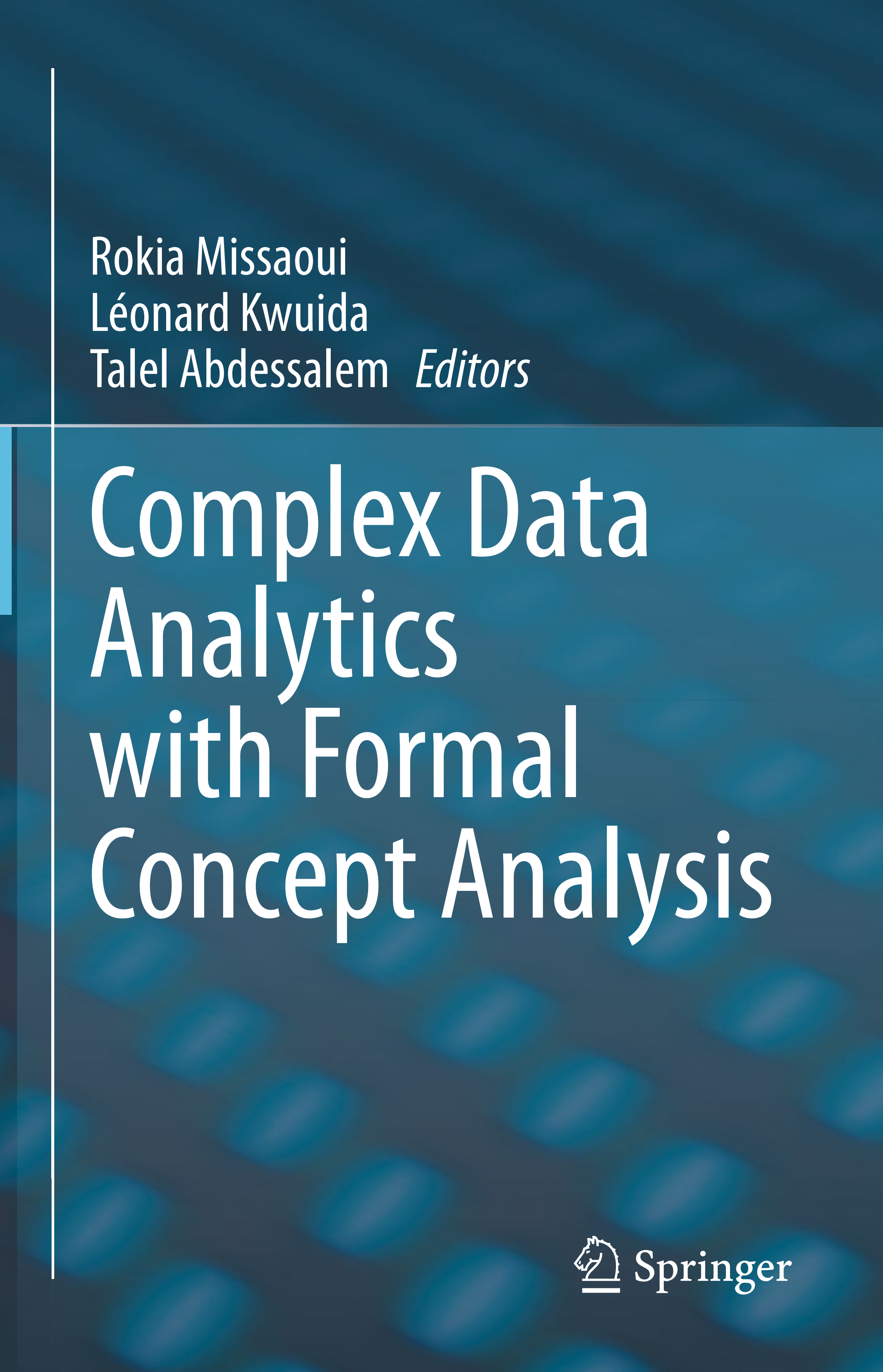


Rokia Missaoui
Léonard Kwuida
Talel Abdessalem *Editors*

Complex Data Analytics with Formal Concept Analysis

Springer

Rokia Missaoui • Léonard Kwuida
Talel Abdessalem
Editors

Complex Data Analytics with Formal Concept Analysis

Springer

Editors
Rokia Missaoui
University of Quebec in Outaouais
Gatineau, QC, Canada

Léonard Kwuida
Business School
Bern University of Applied Sciences
Bern, Switzerland

