



Attribute Exploration Algorithms on Ontology Construction

Qin Ping, Zhang Zhongxiang, Gao Hualing, Wang Ju

► To cite this version:

Qin Ping, Zhang Zhongxiang, Gao Hualing, Wang Ju. Attribute Exploration Algorithms on Ontology Construction. 6th IFIP TC 12 International Conference on Intelligent Information Processing (IIP), Oct 2010, Manchester, United Kingdom. pp.234-244, 10.1007/978-3-642-16327-2_29 . hal-01055073

HAL Id: hal-01055073

<https://inria.hal.science/hal-01055073>

Submitted on 11 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Attribute Exploration Algorithms on Ontology Construction^{*}

Qin Ping^{1,2}, Zhang Zhongxiang¹, Gao Hualing¹, Wang Ju¹

¹ Guangxi Normal University, College of Computer Science and Information Technology, Guilin, 541004, China

² Guangxi Arts Institute, Postgraduate Affairs Office, Nanning, 530022, China
qpapple_5716@163.com

Abstract. Attribute exploration in FCA is proposed by Baader etc. in the past decade and it is an effective tool applying to description logics to construct ontology on Semantic Web. The authors firstly introduce attribute exploration algorithm, then investigate different cases in which the redundant computation may occur. As new results, the improved attribute exploration algorithm is proposed in terms of relevancy. We also give the proof of the completeness of the improved algorithm, and show how the proposed algorithm avoids redundancy and simplifies computation in some certain cases. We finally present the method to construct an ontology based on attribute exploration algorithm(AEOCM) on the open formal context, and specify the implementation procedure of this method in terms of instantiation.

Keywords: Attribute Exploration Algorithms, Description Logics, Formal Concept Analysis, Pseudo Intent, Ontology Construction

1 Introduction

With the development of the semantic web, we have to confront the problem on searching useful information from the mass data. The way to acquire the information which we focus on is making good use of ontology to improve the veracity and validity on searching. An ontology, the backbone of the Semantic Web, is an explicit and formal specification of a conceptualization and is consisted of finite list of terms and the relationships between these terms. A well-defined ontology should be constructed by promising approaches and assures the soundness and completeness. Therefore, how to construct a well-defined ontology becomes a hotspot on the field of intelligent information processing.

Attribute exploration, a simple but useful knowledge acquisition technique, is an important tool from Formal Concept Analysis (FCA), a mathematical theory for concepts and conceptual hierarchies. It is used to acquire knowledge from a domain expert by asking successive questions, find implication between attributes which can

^{*}Supported by The National Natural Science Foundation of China under Grant No.60573010, 60663001; The Innovation Project of Guangxi Graduate Education under Grant No.2008M025. express the knowledge on inclusion between object sets, and get all of the intent and stem base. After the method, applying formal concept analysis to description logics

presented by Franz Baader in 2004-2007, there are some preferable results to modify the attribute exploration to determine a minimal set of implications^[1], extend both the terminological and the assertional part of a description logic knowledge base^[2], extend the definition on conceptual description on the least common subsumer (lcs) to background terminology^[3], and show the attribute exploration algorithm to compute the subsumption hierarchy of all lcs as well as the hierarchy of all conjunctions of subsets of description logic concepts^[4].

Nowadays, many ontologies are constructed by manual or semi-automatic work. Attribute exploration is an efficient and automatic way to generate concept lattice and construct ontology, and makes its goal. With the help of it, we can improve the ontology construction methods, specially on the open semantics by asking the domain expert to get the hierarchical relationship between attributes and objects, and find out all of the knowledge in the domain. It is possible to share and reuse knowledge after the ontology has built and some new knowledge is added easily.

In this paper, we firstly introduce the attribute exploration algorithm, proposed applying formal concept analysis to description logic by Baader, etc. during the past decade. After studying on it, we find out some redundant computation. We propose an improved attribute exploration algorithm in terms of relevancy simplify the computing, and prove its completeness, finally propose an ontology construction method based on attribute exploration algorithm(AEOCM), and specify the implementation procedure of this method in terms of instantiation.

