TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Software Science

Kristo Raun 152921IABM

GREEDY COVERAGE – AN ALGORITHM FOR COVERING A DATASET

Master's Thesis

Supervisor: Ants Torim
PhD

Tallinn 2018

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Kristo Raun 152921IABM

GREEDY COVERAGE – ALGORITM ANDMESTIKU KATMISEKS

magistritöö

Juhendaja: Ants Torim
PhD

Tallinn 2018

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Kristo Raun

07.05.2018

Abstract

Today's decision-makers need to tackle problems quickly, using the best knowledge possibly available to them. Data mining, through its goal of exposing previously unknown information in a dataset, is commonly used as a method to support decision-making. This thesis applies the theory of Formal Concept Analysis (FCA) and the notion of **concept coverage** to look at ways of efficiently displaying the most important parts of the dataset to a decision-maker. Principally, the thesis will examine some existing FCA algorithms and introduce a new algorithm – the **Greedy Coverage**. The algorithms are implemented and executed on several real life datasets, and compared in terms of how well they compute the optimum concepts for covering the data within a dataset. The business applicability of these algorithms is investigated, especially in terms of the notion of concept coverage and the new Greedy Coverage algorithm. Importantly, the Greedy Coverage algorithm outperforms the existing FCA algorithms, and proves valuable in business analysis by offering the data-miner an improved view of the dataset.

This thesis is written in English and is 68 pages long, including 7 chapters, 12 figures and 13 tables.

Annotatsioon

Greedy Coverage – algoritm andmestiku katmiseks

Tänapäeva otsustajad peavad probleeme lahendama kiiresti, kasutades parimat võimalikku teadmist mis neil on. Andmekaevet, mille eesmärk on tuua andmestikust esile info mida varasemalt ei teatud, kasutatakse laialdaselt otsuste langetamise toetamiseks. Käesolev magistritöö tugineb Formaalse Kontsepti Analüüsi (FCA) teooriale ning **kontsepti katvuse** ideele, et uurida kuidas tõhusalt kuvada otsustajale andmestikust kõige olulisemat. Peamiselt vaadeldakse töös mõningaid olemasolevaid FCA algoritme ja tutvustatakse uut algoritmi – **Greedy Coverage**. Algoritmid implementeeritakse ja käivitatakse mitmetel reaalsetel andmestikel. Algoritme võrreldakse selle osas kui hästi nad arvutavad kontsepte mis võimalikult hästi katavad andmestiku. Uuritakse ka algoritmide ärilist otstarvet, iseäranis kontsepti katvuse ning uue algoritmi rakendatavuse seisukohast. Tulem on, et Greedy Coverage edastab kõiki teisi uuritud FCA algoritme ja leiab tõestust algoritmi väärtus ärianalüüsi seisukohast – algoritm annab andmekaevajale parendatud ülevaate andmestikust.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 68 leheküljel, 7 peatükki, 12 joonist, 13 tabelit.

List of abbreviations and terms

1s	Number of “one”-s in a dataset
ARM	Association Rule Mining
FCA	Formal Concept Analysis
GC	Greedy Coverage algorithm
IL	Iceberg Lattice algorithm
LS	Local Stability algorithm
MSM	Monotone Systems Method algorithm
S	Stability algorithm

Table of Contents

List of Figures

List of Tables

1 Introduction

This chapter offers the reader a foundation for understanding this thesis. Firstly, the background of the problem is explained, from the perspective of decision-making, via the area of data mining to the specific notion of data coverage. Then, the problem statements are proposed. The third part introduces the research process, and acknowledges potential considerations that the reader should keep in mind when interpreting the paper. The final subchapter outlines the structure and content of the main portion of the thesis.

1.1 Background

Technology, transportation and communication are continuously advancing, and with them the possibilities for companies and organizations are expanding. It is easier to start a company, smaller companies can more easily compete with established companies, and more companies are finding new ways to provide for the needs of the consumers [1], [2]. The competition is becoming more rapid and a more rapid competition requires faster actions – faster decision-making from leaders. However, the need for faster decisions denotes a caveat: an increase of groundless decisions.

Striking a balance between the opposing forces of **fast decisions** and **grounded decisions** has many names: speed vs quality, efficiency vs effectiveness, tactics vs strategy, doing things right vs doing the right things. Regardless of the name, the problem is apparent – in order to survive the competition, decision-making needs to focus more on speed. Doing so, however, diminishes the importance of thorough investigation, and possibly undermines the correctness and value of the business decision.

How to ensure that decision-makers can tackle problems quickly, using the best knowledge possibly available to them? Answering that question is one of the research outputs of the field of **data mining**. Increasing amounts of data from all parts of a business are recorded and stored. The datasets formed by the data enable businesses to extract new, or previously unknown information and knowledge about their processes [3]. The goal of data mining is to expose this type of novel information from datasets [4].

Data mining covers many domains – from statistics to machine learning [3] – and has many applications – from marketing and sales, to fields as varied as biology and manufacturing [4]. This paper focuses on the method of **Formal Concept Analysis** (FCA). Through its infusion of mathematics and philosophy, the FCA theory divides entities of data into concepts [5], [6], enabling a better understanding of the underlying data. Over the years, the FCA framework has become the basis for many data mining and data analysis methods [7], [8], [9]. One of the FCA theory core attributes is the data visualization method via line diagrams called the **concept lattices** [10], [11], [12]. However, it has been shown in various research that a lattice quickly becomes unreadable as the amount of data grows [13], [14], [15], thus rendering the lattice unusable by the decision-maker.

In order to alleviate the unreadability issue, various algorithms have been proposed and compared [16], [17], [18], most of them applied via setting indices on the FCA concepts, effectively ranking and filtering the most relevant concepts. However, no known study has examined the algorithms and the concepts they calculate based on coverage – the percentage of data, or 1s (“one-s”) in the dataset, which are covered by the concept. Utilizing the **concept coverage** as a metric for ranking concepts and only retaining the most effective concepts could be a potential solution for giving decision-makers a quick and effective overview of the relevant data.

1.2 Problem Statement

Considering the issues introduced in the previous section, the underlying theme of the thesis is to explore how to extract the most relevant data out of a dataset. Grounded in the FCA theory and through the utilization of the coverage metric, the principal questions which this thesis will investigate are:

1. Which FCA algorithms are most effective in covering the largest proportion of a dataset?

2. Is it possible to implement a more efficient algorithm for calculating the optimum concepts for a dataset’s coverage?

3. Do these FCA algorithms have business applications?

1.3 Research Process

The high level steps which need to be taken in order to answer the posed questions, would be as follows:

- Find and implement various FCA algorithms;

- Find relevant and usable datasets on which to apply the algorithms;

- Analyse the results presented by the FCA algorithms from a technical perspective – in other words, how well do the concepts that the algorithms output cover the dataset;

- Explore a dataset from a decision-making analyst point-of-view, by drilling down, comparing and interpreting the results provided by the different algorithms.

These steps also roughly correspond to the outline of the thesis, which will be introduced in Chapter 1.4.

1.3.1 Restrictions and Considerations

While the material presented in this thesis is based on thorough investigation, analysis and fair judgement, it needs to be brought out that due to the spectrum and complexity of the topic, similar research could be undertaken by employing alternative theories and methods instead of FCA. Furthermore, the theory of FCA itself has become wide and varied throughout the years – hence, only a portion of the theory will be presented in this paper.

As a consequence, the potential restrictions that the reader should acknowledge are:

- FCA is just one methodology out of many for data mining and finding the most relevant pieces of data. The domains of both classification and clustering [4] can be considered similar to FCA depending of the context. Likewise, Association Rule Mining (ARM), which deals with finding popular and interesting patterns in a dataset [19], [20], has many analogous premises to FCA [21]. The mentioned disciplines will not be addressed thoroughly in the scope of this paper, but they may serve as potential alternatives for analysing the topic of this thesis. A short overview of ARM will be given in Chapter 2.1.1.

- FCA, at its core, was developed for handling Boolean datasets. Boolean data is logical and thus not always explicit in the physical representation of a business. So, it could be said that focusing the analysis only on Boolean datasets is a rather limited scope and not applicable to actual business cases. While some branches of FCA have investigated exploiting FCA on multi-valued datasets [11], [12], it is not a topic for this paper. However, as also demonstrated in Chapter 3, multi-valued data can quite easily be transformed into Boolean data, and therefore the limitation of Boolean data is considered justified, practical, and well usable also in non-academic applications.

- The set of algorithms investigated in this research paper is relatively small – the limitation came from the size and format of this paper. A similar work encompassing all or most of the FCA algorithms developed throughout the years would be better suited for publishing as a series of articles or possibly even a book.

- The execution speed of the algorithms is important in real life situations, where the amount of data may be vast and the time to make a decision limited. However, execution speeds and possible optimizations are not considered in the scope of this paper.

- Results and analysis presented in this thesis are carried out in an unprejudiced manner; nevertheless, scrutiny must be exercised by the reader, as the business problems vary between business sectors, industries and segments. Interpretations are inherently subjective, and challenging views and understandings by the reader are welcomed by the author.

As mentioned, these bullet-points should be considered by the reader while working with the thesis, but they do not undermine the thesis. Rather, they function as reminders for awareness and possible paths for future work for the author, or also for the reader.

1.4 Outline of the Thesis

The thesis is structured into five chapters:

1) The first, current, chapter provides a background and introduction to the research conducted, including the research process and some considerations;

2) The second chapter explains the theory behind the thesis. Firstly, the principles of FCA are explained, namely the notions of context, concept and lattice; the intent and extent of a concept. Some applications of FCA, as well as its drawbacks, are presented. Four existing FCA algorithms are introduced and implemented. Finally, some similar works are discussed, outlining also the uniqueness and novelty of the research in this thesis;

3) The third chapter explains the methodology of the research. The chapter gives an overview of the datasets used in the research, including the pre-processing applied on the datasets. The idea of concept coverage is examined, as well as the notion of cumulative coverage of a dataset. Finally, the number of useful concepts is investigated, as it is to be used as a baseline in the research portion of the thesis.

4) The fourth chapter introduces a novel FCA algorithm – the Greedy Coverage (GC) – as an algorithm for calculating interesting concepts. The chapter mainly deals with explaining the algorithm's calculation methodology. The algorithm is the original work of the author of this thesis.

5) The fifth chapter is the first, so called *technical* part of the results, showing the results attained by executing the algorithms introduced in the second chapter, together with the GC, on the datasets introduced in the third chapter. The methodologies from the third chapter are used as the basis for interpreting the results. The new GC algorithm outperforms the studied existing algorithms, effectively answering the first and second research question.

6) In addition to a technical assessment, the sixth chapter provides the reader with a qualitative assessment of the results. An overview is given of the Instacart dataset, comparing the concepts calculated by the FCA algorithms in more detail. The outcome is that the GC algorithm, as well as the notion of concept coverage, are applicable in a business setting.

7) The final chapter summarizes the thesis, revisiting all of the most important remarks from each of the thesis' chapters, and proposing directions for future work.

2 Mining Meaningful Data

This chapter introduces the underlying theory of this paper – Formal Concept Analysis (FCA). The reader is given an overview of the fundamental principles of FCA, including the definitions and representations of a formal context, a formal concept and a concept lattice. Some applications of FCA will be discussed, as well as the drawback of a lattice containing many concepts and its potential remedies. Four existing FCA indices are introduced, together with an overview of their inherent algorithms. Finally, some similar works are covered to denote the novelty of the research in this thesis.

2.1 Formal Concept Analysis

Formal Concept Analysis is a mathematical theory which emerged in the 1980s from the study of sets and lattices [12], [5]. The theory describes the philosophical notion of concept as an abstract unit comprising of a set of objects and a set of attributes [5], [10].

In order to introduce the formal concept, firstly a definition of a **formal context** is needed. A formal context is a triple $K := (G, M, I)$, where G is the set of objects, M is a set of attributes and I is a binary relation between G and M, so that if the relation exists between object g from G and m from M, then it is said that object g has attribute m, i.e. $(g, m) \in I$ [5], [10]. Most commonly, a matrix table with binary relations is used for representing a formal context [6]. For the purpose of explanation, a context matrix has been constructed in Table 2.1. The matrix describes some example products of a financial institution, denoted by the *attributes*, or columns – and some customers, denoted by the *objects*, or rows.

Table 2.1. An Example of a Formal Context.

	Deposit Account	**Car Lease**	**Home Loan**	**Credit Card**
Customer 1	**1**	**1**	0	**1**
Customer 2	**1**	0	**1**	**1**
Customer 3	0	**1**	0	0
Customer 4	**1**	**1**	**1**	0

The corresponding sets of the example would be as follows:

G – **the set of objects** – is comprised of customer 1 to 4.

M – **the set of attributes** – is comprised of the products: Deposit Account, Car Lease, Home Loan and Credit Card.

I – **the set of relations** – is marked by a "**1**" in the appropriate row and column, if the relation is true, and by a "0" if the relation is not true. For example, Customer 1 has a deposit account, a car lease agreement and a credit card.

A **formal concept** of a formal context (G, M, I) is a pair (A, B), where A is a subset of G, and B is a subset of M [5]. The set A is called the **extent**, and the set B the **intent** of the formal concept (A, B) [12], [5]. In other words, the *objects* belonging to the concept (A, B) are described by the *extent* of the concept, and the *attributes* of the concept (A, B) are consequently described by the concept's *intent*.

Based on the definition, the concept can be understood as a unique segment of the context, composed of object and attribute relations. An example from our financial institution context would be: customers having both, a deposit account and a car lease agreement. As can be seen from Table 2.2, the concept contains *Customer 1* and *Customer 4*, who both also have additional but different products associated with them. Additionally, both *Deposit Account* and *Car Lease* are contained by other customers, but no other customer has both of them.

Table 2.2. A Formal Concept in a Context.

	Deposit Account	**Car Lease**	**Home Loan**	**Credit Card**
Customer 1	**1**	**1**	0	1
Customer 2	1	0	1	1
Customer 3	0	1	0	0
Customer 4	**1**	**1**	1	0

The amount of concepts a concrete formal context holds is definitive and immune to permutation [10]. In other words, any formal context with the same structure – that is, with the same object-attribute relationships – will always hold the same amount of concepts with the same concept hierarchy. Also, the maximum amount of concepts in a formal context is 2^n, where n is the lower amount of either attributes of objects – whichever is smaller [10]. In the financial institution example, this would be $2^4 = 16$. As can be understood, the amount of concepts grows exponentially, and a relatively small context of 100 objects and 20 attributes could theoretically have $2^{20} = 1\ 048\ 576$ concepts. That is, however, only theoretically likely to occur; actual contexts will generally hold far less concepts, as can be seen also in Chapter 3.1.

An important property of the formal concepts is conceptual hierarchy: formal concepts are naturally ordered, having the subconcept-superconcept relation [10], [12]. The ordered set of all formal concepts of a formal context is called a **concept lattice**, with all concepts having a partial order relation ($\leq$) [5], [6], [12]. Furthermore, any two concepts have a mutual superconcept – **supremum** – and a mutual subconcept – **infimum –** indicating that the concept lattice is a complete lattice [6], [10]. The concept lattice is most commonly depicted using a line diagram, where the more general superconcepts are visualized above the less general subconcepts, and connections are indicated via lines between the concepts. For the purpose of reference, all concept lattices in this thesis were drawn using the program Concept Explorer [22].