Talel Abdessalem
Place Marguerite Perey
Télécom-Paris, Institut Polytechnique
Palaiseau, France

ISBN 978-3-030-93277-0 ISBN 978-3-030-93278-7 (eBook)
https://doi.org/10.1007/978-3-030-93278-7

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Scalable Visual Analytics in FCA

Tim Pattison[1]✉, Manuel Enciso[2], Aaron Ceglar[1], Pablo Cordero[2], Derek Weber[1], Ángel Mora[2], and Michael Broughton[1]

[1] Defence Science and Technology Group (Australia)
`tim.pattison@dst.defence.gov.au`
[2] Universidad de Málaga (Spain)

Abstract. Formal Concept Analysis (FCA) is suitable for use within organisations at different levels of maturity in information management and big data analytics. It takes as input a bigraph, into which both structured and unstructured data can be readily transformed, and produces a multiple-inheritance type hierarchy suitable for formal knowledge representation. Accordingly, FCA has been widely applied in areas such as information retrieval, knowledge discovery and knowledge representation. The multiple-inheritance hierarchy produced by FCA is a complete lattice which can be represented as a labelled, directed, acyclic graph. We adopt a visual analytic approach to FCA by combining computational analysis with interactive visualisation. Scaling FCA to the interactive analysis of large data sets poses three fundamental challenges: the time required to enumerate the vertices, arcs and labels of the lattice digraph; the difficulty of responsive presentation of, and meaningful user interaction with, a large digraph; and the discovery of insightful implications. This chapter briefly surveys potential solutions to these scalability challenges posed by big data volumes, and describes software prototypes and coordinated visualisations which explore some of them.

Keywords: Visual analytics · Lattice drawing · Implications

1 Introduction

This chapter is based on [27] – updated to include the contributions made by three subsequent publications – and [9]. Both have been published in proceedings available via `http://ceur-ws.org`. The Commonwealth of Australia has retained Crown copyright in the former.

The visualisation of implications using the lattice digraph as a substrate, potentially coordinated with complimentary views based on data visualisation, is used to integrate these two works.

Formal Concept Analysis (FCA) [41] derives a multiple-inheritance type hierarchy from a *formal context.* A formal context is a bigraph, consisting of a set of *object* vertices, a set of *attribute* vertices, and edges specified by a binary relation between these two sets. The “types” derived by FCA correspond to maximal bicliques in this bigraph, and are known as *formal concepts.* Each formal concept consists of a set of objects, called its *extent*, and a set of attributes, called its *intent*, which are fully interconnected and jointly maximal: neither objects nor attributes can be added while preserving full interconnection. The set of formal concepts, when partially ordered by set inclusion on their extents, forms a complete lattice. This lattice can be efficiently represented as a single-source, single-sink, labelled, directed acyclic graph (DAG) – henceforth called the *lattice digraph* – whose vertices are formal concepts, and whose adjacency relation is the transitive reduction [1] of the ordering relation.

The resultant multiple-inheritance hierarchy of formal concepts constitutes a useful generalisation of a hierarchy for applications such as the storage and retrieval of data objects using keywords or tags, the representation of a Description Logic subsumption hierarchy [35], or the partial ordering of closed frequent item sets in association mining. Accordingly, FCA has been widely applied in such disparate fields as information retrieval, knowledge discovery and knowledge representation [32, 33].

