

# Multi-level Rule Discovery from Propositional Knowledge Bases

Debbie Richards and Usama Malik

Computing Department,  
Division of Information and Communication Sciences,  
Macquarie University, Australia  
[richards@ics.mq.edu.au](mailto:richards@ics.mq.edu.au)

**Abstract.** This paper explores how knowledge in the form of propositions in an expert system can be used as input into data mining. The output is multi-level knowledge which can be used to provide structure, suggest interesting concepts, improve understanding and support querying of the original knowledge. Appropriate algorithms for mining knowledge must take into account the peculiar features of knowledge which distinguish it from data. The most obvious and problematic distinction is that only one of each rule exists. This paper introduces the possible benefits of mining knowledge and describes a technique for reorganizing knowledge and discovering higher-level concepts in the knowledge base. The rules input may have been acquired manually (we describe a simple technique known as Ripple Down Rules for this purpose) or automatically using an existing data mining technique. In either case, once the knowledge exists in propositional form, Formal Concept Analysis is applied to the rules to develop an abstraction hierarchy from which multi-level rules can be extracted. The user is able to explore the knowledge at and across any of the levels of abstraction to provide a much richer picture of the knowledge and understanding of the domain.

**Keywords:** Formal Concept Analysis, Multi-level Rules, Ripple Down Rules

## 1 Introduction

Discovery of knowledge from databases or other data sources has been the focus of most knowledge discovery research. This emphasis has been natural since much data is readily available and in need of further manipulation, summarisation and interpretation. This paper reports work being conducted to reorganize and discover higher-level knowledge from rule bases rather than data bases. A benefit of starting with knowledge rather than data is that the key features in the data have already been identified. However, knowledge has some peculiarities such as smaller itemsets which are unique and thus do not lend themselves to frequency based algorithms. Our work is similarly motivated to research into finding multi-level association rules from data (e.g. [6],[26]). As these others have found, querying across levels of abstraction is important but not possible when the rules are based merely on the primitive attribute-values used in the original data. Another reason for mining rules is the identification of “interesting” patterns or concepts in amongst the many patterns that may emerge. Interestingness can be considered the extent to which a rule’s support deviates from its predicted behaviour [11]. Our view of interestingness is related to our understanding of a concept as a set of objects and their attributes. Although our approach is currently focused on outputting classification, rather than association rules, like [18] we define an interesting (or most informative) rule to be non-redundant with minimal antecedents and maximal consequents. According to [18] a redundant rule is one that conveys the same or less general information within the same support and confidence levels. The experiments we report do not employ support or confidence measures as we only have one instance of each example. However, based on the work of [5] we are currently exploring if accuracy/usage statistics can be used to supplement our approach to identify the “interesting” concepts. In our approach redundancy exists where we have a repeated concept or where a concept belongs to the same branch, that is, there is a parent-child relationship. These notions can be better understood after our description of the knowledge representation we use and formal concept analysis. Starting with knowledge can be seen as a problem in itself given the difficulties associated with acquiring knowledge particularly where the technique requires a model to be articulated by the domain expert. In this work we use a technique which does not rely on the difficult task of model specification in order to capture data.

This paper offers an approach to knowledge discovery that is focused on knowledge rather than data as input and structured multi-level knowledge rather than single-level knowledge as output. Ripple-down rules (RDR) [2] are used for manual and incremental acquisition of rules from cases at a rate of approximately 1 rule per minute. The rules are then mined using a set-theoretical technique known as Formal Concept Analysis (FCA) [28],[29] to automatically generate a concept lattice. The approach is applicable beyond RDR and can be used on any propositional knowledge-based representation. The approach should be highly attractive to other data mining

techniques that produce association rules which are low-level and simply based on the attribute-value pairs found in the original data.

Firstly, we will introduce the goals of the project and RDR. In section 3 we will describe FCA briefly. In Section 4 we will describe the experiments we have performed using knowledge bases from the domain of chemical pathology. In Section 5 we will discuss our results. Related research and concluding remarks appear in Section 6.

## 2. Introducing Ripple Down Rules

Before describing the work we did, it is necessary to explain our motivation for conducting the work. As mentioned, our motivation was not to uncover knowledge from data. We had already developed a user-driven knowledge acquisition technique and representation that allowed for rapid and incremental development of KB. We have knowledge bases which cover over 50,000 cases which have been developed online and evolved as cases became available. This suited the domain of chemical pathology [4] where RDR was first applied. Our first major system went into routine use with 200 rules and grew to more than 2000 rules over a 4 year period, requiring approximately 100 hours by the domain expert without the mediation of a knowledge engineer.