Figure 2.1 depicts all the concepts from the context introduced in Table 2.1. A concept is marked by a round node, and the connections are marked by a straight edge going from one node to another. The top most concept is the *supremum*, and the bottom most concept is the *infimum*. The *supremum* contains all of the objects of the context, as can be seen when following the descending lines (the *extent*): the concept is superconcept to all other concepts in the lattice. In this example, the *supremum* does not contain any attributes, because there is no product in the context which all customers would have. In such cases, it is said that the *intent* of the *supremum* is an empty set. This is common in most datasets, but may not always hold true, as an attribute may have a relation across all objects. Similarly, the *infimum* contains all of the attributes, as can be seen when following the ascending lines (the *intent*). Again, the *infimum* does not contain any objects in this example, since in the example there is no customer who has a relation with all the products in the context, but in general it is possible and thus an *infimum* containing an object in its *extent* indicates that the object contains all the attributes of the context.

A *concept lattice* provides a visual overview of the context’s underlying concepts and the relations between the concepts, which may not otherwise be so obvious from looking at the context. For example, when examining Figure 2.1, following the path down from the concept marked with “Credit Card”, it can easily be noticed that there are two customers in the concept’s extent: Customer 1 and Customer 2. Indeed, looking at the context from Table 2.1, one can see that those two customers have a credit card. It can also be observed that both Credit Card and Home Loan are marked as subsets of the concept holding the Deposit Account notation. That implies correctly that all customers who have either a credit card or a home loan also have a deposit account in our context. Also, it is important to note that concepts may not always match exactly to an attribute or an object, as can be seen by examining the concept from Table 2.2. In Figure 2.1, the concept is the left-most concept in the centre row and it has no denominators, because no customer or product matches to that concept in its full.

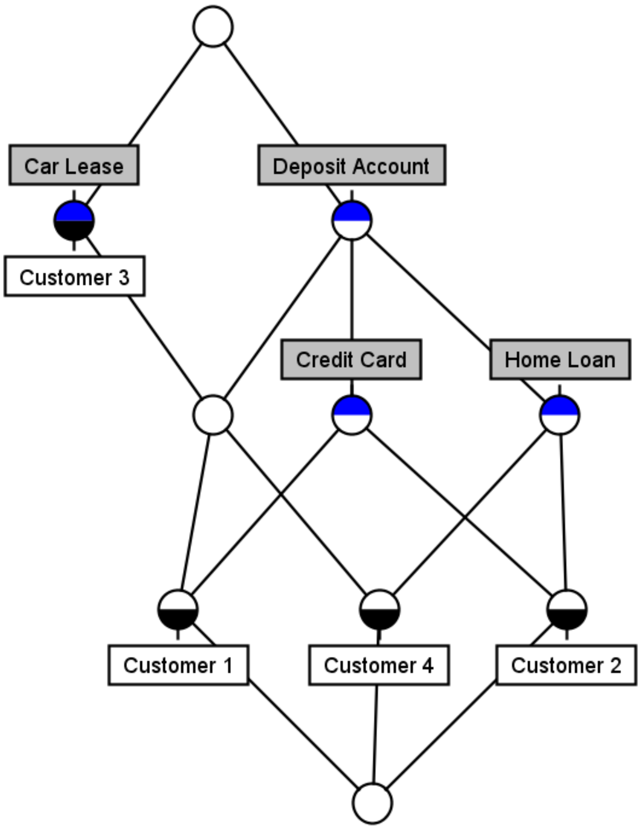


Figure 2.1. Concept Lattice of the Context from Table 2.1.

2.1.1 FCA Applications

Already from its early days, FCA has seen application in various areas. Some of the early research fields included linguistics, text mining, association rule mining (ARM), data analysis, conceptual knowledge processing, and software engineering [5]. More recent years have added further to the spectrum, with additions such as security analysis, web mining, social media mining, software testing, e-learning, bioinformatics, image processing and psychology [6], [8], [23], [24], [25]. Such a wide array of subjects indicates the usefulness and applicability of FCA.

Albeit this thesis does not deal with ARM specifically, a short overview of the notion will be given here to help the reader in understanding some of the analysis and results parts of this thesis, as each of the business-related datasets introduced in Chapter 3.1 is a so called *market basket* – a matrix table of purchases (rows) and products (columns). ARM's main goal is to find popular patterns in such datasets, most commonly so that if the purchase of one product strongly indicates the purchase of another product, then this connection is presented to the analyst of the dataset [19], [20]. While this methodology is useful for drilling down and understanding relationships between specific products, it is computationally heavy, and may provide a long list of associations which do not provide great value to the analyst in understanding the dataset as a whole [20], [21]. Thus, it is expected that if the results and analysis in this thesis will provide similar results as ARM, giving the analyst many associations of products, then the associations have to be very strong and justified in order to indicate value. That's because, as listed in the problem statements in Chapter 1.2, the aim in this research paper is to find methods for quick and extensive overview of the dataset, not a drilled down strong correlation as would be expected from an ARM-focused study.

2.1.2 FCA Drawbacks

Having introduced the FCA theory and discussed briefly its applications, it is time to also visit some of its drawbacks. As discussed previously, a concept lattice is usually a much quicker and clearer way for the viewer to comprehend concepts and their relations, especially when compared to a matrix data table. However, a crucial impediment of the concept lattice is its scalability – as noted in Chapter 2.1, the amount of concepts in a context can grow exponentially as the context's size increases. The financial institution example used previously has 4 objects, 4 attributes and 10 concepts, from a potential maximum of 16 concepts. A sparse dataset introduced and used later in this paper, containing grocery store data of 75 transactions (objects) and 99 commodity groups (attributes) has a concept count of 290, which might not seem like a lot, but as can be seen from Figure 2.2, the concept lattice is unreadable.

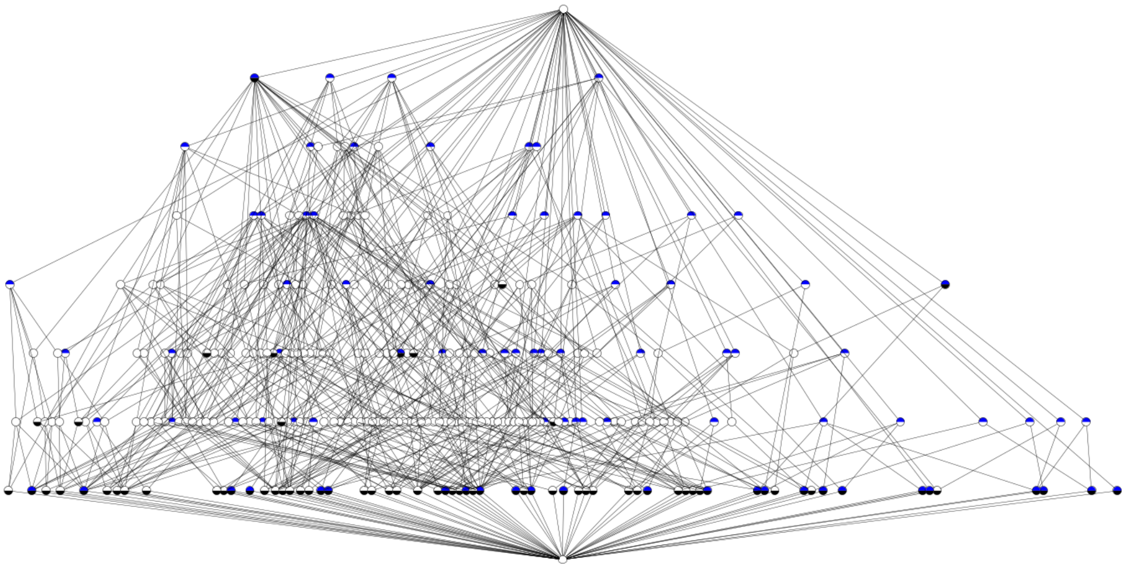


Figure 2.2. Concept Lattice of 290 Concepts.

Naturally, one could drill down to specific concepts, and investigate the relations and implications of a specific subset, but this would be cumbersome and sluggish, and it would not give the data miner the quick overall picture of the dataset, which is largely the purpose of the lattice.

The issue is not new to FCA [8], [13], [14], [15], and various methods have originated to tackle this. Some of the most published ones include:

- **Classification methods** – these methods drastically reduce the amount of concepts by ways of building classes from a set of examples, and using the generated classes as a template in the generation of new examples [26], [27]. These methods have been found to be especially useful in the field of machine learning [28]. However, in data extraction, one should be careful in utilizing these methods, as a large portion of relevant data may be left out already at the creation of the concepts.

- **Nested line diagrams** – this method is purely for simplifying the visualization of the concept lattice by nesting portions of the original lattice together, to form sets of smaller diagrams [5], [7], [12], [29]. The sets of smaller diagrams can, in turn, be presented as a lattice. While useful for visualization, it needs to be noted that such a solution would still hold all concepts of the context and thus would not be useful in determining which pieces of the data are most relevant.

- **Concept indices** [16], [17], [30]. Indices are used as a grading system for concepts, and based on the hierarchy provided by the index, only the best performing concepts are kept. These concepts can be used as a basis for generating a new, smaller and better readable lattice.

With the goal of finding the most relevant and valuable pieces of data in a business context, the notion of concept indices is the preferred choice, as it considers all of the initial data, while compressing the data into a visually comprehensible form. Consequently, the next chapter will focus on introducing various algorithms designed for calculating indices and allowing for a ranking of the concepts within a context.

2.2 Concept Indices

As described in the previous sections, the number of concepts in a formal context may become too large for a viewer to comprehend. In order to extract meaning out of a dataset, some sort of filtering needs to be applied to limit the number of concepts. Understandably, the filtered concepts should be the most interesting, relevant and representative of the dataset. This section will introduce routines for finding the most relevant concepts of a context.

Various algorithms have been developed throughout the years for finding the most meaningful concepts out of a concept set by grading the concepts based on some predefined metrics. A selection has been made in order to fit the constraints of this paper. Two indices with a widespread use and proven efficiency were chosen, namely the **Stability** index [16], [17], [23], [30], [31], [32], [33], [34], [35] and the **Iceberg Lattice** [10], [23], [30], [36], [37], [38]. A few less known but interesting methods were chosen as contrast: **Local Stability** [39] and a **Monotone Systems Method** [40], [41]. Some notable indices which were left out include the MONOCLE method [16], [41], which proved effective in smaller sample datasets but had to be removed due to performance restrictions, and probability and separation, which have good coverage in the academia [16] but have shown to be outperformed by the stability index in comparisons [17].

The implementations of the algorithms shown in the following subchapters are done in **Python 3.6.3** using the **Concepts 0.7.12** module [42].

2.2.1 Concept Stability

Concept Stability (S) of formal concepts is a measure which has been proven effective in dealing with contexts which include noisy data [17], [30]. The intensional stability index used in this work indicates how much the concept intent depends on particular objects of the extent [32]. In other words, it is the probability of preserving the intent of the concept, after removing an arbitrary number of objects from the context [31]. Data which is noisy is more likely to be removed in such cases, and thus the concepts that have higher stability are possibly the ones the viewer may find more interesting and relevant. The implementation in Figure 2.3 is author's implementation based on the pseudocode provided in [31].

```
Stability = {}
CountedSubsets = {}
for i in l:
    TotalSubsets = 2**len(i.extent)
    subsum = 0
    for j in l.downset_union([i]):
        if j != i:
            subsum = subsum + CountedSubsets[j]
    FinalSubset = TotalSubsets-subsum
    CountedSubsets[i] = FinalSubset
    Stability[i] = FinalSubset/TotalSubsets
```

Figure 2.3. Stability Algorithm.

The algorithm moves through the lattice starting from the bottom concept, calculating the subset amount and stability for the concept. The algorithm goes up to the next concept once all of the concept's subconcepts' stabilities have been calculated – this is not visible in Figure 2.3, as the lattice object *l* is pre-sorted. During the calculation, the amount of subsets from subconcepts is deducted when calculating a concept's subset, in order to subtract the subsets of the concept intent which do not generate the concept extent [31]. The resulting stability of a concept is a value from 0 to 1, with 1 showing the highest stability.

2.2.2 Iceberg Lattice

Iceberg Lattice (IL) is one of the most widely addressed methods for keeping only the most relevant concepts in a lattice [10], [34], [36], [37]. The method is applied by setting a limit on the percentage or number of objects that a concept should cover [10], in other words: it is a threshold on the number of objects that should be in the extent of the concept. The implementation in Figure 2.4 was done based on the explanation in [39].

```
IcebergLattice = {}
for i in l:
    IcebergLattice[i] = len(i.extent)/len(c.objects)
```

Figure 2.4. Iceberg Lattice Algorithm.

The name of the method comes from the fact that the most general concepts, which usually have the largest extents, are in the top part of the concept lattice, giving the viewer sort of an iceberg view of the lattice diagram [39]. It is also important to note that a bottom concept needs to be added to the iceberg view of the lattice in order to make the view a complete lattice again [10], since only having the concepts from the top of the lattice would not have a bottom closure.

2.2.3 Local Stability

Local Stability (LS) is an add-on to the calculation of the concept stability, with little evidence of use in research papers thus far. However, it is interesting because of its inherent method of eliminating residual concepts. Namely, applying LS maintains only the concepts which have stability at least as high as all of its upper and lower neighbours [39]. This means that the prerequisite for applying LS is the calculation of the stability index. The reasoning is that maintaining the concepts which have only the highest stability may give a one-sided view of a dataset, as concepts which are relevant but happen to have low stability due to context structure are removed. LS ensures that those concepts remain available for the viewer. The implementation in Figure 2.5 is built on the definition in [39].

One notable property of the LS is that, unlike the rest of the algorithms included in this study, the LS only keeps a certain amount of concepts which pass the filtering. This means that the final number of concepts is possibly far less than the total number of concepts in the dataset. This feat can be observed also in the results part in Chapter 5.1.

```
LocalStability={}
for i in Stability:
    a = True
    for j in i.upper_neighbors:
        if Stability[j] > Stability[i]:
            a = False
            break
    if a == True:
        for j in i.lower_neighbors:
            if Stability[j] > Stability[i]:
                a = False
                break
    if a == True:
        LocalStability[i] = Stability[i]
```

Figure 2.5. Local Stability Algorithm.

2.2.4 Monotone Systems Method

A monotone system is described as a set of elements with a weight function, where the weight function measures the importance of the elements for the system [41]. One such measure for importance employed within monotone systems is called conformity [40], which has been used in FCA restrainedly thus far [41]. The premise of the Monotone Systems Method (MSM) employed here is influenced by the conformity scale: first, the weights of each attribute and object based on its frequency in the dataset are calculated; then, the concepts are ordered based on the area they cover within the context, taking into account also the uniqueness of the concept's coverage [41]. In other words, the highest ranked concepts are the ones that cover a large part of the dataset, but do not share this coverage with too many other concepts from the context. The *best* concepts are the ones which have the highest weights score.

The implementation in Figure 2.6 is done based on the example of [41]. The algorithm first calculates the sum of concepts which have at least one element in both extent and intent. The frequencies of each object and attribute within the concept set is calculated, which give the multipliers for the weight function. Finally, the weight for each concept is calculated by multiplying the object and attribute multipliers, taking into account also the uniqueness of the concept.

```
cCount = len(l)
for con in l:
    if (len(con.extent) == 0) or (len(con.intent) == 0):
        cCount -= 1
propvalues = []
for i in c.properties:
    curval = 0
    for a in l:
        if (len(a.extent)) > 0:
            if i in a.intent:
                curval += 1
    propvalues.append([i, curval])
objvalues = []
for i in c.objects:
    curval = 0
    for a in l:
        if (len(a.intent)) > 0:
            if i in a.extent:
                curval += 1
    objvalues.append([i, curval])
ConformityScale = []
for a in l:
    objsum = 0
    atrsum = 0
    for i in a.intent:
        j = c.properties.index(i)
        objsum = objsum + (cCount - propvalues[j][1]) + 1
    for i in a.extent:
        j = c.objects.index(i)
        atrsum = atrsum + (cCount - objvalues[j][1]) + 1
    ConformityScale.append([a, objsum*atrsum])
ConformityScale = sorted(ConformityScale, key=lambda x: x[1],
reverse=True)
```

Figure 2.6. Monotone Systems Method Algorithm.