Formal Concept Analysis is an analytic technique suitable for organisations at different levels of maturity in information management. For those who primarily retrieve and read unstructured text, the context bigraph is equivalent to the "bag of words" representation common to a number of statistical techniques for natural language processing, such as Latent Semantic Analysis [19]. For those analysing user-tagged data, including various forms of social media, the tags are attributes associated with the media objects of interest. FCA constitutes a form of association mining for structured data such as the membership of people in organisations or communities of interest, producing a set of implications amongst the chosen attributes. For organisations aspiring to automated reasoning, the output of FCA is an empirically-derived subsumption hierarchy whose incorporation into Description Logics has attracted considerable research interest [36].

"Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces," which, inter alia, "seeks to marry techniques from information visualisation with techniques from computational transformation and analysis of data" [40]. We adopt a visual analytic approach to FCA by combining computational analysis with interactive visualisation. Scalability is a key challenge for visual analytics. Algorithms must scale to large data sets, visualisations must make efficient and intelligible use of screen real-estate, and both must be responsive for interactive use. The number of formal concepts derived from a formal context is bounded above by an exponential function of the number of objects and attributes in that context. Consequently, three fundamental challenges confront those who wish to scale FCA to the interactive analysis of large data sets: the time required to enumerate the vertices, arcs and labels of the lattice digraph; the difficulty of meaningful and responsive user interaction with a large lattice digraph; and the discovery of insightful implications.

This chapter focuses on the volume challenge posed by the exploitation of big data, rather than on data velocity, variety or veracity. We assume that the number $|G|$ of objects is larger than the number $|M|$ of attributes, in which case the number of formal concepts is bounded above by $2^{|M|}$. We further assume that while $|G|$ may be very large, either $|M|$ remains moderate or the formal context is relatively sparse, so that the threatened exponential explosion in the number of formal concepts is not realised. Aggregating all objects into a single formal context is justified provided all objects are sampled from the same unknown joint probability distribution (see e.g. [25]) over the attributes. If there is reason to believe that this distribution is instead slowly time-varying, periodically discarding older objects and rebuilding the concept lattice may be more appropriate.

This paper is organised as follows. Section 2 provides a graph-theoretic introduction to Formal Concept Analysis. Section 3 introduces the topic of visual analytics and its application to Formal Concept Analysis. Section 4 undertakes a brief survey of techniques aimed at improving the scalability of FCA for more responsive visualisation and interaction. Section 5 briefly presents three techniques and associated software prototypes through which the Defence Science and Technology Group has addressed selected aspects of FCA scalability. Section 6 describes the use of data visualisation by the Universidad de Málaga to help users find meaningful implications. Section 7 discusses the remaining challenges for scaling these techniques to deal with truly big data volumes.

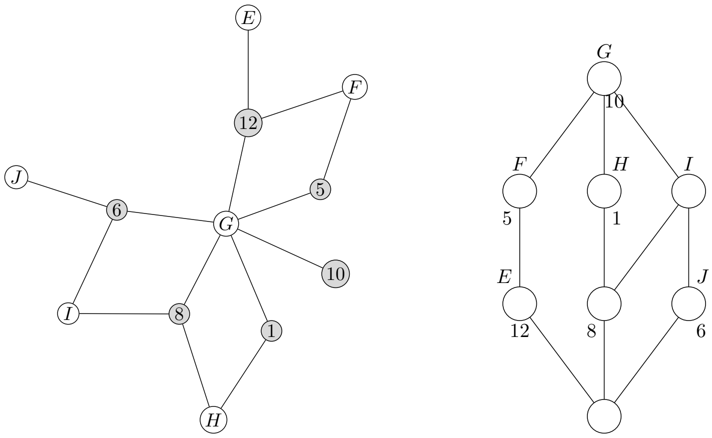
2 Graph-theoretic introduction to FCA

A formal context (G, M, I) is a bipartite graph, or bigraph, with object vertex set G, attribute vertex set M, and undirected edge set $I \subseteq G \times M$. Each edge is adjacent to one object and one attribute vertex. Each object and attribute vertex has a unique label which derives from the domain of application. For an information retrieval domain, for example, the object labels may be document titles and the attribute labels keywords. Figure 1a shows an example formal context, in which the object and attribute vertices have numerical and alphabetic labels respectively.

