# Attribute Exploration Algorithms on Ontology Construction

Qin Ping, Zhang Zhongxiang, Gao Hualing, Wang Ju

## HAL Id: hal-01055073
## https://hal.inria.fr/hal-01055073

Submitted on 11 Aug 2014

# Attribute Exploration Algorithms on Ontology Construction [*]

Qin Ping[1,2], Zhang Zhongxiang[1], Gao Hualing[1], Wang Ju[1]

[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 implement-
tation 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

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 revelancy 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 $\mathbb{K}=(\mathcal{O},\mathcal{A},\mathcal{R})$,where $\mathcal{O}$ is a set of objects, $\mathcal{A}$ is a set of attributes, and $\mathcal{R}\subseteq\mathcal{O}\times\mathcal{A}$ is a relation that associates each object o with the attributes satisfied by o.

**Definition 2.2**[4,6] (Derivation operator) Let $\mathbb{K}=(\mathcal{O},\mathcal{A},\mathcal{R})$ be a formal context. For a set of objects $A\subseteq\mathcal{O}$, we define the intent $A^{\mathbb{Z}}$ of $A$ as follows: $A^{\mathbb{Z}}:=\{ p\in\mathcal{A}|\forall a\in A.(a,p)\in\mathcal{R}\}$.

Similarly, for a set of attributes $B\subseteq\mathcal{A}$, we define the extent $B^{\mathbb{Z}}:=\{o\in\mathcal{O}.|\forall b\in B.(o,b)\in\mathcal{R}\}$.

**Definition 2.3**[6] (Formal concept) Let $\mathbb{K}=(\mathcal{O},\mathcal{A},\mathcal{R})$ be a formal context. A formal concept of $\mathbb{K}$ is a pair $(A,B)$ where $A\subseteq\mathcal{O}$, $B\subseteq\mathcal{A}$ 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 $(A_1,B_1)\mathbb{C}(A_2,B_2):\Leftrightarrow A_1\subseteq A_2(B_1\supseteq B_2)$.

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

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

$$\bigwedge_{t\in T}(A_t,B_t)=(\coprod_{t\in T} A_t,(\bigcup_{t\in T} B_t)''),\quad \bigvee_{t\in T}(A_t,B_t)=((\bigcup_{t\in T} A_t)'',\coprod_{t\in T} B_t).$$

**Definition 2.6**[6] (Implication between attributes) An implication $L\|R$ 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 $L^{\mathbb{Z}}\square R^{\mathbb{Z}}$. We denote the set of all implication that hold in $…$ by $Imp(…)$.

**Definition 2.7**[8] (Pseudo-intent) $P\square M$ is called the pseudo-intent of $(\ddot{\cdot}.,\|,\prec)$ if and only if $P\square P^{\mathbb{ZZ}}$ and $Q^{\mathbb{ZZ}}\square P$ holds for every pseudo-intent $Q\square P$, $Q\square P$.

**Definition 2.8**[8] (Duquenne-Guigues-Base) The set of implications $\{P\|P^{\mathbb{Z}}\rightleftharpoons P|P$ 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 $…:=\{P\|P^{\mathbb{Z}}\rightleftharpoons P|P$ pseudo-intent$\}$is non-redundant and complete.

**Definition 2.9**[4,8] (Lectic order) Let an arbitrary linear order on the set of attributes $\|=\{p_1,\cdots,p_n\}$, say $p_1<\cdots< p_n$. For all $j$, $1\leqslant j\leqslant n$, and $B_1,B_2\square\|$ we define $B_1<_j B_2$ iff $p_j\square B_2\rightleftharpoons B_1$ and $B_1\square\{ p_1,\cdots,p_{j-1}\}=B_2\square\{ p_1,\cdots,p_{j-1}\}$.

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

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

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

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

   (i) $B\square H$;

   (ii) $B_1\|B_2\square℧$ and $B_1\square H$ (strict subset) imply $B_2\square H$.