2.3 Similar Works

One of the key motivations for the subject of this thesis was that no reviewed or known paper had examined the same problems via the same methodology as listed in this thesis. However, there exists various scientific research where FCA algorithms are compared or concept interestingness in measured. The following is a short summary of some of the most relevant and related works:

- On Interestingness Measures of Formal Concepts [16]

  - Summary: The article investigates indices for calculating interesting concepts and compares the indices based on efficiency of computation and applicability to noisy data. An overview of 20 indices is given, together with a comparison between their key features and calculation complexity. The datasets are generated randomly, using different densities, and the comparison of interestingness is done by comparing correlations between the different indices.

  - Differences:

    - A larger number of algorithms (indices) is examined;

    - Contexts used are artificial, not real life;

    - Concept interestingness is measured based on similarity or correlation of the different algorithms.

- Approaches to the Selection of Relevant Concepts in the Case of Noisy Data [17]

  - Summary: The aim of the article is to compare how FCA indices deal with noisy data. Four datasets and their lattices are generated, and different levels of noise are introduced into the datasets. Then, three indices - stability, separation and probability - are applied on the noise-included datasets to see how well the calculated concepts and the respective lattice corresponds to the original lattice of the dataset without noise. It is found that stability is the most efficient in correcting for the noise in datasets.

  - Differences:

    - Focus only on noise-filtering;

    - Contexts are generated, not real life.

- Basic Level in Formal Concept Analysis: Interesting Concepts and Psychological Ramifications [18]

  - Summary: Taking the notion of basic level from psychology, the authors implement five different indices for compressing a set of concepts into more general ones which would better explain the whole dataset. The datasets used in the experiments are both real life and generated. In some cases, the indices calculated quite similar results, and the correlation between the indices was examined in more detailed. The indices are found to produce more natural and perhaps interesting concepts.

    - Differences:

        - Only a novel and specific set of indices is implemented and compared;

        - Algorithms are compared based on similarity or correlation.

Considering the above, the uniqueness of this research can be outlined as follows:

- Comparison of algorithms only on real-life datasets. Further, partially executing the algorithms on business-relevant datasets – namely, market baskets;

- Using context coverage as a comparison metric.

Therefore, the research is deemed relevant. Having explored the background of the FCA theory, its applications and drawbacks, and potential remedies to the drawbacks via concept indices, the next chapter will now introduce the datasets on which the research will be conducted on, as well as the methodology principles of the study.

3 Research Methodology

Having introduced the underlying theory and algorithms which will be used for conducting the research, this chapter will explain the applied methodology for validating and analysing the results. The chapter is divided into three parts: first, the **datasets** will be introduced and explained. Secondly, the notion of the area of a concept, also called the **concept coverage** will be examined. Finally, as also disclosed in the theory part – the **number of concepts** in a lattice can easily become very large, and as a result, analysing the dataset can become enigmatic. Thus, the third part of the chapter will establish some thresholds on what would be an optimum level of concepts to observe in a dataset. Altogether, by the end of this chapter, the foundation for conducting the research is in place, by having introduced what the research will be conducted on (datasets), what will be measured and how (concept coverage), and what will be the measuring points (number of concepts).

3.1 Datasets

In order to assertingly answer the research questions posed by this thesis, several datasets would need to be studied. The aim in choosing the datasets for conducting this research was to have substantially large datasets from different perspectives. In total, five datasets were chosen. As one of the goals of the paper is to verify the business applicability of FCA algorithms calculating concepts with high coverage, the selection preferred datasets originating from a business environment: three of the five datasets come from the retail world and comprise of **market basket data**. The other two datasets, which are not originating from a business environment, involve a set on voting records from the United States congressional voting, and a set on student performance from two Portuguese schools. However, the two datasets can of course be thought of as a model for how a customer segmentation dataset might look like. Thus, one could argue that including those two datasets is also well applicable to a business setting.

The datasets' descriptions and conducted pre-processing steps are given in the following subsections. The metadata in Table 3.1 provides the reader with info on the size of a dataset, a dataset's density and an indication of potential similarity of the data within a dataset.

Table 3.1. Metadata of the Datasets.

Dataset	**# attributes**	**# objects**	**# 1s**	**Coverage by 1s**	**# concepts**	**Concepts-to-1s ratio**
1984 House Votes	18	435	3 856	49.25%	10 644	2.76
Student Performance	22	395	4 573	52.62%	64 422	14.09
Basket Example 1	99	75	408	5.49%	290	0.71
Basket Example 2 – E-Commerce	70	1 134	2 259	2.85%	851	0.38
Basket Example 3 – Instacart	134	600	4 750	5.91%	18 897	3.98

3.1.1 1984 House Votes

The original dataset holds the votes of the United States House of Representatives Congressmen from 1984 on 16 key votes as identified by the Congressional Quarterly Almanac [43]. The following activities were done during the pre-processing of the data:

- The first attribute, whether the Congressman was Democrat or Republican, was transformed into Boolean data type;
- The votes for “yes” were counted as positive answers, marked by 1s. The votes with unknown or “no” were counted as negative answers.

3.1.2 Student Performance

This dataset represents information and grades of students in secondary education from two Portuguese schools [44]. In the original dataset there are two distinct sets: one for Mathematics and the other for Portuguese classes. The attributes in both sets are identical, containing some Boolean, a few nominal, and mostly numeric type attributes. The pre-processing for the Student Performance dataset included the following:

- Only the Mathematics set was kept;
- The following attributes were removed from the dataset: school identifier, student age, parents’ education, parents’ jobs, reason for choosing that particular school, first period grade, second period grade.
- All remaining attributes containing numerical or nominal values were transformed into Boolean values. For example, final grades below 10 were replaced by Boolean “Failed final” attribute.

3.1.3 Basket Example 1

The first dataset of the business-related datasets is a small sample dataset of retail sales, including information about the sales and information about the products [45]. The product list is matching to a grocery store's product list. It needs to be noted, however, that there is little information present about this dataset, thus some reservation might be needed in assessing the results from this analysis. Based on a critical judgement, the dataset appears to be coherent. Pre-processing which was done on the dataset:

- Keeping only defined product category and basket matches;
- Removing duplicates.

3.1.4 Basket Example 2 – E-Commerce

The dataset contains all transactions of a UK-based online retailer from a period of time in 2010 and 2011 [46]. Main articles of the retailer are all-occasion gifts, with many of the customers being wholesalers. The initial dataset is relatively large, containing over 500 000 rows. Thus, in addition to pre-processing, a smaller sample of the dataset was chosen. A mixture of representative and random sample was used to ensure optimal data quality. The pre-processing activities done before sampling were as follows:

- Rows with missing values were removed;
- Duplicate orders were removed;
- Orders which were cancelled were removed.

3.1.5 Basket Example 3 - Instacart

Instacart is a service in major cities in the US, where the consumer can shop for groceries in a variety of stores via the Instacart app and schedule a delivery of the groceries to their door-step [47]. In May 2017, Instacart open sourced a dataset of 3 million orders [48]. The dataset as a whole, according to Instacart, is not representative of their products or users [49]; but it nevertheless provides a highly detailed and large amount of data for analysis. As with the previous dataset, this dataset was also sampled to allow for a reasonable amount of concepts to emerge from the dataset. The full list of pre-processing activities were:

- Instead of products, aisles were mapped to specific orders to reduce noise and variance in data;

- Duplicate order-aisle mappings were removed.

3.2 Concept Coverage

The notion of the concept coverage, also called the area of a concept, is relatively simple to understand – it is the sum of 1s (“one-s”) in the concept. The 1s are important, because they determine the existing relation between the object and the attribute. The absence of 1 means that there is no linkage between the object and the attribute, and while the lack of relation may have implications of its own, it is generally not an insightful statement. It makes sense from a perspective of trying to understand the data and its relations to look at the 1s within the context. Thus, the concept coverage measures the number of the relations described by the concept. In other words, how many relations the concept covers. An example of concept coverage can be observed in Table 2.2: the coverage of the highlighted concept is 4, as the concept covers four 1s of the dataset.

Surprisingly, concept coverage has appeared in relatively few research papers thus far – that includes not only FCA, but also its related fields – at the same time, no paper has mentioned any shortcomings of looking at the concept coverage. The following is a brief summary of the most known mentions of the concept coverage and its importance:

- In the field of clustering, the notion of Coverage Density, which in essence measures also the 1s of important data clusters, is used as a way to discover the optimal number of clusters a dataset should produce [50], [51].

- In the domain of role mining, which is a subdomain of data mining focusing on finding patterns within the rights and roles assigned to users of a system, a similar notion of finding areas covered by different roles is used as a way to distinguish the most dominant roles in the system [52].

- In FCA-related research, the concept coverage has been used as a way to better describe the importance of concepts, indicating that a larger coverage of 1s gives prominence to concepts which are more relatable to common notions [41].

> However, there has been little further exploration into what the area of a concept might actually denote in relation to FCA concepts.

A concept's coverage is useful for understanding how big portion of the dataset the single concept covers. In order to measure the coverage across a whole dataset, the **cumulative coverage** of concepts in regards to the uncovered portion of the dataset will be examined. As an example, another concept from the example introduced in Chapter 2.1 is depicted in Table 3.1. This concept covers also four 1s, as the example in Table 2.2. If the concept from Table 3.1 is calculated *after* the concept from Table 2.2, then the 1s already covered by concept from Table 2.2 (underlined in Table 3.1) cannot be counted towards the cumulative coverage of the dataset. Thus, while the concept coverage of the concept in Table 3.1 is four, the addition to cumulative coverage is three – three 1s which were not covered previously.

Table 3.2. Cumulative Coverage Example. Already counted 1s are underlined.

	Deposit Account	**Car Lease**	**Home Loan**	**Credit Card**
Customer 1	**1**	1	0	**1**
Customer 2	**1**	0	1	**1**
Customer 3	0	1	0	0
Customer 4	1	1	1	0

Finally, in order for a metric to be effective and articulative, it needs to have a method for benchmarking – in other words, there has to be a threshold for using the metric. This will be discussed in the following section.

3.3 Number of Useful Concepts

Line diagrams have been proven to be an effective way for visualization of information [53]. However, as introduced in Chapter 2.2, the number of useful or relevant concepts is important in visualization, because the number of elements to be observed may be incomprehensibly large. Thus, in order for data mining to be effective, there should be set a maximum number of concepts which would keep the context and the lattice still perceivable to the viewer. Naturally, the data mining tasks may be of different urgency, and therefore 3 different levels of maxima will be set, to investigate instances of both more immediate data visualization needs and more exhaustive ones.

Research on human perception and information processing has shown that 8-10 elements are perceivable to an average human, depending of the stimuli [54]. Furthermore, it is natural that if a person is familiarized with the subject, then the amount of perceivable elements would increase. Based on this information, **10 concepts** is taken as the smallest threshold; this should be the number of concepts that would enable the dataset's implications to be immediately comprehensible to the viewer. The largest threshold to observe would be **50 concepts**: while not based on empirical evidence, it is deemed as the largest possibly comprehensible number of concepts in a lattice for a thorough analysis project. As an example, if there was an important decision to be made and the highest possible confidence needed, 50 concepts would be feasible for an experienced data miner to analyse still in a quick and relatively efficient manner. Finally, the threshold of **30 concepts** is taken as a mean of the previous two thresholds, to showcase a possible standard circumstance of a viewer finding an optimum between speed and confidence of the data analysis task.

Thus, to summarize, the setup for conducting the research would be as follows:

1) Execute concept indices from Chapter 2.2 on the concepts of the datasets introduced in Chapter 3.1, producing 5 ordered concept sets for each of the 5 datasets;

2) Calculate the concept coverages for the top 50 concepts from each of the concept sets;

3) Calculate the cumulative coverages of the datasets at each concept;

4) Compare the results at the 10-, 30- and 50-concept marks.

The results based on this outline will be explored in Chapter 5. The next chapter introduces the Greedy Coverage, an algorithm developed by the author during the research in an effort to more efficiently calculate concepts with high cumulative coverage.

4 Greedy Coverage

Following the foundation for the metrics and measurements of the research conducted in this paper, and some initial executions of the existing FCA algorithms, it became evident that a possibly more efficient algorithm could be constructed to find the concepts covering the whole context in a fewer amount of steps. This algorithm is introduced in this chapter and is part of the original work done by the author.

4.1 The Algorithm

One of the main research questions of this thesis is to find the algorithm for covering most effectively the biggest portion of the data. During the research, it was realized that a greedy algorithm which would choose concepts based on their addition to the cumulative coverage of the uncovered dataset may turn out to be a feasible alternative to the indices introduced in Chapter 2.2. Thus, the Greedy Coverage algorithm was developed by the author. The premise of the algorithm is as follows:

1) Calculate the coverage of all concepts within the context;

2) Choose the concept with the highest coverage of 1s;

3) Remove the concept and its covered 1s from further calculations;

4) Recalculate the coverage for all remaining concepts;

5) Repeat steps 2-4 until full coverage of the context has been achieved.

The implementation in Python 3.6 using Concepts 0.7.12 module is shown in Figure 4.1. Some of the variables are explained below to assist in understanding the algorithm:

- GreedyCoverage – holder for the concepts and their respective cumulative coverage values;

- CoverageList – transitory holder for the concepts, used for choosing the next concept to be added to GreedyCoverage list;

- BoolsList – holder for uncovered 1s in the dataset.

```
GreedyCoverage = []
CoverageList = []
BoolsList = []
rowCounter = 0
totalOnes = 0
for val in c.bools:
    totalOnes += sum(val)
for i in l:
    z = [(d[0].index(p), d[1].index(q)) for p in i.extent for q
in i.intent]
    CoverageList.append([i,z])
for i in CoverageList:
    i.append(len(i[1]))
for j in c.bools:
    columnCounter = 0
    for k in j:
        if k == True:
            BoolsList.append((rowCounter, columnCounter))
        columnCounter +=1
    rowCounter += 1
CoverageList = sorted(CoverageList, key=lambda x: x[2])
GreedyCoverage.append(CoverageList.pop())
boolCount = GreedyCoverage[-1][2]
while boolCount < totalOnes:
    for i in GreedyCoverage[-1][1]:
        if i in BoolsList:
            del BoolsList[BoolsList.index(i)]
    for j in CoverageList:
        pointList = []
        for k in j[1]:
            if k in BoolsList:
                pointList.append(k)
            else:
                j[2] -=1
        j[1] = pointList
    CoverageList = sorted(CoverageList, key=lambda x: x[2])
    GreedyCoverage.append(CoverageList.pop())
    boolCount += GreedyCoverage[-1][2]
```

Figure 4.1. Greedy Coverage Algorithm.

Due to its proof-of-concept status, possible optimizations to the algorithm have not been considered within the scope of this paper. Based on the findings from the next chapters, a possible extension to the algorithm and its output is discussed in Chapter 7.1.

5 Results and Interpretation

This chapter presents the results of the research done. The first part looks at the results on each of the datasets separately. The second part of the chapter summarizes the findings and presents some interpretations.

5.1 Results of Algorithm Executions

The research was done on the five datasets introduced in Chapter 3.1, using the five algorithms introduced – the four existing FCA algorithms introduced in Chapter 2.2, and the fifth – Greedy Coverage algorithm – introduced in Chapter 4. All algorithms were run on all datasets, producing an ordered list of concepts for all datasets. Then, calculation was done for determining how big area of the context – or how many of the 1s in the context – are covered in a cumulative manner. As an example, if the 2nd concept covers some 1s which were covered also by the 1st concept, then those 1s do not count towards the cumulative (dataset) coverage.

