinguer des attributs plus importants et d'autres moins importants ce qui traduit la préférence qu'on pourrait avoir pour un (ensemble d') attribut(s) plutôt que d'autres. La prise en compte de ces relations est particulièrement intéressante pour la recherche d'information par treillis pour deux raisons : d'une part elle permet d'effectuer une exploration guidée par des connaissances d'un treillis de concepts et d'autre part elle permet de préciser la façon dont les mots-clés d'une requête doivent être considérés et interprétés afin d'obtenir la réponse la plus satisfaisante.

Mots-clés : Analyse de concepts formels, treillis de concepts, treillis de Galois, formules de dépendance entre attributs, recherche d'information par treillis.

1 Introduction

Dans l'analyse de concepts formels (FCA pour "Formal Concept Analysis"), bien que les attributs soient binaires, ils peuvent être très variés. Cette variété reflète la richesse de l'information que peut représenter un contexte formel. Dans le cadre de la recherche d'information, la façon dont les attributs d'un contexte formel sont exploités diffère en fonction de la nature de l'information à extraire d'une part et des connaissances supplémentaires dont on dispose sur ces attributs d'autre part. De ce fait l'importance d'un ou de plusieurs attributs d'un même contexte peut varier d'une utilisation à une autre. Considérons par exemple un contexte qui représente des bases de données biologiques et leurs attributs. Parmi les attributs nous avons le nom de l'espèce étudiée, les standards (ontologies et vocabulaires contrôlés) avec lesquelles les contenus des bases de données sont compatibles, la fraîcheur des contenus des bases de données (fréquence

de mise à jour, dernière modification), etc. Dans une première exploration des données de ce contexte on pourrait s'intéresser aux bases de données contenant des informations sur *l'espèce humaine*. L'attribut relatif à ce nom d'espèce sera préféré aux autres attributs. Si plusieurs bases de données contiennent des informations sur l'espèce humaine, celle dont les informations ont été mises à jour récemment sera plus intéressante à explorer. Si une base de données contient des informations à jour mais relatives à une espèce autre que l'espèce humaine, elle ne sera pas utile à explorer. On considère dans ce cas que l'attribut *contenu à jour* donne une information supplémentaire dans le cas où il s'agit d'une base de données relative à l'espèce humaine. Dans une deuxième exploration, on s'intéresse aux bases de données contenant les informations les plus récentes dans le domaine de la biologie afin de réaliser une étude statistique. Dans ce cas l'attribut *contenu à jour* devient préféré aux autres attributs. De façon similaire à l'exemple précédent, une information sur le nom de l'espèce traitée dans chaque base de données peut être significative dans le cas où il s'agit d'une base de données dont le contenu est à jour. Ces deux exemples illustrent la variation de l'importance à accorder aux attributs dans un contexte formel en fonction de l'exploitation des données qu'il contient. Dans les deux cas on peut distinguer un attribut principal qui doit être présent pour qu'une base de données soit considérée intéressante, et un (ou plusieurs) attribut(s) secondaires qui donnent des informations complémentaires.

La prise en compte des préférences entre attributs dans un contexte formel peut être bénéfique dans plusieurs applications de l'analyse de concepts formels, telles que la recherche d'information (par treillis) (Carpineto & Romano, 2004; Messai *et al.*, 2005, 2006b). Une première formalisation des préférences entre attributs a été proposée par Belohlávek & Sklenar (2005). Dans l'article en cours nous adaptons cette formalisation et nous l'utilisons pour la recherche d'information par treillis. La suite de l'article est organisée comme suit. Nous rappelons dans la section 2 les définitions de l'analyse de concepts formels nécessaires pour la compréhension du travail présenté. Dans la section 3 nous présentons les formules de dépendances entre attributs. Dans la section 4 nous détaillons l'application de ces formules pour la recherche d'information par treillis et nous illustrons par des exemples. Nous finirons par conclure ce travail en section 5.

2 Analyse de concepts formels (FCA)

Dans cette section nous rappelons brièvement les définitions et les résultats concernant l'analyse de concepts formels nécessaires pour la suite de cet article (la référence de base en la matière est (Ganter & Wille, 1999)).