2 Basic definitions in formal concept analysis

Formal concept analysis (FCA) is a field of applied mathematics that aims to formalize the notions of a concept and a conceptual hierarchy by means of mathematical tools. It facilitates the use of mathematical reasoning for conceptual data analysis and knowledge processing. There are some basic definitions and theorems used in the following.

Definition 2.1^[4,6] (Formal context) A formal context is a triple $\dots = (\dot{\cdot}, ||, \triangleleft)$, where $\dot{\cdot}$ is a set of objects, $||$ is a set of attributes, and $\triangleleft \subseteq \dot{\cdot} \times ||$ is a relation that associates each object o with the attributes satisfied by o .

Definition 2.2^[4,6] (Derivation operator) Let $\dots = (\dot{\cdot}, ||, \triangleleft)$ be a formal context. For a set of objects $A \subseteq \dot{\cdot}$, we define the intent $A^{\mathbb{Z}}$ of A as follows: $A^{\mathbb{Z}} := \{ p \in || \mid \square a \in A. (a, p) \in \triangleleft \}$.

Similarly, for a set of attributes $B \subseteq ||$, we define the extent $B^{\mathbb{Z}} := \{ o \in \dot{\cdot} \mid \square b \in B. (o, b) \in \triangleleft \}$.

Definition 2.3^[6] (Formal concept) Let $\dots = (\dot{\cdot}, ||, \triangleleft)$ be a formal context. A formal concept of \dots is a pair (A, B) where $A \subseteq \dot{\cdot}$, $B \subseteq ||$ such that $A^{\mathbb{Z}} = B$ and $B^{\mathbb{Z}} = A$. We call A the extent, and B the intent of (A, B) .

Definition 2.4^[7] (Subconcept-superconcept-relation) The subconcept-superconcept-relation is mathematized by $(A_1, B_1) \subseteq (A_2, B_2) \iff A_1 \subseteq A_2 \wedge B_1 \subseteq B_2$.

The set of all formal concepts of K together with the defined order relation is denoted by $\hat{=}(K)$.

Definition 2.5^[7] (Concept lattices) Let $\dots = (\cdot, \cdot, \cdot, \cdot)$ be a formal context. Then $\hat{=}$ (...) is a complete lattice, called the concept lattice of $(\cdot, \cdot, \cdot, \cdot)$, for which infimum and supremum can be described as follows:

$$\bigwedge_{i \in I} (A_i, B_i) = (\bigcap_{i \in I} A_i, (\bigcup_{i \in I} B_i)'), \quad \bigvee_{i \in I} (A_i, B_i) = ((\bigcup_{i \in I} A_i)', \bigcap_{i \in I} B_i).$$

Definition 2.6^[6] (Implication between attributes) An implication $L \parallel R$ holds in ... if every object of ... that has all of the attributes in L also has all of the attributes in R , i.e., if $L^{\mathbb{Z}} \subseteq R^{\mathbb{Z}}$. We denote the set of all implication that hold in ... by $Imp(\dots)$.

Definition 2.7^[8] (Pseudo-intent) $P \sqsubseteq M$ is called the pseudo-intent of $(\cdot, \cdot, \cdot, \cdot)$ if and only if $P \sqsubseteq P^{\mathbb{Z}}$ and $Q^{\mathbb{Z}} \subseteq P$ holds for every pseudo-intent $Q \sqsubseteq P$.

Definition 2.8^[8] (Duquenne-Guigues-Base) The set of implications $\{P \parallel P^{\mathbb{Z}} \rightarrow P \mid P \text{ pseudo-intent}\}$. We call this the Duquenne-Guigues-Basis or simply the stem base of the attribute implications.

Theorem 1^[8] The set of implications $\dots := \{P \parallel P^{\mathbb{Z}} \rightarrow P \mid P \text{ pseudo-intent}\}$ is non-redundant and complete.

Definition 2.9^[4,8] (Lectic order) Let an arbitrary linear order on the set of attributes $\parallel = \{p_1, \dots, p_n\}$, say $p_1 < \dots < p_n$. For all j , $1 \leq j \leq n$, and $B_1, B_2 \sqsubseteq \parallel$ we define $B_1 < B_2$ iff $p_j \sqsubseteq B_2 \rightarrow B_1$ and $B_1 \sqsubseteq \{p_1, \dots, p_{j-1}\} = B_2 \sqsubseteq \{p_1, \dots, p_{j-1}\}$.