Below is a short legend for understanding the graphs which depict the results:

- X-axis: **number of concepts**;
    - Goes from 0 to 50 concepts;
    - 10-, 30- and 50-concept marks are indicated by a vertical line.
- Y-axis: **number of 1s**;
- Lines: **the algorithms**:
    - S = Stability;
    - IL = Iceberg Lattice;
    - LS = Local Stability;
    - MSM = Monotone Systems Method;
    - GC = Greedy Coverage.

5.1.1 1984 House Votes

The first algorithm executions were done on the 1984 House Votes dataset. Recalling from Chapter 3.1, this dataset is quite dense, having a high number of 1s – 49.25%. The number of concepts is also quite big with 10 644. The graph of results from the algorithm executions can be seen in Figure 5.1.

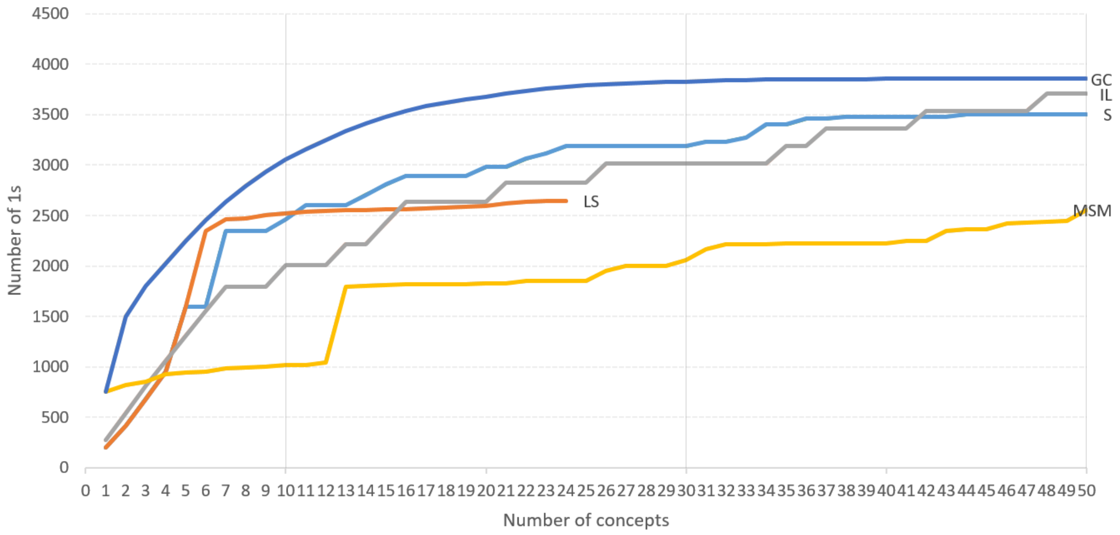


Figure 5.1. The Results of the 1984 House Votes Dataset.

The clear winner in this dataset is the GC algorithm – it covers the most concepts in all checkpoints. In fact, the algorithm covers the whole dataset by the 41st concept. This is quite remarkable, considering the total amount of concepts. The weakest algorithm here is the MSM, which has high coverage with the first concept, but then has a low and steady climb for most of the observed experiment, except the 13th concept, when there is a high spike upwards. Another observance is that the IL is noticeably behind S at the 10-concept mark, closes the gap modestly by the 30-concept mark and surpasses S by the 50-concept mark. LS performs strong in the range of 5-10 concepts, but flats out thereafter and ends its run at the 24th concept – this is expected behaviour, as described in Chapter 2.2.3, the LS algorithm does not include all of the concepts present in the dataset.

5.1.2 Student Performance

The Student Performance dataset had the highest density – 52.62% - and the highest number of concepts, with a whopping 64 422. Looking at the results in Figure 5.2, the GC again outperforms other algorithms and comes out on top. Further, the GC also manages again to reach the total coverage of the context before the 50-concept mark: in this context, all 1s are covered by the 46th concept. In comparison with other algorithms, the GC covers consistently 60-70% more 1s throughout the observed range. The S and IL algorithms perform quite equally until the 25th concept, but then S shows better results at both 30-concept and 50-concept mark. This time, the MSM is not as far behind as in the previous dataset, but it is still clearly losing to both S and IL at all checkpoints. Finally, the LS algorithm, while having more than 50 concepts in this context, performs poorer than any of its rivals at all 3 checkpoints.

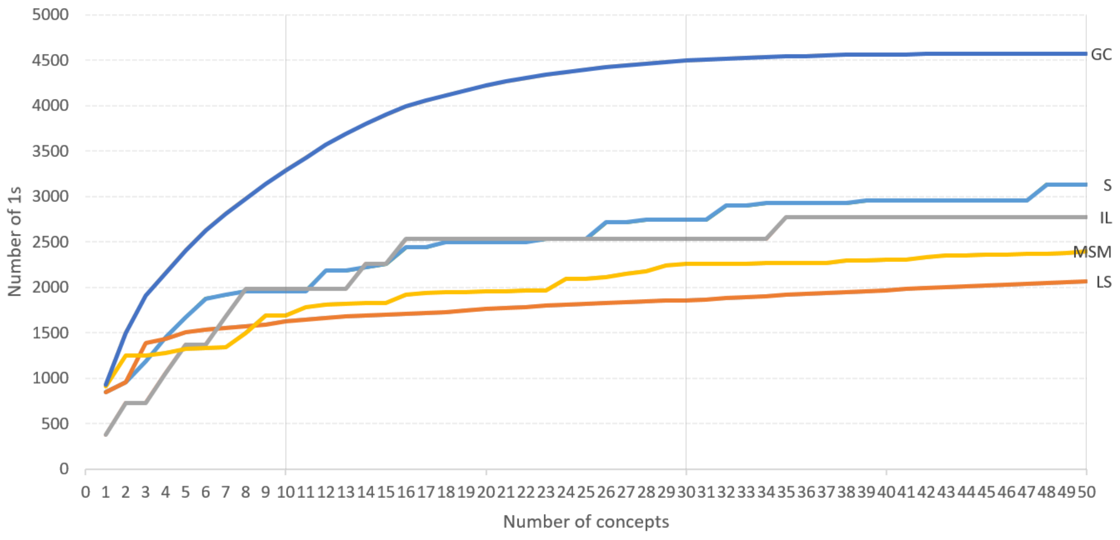


Figure 5.2. The Results of the Student Performance Dataset.

5.1.3 Basket Example 1

The first market basket example is the smallest studied dataset, both in terms of 1s and concepts. Analysing the results from Figure 5.3, the GC is again the algorithm showing the best results at all checkpoints. In fact, the difference between GC and the rest of the algorithms is increasing as the number of concepts increases. This indicates that other algorithms are calculating concepts which are similar to each other, and thus add diminishingly to the cumulative coverage. The second best algorithm this time at the 10-concept mark is the IL, while at 30- and 50-concept mark the second best result is shown by S, albeit both mentioned algorithms are comparingly even throughout the experiment. The MSM shows poor results in the 10-concept mark, but then closes the gap and achieves almost equal results to S and IL by the 50-concept mark. LS algorithm fails to even deliver 10 concepts, thus it is unusable on this dataset for the purposes of this research.

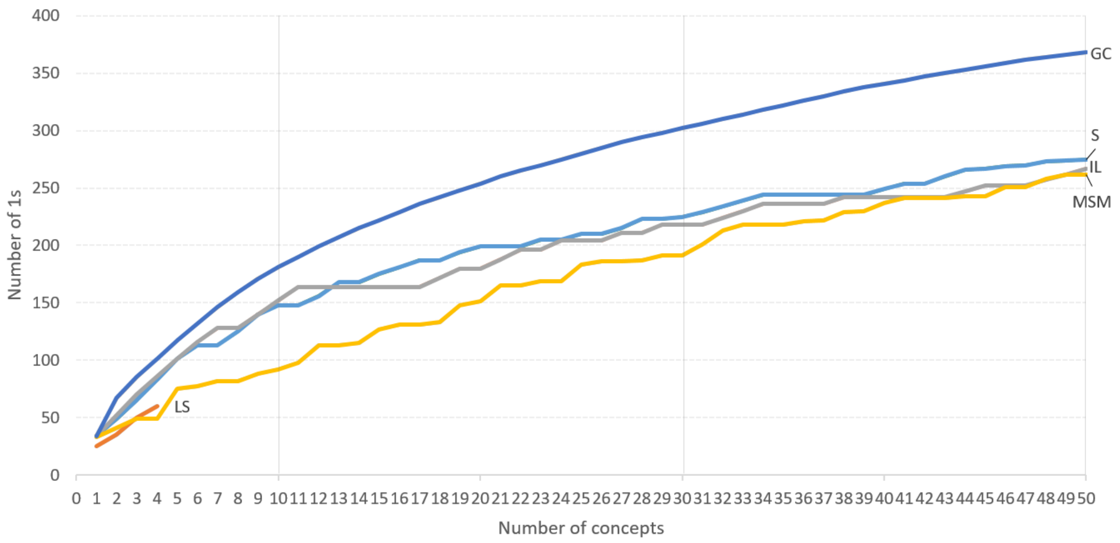


Figure 5.3. The Results of the Basket Example 1 Dataset.

5.1.4 Basket Example 2 – E-Commerce

The E-Commerce Dataset has the lowest density of 1s (2.85%) and the lowest concepts-to-1s ratio (0.38). Furthermore, the dataset has the highest number of objects, giving the lowest amount of 1s per row – in average, two 1s per row. This indicates a low amount of combinations between different products. The results can be observed in Figure 5.4.

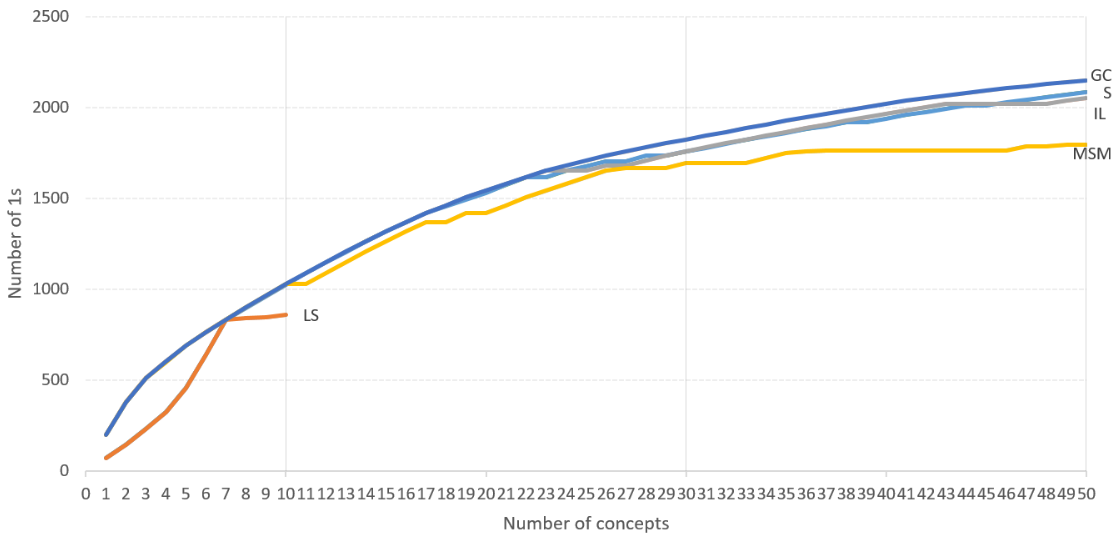


Figure 5.4. The Results of the Basket Example 2 - E-Commerce Dataset.

The results from the dataset exhibit the most similar results between algorithms examined thus far. At the 10-concept mark, GC, IL and MSM show equal results, and the S algorithm is underperforming the top 3 only by a small margin. The clearly weakest link is the LS, which also has the last calculated concept at the 10-concept mark. At the 30-

concept mark, the GC shows the best results, S and IL share the 2nd place, and MSM is starting to lose its momentum and begins to flat out. The 50-concept mark sees also the GC having the highest number of covered 1s, but the difference with S and IL is very small in this dataset. MSM, however, is clearly 4th at this checkpoint.

The possible reasons why this dataset shows different results than examined thus far in the other datasets could be, as mentioned, the low coverage of 1s, the low concept-to-1s ratio, and low amount of 1s per row. Clearly, there has been an impact for all algorithms – a possible learning, thus, would be that for datasets with such structure, S or IL algorithms would give comparable results to GC and may be preferred choices if they provide computational advantages. A more thorough understanding, however, would be a research topic for a different paper.

5.1.5 Basket Example 3 - Instacart

The Instacart dataset has the most 1s – 4 750 – and the highest number of concepts across the three basket example datasets – 18 897. The results in Figure 5.5 display again a strong lead for the GC algorithm at all 3 checkpoints.

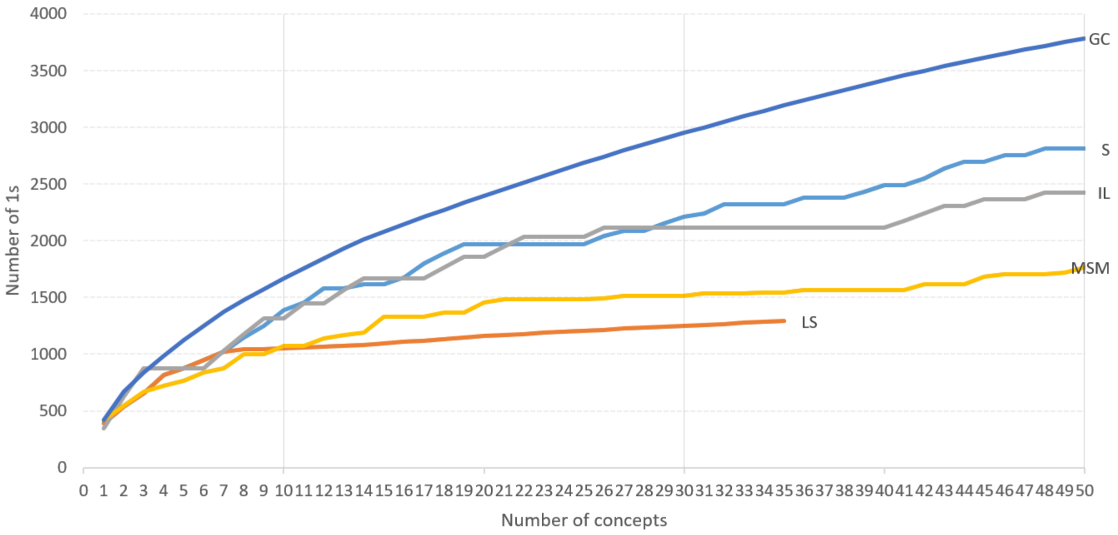


Figure 5.5. The Results of the Basket Example 3 - Instacart Dataset.

Similarly as displayed in the Basket Example 1, the GC lead is increasing as the number of concepts increases, indicating that at smaller number of observed concepts, the algorithms are more comparable than if the number of concepts is increased. The 2nd position at all checkpoints is held by the S algorithm, while up until the 30-concept mark the IL algorithm, which is 3rd in all measured checkpoints, shows quite similar results to S. MSM is 4th in all of the checkpoints, with relatively poor results – at the 30-concept mark, only 51% of the number of 1s covered by GC are covered by the MSM, deteriorating to 47% by the 50-concept mark. Nevertheless, the worst results are displayed by the LS algorithm, which has the least covered 1s at the 10- and 30-concept mark, and the algorithm generates only 35 concepts in this example.