Définition 1 (Treillis)
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* $\top$.

Définition 2 (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 $I \subseteq G \times M$. Un couple $(g, m) \in I$ (notée aussi gIm) signifie que l'objet $g \in G$ possède l'attribut $m \in M$.

Le tableau 1 est un exemple de contexte formel. Les objets sont des bases de données biologiques ($BD1$, $BD2$, ... $BD8$) et les attributs sont de trois types : les espèces concernées par les informations contenues dans les bases de données (les mammifères, les oiseaux, les amphibiens et les poissons), la qualité du contenu des bases de données (à jour, complet) et l'ontologie utilisée comme référence (ontologie 1, ontologie 2). La relation I exprime le fait qu'une base de données est annotée par un attribut (auquel cas la case correspondante dans le tableau est cochée par "x") ou non (auquel cas la case correspondante est vide). Considérons par exemple la base de données $BD1$. Le contenu de $BD1$ a les caractéristiques suivantes : il concerne les espèces amphibiens et poissons (($BD1, Amphibiens$) $\in I$ et ($BD1, Poissons$) $\in I$), il est complet (($BD1, Complet$)$\in I$) et il contient des termes provenant de l'ontologie 2 (($BD1, Ont2$) $\in I$).

TAB. 1 – Le contexte formel $\mathbb{K} = (G, M, I)$ où Ma, Oi, Am et Po sont des abréviations pour Mammifères, Oiseaux, Amphibiens et Poissons respectivement

Attr / Obj	Classe d'espèces				Contenu		Ontologies	
	Ma	Oi	Am	Po	À jour	Complet	Ont1	Ont2
BD1			x	x		x		x
BD2			x	x	x	x		x
BD3	x		x					x
BD4	x	x			x			
BD5	x		x		x	x		
BD6	x	x				x	x	
BD7	x	x			x	x	x	x
BD8	x	x						x

Définition 3
Soit $\mathbb{K}$ 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).

Définition 4 (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 appelées $extension$ (extent) et $intension$ (intent) du concept formel (A, B). L'ensemble des concepts formels associés au contexte formel $\mathbb{K} = (G, M, I)$ est noté $\mathfrak{B}(G, M, I)$

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 $''$) 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 $''$). 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 $(A1, B1)$ et $(A1, B1)$ est dit *subsumé* de $(A2, B2)$.

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 & Monjardet, 1970) et noté par $\underline{\mathfrak{B}}(G, M, I)$.

Le treillis de concepts correspondant au contexte formel donné dans le tableau 1 est représenté à la figure 1.

En FCA, cette représentation des treillis s'appelle "notation réduite". 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 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, g est hérité vers le haut et tous les ancêtres le partagent. Ainsi l'extension A d'un concept (A, B) est obtenue en considérant tous les objets qui apparaissent sur les descendants du nœud n dans le treillis et son intension B est obtenue en considérant tous les attributs qui apparaissent sur les ancêtres du nœud n dans le treillis.

3 Formules de dépendance entre les attributs

3.1 Introduction

Le treillis de concepts généré à partir d'un contexte formel fournit une classification des objets sur la base des attributs qu'ils ont en commun. Les attributs du contexte sont considérés au même niveau d'importance. Cet aspect peut être pénalisant dans le cas des treillis de grande taille où seul une partie du treillis est particulièrement intéressante à explorer. Pour permettre de se focaliser sur une telle partie du treillis, il est judicieux de distinguer des attributs qui sont jugés importants ("principaux") et d'autres qui sont jugés "secondaires".