Theorem 2 For a given attribute set $\parallel = \{p_1, p_2, \dots, p_n\}$, which basic order is $p_1 < p_2 < \dots < p_n$ and $B \sqsubseteq P$, we denote the first element after B on lectic order by B^+ . It satisfies:

(i) existing an positive j is maximal such that $p_j \sqsubseteq B$;

(ii) $B^+ = B \sqcup \{p_1, p_2, \dots, p_{j-1}\} \sqsubseteq \{p_j\}$.

Definition 2.10^[4] (Implication Pseudo-hull) For a subset B of \parallel , the implication pseudo-hull of B with respect to \mathfrak{U} is denoted by $\mathfrak{U}^*(B)$. It is the smallest subset H of \parallel such that

(i) $B \sqsubseteq H$;

(ii) $B_1 \parallel B_2 \sqsubseteq \mathfrak{U}$ and $B_1 \sqsubseteq H$ (strict subset) imply $B_2 \sqsubseteq H$.

Definition 2.11^[8] (respect) A subset $T \sqsubseteq M$ respects an implication $A \parallel B$ if $A \sqsubseteq T$ or $B \sqsubseteq T$. T respect a set \dots of implication if T respect every single implication in \dots .

Definition 2.12 In the concept lattice, the length of maximal chain from concept (A, B) to maximum is \cdot, \cdot , (A, B) is the \cdot, \cdot level node of lattice.

3 Ontology and its construction by the way of FCA

Ontologies, often defined as an explicit specification of conceptualization, are necessary for knowledge representation and knowledge exchange. Generally ontology can be formally defined by (C, P, I, S, E) , where C refers to Class; P refers to property of Class; I refers to instance of Class; S refers to subsumption relation and E refers to other Enriched relation. In the concept lattice, the concept can be mapped to class in the ontology definition; the elements of objects in each concept can be mapped to the instance of ontology; the element of attributes will be mapped to the property of ontology and finally subsuper concept relation is equivalent to subsumption relation in ontology.^[9]

A good design ontology means that they should adequately capture the modeled domain, be understandable for human user and provide good support for machine processing. By a good definition means not only the syntax, but also the semantics.

An automated reasoning over ontologies enables to support the ontology design, integrating and sharing ontologies automatically, determining and establishing relationships among ontologies etc.^[10]

Formal Concept Analysis (FCA) is adopted in the ontology construction process to establish the taxonomic hierarchy correctly and properly, because it yields the mathematization of concepts and component ontology are explained briefly. The way to applying the formal concept analysis to ontology construction, first of all, we should model context in a research domain by topical similarity and subsumption which should expand user's search context and improve the interactive capability of traditional search engines, and develop information context. In the concept lattice, a concept will be described by a set of attributes. In this case, we can get a formal context which is a crucial step. Then we use attribute exploration based on formal context and with the help of the domain expert to construct ontology on the concept lattice and get Duquenne-Guigues Base on knowledge inference.

4 Attribute exploration algorithm

Attribute exploration is a method of acquiring knowledge from a domain expert by asking successive questions to expand ABox and TBox. In many application fields where the formal context is not explicitly given, but only "known" to some domain expert, it has proved to be a successful method for efficiently capturing the expert's knowledge. The superiority of attribute exploration is not only computing intents, pseudo-intents and Duquenne-Guigues Base, but also expanding counterexamples to it by domain expert to acquire complete knowledge about the application domain instead of being restricted by the given formal context.

4.1 Attribute exploration algorithm proposed by Baader etc.

For a given formal context, implications between attributes can express the important knowledge of subsumption between object sets. In some cases, especially impossible to list all of the subsumption, we try to compute the base of these subsumption. Using it, we can compute all of intents and stem base, which can inference all of the subsumption. The attribution exploration algorithm proposed by Baader etc. is as follows:

Algorithm 1(Attribution exploration)^[4]

Initialization: One starts with the empty set of implication, i.e., $\mathfrak{O}_0 := \square$, the empty set of concept intents $\mathbb{I}_0 := \square$, and the empty subcontext \dots_0 of ..., i.e., $\dot{\cdot}_0 := \square$. The lectic smallest subset of \mathbb{I} is $B_0 := \square$.