A sub-context (G', M', I') of the formal context (G, M, I) is a bigraph consisting of a subset $G' \subseteq G$ of its objects, a subset $M' \subseteq M$ of its attributes, and the subset $I' = I \cap (G' \times M')$ of its edges adjacent to those object and attribute vertices.

2.1 Formal concepts

A formal concept consists of a set of objects, called the extent, and a set of attributes, called the intent, which form a maximal biclique in the context bigraph. For example, $(\{\mathtt{5}, \mathtt{12}\}, \{\mathtt{F}, \mathtt{G}\})$ is a formal concept in the formal context of Figure 1a, since both attributes in its intent $\{\mathtt{F}, \mathtt{G}\}$ are connected to both objects in its extent $\{\mathtt{5}, \mathtt{12}\}$, and no other objects or attributes can be added while preserving full inter-connection. Two concepts are said to be comparable iff the extent of one is a subset of the extent of the other.



(a) Bigraph (b) Line diagram

Fig. 1: Bigraph and line diagram (v.i.) for example formal context.

2.2 Concept lattice digraph

FCA produces a set of formal concepts which are, or can be, partially-ordered by extent set inclusion. This partially-ordered set forms a complete lattice [10], which includes *inter alia* a unique maximum element, called the supremum, and a unique minimum element, called the infimum.

This complete lattice can be represented as a DAG, in which each vertex represents a formal concept, and each arc connects a lower neighbour in the partial order to its upper neighbour. This lattice digraph has a single source vertex, corresponding to the infimum of the lattice, and a single sink vertex, corresponding to the supremum. Two concepts are comparable iff there is a directed path between their corresponding vertices in the lattice digraph.

2.3 Line diagram

A line diagram is a layered drawing of the lattice digraph in which the vertical component of each arc is upwards on the page. This convention aids the interpretation of the partial ordering and obviates the need to explicitly indicate the direction of each arc. The source [sink][3] vertex appears at the bottom [top] of the diagram, and all other vertices are assigned to intervening layers. Figure 1b shows the line diagram resulting from FCA of the formal context shown in Figure 1a.

Each concept bears an attribute label in the line diagram, and is said to be an attribute concept, iff its extent is the set of objects adjacent to that attribute in

---

[3] Square brackets are used throughout this paper to indicate that a sentence is true both when read without the bracketed terms and when read with each bracketed term substituted for the term which precedes it.

the context bigraph. Similarly, each concept bears an object label, and is said to be an object concept, iff its intent is the set of attributes adjacent to that object. For example, the top vertex in Figure 1b is an attribute concept for attribute `G` and an object concept for object `10`. Attribute [object] labels are placed above [below] the labelled concept.

A concept inherits the attributes [objects] appearing as labels on comparable concepts above [below] it in the line diagram. The vertex having attribute label set $\{\mathtt{F}\}$ and object label set $\{\mathtt{5}\}$ in Figure 1b corresponds to the concept having extent $\{\mathtt{5}, \mathtt{12}\}$ and intent $\{\mathtt{F}, \mathtt{G}\}$. In addition to its own object and attribute labels, it inherits attribute `G` from its upper neighbour and object `12` from its lower neighbour.