La relation entre ces attributs est donnée par les formules de dépendance entre attributs introduites dans (Belohlávek *et al.*, 2004; Belohlávek & Sklenar, 2005). Dans

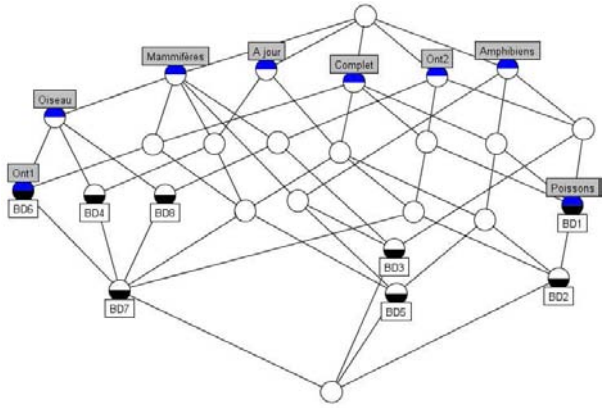

FIG. 1 – Le treillis de concepts correspondant au contexte formel $\mathbb{K} = (G, M, I)$ donné dans le tableau 1

la suite de cette section nous détaillons formellement les dépendances entre attributs et nous montrons leurs effets sur un treillis de concepts. Pour cela nous considérons un contexte formel $\mathbb{K} = (G, M, I)$ où G est un ensemble d'objets, M un ensemble d'attributs, et I une relation d'incidence (sur $G \times M$).

3.2 Définitions

Définition 5 (Attribut principal, attribut secondaire)
Un attribut $m \in M$ est dit principal si sa présence n'est conditionnée par aucun autre attribut. Dans le cas contraire, un attribut $m \in M$ est dit secondaire si sa présence n'est pas possible sans la présence d'au moins un attribut principal.

La relation entre les attributs principaux et les attributs secondaires est donnée par les formules de dépendances entre attributs définies comme suit (Belohlávek & Sklenar, 2005) :

Définition 6 (Formule de dépendance entre attributs)
Soient m, m_1, ... , m_n des attributs dans M. Une formule φ de dépendance entre attributs est de la forme :

$$m \sqsubseteq m_1 \sqcup m_2 \sqcup ... \sqcup m_n$$

m_1, ... , m_n sont les attributs principaux et m est l'attribut secondaire.

Une formule de dépendance entre attributs s'interprète de la façon suivante : m est un attribut secondaire dont la présence n'est possible que si elle est accompagnée par celle d'au moins un des attributs principaux m_1, ... , m_n. Autrement dit, si m apparaît dans l'intension d'un concept formel alors m_1 ou m_2 ou ... ou m_n apparaît aussi dans cette même intension. Cela nous conduit à définir les concepts formels cohérents vis-à-vis d'une formule de dépendance entre attributs comme suit.

Définition 7 (Coherence de concepts formels)
Un concept formel (A, B) *est dit cohérent vis-à-vis d'une formule* φ *de dépendance entre attributs,* $m \sqsubseteq m_1 \sqcup m_2 \sqcup ... \sqcup m_n$*, si et seulement si on a : si* $m \in B$ *alors* $\exists\, i \in \{1, ..., n\}$ *tel que* $m_i \in B$*. On dit aussi que* (A, B) *satisfait* φ *et on note* $(A, B) \models \varphi$*.*

Une formule de dépendance entre attributs exprime des connaissances de domaine non représentées dans un contexte formel. Ces connaissances peuvent provenir soit de ressources externes telles que les ontologies de domaine contenant des relations sémantiques entre les attributs du contexte formel soit tout simplement de l'expertise sur des relations implicites entre ces attributs. La représentation de telles connaissances sous la forme de formules de dépendance entre attributs produit un ensemble de formules, noté $\mathcal{F}$, à appliquer simultanément sur les concepts formels d'un treillis. L'application de telles formules à un treillis de concepts consiste à éliminer les concepts formels incohérents vis-à-vis de ces formules. En d'autres termes il s'agit de réduire le treillis à l'ensemble de concepts satisfaisant les formules dans $\mathcal{F}$.