Iteration: Assume that \dots_i , \mathfrak{O}_i , \mathbb{I}_i , and $B_i (i \geq 0)$ are already computed. Compute B_i'' with respect to the current subcontext \dots_i . Now the expert is asked whether the implication $B_i \parallel B_i'' \Rightarrow B_i$ holds in ...

If the answer is "no", then let $o_i \sqsubset \dot{\cdot}_i$ be the counterexample provided by the expert. Let $B_{i+1} := B_i$, $\mathfrak{O}_{i+1} := \mathfrak{O}_i$, and let \dots_{i+1} be the subcontext of ... with $\dot{\cdot}_{i+1} := \dot{\cdot}_i \sqcup \{o_i\}$. The iteration continues with \dots_{i+1} , \mathfrak{O}_{i+1} , \mathbb{I}_{i+1} , and B_{i+1} .

If the answer is "yes", then $\dots_{i+1} := \dots_i$ and

$$\left\{ \begin{array}{l} (\mathbb{I}_i, \mathfrak{O}_i \sqcup \{ B_i \parallel B_i'' \Rightarrow B_i \}) \quad \text{if } B_i'' \sqsubset B_i \\ \end{array} \right.$$

$$(\coprod_i \square \{B_i\}, \mathfrak{U}_i) \quad \text{if } B_i'' = B_i$$

To find the new set B_{i+1} , we starts with $j=n$, and test whether

$$B_i <_j \mathfrak{U}_{i+1}^* ((B_i \square \{p_1, \dots, p_{j-1}\}) \square \{p_j\}) \quad (*)$$

holds. The index j is decreased until one of the following cases occurs:

(1) $j=0$: In this case, \coprod_{i+1} is the set of all concept intents and \mathfrak{U}_{i+1} the Duquenne-Guigues base of ..., and the algorithm stops.

(2) $(*)$ holds for $j>0$: In this case, $B_{i+1} := \mathfrak{U}_{i+1}^* ((B_i \square \{p_1, \dots, p_{j-1}\}) \square \{p_j\})$, and the iteration is continued.

By computing on this algorithm, we find out that there are some redundancies in the iterative steps. This redundancy happens in computing formula(*). When computing B_{i+1} , we work out the implication pseudo-hull, i.e. $\mathfrak{U}_{i+1}^* ((B_i \square \{p_1, \dots, p_{j-1}\}) \square \{p_j\})$ and lectic order $<_j$. It appear this case that the same set of implication pseudo-hull is computed repeatedly when j don't satisfy formula(*) and then $j:=j-1$. For avoiding the complex computation repeatedly, we try to improve the efficiency on the following.

4.2 the improved attribute exploration algorithm(I)

It is evident that computation the implication pseudo-hull and lectic order is rather complex. In order to resolve this problem, we can modify the way to compute B_{i+1} in terms of relevancy between attribute sets and implication sets, i.e. asking that does B_i^+ (the next element of B_i on lectic order) respect an implication \mathfrak{U}_{i+1} , if the answer is "Yes", let $B_{i+1} := B_i^+$, the iteration is continued; if the answer is "No", let $B_i := B_i^+$, continue to compute the next B_i^+ . Modify the improved algorithm as follows:

Table 1 Improved Attribute Exploration Algorithm

Algorithm 2 Improved attribute exploration	
1	Input: A formal context $\dots = (\cdot, \cdot, \cdot, \cdot)$
2	Initialization
3	$\mathfrak{U}_0 := \square$ {initial empty set of implications}
4	$\coprod_0 := \square$ {initial empty set of intents}
5	$\dots_0 := \square$ {initial formal context, possibly empty set of objects}
6	$\cdot_0 := \square$ {initial empty set of objects}
7	$B_0 := \square$ {initial empty set of lectic order}
8	while $B_i \neq \cdot$ do {assume $\dots, \mathfrak{U}_i, \coprod_i, B_i (i \geq 0)$ are already computed}
9	Compute B_i'' w.r.t. \dots_i
1	Ask the expert if $B_i \rightarrow B_i'' \setminus B_i$ holds w.r.t. \dots ?
1	if no then the expert provides a counterexample $o_i \in \cdot$.
1	$\cdot_{i+1} := \cdot_i \cup \{o_i\}$
1	$B_{i+1} := B_i$
1	$\mathfrak{U}_{i+1} := \mathfrak{U}_i$
1	$\coprod_{i+1} := \coprod_i$
1	$\dots_{i+1} \{ \dots_i = (\cdot_i, \cdot_i, \cdot_i, \cdot_i), \cdot_i \subseteq \cdot_i, \cdot_i = \cdot_i, \cdot_i = \cdot_i \square (\cdot_i \times \cdot_i); \dots_{i+1} \text{ is a subcontext of } \dots \}$
1	else {yes}
1	$\dots_{i+1} := \dots_i$
1	if $B_i'' \neq B_i$ then { B_i is a pseudo-intent}

```

2      ( $\Pi_{i+1}, \mathfrak{U}_{i+1}$ ) := ( $\Pi_i, \mathfrak{U}_i \cup \{B_i \rightarrow B_i'' \setminus B_i\}$ )
~ 2      else  $\{B_i'' = B_i, B_i \text{ is a intent}\}$ 
2      ( $\Pi_{i+1}, \mathfrak{U}_{i+1}$ ) := ( $\Pi_i \cup \{B_i\}, \mathfrak{U}_i$ )
~ 2      end if
~ 2      Compute  $B_{i+1}$ 
~ 2      while  $B_i \neq \parallel$  then
~ 2      Does  $B_i^+$  respect an implication  $\mathfrak{U}_{i+1}$ ?
~ 2      if yes then
~ 2       $B_{i+1} := B_i^+$ 
~ 2      break
~ 3      else  $\{no\}$ 
~ 3       $B_i := B_i^+$ 
~ 3      end if
~ 3      end while
~ 3      end if
~ 3      end while
~ 3      Output:  $\Pi_i, \mathfrak{U}_i$ 

```

Theorem3 The improved attribute exploration algorithm is complete.

Proof: According to the algorithm, we can obtain the set $\{B_0 = \square, B_1, B_2, \dots, B_m = \parallel\}$, and $B_0 < B_1 < B_2 < \dots < B_m$.

(1) **Initialization:** Take $B_0 = \square$ as beginning, classified discussion as follows:

If $\square'' = A_0(A_0 \square \square)$, then \square is a pseudo-intent, $\mathfrak{U}_1 = \{\square \parallel A_0\}$. The first element respecting with \mathfrak{U}_1 after \square is A_0 , so $B_1 = A_0$ is intent. On the other hand, as \square is a pseudo-intent, any non-empty pseudo-intent must include A_0 , similarly, A_0 must be included in any non-empty intent. It's known that there is any element neither intent nor pseudo-intent.

If $\square'' = \square$, \square is a intent ($B_0 = \square$), $\mathfrak{U}_1 = \square$, the first element after \square is $\{p_n\}$, $\{p_n\}$ respects \mathfrak{U}_1 , $B_1 = \{p_n\}$. As $\{p_n\}$ is the next element of \square on the lectic order. there is not any element between B_0 and B_1 . Now we only have to prove B_1 is intent or pseudo-intent: if $\{p_n\}'' = \{p_n\}$, $\{p_n\}$ is intent; if $\{p_n\}'' \square \{p_n\}$, $\{p_n\}$ is pseudo-intent.

Above all, B_0 and B_1 is intent or pseudo-intent. Moreover there is no other intent or pseudo-intent between B_0 and B_1 .

(2) **Inductive argument:** On the assumption that all of the intents and pseudo-intents less than or up to B_i on lectic order have been computed, denoting that B_1, B_2, \dots, B_i and ensure there is no intent or pseudo-intent between adjacent elements. Now $\mathfrak{U}_{i+1} = \{P \parallel P'' \nrightarrow P \mid P \text{ is pseudo-intent}, P \leq B_i\}$. According to the algorithm, testing from the first element after B_i on lectic order one by one, until acquiring the first element B respecting \mathfrak{U}_{i+1} , let $B_{i+1} = B$, now we should prove that : (i) B_{i+1} is intent or pseudo-intent; (ii) There is no intent or pseudo-intent between B_i and B_{i+1} .