**Definition 2.11**[8] (respect) A subset $T\square M$ respects an implication $A\|B$ if $A\square T$ or $B\square T$. T respect a set $\cdots$ of implication if T respect every single implication in $\cdots$.

**Definition 2.12** In the concept lattice, the length of maximal chain from concept $(A,B)$ to maximum is $\dot{\cdot}\dot{\cdot}$, $(A,B)$ is the $\dot{\cdot}\dot{\cdot}$ 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.

$(\coprod_{i+1}, \mho_{i+1}):=$

An automated reasoning over ontologies enables to support the ontology design, intergrating 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., $\mho_0:=\square$, the empty set of concept intents $\coprod_0:=\square$, and the empty subcontext $\ldots_0$ of $\ldots$, i.e., $\therefore_0:=\square$. The lectic smallest subset of $\|$ is $B_0:=\square$.

**Iteration:** Assume that $\ldots_i$, $\mho_i$, $\coprod_i$, and $B_i(i \geqslant 0)$ are already computed. Compute $B_i''$ with respect to the current subcontext $\ldots_i$. Now the expert is asked whether the implication $B_i\| B_i'' \rightleftharpoons B_i$ holds in $\ldots$.

If the answer is "no", then let $o_i\square^{\therefore}$ be the counterexample provided by the expert. Let $B_{i+1}:= B_i$, $\mho_{i+1}:=\mho_i$, and let $\ldots_{i+1}$ be the subcontext of $\ldots$ with $\therefore_{i+1}:= \therefore_i \square\{o_i\}$. The iteration continues with $\ldots_{i+1}, \mho_{i+1}, \coprod_{i+1}$, and $B_{i+1}$.

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

$$\left\{ (\coprod_i, \mho_i\square\{ B_i\| B_i'' \rightleftharpoons B_i \}) \quad \text{if } B_i'' \square B_i \right.$$

$$(\coprod_i \square \{B_i\}, \mho_i) \qquad\qquad \text{if } B_i''=B_i$$

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

$$B_i <_j \mho^*{}_{i+1}((B_i \square \{ p_1, \cdots p_{j\text{-}1}\}) \square \{ p_j\}) \qquad\qquad (*)$$

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 $\mho_{i+1}$ the Duquenne-Guigues base of …, and the algorithm stops.

(2) $(*)$ holds for $j>0$: In this case, $B_{i+1}:=\mho^*{}_{i+1}((B_i \square \{ p_1, \cdots p_{j\text{-}1}\}) \square \{ p_j\})$, 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. $\mho^*{}_{i+1}((B_i \square \{p_1, \cdots p_{j\text{-}1}\}) \square \{p_j\})$ 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 $j:=j\text{-}1$. 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 $\mho_{i+1}$, 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 …=(∵.,‖,≼)
2     Initialization
3        𝕆₀:=□                    {initial empty set of implications}
4        ∐₀:=□                    {initial empty set of intents}
5        …₀:=□                    {initial formal context, possibly empty set of objects}
6        ∵.₀:=□                   {initial empty set of objects}
7        B₀:=□                    {initial empty set of lectic order}
8     while Bᵢ≠‖ do              {assume …ᵢ, 𝕆ᵢ, ∐ᵢ, Bᵢ(i≥0) are already computed}
9        Compute Bᵢ″ w.r.t. …ᵢ
10       Ask the expert if Bᵢ→Bᵢ″\Bᵢ holds w.r.t … ?
11       if no then the expert provides a counterexample oᵢ∈∵.
12          ∵.ᵢ₊₁:=∵.ᵢ∪{oᵢ}
13          Bᵢ₊₁:=Bᵢ
14          𝕆ᵢ₊₁:=𝕆ᵢ
15          ∐ᵢ₊₁:=∐ᵢ
16          …ᵢ₊₁ {…ᵢ=(∵.ᵢ,‖ᵢ,≼ᵢ) ,∵.ᵢ⊆∵.,‖ᵢ=‖,≼ᵢ=≼□(∵.ᵢ×‖ᵢ); …ᵢ₊₁ is a subcontext of …}
17       else                     {yes}
18          …ᵢ₊₁:=…ᵢ
19          if Bᵢ″≠Bᵢ then         {Bᵢ is a pseudo-intent}
```