3.3 Notations

Dans un treillis de concepts $\underline{\mathfrak{B}}(G, M, I)$, on note $\mathfrak{B}_{\mathcal{F}}(G, M, I)$ l'ensemble de concepts formels cohérents vis-à-vis d'un ensemble $\mathcal{F}$ de formules de dépendances entre attributs. La partie du treillis constituée par l'ensemble de concepts $\mathfrak{B}_{\mathcal{F}}(G, M, I)$ est notée $\underline{\mathfrak{B}}_{\mathcal{F}}(G, M, I)$. $\underline{\mathfrak{B}}_{\mathcal{F}}(G, M, I)$ constitue une projection du treillis $\underline{\mathfrak{B}}(G, M, I)$ conformément aux formules de $\mathcal{F}$. Le parcours des concepts de $\underline{\mathfrak{B}}_{\mathcal{F}}(G, M, I)$ revient à explorer les données du contexte formel $\mathbb{K} = (G, M, I)$ selon l'ensemble de formules $\mathcal{F}$. Cette exploration peut se voir comme guidée par des connaissances du domaine exprimées sous forme de formules de dépendance entre attributs.

Le treillis $\underline{\mathfrak{B}}_{\mathcal{F}}(G, M, I)$ dépend de l'ensemble de formules de dépendance entre attributs $\mathcal{F}$, qui dépend à son tour des contraintes et des connaissances qu'on désire exprimer via ces formules.

3.4 Exemple

Considérons comme exemple le contexte formel des bases de données biologiques donné dans le tableau 1 et supposons que l'on s'intéresse aux bases de données contenant des informations sur les espèces ne vivant pas dans l'eau. Cela revient à considérer les attributs *Mammifères* et *Oiseaux* comme attributs principaux. Les attributs relatifs aux contenus et aux ontologies sont des attributs secondaires. En terme de formules de dépendance entre attributs cela donne l'ensemble $\mathcal{F}_1$ de formules suivantes :

φ_1 *Contenu à jour* $\sqsubseteq$ *Mammifère* $\sqcup$ *Oiseaux*
φ_2 *Contenu Complet* $\sqsubseteq$ *Mammifère* $\sqcup$ *Oiseaux*
φ_3 *Ontologie 1* $\sqsubseteq$ *Mammifère* $\sqcup$ *Oiseaux*
φ_4 *Ontologie 2* $\sqsubseteq$ *Mammifère* $\sqcup$ *Oiseaux*

En appliquant ces formules au treillis $\underline{\mathfrak{B}}(G, M, I)$ donné à la figure 1 les concepts incohérents vis-à-vis de cet ensemble de formules sont écartés. Un exemple de concept incohérent est ($\{BD1, BD2, BD5, BD6, BD7\}$,$\{Complet\}$). Son incohérence est due

au fait que son intension contient l'attribut secondaire *Complet* et ne contient aucun des deux attributs principaux *Mammifères* et *Oiseaux*. La formule φ_2 n'est donc pas satisfaite par ce concept. L'interprétation de cette incohérence est la suivante : l'attribut *Complet* signifie que les contenus des bases de données $BD1$, $BD2$, $BD5$, $BD6$ et $BD7$ (extension du concept formel) sont complets, cependant il n'y a aucune information sur l'objet de ces contenus. Ceci fait que le concept n'est pas assez informatif quand aux espèces décrites dans les bases de données $BD1$, $BD2$, $BD5$, $BD6$ et $BD7$.

Un exemple de concept cohérent est ($\{BD5, BD6, BD7\}$,{*Mammifères, Complet*}). Dans l'intension de ce concept, l'attribut secondaire *Complet* est accompagné de l'attribut principal *Mammifères*. Le treillis $\underline{\mathfrak{B}}_{\mathcal{F}_1}(G, M, I)$ formé par l'ensemble de concepts cohérents vis-à-vis de l'ensemble de formules $\mathcal{F}_1 = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ est donné à la figure 2. Dans ce treillis, on remarque que les attributs principaux apparaissent en haut

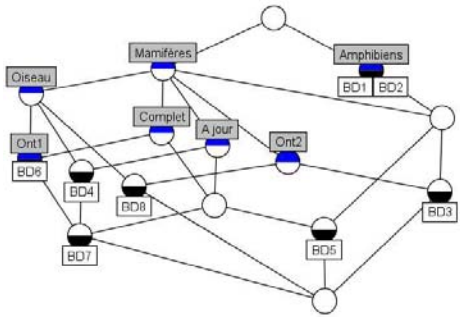

FIG. 2 – Le treillis de concepts $\underline{\mathfrak{B}}_{\mathcal{F}_1}(G, M, I)$ résultant de l'application de l'ensemble $\mathcal{F}_1$ de formules de dépendance entre attributs au treillis $\underline{\mathfrak{B}}(G, M, I)$ donné à la figure 1

du treillis et que les attributs secondaires apparaissent toujours en dessous des attributs principaux. Cela s'explique par le fait que chaque attribut principal caractérise une classe de concepts (tous les concepts ayant une intension qui contient cet attribut) alors que les attributs secondaires constituent des spécialisations dans chacune des classes de concepts définies par les attributs principaux.