Case 1: If $B = B_i^+$, i.e. B_i^+ , the first element after B_i on lectic order, respects \mathfrak{U}_{i+1} .

Considering pseudo-intent Q , satisfying $Q \square B_i^+$, we know that \mathfrak{U}_{i+1} includes the implication $Q \parallel Q'' \nrightarrow Q$ according to the inductive assumption above, i.e. $Q = B_k$, $k \leq i$, B_i^+ respects \mathfrak{U}_{i+1} . It means that $Q \square B_i^+$ imply $Q'' \nrightarrow Q \square B_i^+$, i.e. $Q'' \square B_i^+$. Now if $(B_i^+)'' = B_i^+$, B_i^+ is intent; if $(B_i^+)'' \square B_i^+$, B_i^+ is pseudo-intent. So (i) and (ii) hold.

Attribute Exploration Algorithms on Ontology Construction * 7

Case 2: If $B \sqsubseteq B_i$, i.e. B_i doesn't respect \mathcal{O}_{i+1}

We should prove B_i is neither intent nor pseudo-intent. Using reduction to absurdity, considering pseudo-intent Q , satisfying $Q \sqsubseteq B_i$, we know that \mathcal{O}_{i+1} includes the implication $Q \sqsubseteq Q'' \Rightarrow Q$ according to the inductive assumption above, i.e. $Q = B_k$, $k \leq i$, B_i respects \mathcal{O}_{i+1} . It is evident that it contradicts with the assumption that B_i doesn't respect \mathcal{O}_{i+1} . So B_i is neither intent nor pseudo-intent.

Now testing the next element of B_i , B_i either doesn't respect \mathcal{O}_{i+1} , or is neither intent nor pseudo-intent. There is a minimal element B on lexic order, respecting \mathcal{O}_{i+1} . Now let $B_{i+1} = B$, there is no intent or pseudo-intent between B_i and B_{i+1} . We should prove B_{i+1} is intent or pseudo-intent, considering pseudo-intent A , satisfying $A \sqsubseteq B_{i+1}$, i.e. $A \leq B_{i+1}$. We know that and \mathcal{O}_{i+1} includes the implication $A \sqsubseteq A'' \Rightarrow A$ according to the inductive assumption above, i.e. $A = B_l$, $l \leq i$, B_{i+1} respects \mathcal{O}_{i+1} . It means that $A \sqsubseteq B_{i+1}$ imply $A'' \sqsubseteq A \sqsubseteq B_{i+1}$, i.e. $A'' \sqsubseteq B_{i+1}$. Now if $(B_{i+1})'' = B_{i+1}$, B_{i+1} is intent; if $(B_{i+1})'' \sqsubseteq B_{i+1}$, B_{i+1} is pseudo-intent. So (i) and (ii) hold.

Therefore, the improved attribute exploration algorithm is complete. \square

4.3 Analysis and comparison of the two attribute exploration algorithms

Comparing with the two different algorithms, we can make this conclusion that the improved attribute exploration algorithm neither needn't compute implication pseudo-hull, nor judge whether the computed pseudo-hull satisfy formula(*) based on the definition of lexic order, but judges the next attribute set on lexic order from the view of respect to acquire the next intent or pseudo-intent. The computation with respect, implementing on the way of subsumption between sets, effectively simplify the complexity. It also avoids redundancy computation the same set repeatedly, which sometimes happened in computing the next B_{i+1} in the attribute exploration algorithm proposed by Baader etc. We also prove the improved one is complete.

5 Ontology construction based on attribute exploration algorithm

According to the results of research, we propose a model of the method of ontology construction based on attribute exploration algorithm (AEOCM). This model includes four modules: preprocessing module, attribute exploration module, generating concept lattice module. The figure of this model is as following Fig. 1.