The project described in this paper was driven by two primary goals. The first was multi-level knowledge discovery and the second was restructuring of the knowledge base to make the knowledge base more compact. Both of these goals were interrelated as our approach using FCA discovers relationships between primitive concepts (rules) which could be used either in the identification of a higher level concept or in combining two (or more) rules together. For example, if we have three rules which state:

```
IF barks=yes, suckles_young=yes, warm_blooded=yes
THEN animal=dog

IF miaows=yes, suckles_young=yes, warm_blooded=yes
THEN animal=cat

IF miaows=yes, suckles_young=yes, warm_blooded=yes
THEN animal=feline
```

we can find the set intersections to suggest that perhaps a feline and a cat are the same (and remove one of them). We can show the higher level concept to a human based on the intersection of the cat and dog concept from which we could solicit from a human expert the name “mammal” to add into our background knowledge. Wherever this pattern is found it can be identified with the abstract term. This strategy fits with our incremental and user-manageable KA technique. The result is a new higher-level concept (rule):

```
IF suckles_young=yes, warm_blooded=yes
THEN mammal
```

We chose these goals due to the nature of RDR. RDR uses a rule-based exception structure for knowledge representation and cases to drive knowledge acquisition as well as to allow automatic validation of new rules. Exception rules are used to locally patch a rule that gives a wrong conclusion to a case. The case that prompts a new rule to be added is known as the rule’s cornerstone case. The case thus provides the context in which the knowledge applies. Rules are only ever added, never deleted. As shown in Figure 1, new rules may be modifications/exceptions to a previous rule. Studies have shown that RDR KB are comparable in size to KB developed using machine learning algorithms such as C4.5 and Induct [16]. Manually acquired KB are even more compact as it appears that experts are good at selecting which features in a case justify a particular classification. The partial Multiple Classification RDR (MCRDR) [13] KB shown in Figure 1 demonstrates the incremental nature of development and the possibility of redundancy. The example is taken from a real Glucose KB, referred to as GA, and includes the use of unconditional stopping rules which are used to override previous classifications with a null conclusion. The full KB has 25 rules, Figure 1 shows the rules which have exceptions to demonstrate the MCRDR structure. In section 5 we show how these 24 rules have been reduced to 8 rules and how they can be displayed as a concept lattice. Redundancy and scattered classes occur because rules are added as cases arise and because the expert is free to choose any string as a rule condition (providing it covers the current case and does not cover cornerstone cases) or conclusion. Thus two different strings “representing” the same condition or conclusion will be regarded as different. This makes understanding of the knowledge and the task of pattern identification even harder as concepts which are the same or similar (e.g. classes CL3 and CL4 in the legend in Figure 1) will be spread across possibly thousand of rules. By

finding intersections of shared rule conditions using FCA we can automatically suggest classes to be merged or further differentiated. Minimal analysis of the domain is required. KA can begin once cases are set up or found.

Since the knowledge acquired uses the primitive terms in the cases we were interested in finding higher level concepts to give us further insights into our cases. Higher-level concepts can suggest intermediate rules, like the mammal rule given above. The domain expert may find such abstractions more natural when specifying rules. We allow the user to explore the knowledge at multiple-levels of abstraction through selection of aspects of interest to them which are then displayed as a concept lattice generated using FCA. An example of a lattice is given in Section 5. Pilot studies [20] have shown that the concept lattice is valuable for explanation and learning purposes.