4 Prise en compte des formules de dépendance entre attributs dans la recherche d'information par treillis

La recherche d'information a été considérée comme un domaine d'application privilégié (Godin *et al.*, 1995; Carpineto & Romano, 1996) depuis les débuts de l'analyse de concepts formels (Wille, 1982). En plus du mode classique de recherche d'information par requête, l'analyse de concepts formels offre la possibilité de naviguer dans le corpus de recherche (représenté dans un treillis de concepts) (Ferré & Ridoux, 2001). Plus tard, des méthodes de recherche d'information par treillis ont été mises en place (Carpineto & Romano, 2004; Messai *et al.*, 2006a). Ces propositions visent à améliorer les performances de la recherche d'information par treillis en intégrant au mieux

les connaissances du domaine. Dans (Messai *et al.*, 2006a), la prise en compte de ces connaissances est assurée par le module de raffinement de requêtes à partir d'ontologies de domaine. Cette idée a permis de minimiser le problème du silence dans la réponse à une requête donnée en considérant les relations sémantiques entre les mots clés de la requête et les attributs du contexte formel. Dans ce paragraphe, nous présentons une autre extension de la recherche d'information par treillis. Elle consiste à considérer les relations sémantiques pouvant exister entre les attributs d'un même contexte formel. Ces relations sont représentées sous forme de formules de dépendance entre attributs.

L'utilisation des formules de dépendance entre attributs vient perfectionner la recherche d'information par treillis dans ses deux modes : navigation et requête.

4.1 Recherche d'information par navigation

Dans le cas de la recherche par navigation, la réorganisation du treillis conformément aux contraintes exprimées dans les formules de dépendance entre attributs réduit l'espace de navigation aux concepts cohérents vis-à-vis de ces contraintes. De cette façon on peut considérer que l'ensemble de formules de dépendance entre attributs appliqué à un contexte formel est une préparation de ce contexte à la recherche par navigation conformément aux contraintes données dans ces formules. On peut ainsi parler de navigation guidée par des connaissances de domaine. Considérons par exemple le treillis $\underline{\mathfrak{B}}_{\mathcal{F}_1}(G, M, I)$ donné à la figure 2. Ce treillis résulte de l'application de l'ensemble $\mathcal{F}_1$ de formules de dépendance entre attributs (détaillé à la section 3.4) au treillis $\underline{\mathfrak{B}}(G, M, I)$ donné à la figure 1. La navigation dans $\underline{\mathfrak{B}}_{\mathcal{F}_1}(G, M, I)$ revient à explorer les concepts formels cohérents vis-à-vis de l'ensemble $\mathcal{F}_1$ de formules de dépendance entre attributs.

4.2 Recherche d'information par requête

Dans le cas de la recherche d'information par requête, les formules de dépendance entre attributs ($\mathcal{F}$) peuvent être considérées de deux façons. La première façon consiste à appliquer les formules de dépendance entre attributs au contexte formel considéré auquel cas on parle de formules globales. La deuxième consiste à limiter ces formules aux attributs de la requête auquel cas on parle de formules locales. Dans la suite nous détaillons l'application des deux types de formules.