A notable peripheral observation, however, can be made in this dataset: at the 3rd concept, IL shows greater cumulative coverage than GC. This is, in fact, the only measuring point out of the 250 across the 5 datasets where GC is outperformed. Some possible learnings are investigated in Chapters 6.1.2 and 6.1.3, but one clear takeaway is that, depending of the goal concept by which to calculate coverage, a more efficient algorithm than GC is plausible. This is also briefly introduced as a possible future work in Chapter 7.1.

5.2 Interpretation and Summary of Results

Having executed the algorithms on all of the datasets and explored the individual results, it is time to make some conclusions. Table 5.1 displays information about each of the algorithm and dataset combination, including the checkpoints at 10-, 30-, and 50-concept mark. The number in the cells indicates the *position* of the algorithm, with 1 denoting that the algorithm showed best results and 5 indicating the worst results. A dash (-) shows that the algorithm did not compute enough concepts for that specific checkpoint.

Table 5.1. Ranking of Algorithms.

	House Votes			**Student Perf.**			**Basket 1**			**Basket 2**			**Basket 3**		
	10	**30**	**50**	**10**	**30**	**50**	**10**	**30**	**50**	**10**	**30**	**50**	**10**	**30**	**50**
S	3	2	3	3	2	2	3	2	2	4	2	2	2	2	2
IL	4	3	2	2	3	3	2	3	3	1	2	3	3	3	3
LS	2	-	-	5	5	5	-	-	-	5	-	-	5	5	-
MSM	5	4	4	4	4	4	4	4	4	1	4	4	4	4	4
GC	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

It becomes clear that the algorithms performed in a relatively stable manner across all datasets and checkpoints. **LS** was the worst performer, being unusable in more than half of the cases, and showing the worst results in almost all of the cases where the results were usable. **MSM** showed consistently 4th best results, with slight deviations in two of the fifteen measuring points. **IL**, while showing some variance and achieving quite good results in 3 out of 5 datasets for the 10-concept mark, was mostly the 3rd best algorithm. Similarly, **S** showed some variance, but was 2nd in most of the checkpoints, including all 30-concept checkpoints. Finally, **GC** showed an apparent superiority, achieving the best results across all datasets and checkpoints.

One possibly interesting finding is that looking at the Figures 5.1 to 5.5, it can be noticed that all algorithms, except for GC, exhibit spikes in the amount of 1s they cover per number of concepts. That is, the algorithms' calculations alternate between concepts which do not add many 1s to the cumulative coverage, and concepts which may raise the number of 1s covered in big jumps. Diametrically, the GC algorithm displays a clear tail: every following concept adds as much or less to the total coverage than the concept that was calculated before it. Naturally, this is reasonable since it was the way how the algorithm was developed, but a possible learning is that an algorithm which calculates a more effective concept after a less effective concept cannot be the optimum algorithm, if the viewer may be interested in any of the concepts. Put differently, in a range $(1, 2, 3, \dots, n-1, n)$, if the viewer is only interested in coverage by *n*, then the order of concepts in the range does not matter. But if any arbitrary concept *k* may be observed as well, where $1 \leq k \leq n$, and the aim is to have highest possible coverage at each point of the range, then the cumulative coverage added by k+1 cannot be higher than the cumulative coverage added by *k*. However, this would have to be researched in further detail in order to provide value in development of coverage algorithms.

All in all, the conclusion based on the produced results is that the developed GC algorithm is technically optimal for covering the largest number of 1s in a dataset with the least amount of concepts. The results effectively answer questions 1 and 2 from Chapter 1.2. That is, while existing FCA algorithms are moderately effective at covering a dataset, the new GC algorithm consistently outperforms the existing algorithms and shows thus higher efficiency. The third research question remains unanswered, and will be addressed in the following chapter.

6 Concept Coverage Effect on Business Value

One question from the research questions proposed in Chapter 1.2 remains unanswered at this point – do FCA algorithms which efficiently calculate concept coverage have business application? Having compared the different algorithms from a technical perspective in the previous chapter, this chapter will look at one concrete dataset in an attempt to analyse the specific concepts that the different algorithms extracted. Further, as the Greedy Coverage algorithm performed best in covering the 1s of a dataset, then it will be investigated what differences there are between the concepts found by GC and other algorithms; and what implications do those findings hold in relation to business value. The concept lattices which can be construed from the compressed amount of concepts by each of the algorithms are visualized and discussed as well. Finally, some conclusions will be made, and by the end of the chapter all research questions posed in the beginning of the thesis will be answered.

6.1 An In-depth Look at the Instacart Dataset

This chapter will dive deeper into the specific concepts found for the 3rd basket example – the Instacart dataset. The reason for choosing that particular dataset is that its data is the most recent of all the datasets analysed and the dataset has a clear relation to business value – listing the sales of products. It is also the largest dataset out of the basket datasets, both in terms of 1s and concepts generated.

The methodology for analysis is to look at the structure and metrics of the dataset, to provide a general overview of the dataset and how the different algorithms compare in some metrics. A more specific look will then be taken into the concept intents – aisles – generated by the algorithms, to reveal patterns in achieving a high coverage of the dataset. Combinations of aisles will be investigated, to see what the different concept intents provided by the algorithms imply. Finally, the concept lattices will be constructed and analysed, rounding up the analysis in this chapter.

Only the 10-concept mark will be looked at more thoroughly, as it was visible in Chapter 5.1.5 that the results between the measuring points did not differ greatly. Indeed, the analysis conducted in the following sections was done also on the 30- and 50-concept marks, but due to similar findings the analysis is not included in this chapter. Similar tables as were constructed for the 10-concept analysis can be found in Appendix 1 – Additional Tables from Instacart Analysis for the 30- and 50-concept analysis.

6.1.1 Dataset Overview and Metrics

In order to follow the analysis part in an understandable manner, it is first important to explain some parts about the dataset and its structure, and how the concepts will be denominated in the analysis. The data in the dataset is organized as follows:

- The **columns** represent **aisles** – the equivalent of *product categories*. In some of the further sections, *products* may also refer to *aisles* in terms of the dataset. The actual names of the aisles will not be used in the analysis – instead, the ids of the aisles will be denoted, as the actual description of the aisles does not play a role in this particular analysis. A full list of the aisles with descriptions can be found in Appendix 2 – Instacart Aisle Ids and Descriptions.

- The **rows** represent **purchases**. One purchase may include products from 1…n aisles. Also, a single purchase from an aisle may contain n amount of different products belonging to that aisle, but for the purposes of this research, it was not distinguished whether several products were purchased or not – as long as there was a purchase from that aisle, it was marked with a 1.

Due to the previous, the concepts in the next sections will be listed based on their *intent* – the aisles, or the columns of the dataset. The reasoning is that the concepts with the highest coverage, i.e. the ones that the algorithms have calculated, are more likely to include one or few aisles and many purchases. The probability of one purchase including products from almost all aisles is low, but a single aisle or a combination of aisles may be exhibited in many of the purchases, as certain products are more likely to be purchased frequently.

Comparisons of some general metrics are listed in Table 6.1. The columns represent the metrics for each of the algorithms, the total and unique aisles can be explained as follows:

- **Total aisles** – the subtotal amount of attributes across the intents of all 10 concepts – an aisle is unique within one concept, but can theoretically occur in every concept;

- **Unique aisles** – the unique attributes across all 10 concepts' intents.

Table 6.1. Instacart Dataset 10-Concept Metrics.

	S	**IL**	**LS**	**MSM**	**GC**
Total aisles	15	15	28	20	12
Unique aisles	7	6	14	5	11
Unique % out of total	47%	40%	50%	25%	92%
Total aisles per concept	1.5	1.5	2.8	2.0	1.2
Unique aisles per concept	0.7	0.6	1.4	0.5	1.1

Observing the metrics, it becomes clear that the algorithms produced quite different outputs. A high number of total aisles seems to indicate a poor result in coverage, as the LS which performed worst in terms of coverage by the 10th concept, had the highest number of total aisles with 28. The lowest amount of total aisles was displayed by GC, which also performed the best in terms of coverage. A high amount of unique aisles could also be used as a possible indication of a high coverage, except for the anomaly of LS which shows the highest amount of unique aisles.

Possible explanation for a lower number of total aisles producing a higher dataset coverage could be that as the number of total aisles increases, so too increases the number of overlaps across concepts. This is further indicated by a low amount of unique aisles in most of the algorithms performing worse than GC. Naturally, only looking at the summarized metrics of the amount of aisles cannot be used as conclusive evidence for these assumptions – therefore, a more detailed look at the specific concepts and aisles calculated by the algorithms will be looked at next.

6.1.2 Concepts and Their Cumulative Coverage

The first ten concepts calculated by each algorithm are shown in Table 6.2. The concepts are depicted via their intents – the aisle ids that comprise the concept. The *a* stands for *aisle*, and the number following the *a* stands for the id of the aisle. If there are several aisle ids in a single cell, then it means that the concept is a combination of those aisles. As an example, the first concept returned by the Stability algorithm is represented as *a24, a83, a123* – meaning that those 3 aisles comprise the intent of the concept.

Table 6.2. The First 10 Concepts from the Instacart Dataset.

Concept	S	IL	LS	MSM	GC
1	a24, a83, a123	a24	a24, a83, a123	a24, a83	a24, a83
2	a115	a83	a115	a24, a83, a123	a123
3	a24, a123	a123	a24, a123	a24, a123	a24, a120
4	a24, a83	a24, a83	a24, a83	a83, a123	a115
5	a123	a24, a123	a123	a83	a21
6	a83	a83, a123	a83	a24	a91
7	a24	a120	a24	a123	a84
8	a24, a120	a115	a13, a21, a24, a78, a83, a107, a123	a24, a120	a107
9	a107	a21	a19, a24, a32, a116	a24, a83, a120	a31
10	a21	a24, a83, a123	a14, a24, a83, a84, a123, a129	a21, a24, a83	a112

The first observation is that there are repeating aisles which make up the top concepts for all algorithms: *a24*, *a83*, and *a123* appear in some combination in the top 3 of all algorithms. These are also the aisles which appear in most purchases, and products from the aisles are also frequently bought together. However, there's also an aisle which does not have strong connection to any other aisle: *a115* appears in 4 out of 5 algorithms' results, but it appears only as a single entity, not being combined with any other aisle. This variance indicates that the dataset is diversified and includes both, products which are frequently purchased together with others, and also products which are not as related to the purchases of other products. In order to take a more exhaustive look at the concepts and aisle combinations, Table 6.3 lists the concepts' coverages (Con Cov), the cumulative coverage achieved by concept *x* (Cum Cov), and the percentage of how many new 1s of the dataset were covered by the concept (Con add%).

Table 6.3. Concept and Cumulative Coverages per Algorithm.

	S			IL			LS			MSM			GC		
#	Con Cov	Cum Cov	Con add%	Con Cov	Cum Cov	Con add%	Con Cov	Cum Cov	Con add%	Con Cov	Cum Cov	Con add%	Con Cov	Cum Cov	Con add%
1	390	390	100%	344	344	100%	390	390	100%	416	416	100%	416	416	100%
2	145	535	100%	278	622	100%	145	535	100%	390	546	33%	253	669	100%
3	382	657	32%	253	875	100%	382	657	32%	382	668	32%	246	837	68%
4	416	813	38%	416	875	0%	416	813	38%	314	722	17%	145	982	100%
5	253	875	25%	382	875	0%	253	875	25%	278	765	15%	140	1 122	100%
6	278	945	25%	314	875	0%	278	945	25%	344	840	22%	129	1 251	100%
7	344	1 020	22%	155	1 030	100%	344	1 020	22%	253	875	14%	122	1 373	100%
8	246	1 143	50%	145	1 175	100%	35	1 040	57%	246	998	50%	105	1 478	100%
9	105	1 248	100%	140	1 315	100%	8	1 046	75%	234	998	0%	94	1 572	100%
10	140	1 388	100%	390	1 315	0%	12	1 052	50%	216	1 070	33%	93	1 665	100%

Two clear observations can be made from this data. First of all, GC is dominantly better at choosing concepts which add 100% of their coverage to the cumulative coverage. This is an indicator that all other algorithms keep calculating concepts with 1s which have already been partially, mostly, or totally covered, thus providing little value in giving new insights about the data. This proves the statement from the previous section that the other algorithms except for GC calculated concepts which overlapped. Secondly, single-aisle concepts seem to be much more effective in adding to the cumulative coverage – this can be observed from the concepts of GC, as well as IL. The former has the highest cumulative coverage by the 10th concept, and only two concepts with aisle combinations; the latter has additional coverage only from single-aisle concepts.

Thus, the second observation proposes an interesting question: is it always more efficient to calculate single-attribute concepts? One could argue this from a logical point-of-view and say that no two concepts with single-attributes can overlap, and thus single-attribute concepts are indeed more efficient. Looking at the data seems to partially suggest so as well, especially when looking specifically at the aisles *a24*, *a83* and *a123*. IL calculates the three aisles separately as the first 3 concepts, and consequently has the highest cumulative coverage by the 3rd concept; as noted previously, this is the only observed point in the whole research when GC is surpassed in terms of cumulative coverage. So what, if any, is the benefit of GC calculating the concept with the *a24*, *a83* combination?

The next section will look more thoroughly at the aisle combinations in concepts generated by the algorithms. The implications of such concepts will be examined, as well as the reasoning whether the specific combinations chosen by GC are better than other combinations in some ways.

6.1.3 Attribute Combinations in Concepts

In the Instacart dataset, all algorithms calculated some concepts with multiple attributes. As discussed, GC calculated the lowest amount of such concepts, while having the highest cumulative coverage of the dataset by the 10th concept – indicating, that single-attribute concepts may be more efficient in covering a dataset efficiently. To investigate this further, a list of all concepts with more than one attribute have been listed in Table 6.4.

The table is to be read as follows: the concepts, and the algorithms where the concepts are present are listed in the first 2 columns. The main, middle part of the table shows the *concept coverage* of the *aisle*. As an example, the concept *a24, a83, a123* covers 38% of all of the 1s of the aisle *a24*; the concept *a24, a123* covers 56% of the aisle *a24*. Finally, the last two columns show the *minimum* and *maximum* concept coverages of the concept across all the aisles in the concept's intent.

Table 6.4. Multiple-Aisle Concepts and Their Aisle Coverage.

Concepts	**Algorithms**	**a13**	**a14**	**a19**	**a21**	**a24**	**a32**	**a78**	**a83**	**a84**	**a107**	**a116**	**a120**	**a123**	**a129**	**MIN**	**MAX**
a24, a83, a123	S, IL, LS, MSM					38%			47%					51%		38%	51%
a24, a123	S, IL, LS, MSM					56%								75%		56%	75%
a24, a83	S, IL, LS, MSM, GC					60%			75%							60%	75%
a24, a120	S, MSM, GC					36%							79%			36%	79%
a21, a24, a83	MSM				51%	21%			26%							21%	51%
a83, a123	IL, MSM								56%					62%		56%	62%
a24, a83, a120	MSM					23%			28%				50%			23%	50%
a13, a21, a24, a78, a83, a107, a123	LS	20%			4%	1%		6%	2%		5%			2%		1%	20%
a19, a24, a32, a116	LS			5%		1%	4%					2%				1%	5%
a14, a24, a83, a84, a123, a129	LS		9%			1%			1%	2%				1%	5%	1%	9%

Two cells in the last two columns are highlighted: these are the concepts with the highest minimum, and highest maximum concept coverage of the aisle. These happen to be precisely the two concepts with multi-aisle combinations which the GC algorithm has calculated. The indications behind a high minimum and high maximum can be described as follows:

- **A high minimum** shows that all the aisles within the concept have high correlation between each other – in other words, all products in the intent strongly depend on the other product(s). In the case of *a24, a83,* 60% of the cases when *a24* is bought, *a83* is bought as well. Similarly, and even more notably, 75% of the cases *a83* is bought, *a24* is bought as well.