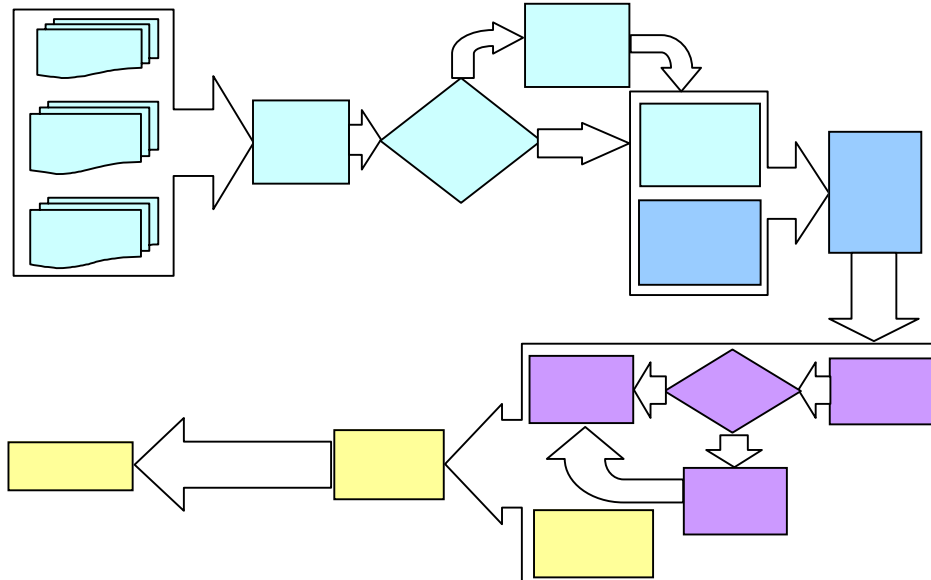
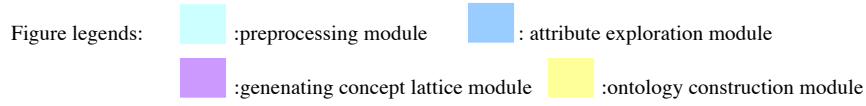


Fig. 1. the model of ontology construction based on attribute exploration algorithm (AEOCM)



5.1 Preprocessing module

In practice, formal context is generally not given, but extracted from data sources. As a result, we need to analysis data resources and resort to respective strategies and algorithms for different data resources to extract formal context. The main processes of preprocessing module are extracting the multi-value formal context which contains attributes and objects from data source files including text files, relational databases, XML data types and so on, and transforming multi-value formal context to single one. The preprocessing module includes two submodules: the submodule of extracting formal context, the submodule of transforming formal context .

5.2 Attribute exploration module

The main processes of attribute exploration module are computing intent, pseudo-intent, Duquenne- Guigues base by the ways of the improved attribute exploration algorithm after asking the expert successive questions, in order to generate formal concept lattice. The way of this module to distinguish with other algorithms computing intent and pseudo-intent, is asking the expert whether the implication holds in the given context, if the answer is “yes”, then doing the next computing in the formal context \dots_i ; if the answer is “no”, then providing counterexample to \dots_i . In this case, it ensures that the new objects are dynamically added to the formal context and what is never expressed explicitly but on the experts’ mind, making the experts expand the knowledge base easily.

5.3 Generating concept lattice module

This module is mainly to generate concept lattice by the structure concept lattice algorithm according to the computed intents and draw the respective hasse graph. The up to down structure concept lattice algorithm is as follows:

Input: All the formal concept $\hat{=}(...)$.

Initialization: $\square(X,Y)\square\hat{=}(...)$ access to the set $C_{|Y|}$ according to the size of $|Y|$, then we will get the sets, such as $C_0, C_1, \dots, C_{|P|}$, $C_0 = (||, ||\mathbb{Z})$ is maximum.

Iteration: Assume $C_k = (A_k, B_k)$ is already computed.

Start from the k ($k \geq 0$) level, compute as following method:

$\square(X,Y)\square C_k$, if finding (X_I, Y_I) in C_{k+1} , satisfy $Y \square Y_I$, connect (X,Y) with (X_I, Y_I) ; if finding (X_2, Y_2) in C_{k-i} ($1 \leq i < k$), satisfy $Y_2 \square Y$ and $X_2 < X$, connect (X,Y) with (X_2, Y_2) . The iteration is continued.