4.2.1 Application des formules globales

L'application de l'ensemble de formules de dépendances (noté $\mathcal{F}$) a été détaillée précédemment. Elle produit un treillis contraint $\underline{\mathfrak{B}}_{\mathcal{F}}(G, M, I)$. Une fois le treillis $\underline{\mathfrak{B}}_{\mathcal{F}}(G, M, I)$ obtenu, les requêtes peuvent être traitées. Nous rappelons ici qu'une requête est représentée par un concept formel virtuel dont l'intension est formée par les attributs (mots clés) associés à la requête (Messai *et al.*, 2006b). Formellement ce concept est défini de la façon suivante :

Définition 8 (Concept requête)
Une requête Q est un couple $(\{x\}, \{x\}')$ où $\{x\}'$ est un ensemble d'attributs décrivant

les objets à retrouver et x *est un objet virtuel supposé satisfaire tous les attributs de* $\{x\}'$.

L'algorithme de recherche d'information par treillis, BR-Explorer (Messai *et al.*, 2006b), est applicable dans ce cas. Il consiste à insérer la requête Q considérée dans le treillis de concepts avec un algorithme de classification incrémental et à chercher ensuite les objets pertinents dans les extensions des super-concepts de Q dans le treillis résultant. L'application de l'algorithme BR-Explorer sur un treillis contraint par des formules de dépendance entre attributs augmente la précision du résultat obtenu. Pour illustrer cette idée, considérons une requête recherchant les bases de données contenant des informations sur les mammifères et ayant un contenu complet et à jour. Les attributs de cette requête sont donc {*Mammifères, Complet, À jour*}. Le treillis obtenu et les étapes de l'exécution de l'algorithme de recherche des bases de données pertinentes pour cette requête sont représentés à la figure 3.

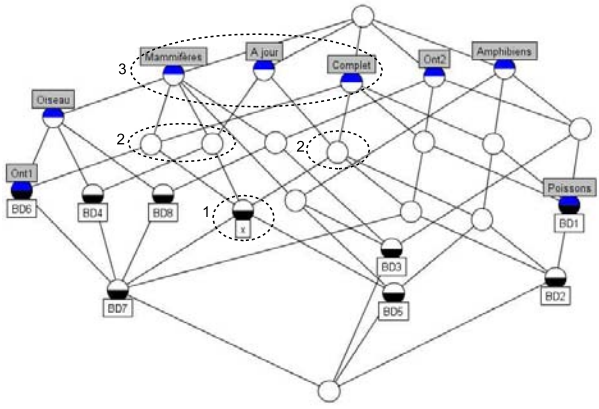

FIG. 3 – Déroulement de l'algorithme BR-Explorer sur le treillis resultant de l'insertion de Q= ($\{x\}$,{ *Mammifères, Complet, À jour*}) dans le treillis $\underline{\mathfrak{B}}(G, M, I)$

Le résultat obtenu en réponse est le suivant :

1- $BD5$: *Mammifères, À jour, Complet*
   $BD7$: *Mammifères, À jour, Complet*
2- $BD4$: *Mammifères, À jour*
   $BD6$: *Mammifères, Complet*
   $BD2$: *À jour, Complet*
3- $BD3$: *Mammifères*
   $BD8$: *Mammifères*
   $BD1$: *Complet*

Dans ce résultat on retrouve $BD2$ au rang 2 alors qu'elle ne contient aucune information sur les mammifères. Cela vient du fait que dans la requête il n'est pas explicitement mentionné que l'attribut *Mammifère* est l'objet principal de la recherche. Cette distinction entre attributs d'une même requête peut être effectuée en utilisant les formules de

dépendance entre attributs.

Pour illustrer cela, considérons la même requête Q= ($\{x\}$,{ *Mammifères, Complet, À jour*}) dans le cas du treillis sous contraintes, $\underline{\mathfrak{B}}_{\mathcal{F}}(G, M, I)$, donné à la figure 2. Dans l'ensemble de formules $\mathcal{F}$ considéré, les formules φ_1 et φ_2 précisent que l'attribut *Mammifères* est un attribut principal et que les attributs *Complet* et *À jour* sont des attributs secondaires. L'insertion de la requête dans ce treillis et l'execution de l'algorithme BR-Explorer sur le treillis résultant sont schématisées à la figure 4.