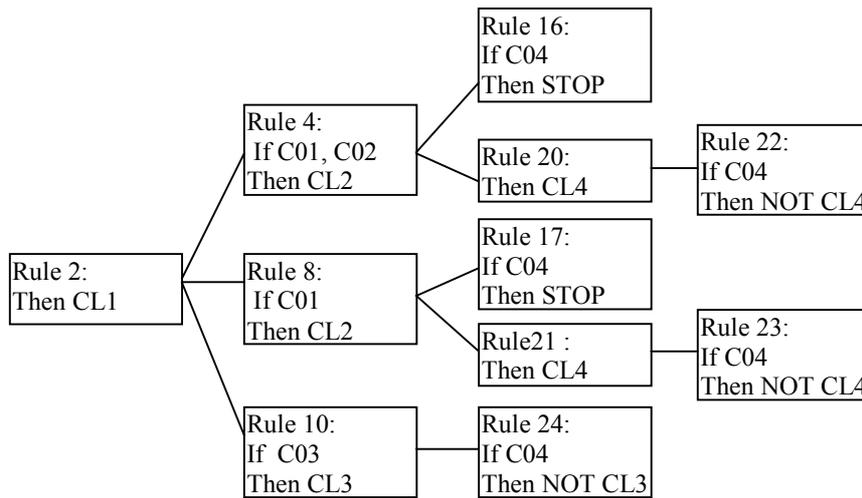


Figure 1: A partial MCRDR KB for the Glucose domain

CL1 = {\$Report}[Glucose Comment] CL2 = [Glucose Comment][The mean blood glucose has been calculated from the HbA1c result. Glucose has not been measured on this specimen.] CL3 = [Glucose Comment][Mean blood glucose is not calculated when the HbA1c is below 6.0%.] CL4 = [Glucose Comment][The mean blood glucose has been calculated from the HbA1c result. It is an index of the average blood glucose. Glucose has not been measured on this specimen.] CO1 = (HbA1c >= 6.0)                      CO2 = (Diabetic=Y) CO3 = (HbA1c <= 5.9)                      CO4 = (NoComment=Y)
--

### 3. Introducing Formal Concept Analysis

Formal Concept Analysis is a theory of data analysis and knowledge processing based on the idea of a formal context and formal concept. It provides methods to visualize data in the form of concept lattices. A formal concept is just a pair of sets, made up of the set of objects, the extension and the set of attributes shared by those objects, the intension. A formal context  $\kappa = (\mathbf{G}, \mathbf{M}, \mathbf{I})$  consists of a set of formal objects ( $\mathbf{G}$ ), a set of formal attributes ( $\mathbf{M}$ ) and a binary relation ( $\mathbf{I}$ ) between  $\mathbf{G}$  and  $\mathbf{M}$ .  $(\mathbf{g}, \mathbf{m}) \in \mathbf{I}$  means object  $\mathbf{g} \in \mathbf{G}$  is in relation  $\mathbf{I}$  with an attribute  $\mathbf{m} \in \mathbf{M}$ . A formal context is usually represented by a cross table.

In our use of FCA we treat each rule as an object comprising a set of attributes based on the rule conditions, we discuss this further in Section 4.2. Figure 2 shows the formal context based on the Glucose KB given in Figure 1. The RDR exception structure has been *flattened*, as described in Section 4.1. In the process rules 4 and 8 have been replaced by rules 16 and 17. Conclusion CL2 is no longer given as it is unconditionally overridden by CL4 in rules 20 and 21. The default rule in Rule 2 which concludes CL1 has not been included as it is not considered potentially interesting. Other propositional knowledge representations that do not use exceptions can be mapped directly to the formal context. The set of objects/rules,  $\mathbf{G} = \{10, 16, 17, 20, 21, 22, 23, 24\}$  and attributes are the rule conditions,  $\mathbf{M} = \{\text{HbA1c} \geq 6.0, \text{Diabetic} = \text{Y}, \text{HbA1c} \leq 5.9, \text{NoComment} = \text{Y}\}$ .

Rule No	HbA1c>=6.0	Diabetic=Y	HbA1c<=5.9	NoComment=Y
10-CL3			X	
16-STOP	X	X		X
17-STOP	X			X
20-CL4	X	X		
21-CL4	X			
22-Not CL4	X	X		X
23-Not CL4	X			X
24-Not CL3			X	X

**Figure 2 – Crosstable for the Glucose KB in Figure 1**

For a set  $X \subseteq G$  of objects we define:  $X' := \{m \in M \mid (g,m) \text{ for all } g \in X\}$

For a set  $Y \subseteq M$  of attributes we define:  $Y' := \{g \in G \mid (g,m) \text{ for all } m \in Y\}$

A formal concept of context  $(G, M, I)$  is a pair  $(X, Y)$  with  $X \subseteq G$  and  $Y \subseteq M$  such that  $X' = Y$  and  $Y' = X$ .  $X$  is called the *extent* and  $Y$  is called the *intent* of the concept  $(X, Y)$ , i.e.  $Y$  consists of those attributes which apply to all objects in  $X$  and all objects in  $X$  have each attribute in  $Y$ . Formal concepts for the context in Fig. 2 include:

- $(\{20\}, \{HbA1c \geq 6.0, Diabetic=Y\})$
- $(\{16, 17, 22, 23, 24\}, \{NoComment=Y\})$
- $(\{16, 22\}, \{HbA1c \geq 6.0, Diabetic=Y, NoComment=Y\})$

Concept 1 is called an *object concept* as its extent consists of only one object. Similarly concept 2 is an *attribute concept* consisting of one attribute only. The most famous algorithm for the purpose of concept generation from a context table is Ganter's algorithm [10]. While we investigated some incremental algorithms [e.g.15] our results are based on Ganter's algorithm.