2.4 Anchoring Implications

Definition 1. *An* **implication** $\mathcal{L} \to \mathcal{R}$ *on the formal context* (G, M, I) *consists of an* **antecedent** $\mathcal{L} \subset M$ *and a* **consequent** $\mathcal{R} \subseteq M$ *where* $\mathcal{L} \subset \mathcal{R}$ *and* $\mathcal{L}'' = \mathcal{R}$*.*

The antecedent and consequent are also known respectively as the left-hand side (LHS) and right-hand side (RHS) of the implication. To avoid repeating the attributes in $\mathcal{L} \subset \mathcal{R}$ on the right-hand side, an implication is often written in the abbreviated form $\mathcal{L} \to \mathcal{R} \setminus \mathcal{L}$, indicating that objects which have all attributes in $\mathcal{L}$ also have all attributes in $\mathcal{R} \setminus \mathcal{L}$. The antecedent $\mathcal{L}$ and abbreviated consequent $\mathcal{R} \setminus \mathcal{L}$ partition the consequent $\mathcal{R}$.

The implications of a formal context, such as that in Table 1, can be enumerated algorithmically and tabulated as per Table 2 for exploration by the user. An interactive table view constitutes a familiar interface through which the user can sort, filter, inspect and visually compare rows, which in this case correspond to implications. Computational assistance, such as highlighting set intersections or differences, is advisable when comparing attribute sets containing more than a handful of elements. Implications can be sorted, for example, by the cardinality or lectic order of their antecedent, consequent or abbreviated consequent sets, or filtered by constraining set membership. Sorting and filtering operations can be useful for managing potentially large numbers of implications. A range of objective measures has also been defined [20] for directing the user's attention to "interesting" implications; these interestingness scores can be added to the table, and used thereafter for sorting and filtering.

In the following discussion, we show how implications can also be related to the concept lattice for the purposes of view coordination. !Suggest why such coordination might be useful. !Perhaps a generic statement about the benefit of considering multiple perspectives. !Exploration or decision-making which requires consideration of multiple factors simultaneously, especially where those factors are not easily combined or overlaid in a single view. !Also useful for educational purposes, so that users can relate the resultant implications back to the lattice digraph or even the input formal context.

Observation 1 *The consequent* $\mathcal{R} = \mathcal{L}''$ *of an implication is the intent of a formal concept, whereas the antecedent* $\mathcal{L} \subset \mathcal{L}''$ *is not.*

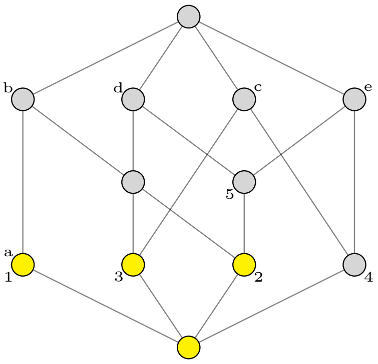


Fig. 2: Concept lattice for the context in Table 1 with concepts coloured yellow whose intents are consequents of implications in the canonical basis.

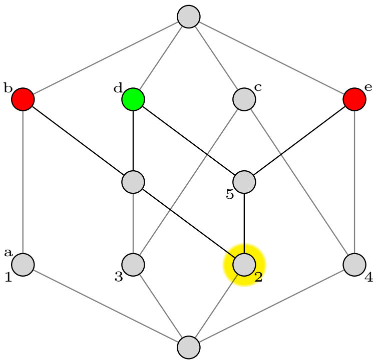


Fig. 3: Concept lattice with attribute concepts for members of the antecedent and abbreviated consequent coloured red and green, respectively.

The digraph vertex corresponding to the formal concept having intent $\mathcal{R}$ serves as a graphical anchor for the implication $\mathcal{L} \to \mathcal{R}$ in a drawing of the concept lattice. User interaction with the implication can be mediated by interaction with its consequent concept in the lattice digraph.

Figure 2 shows the concept lattice for the clarified formal context in Table 1 with the concepts coloured yellow whose intents are the consequents of the implications tabulated in Table 2. Note that implications 2 and 3 have the same consequent, as do implications 5 and 6, demonstrating that there can be a one-to-many relationship between consequents and implications. Since selection of the consequent is not sufficient to uniquely specify an implication of interest, additional interaction is required to choose between the unique antecedents. Unfortunately, these antecedents do not have corresponding graphical elements in a drawing of the lattice digraph with which the user might otherwise interact more directly.