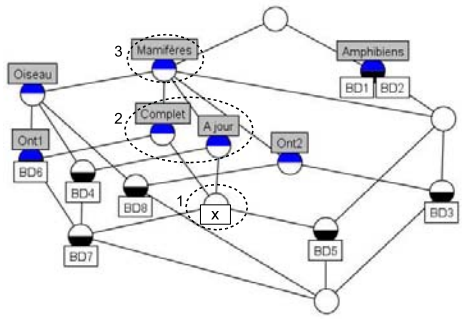

FIG. 4 – Déroulement de l'algorithme BR-Explorer sur le treillis resultant de l'insertion de Q= ($\{x\}$,{*Mammifères, Complet, À jour*}) dans le treillis $\underline{\mathfrak{B}}_{\mathcal{F}_1}(G, M, I)$

La réponse obtenue dans ce cas est la suivante :

1- $BD5$: *Mammifères, À jour, Complet*
   $BD7$: *Mammifères, À jour, Complet*
2- $BD4$: *Mammifères, À jour*
   $BD6$: *Mammifères, Complet*
3- $BD3$: *Mammifères*
   $BD8$: *Mammifères*

On remarque dans cette réponse que les bases de données $BD1$ et $BD2$ ont été écartées, les concepts qui les contiennent ne sont pas cohérents vis-à-vis de l'ensemble $\mathcal{F}_1$ de formules de dépendance entre attributs considérés et plus particulièrement vis-à-vis des formule φ_1 et φ_2 puisque dans le cas de $BD2$ les attributs secondaires *À jour* et *Complet* ne sont accompagnés par aucun attribut principal et dans le cas de $BD1$ l'attribut secondaire *Complet* n'est accompagné d'aucun attribut principal.

Il est à noter ici que tout concept requête, soumis à un treillis $\underline{\mathfrak{B}}_{\mathcal{F}}(G, M, I)$ pour un ensemble de formules de dépendance entre attributs $\mathcal{F}$, doit satisfaire toutes les formules de $\mathcal{F}$. Ceci est les cas de l'exemple de requêtes traité ci-dessus.

4.2.2 Application des formules locales

Les formules locales ne concernent pas tous les attributs du contexte formel. Elles sont à vérifier uniquement par le concept requête. Celui-ci est ensuite inséré dans le treillis $\underline{\mathfrak{B}}(G, M, I)$. Pour que la réponse obtenue vérifie les contraintes considérées, il est possible de procéder des deux manières suivantes. La première consiste à récupérer

la réponse donnée par *BR-Explorer* appliqué au treillis $\underline{\mathfrak{B}}(G, M, I)$ et écarter par la suite les objets qui ne satisfont pas les formules considérées. Considérons à nouveau l'exemple détaillé à la section 3.4. Les objets qui ne satisfont pas l'ensemble $\mathcal{F}_1$ sont $BD2$ et $BD1$. Ils sont filtrés de la réponse à la requête considérée. La deuxième consiste à considérer ces formules au fur et à mesure de la recherche des objets pertinents dans le treillis. À chaque étape de la recherche on teste si le concept en cours satisfait les formules de dépendance entre attributs auquel cas on considère les objets dans son extension et on continue la recherche dans ses subsumants. Dans le cas où le concept en cours ne satisfait pas l'une des formules, il est inutile de considérer ses subsumants dans le treillis. En effet, d'après la définition de la cohérence de concepts donnée plus haut, l'intension d'un concept incohérent ne contient que des attributs secondaires. Et comme les subsumants d'un concept C ont des intensions incluses dans l'intension de C, ils seront incohérents. L'illustration de cette idée est donnée à la figure 5. À la deuxième itération de l'algorithme, un concept qui ne satisfait pas l'ensemble de formules $\mathcal{F}_1$ (le concept représenté par un nœud barré). Ce concept est ignoré et ses subsumants dans le treillis ne sont pas considérés dans la suite des itérations de l'algorithme.