## 4 The Experiments

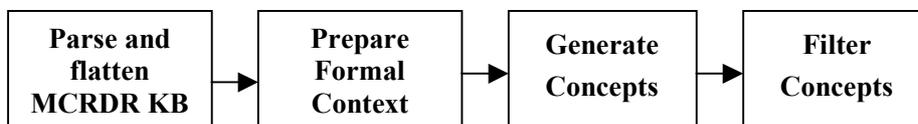
The input data to our experiments were 13 deployed KB developed using the commercial version of MCRDR from the domain of chemical pathology. We had a number of KB for different subdomains such as glucose, bio-chemistry, haematology, and microbiology. The KB ranged in size from 3 rules to 319 rules. The number of attributes ranged from 8 to 182. The largest RDR KB currently in use has over 7,000 rules. We did not have access to this KB in this study, but plan to apply our approach to such large KB when they are made available. While cases are an important part of the RDR KA approach we did not have access to these. Generating metaknowledge involves the following four step process as depicted in Figure 3:

**Step 1:** Parse the KB file and convert the decision list structure into a binary decision table. (i.e. a crosstable or formal context). Since initial experiments produced too many concepts we have included pruning rules in this step.

**Step 2:** Prepare a formal context with rules as objects and conditions on these rules as attributes.

**Step 3:** Generate concepts.

**Step 4:** Prune uninteresting concepts and print remainder in a suitable format for expert to analyze.



**Figure 3: The Meta-Knowledge Generation Process**

### 4.1: Parsing

There are two objectives of parsing:

- To remove the RDR tree structure so that we can generate a cross table for FCA.
- Prune redundant rules.

The parsing rules we used for processing the data had to consider the type of rule (modify, add or remove) and whether the rule included conditions (conditional or unconditional). The combination of these two characteristics resulted in the following set of rules:

- Unconditional modify* – use conclusion from this node and pick up all conditions from all parents, delete parent;
- Conditional modify* – use conclusion and conditions from this node and pick up all conditions from all parents, no change to parent;
- Unconditional add* – this rule will always fire, typically used for a default head rule;
- Conditional add* – use conclusion and conditions from this node. No parents exist;
- Unconditional remove* – flag the parent rule as deleted so that it is not included in concept generation;
- Conditional remove* – use the negation of the parent’s conclusion, pick up conditions from this node and from all predecessors (parents).

Remove rules are known as *stopping rules*, as shown in Figure 1. Stopping rules have a NULL Conclusion. RDR KB tend to have a large number of stopping rules. This does not have a noticeable affect on inferencing, but it does impact the usefulness of some explanations (rule traces). By the end of this phase we have a list of rules. Associated with each is a set of attributes and a conclusion. Next we generate the formal context.

#### 4.2: Generating a Formal Context

A simple technique for generating concepts using FCA on MCRDR KBs is to treat each rule as an object and the rule conditions as attributes. Each condition is actually an attribute–value pair but our naïve approach produces similar results to Ganter’s approach to conceptual scaling and the handling of multi-valued contexts [8],[9]. Each row in the crosstable corresponds to a rule in the MCRDR KB. Each object/row is identified by its rule number and the conclusion or conclusion code. Each column in the crosstable, except for the first column which contains the object id, corresponds to a rule condition. We put a cross when a particular rule has these conditions otherwise we leave it empty. The context table is used to generate concepts.

#### 4.3: Generating Concepts

From Section 3 the generation of new concepts can be simply seen as the result of finding the intersections of objects and their attributes. In these experiments we used Ganter’s algorithm NEXTCONCEPT [10] to generate concepts. It computes all concepts  $\mathcal{L}(G,M,I)$  from context  $(G, M, I)$  in lexicical order in  $\mathcal{O}(|G|^2 \times |M| \times \mathcal{L}(G, M, I))$  [15]. We filter concepts as we generate them, however, we could also leave filtering as a separate final step as outlined below.

#### 4.4: Filtering Concepts

Two criterions are employed to filter concepts.

- First we don’t want single extent concepts. This is because they only tell us that this rule has these attributes, something the expert knows already.
- Since we are interested in cross comparisons of rules in different branches we prune concepts that consists of rules with parent child relation i.e. belonging to same branch.

Finally we print out concepts along with conclusions of rules so that the expert can analyze them.