- **A high maximum** shows that at least one of the aisles is highly dependent on the other(s). In the case of *a24, a120,* the combination covers 79% of the cases where *a120* is bought, indicating that 4/5 times the product is bought together with *a24,* while a similar feat cannot be observed the other way around – *a24* is only bought in 36% of the cases when *a120* is bought.

Thus, it can be concluded that while GC does not calculate only single-attribute concepts – which may be more efficient in achieving a high dataset coverage, as seen in the previous section – when GC does calculate concepts with multiple attributes, then these concepts stand out in terms of having a strong bond between the attributes. A much more extensive research should be done to prove whether this is a consistent pattern or not, but it proves at least the algorithm’s applicability on determining business value.

To summarize, considering a dataset representing a market basket, and in an effort to understand which items are generating the most business value, other FCA algorithms seem to show more combinations of products, but this would not show uniquely which items are the main drivers of the business. On the other hand, it would be possible to only look at the sales figures and list the products based on that, but this would leave out strong relations between different products, should there be any. GC algorithm combines the best of both of these views, generating a concept list displaying the top selling products, together with sets of products which display a strong affinity.

6.1.4 Lattices Comparison

As a final step in analysing the business effect of concept coverage and the GC algorithm, it is time to look at the impact on the visual representation of data. Visual overview is one of the key benefits of FCA, since it is easy to draw a certain line diagram – a concept lattice – from the concepts of the context. The concept lattices, drawn for all 3 checkpoints and 5 algorithms, are displayed in Table 6.6, with the columns representing the checkpoints and each row showing the lattices of the concepts calculated by a specific algorithm. The concept intents – aisles – are used as basis for drawing the lattices, and the aisles themselves are marked by a label above the concept drawn in the lattice.

The lattices drawn from the 10 concepts are all quite easily readable and understandable. The S and IL both have a few concepts directly connecting the supremum and infimum, and some concepts displaying interlinkage: these come from the cases where 1 aisle is present in several calculated concepts. The LS displays a variety of aisles in the bottom concepts. As shown in Table 6.1, LS also had the most unique aisles, so it makes sense that the LS diagram is more cramped by the aisles. Furthermore, LS has the highest lattice height – 5 – already with 10 concepts, as shown in Table 6.5. The MSM displays only intertwined concepts, as the algorithm calculated the fewest unique aisles, and all aisles which were calculated in single-aisle concepts were also present in some multi-aisle concept. Finally, the concepts calculated by the GC algorithm provide a lattice which is almost flat in its structure, with the lowest lattice height of 3. The only aisle showing interconnectedness is *a24*, which was present in two of the concepts calculated by GC. It is also the only additional concept calculated by any of the algorithms at this point. This would, in general, add to the complexity of understanding of the diagram, but since there are so few concepts in these lattices, it is assumed there is no disadvantage due to this. Also, while the lattice of the GC does not appear as concept lattices commonly do, with multiple layers of concepts, it does provide exact reference to the way the GC calculated the most effective concepts: as single-aisle entities.

Table 6.5. 10-Concept Lattice Metrics.

	S	**IL**	**LS**	**MSM**	**GC**
Concepts	12	12	12	12	13
Edges	18	19	17	19	21
Lattice height	4	4	5	4	3

The 30-concept lattices already add additional complexity in deciphering their meaning. Nevertheless, S, IL and GC lattices remain quite comprehensible. MSM has comparably slightly more edges (Table 6.7), and due to the low number of unique aisles, it also increases the difficulty in reading the diagram. MSM has also calculated 2 extra concepts which do not map to any of the actual concepts calculated, which further adds to the difficulty at this point in understanding the lattice. Finally, the LS lattice has more than doubled the amount of concepts it actually calculated, due to the large intents of the concepts and their interconnectedness. The amount of edges presented by LS is also more than double that of other algorithms' lattices at this point, thus showing that while LS covers the lowest amount of 1s in the context, it also presents the observer with the most incomprehensible lattice.

The 50-concept lattices are difficult to interpret quickly and clearly, but they do give some indications regarding the underlying data. For example, looking at the S, IL or GC lattices, it is clear to see that there are some aisles which perform best on their own, and some aisles which perform best together with other aisles. In other words, from the business value perspective, it makes sense to look at those aisles as groups or clusters of information, and the rest of the aisles as individual entities, not so much dependent nor influential to the other aisles in terms of sales. The MSM lattice remains partly understandable, but hard to be deciphered in an efficient manner. The LS lattice is arguably providing very little visual value to the analyst by this point. Notably, GC has the lowest amount of concepts, edges and lattice height, indicating simplicity (Table 6.8).

Table 6.6. Lattices of the Calculated Concepts.

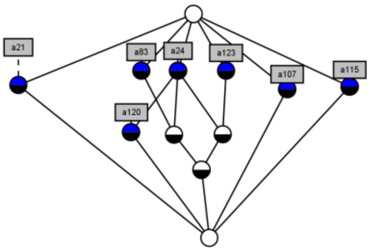
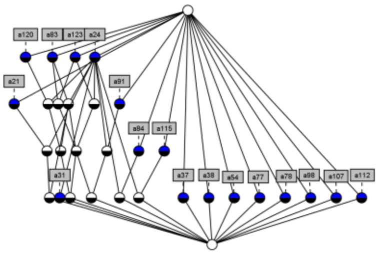
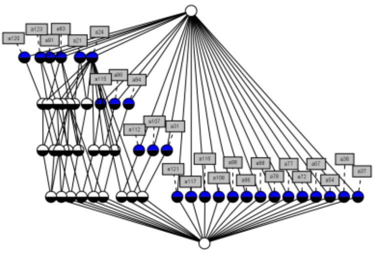
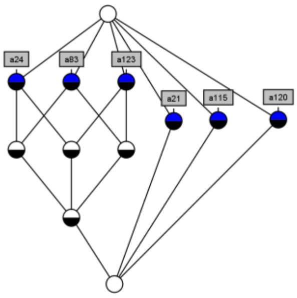
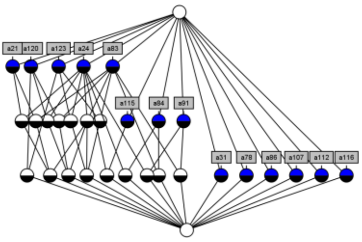
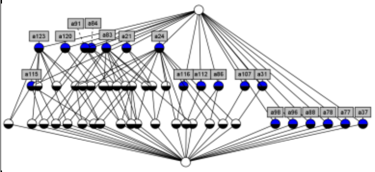
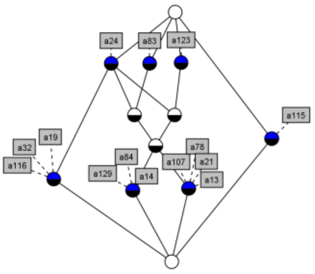
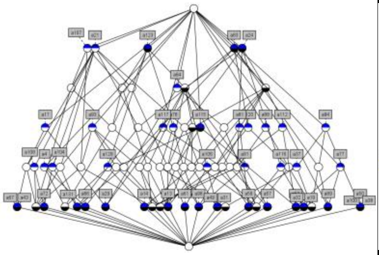
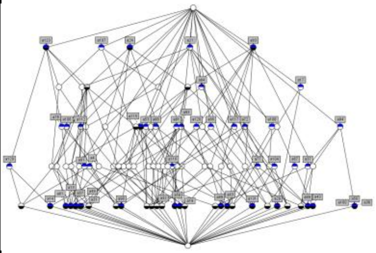
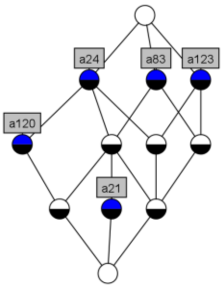
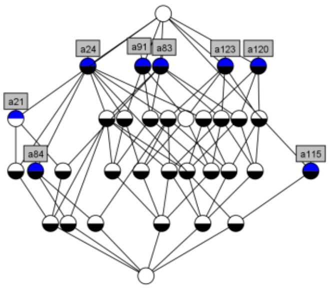
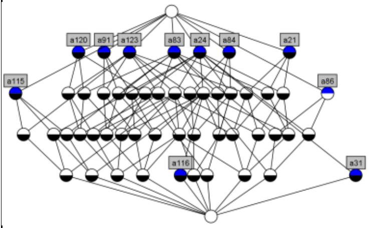
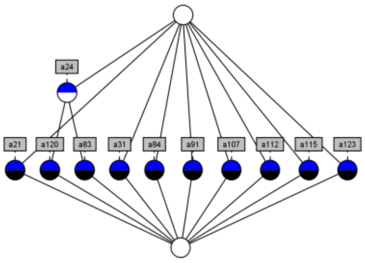
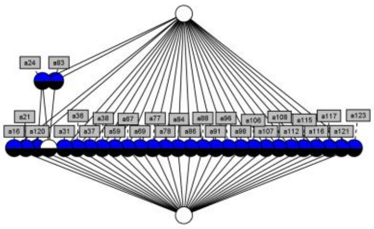
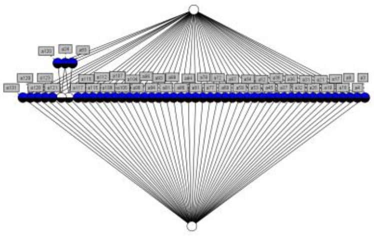
10 concepts	30 concepts	50 concepts
S 	S 	S 
IL 	IL 	IL 
LS 	LS 	LS 
MSM 	MSM 	MSM 
GC 	GC 	GC 

Table 6.7. 30-Concept Lattice Metrics.

	S	IL	LS	MSM	GC
Concepts	32	32	79	34	32
Edges	58	64	177	74	59
Lattice height	5	4	6	5	3

Table 6.8. 50-Concept Lattice Metrics.

	S	IL	LS	MSM	GC
Concepts	52	52	95	53	52
Edges	100	112	220	123	99
Lattice height	5	4	6	5	3

To summarize, while the reading complexity increases as the number of concepts increase in the lattice, the algorithms performing best in the coverage of the context: the GC, S and IL, calculate concepts which form lattices that remain visually readable. The worst performers in terms of coverage of 1s also provide the analyst with lattices which prove of little value due to their interpretation complexity.

6.2 Conclusions from the Analysis

The previous sections provided an insight into potential business applications of the notion of concept coverage, and the new Greedy Coverage algorithm. This section will conclude the analysis part of this thesis by revisiting some of the emerged ideas from the previous chapter.

Firstly, the metrics of the different FCA algorithms were compared in order to find out what are the possible separators of an algorithm calculating a high coverage. The main indicator seemed to be a low number of total attributes in the concepts' intents, which indicated a high amount of **overlap** in the algorithms which performed worse than the GC algorithm.

Thus, secondly, the concepts and their cumulative coverage were studied in higher detail. The proposition about overlapping was confirmed, and a further observation was made – **single-aisle** concepts seemed to have **higher effectiveness** in adding 1s to dataset coverage than concepts with a combination of aisles.

The third part looked at the attribute combinations and the notable discovery was that the multi-aisle concepts calculated by the GC were, in fact, prominent among the other similar concepts, due to their high display of affinity between the aisles that they contained. Importantly, this discovery proved that the GC algorithm and the notion of concept coverage have **relevance in a business setting**: for the studied dataset, the GC algorithm chose concepts which covered the aisles with highest business value, and also aisle combinations which were significantly strong.

Finally, the lattices construed by the aisles from the calculated concepts were compared. The main finding here was that the higher coverage of a dataset by an algorithm seemed to provide lattices with higher **readability**.

Naturally, finding business value will depend a lot on the type of data examined and the business problem or opportunity at hand. Some of the findings from the Instacart dataset are not applicable to, for example, more dense datasets such as the Student Performance dataset. An example of the GC lattices for the Student Performance dataset can be observed in Appendix 3 – GC Lattices for Student Performance Dataset. But as listed in Chapter 3.1, the main focus for looking at business applicability in this thesis is on the market basket datasets. A thorough look at further datasets with a different structure, density and business focus may give additional input into the usability of concept coverage and the Greedy Coverage algorithm. Ultimately, based on the observations done in this chapter, it can be concluded that FCA algorithms and concept coverage do have business applications, providing the data miner with an improved view of the dataset.

7 Summary

The thesis set out to find Formal Concept Analysis (FCA) algorithms' effectiveness in terms of coverage, and whether such algorithms have business applications. This was done through first introducing the FCA theory and its use of line diagrams called concept lattices to visually present data to the data miner. Then, the weakness of lattice readability was discussed, and four existing FCA algorithms which have previously been used as a way to rank concepts and compress the lattice to improve its readability were introduced. The third chapter explained the methodology for research, including the datasets on which the research would be done on, the notion of concept coverage and how it is measured, and the benchmarks on which to measure the effectiveness of the algorithms.

Inspired by the theory and building on the methodology, a new algorithm – Greedy Coverage (GC) – was developed by the author, and the algorithm was introduced in Chapter 4. Chapter 5 applied everything previously introduced: the 5 algorithms, on the 5 datasets, and compared the results in the 3 measuring points. The new GC algorithm proved superior to all other four algorithms across all datasets and measuring points, in terms of ranking concepts based on the coverage of the dataset. Thus, one part of the research objective was covered. The sixth chapter looked more deeply into one of the datasets and the results, investigating the potential application for a business setting. In summary, the GC algorithm was deemed as effective in providing the data miner with an improved view of the dataset.

7.1 Future Work

Three key approaches to future work have been identified by the author:

1. A technical approach, investigating the effects of concept coverage and the performance of the algorithms on **a wider spectrum of concept indices**. Together with the new GC algorithm, only five indices were compared within this study, while potentially over twenty indices [16] could be investigated, providing a more substantial comparison of the algorithms and probably some novel insights into how the algorithms perform in terms of generating readable lattices.

2. A more business oriented approach, experimenting with the algorithms on **more datasets** containing different business objectives. This thesis focused on the market basket datasets, while potential extensions could be in datasets such as:

    a. Customer segmentation – which are the most definitive characteristics of loyal and high value customers?

    b. Supply chain management – what is the performance of the company's suppliers and their products?

    c. Portfolio management – what are the key differentiating traits of projects that are financially successful?

3. Finally, the third approach would focus on improvements or additions to the GC algorithm and the way **how optimum coverage concepts are found**. As witnessed in the Instacart dataset case, there was one point where the Iceberg Lattice had higher cumulative coverage than the GC, laying the groundwork for a further study of how to best find and rank the concepts. One possible thought could be a sort of K-Coverage algorithm, which would find the highest coverage by the K*th* concept.

To summarize, the field of concept coverage, especially in terms of Formal Concept Analysis, seems to be quite unexplored and has plenty to offer for future research.

References

[1] A. Bounds, “Number of UK start-ups rises to new record,” 12 October 2017. [Online]. Available: https://www.ft.com/content/cb56d86c-88d6-11e7-afd2-74b8ecd34d3b. [Accessed 24 April 2018].

[2] A. Airaksinen, H. Luomaranta, P. Alajääskö and A. Roodhuijzen, “Statistics on small and medium-sized enterprises,” Eurostat, September 2015. [Online]. Available: http://ec.europa.eu/eurostat/statistics-explained/index.php/Statistics_on_small_and_medium-sized_enterprises. [Accessed 24 April 2018].

[3] F. Provost and T. Fawcett, Data Science for Business: What you need to know about data mining and data-analytic thinking, Sebastopol, CA, USA: O'Reilly Media, Inc., 2013.

[4] I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed., San Francisco, CA, USA: Morgan Kaufmann Publishers, 2005.

[5] R. Wille, “Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies,” in *Formal Concept Analysis*, Berlin/Heidelberg, Germany, Springer, 2005, pp. 1-33.