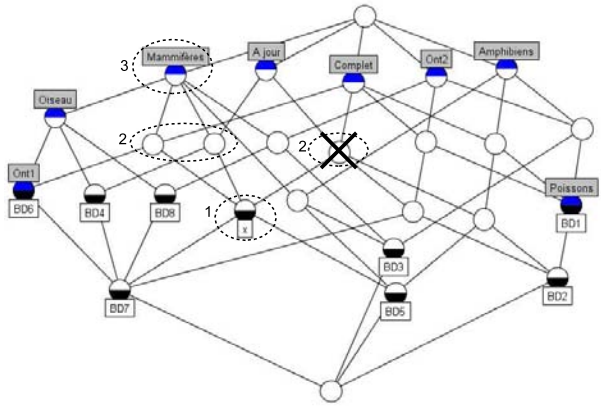

FIG. 5 – Application des formules locales lors du traitement d'une requête

Contrairement aux formules globales qui sont appliquées une fois pour toute au départ, les formules locales sont spécifiées pour chaque requête et leurs application n'affecte pas le treillis. Elles peuvent être considérées comme une partie de la requête qui donne la façon dont ses attributs (mots-clés) doivent être interprétés et qui précise que certains attributs sont préférés à certains autres. Cette façon de considérer les formules de dépendance entre attributs augmente l'expressivité dans les requêtes.

5 Conclusion

La prise en compte des relations entre attributs permet d'effectuer une exploration du treillis de concepts conformément à des contraintes exprimées dans des formules

de dépendances entre attributs. Ces formules reflètent les préférences qu'on donne à un ou plusieurs attributs et permettent ainsi de différencier les attributs principaux et des attributs secondaires. Dans le cas de la recherche d'information par treillis, la prise en compte des dépendances entre attributs permet de minimiser le bruit en filtrant les concepts incohérents vis-à-vis des formules et de restreindre la recherche aux concepts cohérents. De cette façon on peut considérer que les formules de dépendance entre attributs permettent une exploration du treillis guidée par les connaissances du domaine représentées sous forme de formules de dépendances entre attributs.

Références

BARBUT M. & MONJARDET B. (1970). *Ordre et classification : algèbre et combinatoire*.

BELOHLÁVEK R. & SKLENAR V. (2005). Formal concept analysis constrained by attribute-dependency formulas. In B. GANTER & R. GODIN, Eds., *ICFCA*, volume 3403 of *Lecture Notes in Computer Science*, p. 176–191 : Springer.

BELOHLÁVEK R., SKLENAR V. & ZACPAL J. (2004). Formal concept analysis with hierarchically ordered attributes. *International Journal of General Systems*, **33**(4), 283 – 294.

CARPINETO C. & ROMANO G. (1996). A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning*, **24**(2), 95–122.

CARPINETO C. & ROMANO G. (2004). *Concept Data Analysis : Theory and Applications*. John Wiley & Sons.

FERRÉ S. & RIDOUX O. (2001). Searching for objects and properties with logical concept analysis. In H. S. DELUGACH & G. STUMME, Eds., *International Confonference on Conceptual Structures, ICCS 2001*, volume 2120 of *LNCS*, p. 187 – 201 : Springer Verlag.

GANTER B. & WILLE R. (1999). *Formal Concept Analysis*. Springer, mathematical foundations edition.

GODIN R., MINEAU G. W. & MISSAOUI R. (1995). Méthodes de classification conceptuelle basées sur les treillis de Galois et applications. *Revue d'intelligence artificielle*, **9**(2), 105–137.

MESSAI N., DEVIGNES M.-D., NAPOLI A. & SMAIL-TABBONE M. (2006a). 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.

MESSAI N., DEVIGNES M.-D., NAPOLI A. & SMAÏL-TABBONE M. (2005). Querying a bioinformatic data sources registry with concept lattices. In *13th International Conference on Conceptual Structures, ICCS 05, Kassel, Germany, July 18-22, 2005*, p. 323–336.

MESSAI N., DEVIGNES M.-D., NAPOLI A. & SMAÏL-TABBONE M. (2006b). Br-explorer : An fca-based algorithm for information retrieval. In *Fourth Internationnal Conference on Concept Lattices and their Applications, CLA 2006, October 30th - November 1st, Yasmine Hammamet, Tunisia*, p. 285–290.

WILLE R. (1982). Restructuring lattice theory : an approach based on hierarchies of concepts. *Ordered sets*, p. 445–470.