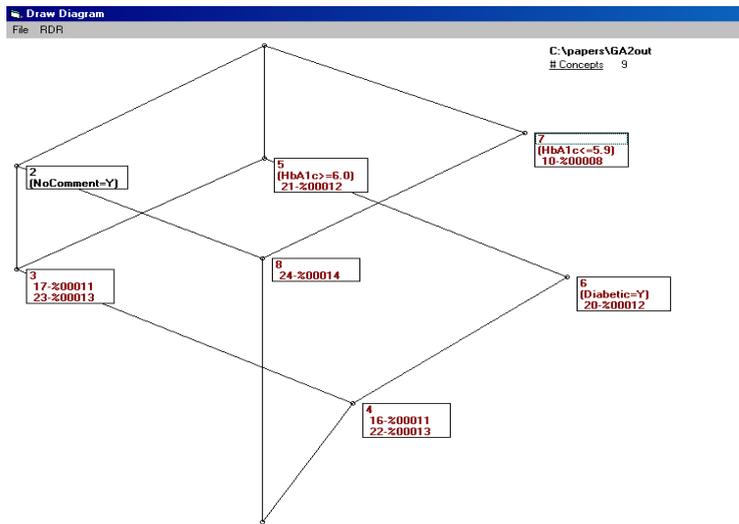
### 5 Results

Results for five of the KB are given in Table 1. While we conducted our experiments on all 13 KB, for space we have randomly selected five of different sizes, each covering different subdomains within chemical pathology. The first column shows the KB id. The second column shows the original number of rules in the KB. The second and third columns show the number of rules output from Steps 1 and 3, respectively. The reduction from Step 1 is due to the removal of stopping rules. Step 2 is simply a formatting stage. The fourth column shows how many concepts from Step 3 passed through the filter for human review in step 4. The fifth column shows the number of concepts that meet our “interesting” concept criteria of minimal antecedent and maximal consequent to convey equivalent information. Finally we show the percentage of interesting concepts to original concepts from step 3. We can see a substantial reduction in the original size of the KB. A more compact and restructured KB can be produced from the

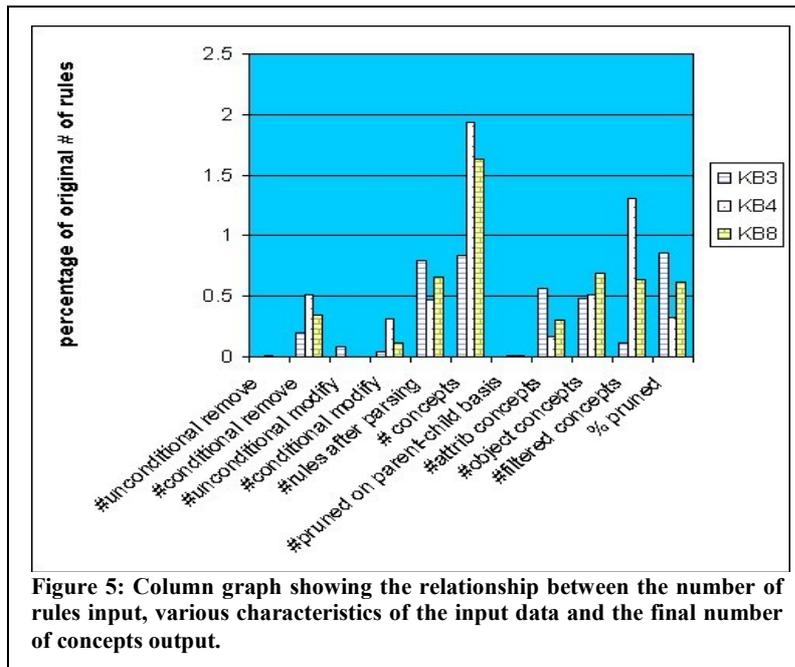
concept output from step 4. The interesting concepts can be used as the initial concepts, to further encourage an optimal KB organization. The smaller set of interesting concepts can be explored using a concept lattice. Without pruning and identification of the interesting concepts, the lattices were too large to be comprehensible. In Figure 4 we see the remaining 8 concepts after pruning the GA Glucose KB. An introduction to how lattices are generated and used is given in [22].