Table 1: Example formal context.

	a	b	c	d	e
1	×	×			
2		×		×	×
3		×	×	×	
4			×		×
5				×	×

Table 2: Duchenne-Guigues canonical basis for implications on the formal context of Table 1.

Serial	$\mathcal{L}$	$\to$	$\mathcal{R}$	$\mathcal{R} \setminus \mathcal{L}$	$\lvert\mathcal{R}'\rvert$
1	$\{a\}$	$\to$	$\{a, b\}$	$\{b\}$	1
2	$\{b, c\}$	$\to$	$\{b, c, d\}$	$\{d\}$	1
3	$\{c, d\}$	$\to$	$\{b, c, d\}$	$\{b\}$	1
4	$\{b, e\}$	$\to$	$\{b, d, e\}$	$\{d\}$	1
5	$\{a, b, d\}$	$\to$	$\{a, b, c, d, e\}$	$\{c, e\}$	0
6	$\{b, c, d, e\}$	$\to$	$\{a, b, c, d, e\}$	$\{a\}$	0

The concept whose intent is the consequent $\mathcal{R}$ of an implication is the meet of the attribute concepts for the attributes $m \in \mathcal{L} \subset \mathcal{R}$. For a given implication, the attribute concept for each attribute $m \in \mathcal{R}$ can be coloured according to whether the attribute belongs to the antecedent $\mathcal{L}$ or the abbreviated consequent $\mathcal{R} \setminus \mathcal{L}$. Following international maritime and aeronautical conventions for left and right, we colour these red and green respectively. Figure 3 shows the attribute concepts for the antecedents of implication 4 from Table 2 coloured red and those for the abbreviated consequent coloured green. The concept whose intent is the consequent is shown with yellow halo rather than fill to allow for the possibility that it may also be the attribute concept for an antecedent or abbreviated consequent attribute. For example, the consequent concept for implication 1 is also the attribute concept for antecedent attribute a, for which our colour code prescribes red fill. The convergence at the haloed vertex of downward paths from the two red vertices confirms that the consequent is the meet of the attribute concepts for members of the antecedent. An upward path from the consequent to the green vertex confirms that attribute d is implied by the presence of the antecedent attributes b, e.

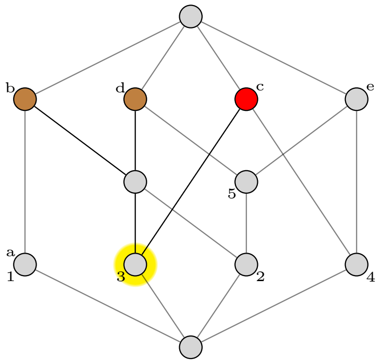


Fig. 4: Concept lattice with the consequent for implications 2 and 3 of Table 2 highlighted with yellow halo and attributes common to the antecedents of both coloured red. The antecedents and abbreviated consequents differ with respect to the brown attributes.

Figure 4 shows the concept lattice with the consequent for implications 2 and 3 of Table 2 highlighted with yellow halo. In this case, selecting the consequent concept is not sufficient to unambiguously identify the implication of interest. For both implications the antecedent includes attribute c, which is accordingly coloured red. Any attributes in $\mathcal{R}$ which are in neither antecedent are necessarily in the abbreviated consequents of both, and would therefore be coloured green; in this case there are no such attributes. The attribute concepts for the remaining two attributes b and d are coloured brown to indicate that at least one of them must be explicitly assigned to either the antecedent or abbreviated consequent in order to distinguish between the possible implications. In this case, assigning either brown vertex is sufficient for this purpose. The purpose of visually distinguishing these vertices is to invite the user to make a binary choice; an alternate representation such as ◐ might better signify and facilitate this choice.