[6] F. Škopljanac-Mačina and B. Blašković, “Formal Concept Analysis – Overview and Applications,” *Procedia Engineering,* vol. 69, pp. 1258-1267, 2014.

[7] J. Poelmans, S. O. Kuznetsov, D. I. Ignatov and G. Dedene, “Formal Concept Analysis in Knowledge Processing: A survey on models and techniques,” *Expert Systems with Applications,* vol. 40, no. 16, pp. 6601-6623, 2013.

[8] P. K. Singh, C. A. Kumar and A. Gani, “A Comprehensive Survey on Formal Concept Analysis, Its Research Trends and Applications,” *International Journal of Applied Mathematics and Computer Science,* vol. 26, no. 2, pp. 495-516, 2016.

[9] U. Priss, “Formal Concept Analysis in Information Science,” *Annual Review of Information Science and Technology,* vol. 40, no. 1, pp. 521-543, 2007.

[10] B. Ganter and S. Obiedkov, Conceptual Exploration, Berlin/Heidelberg, Germany: Springer, 2016.

[11] R. Wille, “Concept Lattices and Conceptual Knowledge Systems,” *Computers & Mathematics with Applications,* vol. 23, no. 6-9, pp. 493-515, 1992.

[12] B. Ganter and R. Wille, “Applied Lattice Theory: Formal Concept Analysis,” in *In General Lattice Theory*, Birkhäuser, 1997.

[13] K. S. K. Cheung and D. Vogel, “Complexity Reduction in Lattice-Based Information Retrieval,” *Information Retrieval,* vol. 8, no. 2, pp. 285-299, 2005.

[14] R. Belohlavek and V. Vychodil, “Reducing the Size of Fuzzy Concept Lattices by Hedges,” in *Proceedings of the 14th IEEE International Conference on Fuzzy Systems*, Reno, NV, USA, 2005.

[15] S. Andrews and C. Orphanides, “Analysis of Large Data Sets Using Formal Concept Lattices,” in *Proceedings of the 7th International Conference on Concept Lattices and Their Applications,* Sevilla, Spain, 2010.

[16] S. O. Kuznetsov and T. Makhalova, “On Interestingness Measures of Formal Concepts,” *Information Sciences,* Vols. 442-443, pp. 202-219, 2018.

[17] M. Klimushkin, S. Obiedkov and C. Roth, “Approaches to the Selection of Relevant Concepts in the Case of Noisy Data,” in *Proceedings of the 8th international conference on Formal Concept Analysis,* Agadir, Morocco, 2010.

[18] R. Belohlavek and M. Trnecka, “Basic Level in Formal Concept Analysis: Interesting Concepts and Psychological Ramifications,” in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence,* Beijing, China, 2013.

[19] M. Kaur and S. Kang, “Market Basket Analysis: Identify the Changing Trends of Market Data Using Association Rule Mining,” *Procedia Computer Science,* vol. 85, pp. 78-85, 2016.

[20] J. Hipp, U. Güntzer and G. Nakhaeizadeh, “Algorithms for Association Rule Mining – A General Survey and Comparison,” *ACM SIGKDD Explorations Newsletter,* vol. 2, no. 1, pp. 58-64, 2000.

[21] L. Lakhal and G. Stumme, “Efficient Mining of Association Rules Based on Formal Concept Analysis,” in *Formal Concept Analysis,* Berlin/Heidelberg, Germany, Springer, 2005, pp. 180-195.

[22] “The Concept Explorer,” [Online]. Available: http://conexp.sourceforge.net. [Accessed 11 December 2017].

[23] J. Poelmans, D. I. Ignatov, S. O. Kuznetsov and G. Dedene, “Formal Concept Analysis in Knowledge Processing: A survey on applications,” *Expert Systems with Applications,* vol. 40, no. 16, pp. 6538-6560, 2013.

[24] F. Strok, “GTAC 2016: Using Formal Concept Analysis in Software Testing,” [Online]. Available: https://www.youtube.com/watch?v=xpOq-IzqQhM. [Accessed 25 February 2018].

[25] A. K. Sarmah, S. M. Hazarika and S. K. Sinha, “Formal Concept Analysis: Current Trends and Directions,” *Artificial Intelligence Review,* vol. 44, no. 1, pp. 47-86, 2015.

[26] O. Prokasheva, A. Onishchenko and S. Gurov, “Classification Methods Based on Formal Concept Analysis,” in *Formal Concept Analysis Meets Information Retrieval Workshop co-located with the 35th European Conference on Information Retrieval*, Moscow, Russia, 2013.

[27] M. Trabelsi, N. Meddouri and M. Maddouri, “New Taxonomy of Classification Methods Based on Formal Concepts Analysis,” in *Proceedings of the 5th International Workshop ”What can FCA do for Artificial Intelligence?”*, The Hague, Netherlands, 2016.

[28] M. E. Cintra, M.-C. Monard and H. D. A. Camargo, “FCA-BASED RULE GENERATOR, a framework for the genetic generation of fuzzy classification systems using formal concept analysis,” in *IEEE International Conference on Fuzzy Systems*, Istanbul, Turkey, 2015.

[29] B. Ganter, Formal Concept Analysis: Methods and Applications in Computer Science, Magdeburg, Germany: Springer, 2003.

[30] N. Jay, F. Kohler and A. Napoli, “Analysis of Social Communities With Iceberg and Stability-Based Concept Lattices,” in *International Conference on Formal Concept Analysis*, Montréal, Canada, 2008.

[31] C. Roth, S. Obiedkov and D. G. Kourie, “On Succinct Representation of Knowledge Community Taxonomies With Formal Concept Analysis,” *International Journal of Foundations of Computer Science,* vol. 19, no. 2, pp. 383-404, 2008.

[32] A. Buzmakov, S. O. Kuznetsov and A. Napoli, “Concept Stability as a Tool for Pattern Selection,” in *What can FCA do for Artificial Intelligence?*, Prague, 2014.

[33] S. O. Kuznetsov, “On Stability of a Formal Concept,” *Annals of Mathematics and Artificial Intelligence,* vol. 49, no. 1-4, pp. 101-115, 2007.

[34] A. Buzmakov, *Formal Concept Analysis and Pattern Structures for mining Structured Data,* Lorraine, France: Universite de Lorraine, 2015.

[35] S. Kuznetsov, S. Obiedkov and C. Roth, “Reducing the Representation Complexity of Lattice-Based Taxonomies,” in *Conceptual Structures: Knowledge Architectures for Smart Applications,* Sheffield, UK, 2007.

[36] N. Jay, F. Kohler and A. Napoli, “Using Formal Concept Analysis for mining and interpreting patient flows within a healthcare network,” in *Fourth International Conference on Concept Lattices and Their Applications,* Hammamet, Tunisia, 2006.

[37] D. T. Smith, “A Formal Concept Analysis Approach to Data Mining: The QuICL Algorithm for Fast Iceberg Lattice Construction,” *Computer and Information Science,* vol. 7, no. 1, pp. 10-32, 2014.

[38] G. Stumme, “Efficient Data Mining Based on Formal Concept Analysis,” in *International Conference on Database and Expert Systems Applications*, Aix-en-Provence, France, 2002.

[39] S. Obiedkov, “Introduction to Formal Concept Analysis,” [Online]. Available: https://www.coursera.org/learn/formal-concept-analysis. [Accessed 25 November 2017].

[40] L. Võhandu, R. Kuusik, A. Torim, E. Aab and G. Lind, “Some Algorithms for Data Table (Re)ordering Using Monotone Systems,” in *Proceedings of the 5th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases,* Madrid, Spain, 2006.

[41] A. Torim, *Formal Concepts in the Theory of Monotone Systems,* Tallinn, Estonia: TUT Press, 2009.

[42] “Concepts,” [Online]. Available: https://pypi.org/project/concepts. [Accessed 4 February 2018].

[43] J. Schlimmer, “UCI Machine Learning Repository,” [Online]. Available: https://archive.ics.uci.edu/ml. [Accessed 23 February 2018].

[44] P. Cortez and A. Silva, “Using Data Mining to Predict Secondary School Student Performance,” in *Proceedings of 5th FUture BUsiness TEChnology Conference,* Porto, Portugal, 2008.

[45] “Dataset - MBA,” [Online]. Available: https://www.kaggle.com/arnasca1965/dataset-mba. [Accessed 18 March 2018].

[46] D. Chen, S. L. Sain and K. Guo, “Data Mining for the Online Retail Industry: A case study of RFM model-based customer segmentation using data mining,”

Journal of Database Marketing and Customer Strategy Management, vol. 19, no. 3, pp. 197-208, 2012.

[47] “Instacart: Groceries Delivered From Local Stores,” [Online]. Available: https://www.instacart.com. [Accessed 1 April 2018].

[48] “Instacart Market Basket Analysis,” [Online]. Available: https://www.kaggle.com/c/instacart-market-basket-analysis. [Accessed 18 March 2018].

[49] J. Stanley, “3 Million Instacart Orders, Open Sourced,” [Online]. Available: https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2. [Accessed 1 April 2018].

[50] H. Yan, K. Chen and L. Liu, “Efficiently Clustering Transactional Data with Weighted Coverage Density,” in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, Arlington, VA, USA, 2006.

[51] H. Yan, K. Chen, L. Liu and J. Bae, “Determining the Best K for Clustering Transactional Datasets: A Coverage Density-based Approach,” *Data & Knowledge Engineering*, vol. 68, no. 1, pp. 28-48, 2009.

[52] M. Frank, D. Basin and J. M. Buhmann, “A Class of Probabilistic Models for Role Engineering,” in *Proceedings of the 15th ACM conference on Computer and communications security*, Alexandria, VA, USA, 2008.

[53] P. Eklund, J. Ducrou and P. Brawn, “Concept Lattices for Information Visualization: Can Novices Read Line-Diagrams?,” in *Proceedings of the 2nd international conference on formal concept analysis*, Sydney, Australia, 2004.

[54] M. Ward, G. Grinstein and D. Keim, “Human Perception and Information Processing,” in *Interactive Data Visualization Foundations*, Natick, MA, USA, A. K. Peters, Ltd., 2010, pp. 73-128.

Appendix 1 – Additional Tables from Instacart Analysis

Table Ap1.1 - Instacart Dataset 50-Concept Metrics. *LS metrics for 35 concepts

	S	IL	LS	MSM	GC
Total aisles	88	88	160	117	52
Unique aisles	26	19	48	11	48
Unique % out of total	30%	22%	30%	9%	92%
Total aisles per concept	1.8	1.8	4.6*	2.3	1.0
Unique aisles per concept	0.5	0.4	1.4*	0.2	1.0

Table Ap1.2 - Additional Concepts from 10- to 30-Concept Mark

Concept	S	IL	LS	MSM	GC
11	a77	a91	a21, a24, a61, a106, a123	a24, a120, a123	a24
12	a91	a24, a120	a24, a88, a93, a107, a112, a123	a24, a83, a91	a86
13	a24, a115	a84	a17, a21, a43, a67	a24, a91	a116
14	a120	a107	a24, a57, a91, a112, a123	a21, a83	a78
15	a21, a24, a83	a21, a24	a21, a83, a93, a104, a131	a115	a83
16	a37	a24, a91	a21, a72, a104, a108	a24, a83, a120, a123	a77
17	a84	a21, a83	a24, a77, a83, a89, a94	a24, a83, a91, a123	a37
18	a112	a31	a24, a59, a81, a83, a116, a117	a120	a98
19	a78	a112	a21, a37, a78, a94, a107, a123	a120, a123	a96
20	a24, a84	a24, a84	a37, a83, a88, a94, a120	a24, a84	a88
21	a24, a91	a86	a17, a29, a84, a117	a21, a24	a38
22	a24, a83, a91, a123	a116	a4, a21, a107, a123, a129	a24, a91, a123	a67
23	a83, a123	a24, a115	a38, a50, a94, a100	a24, a83, a84	a121
24	a24, a83, a120	a120, a123	a17, a21, a93, a107, a108	a24, a115	a16
25	a24, a83, a120, a123	a83, a120	a53, a77, a112, a115	a83, a120	a117
26	a24, a31	a78	a24, a81, a83, a93, a115, a120, a123	a83, a91	a108
27	a54	a24, a83, a120	a24, a31, a49, a83, a84, a106, a123	a91	a69
28	a21, a24	a83, a91	a21, a24, a81, a83, a84, a88, a120	a24, a84, a123	a59
29	a98	a83, a84	a4, a86, a93, a115	a83, a91, a123	a106
30	a38	a21, a24, a83	a24, a78, a83, a91, a98, a115, a117, a123	a83, a120, a123	a36

Table Ap1.3 – 30-Concept Mark Concepts and Cumulative Coverages per Algorithm

	S			IL			LS			MSM			GC		
#	Con Cov	Cum Cov	Con add%	Con Cov	Cum Cov	Con add%	Con Cov	Cum Cov	Con add%	Con Cov	Cum Cov	Con add%	Con Cov	Cum Cov	Con add%
11	64	1 452	100%	129	1 444	100%	10	1 058	60%	207	1 070	0%	344	1 756	26%
12	129	1 581	100%	246	1 444	0%	12	1 066	67%	201	1 137	33%	89	1 845	100%
13	172	1 581	0%	122	1 566	100%	8	1 074	100%	192	1 166	15%	87	1 932	100%
14	155	1 613	21%	105	1 671	100%	15	1 083	60%	188	1 188	12%	80	2 012	100%
15	216	1 613	0%	194	1 671	0%	15	1 095	80%	145	1 333	100%	278	2 082	25%
16	63	1 676	100%	192	1 671	0%	16	1 111	100%	204	1 333	0%	64	2 146	100%
17	122	1 798	100%	188	1 671	0%	15	1 120	60%	196	1 333	0%	63	2 209	100%
18	93	1 891	100%	94	1 765	100%	18	1 132	67%	155	1 365	21%	63	2 272	100%
19	80	1 971	100%	93	1 858	100%	18	1 146	78%	170	1 365	0%	62	2 334	100%
20	184	1 971	0%	184	1 858	0%	15	1 158	80%	184	1 457	50%	61	2 395	100%
21	192	1 971	0%	89	1 947	100%	12	1 168	83%	194	1 482	13%	60	2 455	100%
22	196	1 971	0%	87	2 034	100%	15	1 176	53%	183	1 482	0%	59	2 514	100%
23	314	1 971	0%	172	2 034	0%	12	1 187	92%	186	1 482	0%	58	2 572	100%
24	234	1 971	0%	170	2 034	0%	15	1 199	80%	172	1 482	0%	58	2 630	100%
25	204	1 971	0%	168	2 034	0%	12	1 208	75%	168	1 482	0%	57	2 687	100%
26	140	2 041	50%	80	2 114	100%	21	1 216	38%	156	1 493	7%	56	2 743	100%
27	49	2 090	100%	234	2 114	0%	21	1 227	52%	129	1 515	17%	54	2 797	100%
28	194	2 090	0%	156	2 114	0%	21	1 231	19%	171	1 515	0%	53	2 850	100%
29	63	2 153	100%	144	2 114	0%	12	1 239	67%	156	1 515	0%	51	2 901	100%
30	60	2 213	100%	216	2 114	0%	24	1 249	42%	162	1 515	0%	50	2 951	100%