**TABLE: 1: Results from each step**

KB	# Rules Input	#Rules from Step 1	# Concepts Step 3	# Concept Step 4	# interesting concepts	%pruned
ANA	107	55	90	42	19	0.79
GA	25	20	21	8	3	0.85
GTT	19	14	23	13	7	0.70
GLU	319	154	616	416	310	0.49
IRON	90	59	147	84	57	0.61



**Figure 4: A concept lattice for the pruned Glucose (GA) KB. Attribute values which belong to a concept are reached by traversing ascending paths. Shared nodes indicate higher level concepts.**



**Figure 5: Column graph showing the relationship between the number of rules input, various characteristics of the input data and the final number of concepts output.**

In Figure 5 statistics of three different size KBs 3 (GA), 4 (GLU) and 8 (IRON) are given. All data have been converted into a percentage of the original number of rules input as given in Table 1. From the chart we compare the affect of unconditional removes or modifies on the number of rules after parsing. Since there are so few in each KB it is not possible to draw any conclusions. If there were many one would expect a corresponding decrease in the number of rules after parsing. There does appear to be a relationship between the number of conditional removes and modifies and a decrease in the number of rules after parsing i.e. more conditional removes and/or modifies results in fewer rules output. All KB are smaller than the original KB after parsing. One(1) on the y-axis signifies 100% of the original number

of rules. From this small sample it appears that despite the reduction in size from parsing, the number of concepts generated from the parsed rules is greater for KB with more conditional removes and modifies. Our strategy to ignore parent-child concepts does not seem to have been terribly useful since there were 0, 2 and 1 such concepts in KBs 3, 4 and 8, respectively. The relationship between the number of object concepts (one object shared by many attributes) and final number of concepts output is not clear, but since these are primitive concepts it is obvious they are not new or interesting. Our decision to prune attribute concepts (only one attribute shared by many objects), however, does seem to substantially reduce the number of filtered concepts. The sample confirms some of our intuitions but also compels us to consider further pruning strategies. We particularly need to combine similar strings or attribute values. For this purpose we will use clustering techniques and our nearest neighbour algorithm in conjunction with FCA. Initial work which manually mapped similar attributes resulted in even more rules sharing conditions (as expected) but resulted in many more concepts (not what we wanted). Our current focus is on getting access to usage statistics so that we can apply confidence and support measures to further prune concepts. Once we have a small number of concepts, say five to twenty concepts for each KB, we will meet with domain experts who will assess whether the concepts are in fact interesting.

## 6 Related Work and Conclusion

Like most techniques for multi-level mining of rules (e.g. [6]) our approach using FCA involves pruning or filtering of rules. There is, however, no transformation resulting in loss of information in those rules that are kept. Essentially, filtering implements the semantics of the KB given by the domain expert. That is, only objects (rules) identified for removal are ignored in the generation of the formal context and formal concepts. Our approach is like the use of restricted contexts [8].

Before we adopted FCA for higher-concept generation we explored the use of Rough Set Theory [19] since that technique also does not rely on frequencies of cases. In fact duplicate examples are discarded as a first step in the development of reducts (rules). However, we found that since rules have already discarded most (if not all) irrelevant attributes we lost too much information and classification accuracy when the rules output were run against the test sets [20]. Like Fortin [6] the concept hierarchies we produce are dynamically adjusted. Lattices can be generated for each query. Alternatively, algorithms for the incremental update of lattices can be used which modify only those parts of the concept lattice affected by the new query or changes in the KB itself. Using FCA multi-level knowledge is not only discovered but also structured. The structure contains valuable knowledge which is often not known or easily articulated by experts but which adds clarity and improves understanding. The abstraction hierarchy generated by FCA can be viewed as an ontology in that it represents a shared conceptualization [12] of a particular domain. Our work which combines multiple knowledge bases to develop a shared knowledge repository [21] fits particularly well with the notion of a shared conceptualization in addition to supporting queries across concept hierarchies. Omelayenko [17] has surveyed a number of approaches to learning ontologies. He divides the approaches broadly into two main categories depending on whether they involve machine-learning (ML) or manual construction. Omelayenko decides that the combination of both techniques will offer the best results by combining the speed of machine learning with the accuracy of a human. We are inclined to agree with his conclusion. Kietz et al. [14] offer an algorithm that uses two parameters: support and confidence for a rule to semiautomatically develop an ontology from text. As Omelayenko points out, ML results in flat homogeneous structures often in propositional form. He cites work from the RDR group [25], which was seeded by our FCA work, which learns different relationships between classifications: subsumption in marginal cases, mutual exclusivity and similarity and then develops a taxonomic hierarchy between the classes.