```
              (∐_{i+1}, ℧_{i+1}):= (∐_i, ℧_i∪{B_i→B_i''\B_i})
          else                        {B_i''=B_i , B_i is a intent}
              (∐_{i+1}, ℧_{i+1}):= (∐_i∪{B_i}, ℧_i)
          end if
          Compute B_{i+1}
      while B_i≠‖ then
          Does B_i^+ respect an implication ℧_{i+1}?
          if yes then
              B_{i+1}:=B_i^+
              break
          else                         {no}
              B_i:=B_i^+
          end if
        end while
      end if
    end while
  Output: ∐_i, ℧_i
```

**Theorem3** The improved attribute exploration algorithm is complete.

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

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

If $\square''=A_0(A_0\square\square)$, then $\square$ is a pesudo-intent, $℧_1=\{\square\|A_0\}$. The first element respecting with $℧_1$ after $\square$ is $A_0$, so $B_1=A_0$ is intent. On the other hand, as $\square$ is a pesudo-intent, any non-empty pesudo-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 pesudo-intent.

If $\square''=\square$, $\square$ is a intent ($B_0=\square$), $℧_1=\square$, the first element after $\square$ is $\{p_n\}$, $\{p_n\}$ respects $℧_1$, $B_1=\{p_n\}$. 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 pesudo-intent: if $\{p_n\}''=\{p_n\}$, $\{p_n\}$ is intent; if $\{p_n\}''\square\{p_n\}$, $\{p_n\}$ is pesudo-intent.

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

**(2) Inductive argument**: On the assumption that all of the intents and pesudo-intents less than or up to $B_i$ on lectic order have been computed, denoting that $B_1, B_2, \cdots, B_i$ and ensure there is no intent or pesudo-intent between adjacent elements. Now $℧_{i+1}=\{P\|P''\rightleftarrows P|P$ is pesudo-intent, $P\leqslant B_i\}$. 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 $℧_{i+1}$, let $B_{i+1}=B$, now we should prove that : (i) $B_{i+1}$ is intent or pesudo-intent; (ii)There is no intent or pesudo-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 $℧_{i+1}$.

Considering pesudo-intent $Q$, satisfying $Q\square B_i^+$, we know that $℧_{i+1}$ includes the implication $Q\|Q''\rightleftarrows Q$ according to the inductive assumption above, i.e. $Q = B_k, k\leqslant i$, $B_i^+$ respects $℧_{i+1}$. It means that $Q\square B_i^+$ imply $Q''\rightleftarrows Q\square B_i^+$, 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 pesudo-intent. So (i) and (ii) hold.

Text file

Relational database    Extract    Generate formal context    Is it a multivalue formal context?    No    Former formal context    Compute

Transform multivalue formal context    Yes

Intent, pesudo-intent, D&G base

XML data type    Attribute exploration algorithm

Ontology    add,modify manually    Logical concept lattice    Is it    Initial concept lattice