Table Ap1.4 - Additional Concepts from 30- to 50-Concept Mark

Concept	S	IL	LS	MSM	GC
31	a31	a24, a31	a17, a21, a67, a83, a86	a21	a54
32	a116	a24, a120, a123	a24, a53, a72, a81, a83, a84, a123	a24, a115, a123	a32
33	a24, a107	a91, a123	a4, a53, a72, a106	a21, a83, a123	a53
34	a21, a24, a83, a123	a24, a86	a24, a63, a83, a86, a108, a123	a83, a84	a131
35	a21, a83	a21, a123	a69, a72, a83, a93	a91, a123	a81
36	a67	a84, a123	-	a84	a9
37	a24, a83, a84	a83, a115	-	a24, a83, a84, a123	a52
38	a24, a112	a24, a83, a91	-	a21, a24, a83, a123	a72
39	a72	a115, a123	-	a115, a123	a93
40	a96	a24, a116	-	a84, a123	a26
41	a24, a115, a123	a77	-	a83, a115	a17
42	a121	a98	-	a24, a83, a86	a94
43	a86	a37	-	a24, a84, a120	a19
44	a88	a24, a112	-	a24, a83, a115	a3
45	a24, a86	a96	-	a24, a31	a128
46	a108	a24, a83, a84	-	a24, a86	a129
47	a24, a83, a86	a83, a86	-	a21, a24, a123	a4
48	a117	a88	-	a21, a123	a45
49	a24, a120, a123	a24, a91, a123	-	a83, a86	a104
50	a24, a83, a91	a24, a107	-	a24, a83, a116	a120

Table Ap1.5 – 50-Concept Mark Concepts and Cumulative Coverages per Algorithm

	S			IL			LS			MSM			GC		
#	Con Cov	Cum Cov	Con add%	Con Cov	Cum Cov	Con add%	Con Cov	Cum Cov	Con add%	Con Cov	Cum Cov	Con add%	Con Cov	Cum Cov	Con add%
31	94	2 237	26%	140	2 114	0%	15	1 256	47%	140	1 536	15%	49	3 000	100%
32	87	2 324	100%	207	2 114	0%	21	1 266	48%	159	1 536	0%	49	3 049	100%
33	120	2 324	0%	138	2 114	0%	12	1 277	92%	159	1 536	0%	49	3 098	100%
34	164	2 324	0%	138	2 114	0%	18	1 284	39%	144	1 546	7%	48	3 146	100%
35	188	2 324	0%	136	2 114	0%	16	1 291	44%	138	1 546	0%	47	3 193	100%
36	59	2 383	100%	136	2 114	0%	-	-	-	122	1 566	16%	46	3 239	100%
37	186	2 383	0%	134	2 114	0%	-	-	-	160	1 566	0%	46	3 285	100%
38	126	2 383	0%	201	2 114	0%	-	-	-	164	1 566	0%	45	3 330	100%
39	45	2 428	100%	132	2 114	0%	-	-	-	132	1 566	0%	44	3 374	100%
40	62	2 490	100%	128	2 114	0%	-	-	-	136	1 566	0%	42	3 416	100%
41	159	2 490	0%	64	2 178	100%	-	-	-	134	1 566	0%	42	3 458	100%
42	58	2 548	100%	63	2 241	100%	-	-	-	150	1 616	33%	40	3 498	100%
43	89	2 637	100%	63	2 304	100%	-	-	-	141	1 616	0%	39	3 537	100%
44	61	2 698	100%	126	2 304	0%	-	-	-	150	1 616	0%	38	3 575	100%
45	138	2 698	0%	62	2 366	100%	-	-	-	140	1 686	50%	37	3 612	100%
46	56	2 754	100%	186	2 366	0%	-	-	-	138	1 705	14%	37	3 649	100%
47	150	2 754	0%	124	2 366	0%	-	-	-	153	1 705	0%	35	3 684	100%
48	57	2 811	100%	61	2 427	100%	-	-	-	136	1 705	0%	35	3 719	100%
49	207	2 811	0%	183	2 427	0%	-	-	-	124	1 717	10%	35	3 754	100%
50	201	2 811	0%	120	2 427	0%	-	-	-	141	1 764	33%	155	3 786	21%

Table Ap1.6 – Top 5 Multiple-aisle Concepts from 50-Concept Mark, Sorted by Highest MIN

#	Concept Intent	MIN	MAX
1	a24, a83	60%	75%
2	a24, a123	56%	75%
3	a83, a123	56%	62%
4	a24, a83, a123	38%	51%
5	a24, a120	36%	79%

Table Ap1.7 – Top 5 Multiple-aisle Concepts from 50-Concept Mark, Sorted by Highest MAX

#	Concept Intent	MIN	MAX
1	a24, a120	36%	79%
2	a24, a86	20%	78%
3	a24, a123	56%	75%
4	a24, a83	60%	75%
5	a24, a84	27%	75%

Appendix 2 – Instacart Aisle Ids and Descriptions

Table Ap2.1 - Instacart Aisle Ids and Descriptions [48]

a1	prepared soups salads
a2	specialty cheeses
a3	energy granola bars
a4	instant foods
a5	marinades meat preparation
a6	other
a7	packaged meat
a8	bakery desserts
a9	pasta sauce
a10	kitchen supplies
a11	cold flu allergy
a12	fresh pasta
a13	prepared meals
a14	tofu meat alternatives
a15	packaged seafood
a16	fresh herbs
a17	baking ingredients
a18	bulk dried fruits vegetables
a19	oils vinegars
a20	oral hygiene
a21	packaged cheese
a22	hair care
a23	popcorn jerky
a24	fresh fruits
a25	soap
a26	coffee
a27	beers coolers
a28	red wines
a29	honeys syrups nectars
a30	latino foods
a31	refrigerated
a32	packaged produce
a33	kosher foods
a34	frozen meat seafood
a35	poultry counter
a36	butter
a37	ice cream ice
a38	frozen meals
a39	seafood counter
a40	dog food care
a41	cat food care
a42	frozen vegan vegetarian
a43	buns rolls
a44	eye ear care
a45	candy chocolate
a46	mint gum
a47	vitamins supplements
a48	breakfast bars pastries
a49	packaged poultry
a50	fruit vegetable snacks
a51	preserved dips spreads
a52	frozen breakfast

a53	cream
a54	paper goods
a55	shave needs
a56	diapers wipes
a57	granola
a58	frozen breads doughs
a59	canned meals beans
a60	trash bags liners
a61	cookies cakes
a62	white wines
a63	grains rice dried goods
a64	energy sports drinks
a65	protein meal replacements
a66	asian foods
a67	fresh dips tapenades
a68	bulk grains rice dried goods
a69	soup broth bouillon
a70	digestion
a71	refrigerated pudding desserts
a72	condiments
a73	facial care
a74	dish detergents
a75	laundry
a76	indian foods
a77	soft drinks
a78	crackers
a79	frozen pizza
a80	deodorants
a81	canned jarred vegetables
a82	baby accessories
a83	fresh vegetables
a84	milk
a85	food storage
a86	eggs
a87	more household
a88	spreads
a89	salad dressing toppings
a90	cocoa drink mixes
a91	soy lactosefree
a92	baby food formula
a93	breakfast bakery
a94	tea
a95	canned meat seafood
a96	lunch meat
a97	baking supplies decor
a98	juice nectars
a99	canned fruit applesauce
a100	missing
a101	air fresheners candles
a102	baby bath body care
a103	ice cream toppings
a104	spices seasonings
a105	doughs gelatins bake mixes
a106	hot dogs bacon sausage
a107	chips pretzels
a108	other creams cheeses
a109	skin care
a110	pickled goods olives

a111	plates bowls cups flatware
a112	bread
a113	frozen juice
a114	cleaning products
a115	water seltzer sparkling water
a116	frozen produce
a117	nuts seeds dried fruit
a118	first aid
a119	frozen dessert
a120	yogurt
a121	cereal
a122	meat counter
a123	packaged vegetables fruits
a124	spirits
a125	trail mix snack mix
a126	feminine care
a127	body lotions soap
a128	tortillas flat bread
a129	frozen appetizers sides
a130	hot cereal pancake mixes
a131	dry pasta
a132	beauty
a133	muscles joints pain relief
a134	specialty wines champagnes

Appendix 3 – GC Lattices for Student Performance Dataset

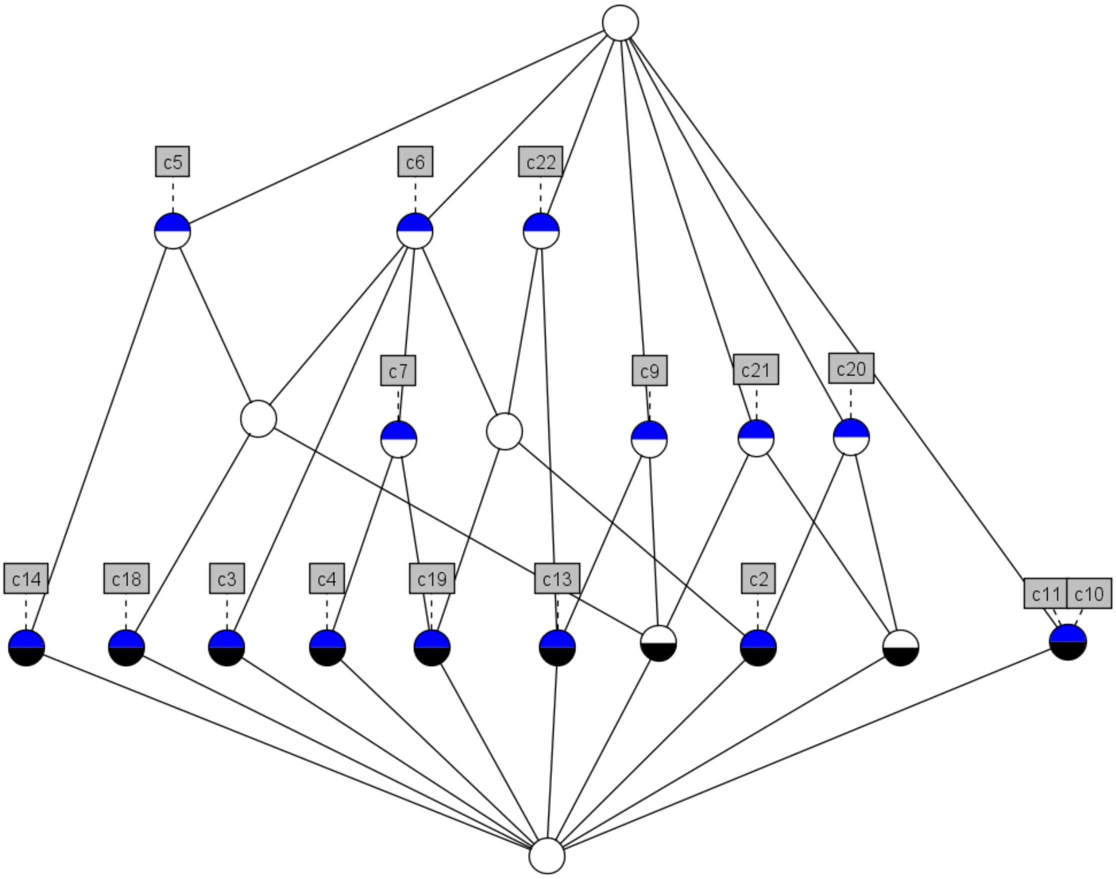


Figure Ap3.1 – 10-Concept Lattice of Student Performance Dataset by GC

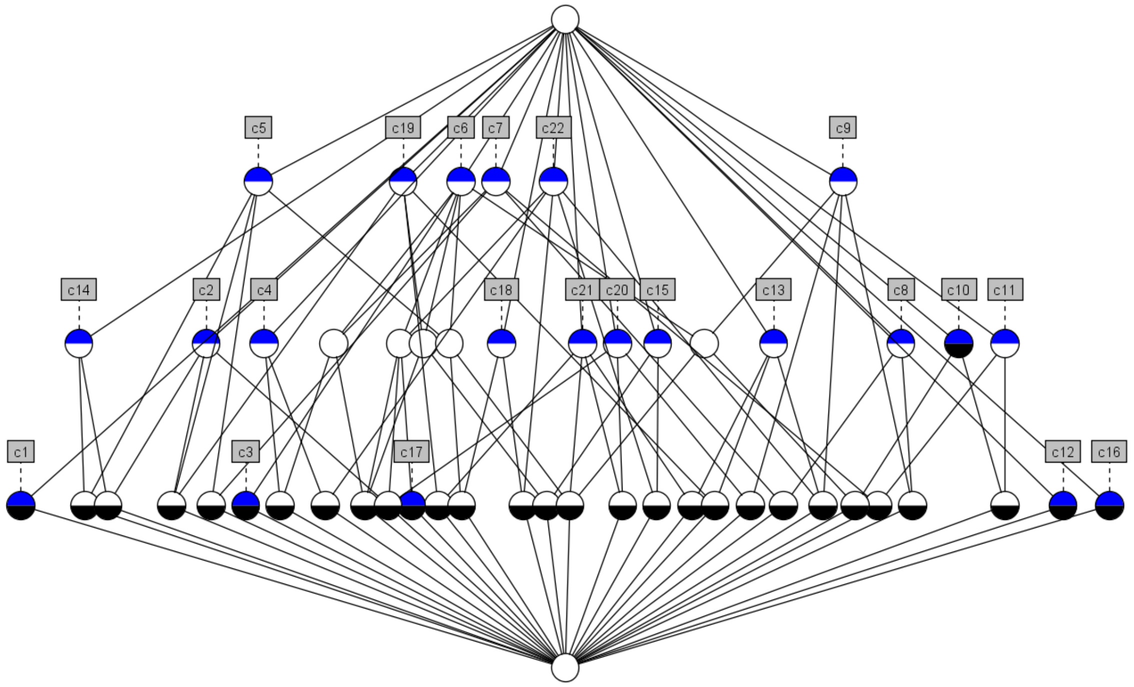


Figure Ap3.2 – 30-Concept Lattice of Student Performance Dataset by GC

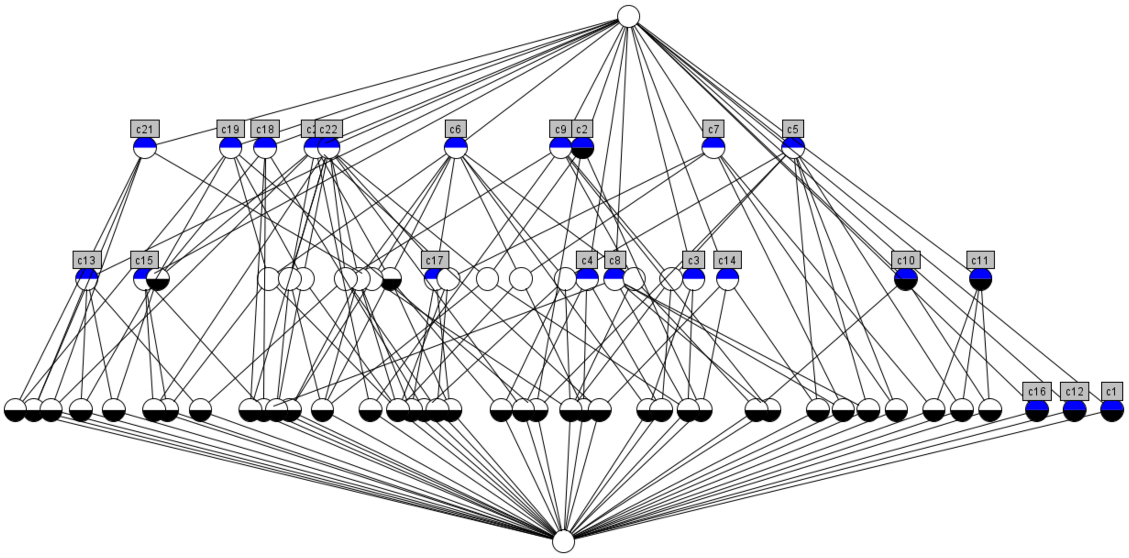


Figure Ap3.3 – 50-Concept Lattice of Student Performance Dataset by GC