Until $k = |||$, the algorithm stops.

Output: Concept lattice $\hat{=}(...), \leq, \geq$.

5.4 Ontology construction module

This module mainly contains constructing an ontology from above concept lattice according to the ontology generating rules, and adding other properties and restrictions practically, finally generating the ultimate ontology.

The respective ontology generating rules are as follows:

Table 2: The respective ontology generating rules

Lattice	Description Logic	Ontology
Attributes	Concepts	Classes
Objects	Instances	Individuals
Hierarchy of lattice	Subsumptions and equivalences between concepts	Hierarchy between classes and individuals
Can't be gotten directly	Hierarchy among Roles	The relations among Object Properties
Can't be gotten directly	Number restriction operations	Data Properties define datatype

From the table1, we can make a conclusion that the classes, individuals and their hiberarchy can be found in the lattice directly, however, object properties and data properties are added manually by the constructor.

6 Conclusion

In recent years, the automatic methods on ontology construction become a research hotspot. Attribute exploration in FCA is an efficient tool to generate concept lattice and construct ontology, and assure to share and reuse knowledge. Through asking a domain expert question, we can add new knowledge to generate the concept lattice structure on concept hierarchy to construct ontology. In this paper, we firstly have investigated the existing attribute exploration algorithm and find its redundancy computation, then have proposed an improved attribute exploration algorithm to simplify the complexity effectively. Applying this algorithm to ontology construction, we also have proposed a model of ontology construction based on attribute

exploration algorithm (AEOCM), and elaborate the whole implement procedures in detail according to each module. Our experience in the research has convinced us that the study in this direction is indeed significant and should have a good prospect both in the application and the theory. We will continue our works to find more efficient algorithms and try to implement it in practice to construct ontology and its automatic inference.

References

1. Gerd Stumme. Attribute exploration with background implications and exceptions[C]//In Hans-Herrmann Bock, Wolfgang Polasek, editors, Data analysis and information systems. Proceedings of the 19th annual conference of Gesellschaft für Klassifikation e.v. University of Basel, March 8-10, 1995. Berlin, Heidelberg: Springer, 1996:457-469.
2. Franz Baader, Bernhard Ganter, Ulrike Sattler, et al. Completing Description Logic Knowledge Bases using Formal Concept Analysis, LTCS-Report 06-02[R]. Germany, Dresden University of Technology, 2006.
3. F. Baader, Baris Sertkaya, Anni-Yasmin Turhan. Computing the least common subsumer w.r.t. a background terminology[J]. Journal of Applied Logic, 5 (2007):392-420.
4. F. Baader, B. Sertkaya. Applying formal concept analysis to description logics[C]//In P. Eklund, editor. Computer Science. Proceedings of the 2nd International Conference on Formal Concept Analysis (ICFCA), Sydney, Berlin- Heidelberg:Springer-Verlag, 2004, 2961: 261- 286.
5. F. Baader. Computing a minimal representation of the subsumption lattice of all conjunctions of concepts defined in a terminology[J]. Proc. Intl. KRUSE Symposium, 1995, 168-178.
6. Baris Sertkaya. Formal Concept Analysis Methods for Description Logics(D)
7. Rudolf Wille. Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies [J]. Berlin- Heidelberg:Springer-Verlag, 2005, 3626:1-33.
8. B. Ganter, R. Wille. Formal Concept Analysis: Mathematical Foundations[M]. Berlin: Springer-Verlag, 1999:79-90.
9. Haibo Jia, Julian Newman, Huaglory Tianfield. A new Formal Concept Analysis based learning approach to Ontology building[J]. <http://www.mtsr.ionio.gr/proceedings/haibo.pdf>
10. Marek Obitko, Vaclav Snasel, Jan Smid. Ontology Design with Formal Concept Analysis[J]. CLA 2004, 111-119, 2004
11. Zhang wenxiu, Wei ling, Qi jianjun. The attribute reduction theory and approach of concept lattice[J]. SCIENCE IN CHINA ser.E Information Sciences, 2005, 35 (6):628-639.