Former ontology    Yes    No

Generate concept lattice

Compile

Ontology transform rules

Case 2: If $B \square B_i^+$, $B_i^+$ doesn't respect $\mho_{i+1}$

We should prove $B_i^+$ is neither intent nor pesudo-intent. Using reduction to absurdity, considering pesudo-intent $Q$, satisfying $Q \square B_i^+$, we know that $\mho_{i+1}$ includes the implication $Q \| Q'' \rightleftarrows Q$ according to the inductive assumption above, i.e. $Q = B_k$, $k \leq i$, $B_i^+$ respects $\mho$. It is evident that it contradicts with the assumption that $B_i^+$ doesn't respect $\mho_{i+1}$. So $B_{i+1}$ is neither intent nor pesudo-intent.

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

Therefore, the improved attribute exploration algorithm is complete. ☐

### 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 pesudo-hull, nor judge whether the computed pesudo-hull satisfy formula(∗) based on the definition of lectic order, but judges the next attribute set on lectic order from the view of respect to acquire the next intent or pesudo-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, ontology construction module. The figure of this model is as following Fig. 1.
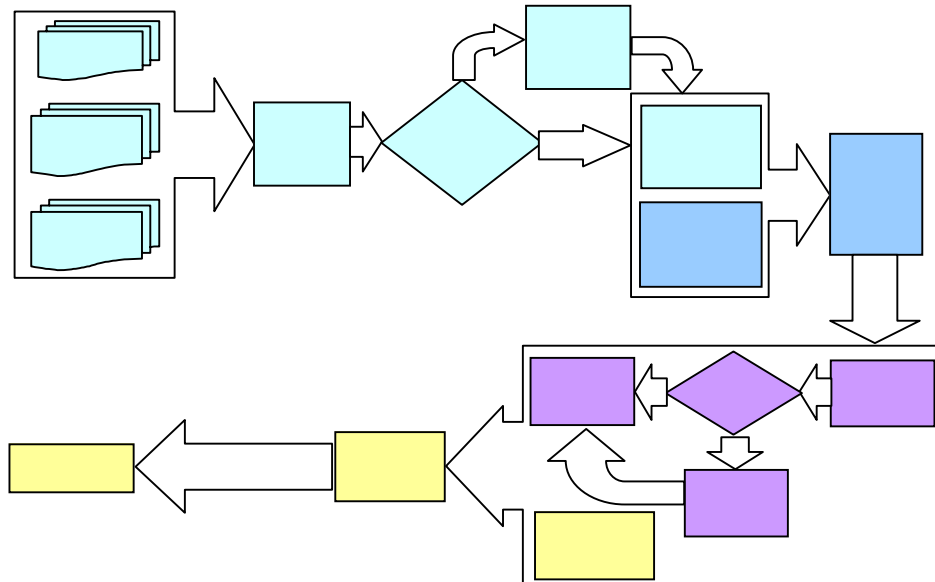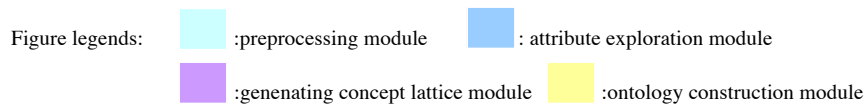
Fig. 1.  the model of ontology construction based on attribute exploration algorithm (AEOCM)

Figure legends:　　[　]　:preprocessing module　　　[　] : attribute exploration module

[　]　:genenating concept lattice module　　[　] :ontology construction module

## 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, pesudo-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 pesudo-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 …$_i$; if the answer is "no", then providing counterexample to …$_i$ . 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 $\doteq(\dots)$.

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

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

Start from the $k$ $(k\geqslant0)$ level, compute as following method:

$\square(X,Y)\square C_k$, if finding $(X_1,Y_1)$ in $C_{k+1}$, satisfy $Y\square Y_1$, connect $(X,Y)$ with $(X_1,Y_1)$; if finding $(X_2,Y_2)$ in $C_{k-i}(1\leqslant i<k)$, satisfy $Y_2\square Y$ and $X_2\prec X$, connect $(X,Y)$ with $(X_2,Y_2)$. The interation is continued.

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

**Output:** Concept lattice $<\doteq(\dots),\leqslant>$.

### 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 |
| Hiberarchy of lattice | Subsumptions and equivalences between concepts | Hiberarchy 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 fur Klassifikation e.v. University of Basel, March 8-10,1995.Berlin, Heidelberg: Springer,1996:457-469.
2.  Franz Baader, Bernhard Ganter, Ulrike Sattler, etal. 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.