Our goal to develop an approach that supports multiple levels of abstraction is shared by the work of [26]. They are looking at the seamless integration of knowledge bases and databases. The ParkaDB approach supports the development of high and multi-level classification rules. An ontology (in the form of a concept hierarchy) together with frequency counts are used to determine which concepts should be included in a rule. The ontology provides the background knowledge to guide the discovery process. A number of similar approaches that use concept taxonomies (e.g. [24]) have also been developed but these are based on traditional relational database technology and require transformation to a generalized table as part of the preprocessing which can result in over-generalisation. DeGraaf et al. [11] use a prespecified taxonomy to suggest interesting association rules, which are rules that do not fit with the taxonomy. In this way interestingness is the notion of unexpected. Intuitively it is appealing but the approach obviously relies on the development of a complete and valid taxonomy. ParkDB does not require such preprocessing but supports dynamical generalization of data without over generalization. Our approach is different to all of these in

that we do not use a concept hierarchy to develop rules. Instead we use rules to develop a concept hierarchy which may lead to higher level rules being uncovered. Thus we avoid the substantial effort required in first developing the hierarchies and the difficult task of validating them. Given any string or substring and using set intersection, term subsumption and lexicographical ordering we are able to find all combinations using that string. Some concepts will include attributes and/or objects at different levels of abstraction such as the objects mammal and dog will appear in individual and shared concepts. The combination of multi-level concepts supports queries at multiple levels.

Rules are sometimes used as background knowledge (eg. [1], [27]) to guide the discovery of knowledge from data. Some approaches use templates to guide the mining process. Shen et al.[23] specifies abstract forms of rules which are used in metaqueries. To some extent the RDR approach is concurrently data and knowledge based in that it combines the use of case and rule-based reasoning. A key difference between our approach to data mining and others is that we do not rely on large amounts of data or large number of cases for rule development. Where data is plentiful ML algorithms will develop rules more quickly and have higher accuracy than manually developed KB. Even better results will be achieved if the chosen set of cases are well-structured and representative of stereotypical cases. However, where large numbers of classified cases do not exist or where we want to provide an incremental way of developing and maintaining rules the technique offered by RDR works very well in building compact KB that mature quickly (i.e. they cover the domain with small error rates) [4].

Knowledge mining is offered in this paper as a way of extracting added-value from an organization's knowledge sources. The complete approach using RDR and FCA could be applied where limited cases exist or they emerge over time and a human expert is available to perform knowledge acquisition. Where large amounts of data exist data mining techniques can be applied to develop a first round of primitive rules which are then used as input by FCA to develop an abstraction hierarchy from which multi-level rules can be extracted. As a result the user is able to explore the knowledge at and across any of the levels of abstraction to provide a much richer picture of the knowledge and understanding of the domain.

#### Acknowledgements

The authors would like to thank Pacific Knowledge Systems (PKS) for access to the research knowledge bases. Particular thanks to Les Lazarus and Michael Harries from PKS for their time and assistance.

#### References

- [1] Cai, Y., Cerone, N. and Han, J. (1991) Attribute-oriented induction in relational databases. In *Knowledge Discovery in Databases* AAAI/MIT Press.
- [2] Compton, P. and Jansen, R., (1990) A Philosophical Basis for Knowledge Acquisition. *Knowledge Acquisition* 2:241-257.
- [3] Deogun, J., Raghavan, V. and Sever, H. (1998) Association Mining and Formal Concept Analysis, In *Proceedings Sixth International Workshop on Rough Sets, Data Mining and Granular Computing*, Vol 1: 335-338.
- [4] Edwards, G., Compton, P., Malor, R., Srinivasan, A. and Lazarus, L. (1993) PEIRS: a Pathologist Maintained Expert System for the Interpretation of Chemical Pathology Reports *Pathology* 25: 27-34.
- [5] Edwards, G., Kang, B., Preston, P. and Compton, P. (1995) Prudent Expert Systems with Credentials: Managing the expertise of Decision Support Systems *Int. Journal Biomedical Computing* 40:125-132.
- [6] Fortin, S., Liu, L. and Goebel, R. (1996) Multi-Level Association Rule Mining: An Object-Oriented Approach Based on Dynamic Hierarchies, *Technical Report TR 96-15*, Dept. of Computing Science, University of Alberta.
- [7] Gaines, B.R. and Compton, P. (1993). *Induction of meta-knowledge about knowledge discovery*. IEEE Transactions on Knowledge and Data Engineering 5(6) 990-992.
- [8] Ganter, B. (1988) Composition and Decomposition of Data In H. Bock (ed), *Classification and Related Methods of Data Analysis* North-Holland, Amsterdam, 561-566.
- [9] Ganter, B. and Wille, R. (1989) Conceptual Scaling In F. Roberts (ed), *Applications of Combinatorics and Graph Theory to the Biological Sciences* Springer, New York, 139-167.
- [10] Ganter, B. and Wille, R. (1999) *Formal Concept Analysis: Mathematical Foundations* Springer, Berlin.
- [11] Graaf, de, J., M., Kusters, W., A. and Witteman, J., J., W. (2000) Interesting Association Rules in Multiple Taxonomies, *Proceedings of the Twelfth Belgium-Netherlands Artificial Intelligence Conference*.
- [12] Gruber, T. R., (1993) Toward Principles for the Design of Ontologies Used for Knowledge Sharing *Knowledge Systems Laboratory*, Stanford University.
- [13] Kang, B., Compton, P. and Preston, P. (1995) Multiple Classification Ripple Down Rules: Evaluation and Possibilities *Proceedings 9th Banff Knowledge Acquisition for Knowledge Based Systems Workshop* Banff. Feb 26 - March 3 1995, Vol 1: 17.1-17.20.
- [14] Kietz, J.-U., Maedche, A. and Volz, R. (2000) A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet. WS "Ontologies and Text", co-located with *EKA'2000*, Juan-les-Pins, French Riviera, October 2-6, 2000
- [15] Lindig, C. (2000) Fast Concept Analysis In G. Stumme (ed) *Working with Conceptual Structures – Contributions to ICCS'2000* Shaker-Verlag, Aachen, 2000, 152-161.

- [16] Mansuri, Y., Kim, J.G., Compton, P. and Sammut, C. (1991). A comparison of a manual knowledge acquisition method and an inductive learning method *Australian Workshop on Knowledge Acquisition for Knowledge Based Systems*, Pokolbin (1991), 114-132.
- [17] Omelayenko B., (2001) Learning of Ontologies for the Web: the Analysis of Existent Approaches, In: Proceedings of the International Workshop on Web Dynamics, held in conj. with the 8th International Conference on Database Theory (ICDT'01), London, UK, 3 January 2001.
- [18] Pasquier, N. (2000): Mining association rules using formal concept analysis. In: Stumme, G. (Ed.): *Working with Conceptual Structures*. Contributions to ICCS 2000. Verlag Shaker Aachen, 259-264.
- [19] Pawlak, Zdzislaw (1991) *Rough Sets: Theoretical Aspects of Reasoning about Data* Kluwer Academic Publishers, Dordrecht.
- [20] Richards, D. (1998) Using AI to Resolve the Republican Debate In Slaney, J. (ed) *Poster Proceedings of Eleventh Australian Joint Artificial Intelligence Conference AI'98* 13-17 July 1998, Griffith University, Nathan Campus, Brisbane, Australia, 121-133.
- [21] Richards, D. (2000) Reconciling Conflicting Sources of Expertise: A Framework and Illustration In *Proceedings of the 6<sup>th</sup> Pacific Knowledge Acquisition Workshop* P. Compton, A. Hoffmann, H. Motoda and T. Yamaguchi (eds) Sydney December 11-13, 2000, 275 - 296.
- [22] Richards, D and Compton, P, (1997) Combining Formal Concept Analysis and Ripple Down Rules to Support the Reuse of Knowledge *Proceedings Software Engineering Knowledge Engineering SEKE'97*, Madrid 18-20 June 1997, 177-184.
- [23] Shen, W., Ong, K., Mitbender, B. and Zaniolo, C. (1995) Metaqueries for Data Mining In U. Fayad, Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (eds) *Advances in Knowledge Discovery and Data Mining AAAI/MIT Press*.
- [24] Singh, L., Scheuermann, P. and Chen, B. (1997) Generating Association Rules from SemiStructured Documents Using an Extended Concept Hierarchy. In *Proceedings of the 6th International Conference on Information and Knowledge Management*. Las Vegas, Nevada, USA, 1997, pp. 193-200.
- [25] Suryanto, H., and Compton, P. (2000) Learning Classification Taxonomies from a Classification Knowledge Based System. In S. Staab, A. Maedche, C. Nedellec, P. Wiemer-Hastings (eds) *Proceedings of the Workshop on Ontology Learning, 14<sup>th</sup> Conference on Artificial Intelligence (ECAI'00)* August 20-25, Berlin.
- [26] Taylor, M., Stoffel, K. and Hendler J. (1997) Ontology-based Induction of High Level Classification Rules In *SIGMOD Data Mining and Knowledge Discovery workshop proceedings*. Tuscon, Arizona, 1997.
- [27] Walker, A. (1980) On Retrieval from a Small Version of a Large Database, In *VLDB Conference Proceedings*.
- [28] Wille, R. (1982) Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts In I. Rival (ed), *Ordered Sets* Reidel, Dordrecht, Boston, 445-470,.
- [29] Wille, R. (1992) Concept Lattices and Conceptual Knowledge Systems *Computers Math. Applic.* (23) 6-9: 493-515.