Relating implications to formal concepts, and antecedents and abbreviated consequents to their constituent attributes, has the benefit that this approach is compatible with coordination between views of the lattice digraph and context bigraph. Attribute vertices in the context bigraph should be coloured the same as the corresponding attribute concepts in the lattice digraph. An alternative approach is to add to the lattice digraph a vertex for the antecedent of each implication, which would further clutter the digraph drawing. Insertion of some arcs and deletion of others would also be necessitated by their revised interpretation solely in terms of the transitively reduced subsumption relation between attribute sets. Importantly concept meets – by which the user can confirm the correctness of implications – could no longer be read from this diagram.

Canonical basis

Definition 2. *A set of implications on a context* (G, M, I) *is* **sound** *if it contains only valid implications,* **complete** *if every implication follows from that set, and* **non-redundant** *if no implication in the set follows from the others in the set.*

Definition 3. $\mathcal{P} \subseteq M$ *is called a* **pseudo-intent** *of* (G, M, I) *iff* $\mathcal{P} \subset \mathcal{P}''$ *and* $\mathcal{Q}'' \subset \mathcal{P}$ *holds for every pseudo-intent* $\mathcal{Q} \subset \mathcal{P}$.

Theorem 1 ([15]). *The set of implications*

$$\{\mathcal{P} \to \mathcal{P}'' | \mathcal{P} \ \textit{pseudo-intent}\}$$

is sound, complete and non-redundant.

This set of implications is called the Duchenne-Guigues canonical basis. Table 2 lists the implications in this basis for the example formal context of Table 1.

Observation 2 *Let* $\mathcal{P}$ *and* $\mathcal{Q} \subset \mathcal{P}$ *be pseudo-intents. Then* $\mathcal{Q} \subset \mathcal{Q}'' \subset \mathcal{P} \subset \mathcal{P}''$.

Antecedent $\mathcal{P}$ lies between consequents $\mathcal{Q}''$ and $\mathcal{P}''$ of implications in the canonical basis. These consequents are the intents of comparable concepts, with $\mathcal{Q}''$ and $\mathcal{P}''$ corresponding to the super- and sub-concepts respectively. The antecedent $\mathcal{Q}$ for the super-concept is a subset of the antecedent $\mathcal{P}$ for the sub-concept. Not all pseudo-intents $\mathcal{P}$ need have a pseudo-intent $\mathcal{Q} \subset \mathcal{P}$; with reference to Figure 2, this can be inferred from the observation that not all yellow concepts have yellow super-concepts.

Valid implications All valid implications for a context can be derived from the canonical basis, or any other complete set of implications, by combining implications using Armstrong's rules [3]. The resultant additional implications are said to follow from that complete set. For example, the new implication $\{a, d\} \to \{b, c, e\}$ follows from implications 1 and 5 in Table 2. From the lattice digraph in Figure 2, it can be seen that the infimum, whose intent contains all attributes, is the meet of attribute concept a with attribute concepts c, d or e. This observation confirms the validity of the new implication and shows that at least six additional implications also follow: $\{a, c\} \to \{b, d, e\}$, $\{a, e\} \to \{b, c, d\}$, $\{a, c, d\} \to \{b, e\}$, $\{a, c, e\} \to \{b, d\}$, $\{a, d, e\} \to \{b, c\}$ and $\{a, c, d, e\} \to \{b\}$. This example suggests that the set of all valid implications on a formal context which has a suitably large number of attributes will be too numerous for detailed inspection of each by the user to be feasible.

If a formal context constitutes an incomplete or unrepresentative sample from the object population, implications which are provisionally valid may be falsified by the addition of new objects. Such new objects may be sampled automatically from the population or chosen by a domain expert in order to expedite the acquisition of domain knowledge. Falsifying an implication also falsifies all implications which previously followed from it. For the purposes of manually verifying implications, it is therefore only necessary to inspect a complete and non-redundant set. The user may choose to systematically verify each implication in the set, explore the set of implications to identify those most in need of verification, or simply explore the implications to understand and absorb the domain knowledge they encode.

Allowing the user to sort and filter a complete and non-redundant set of implications in order to narrow down to those of interest, will scale to somewhat larger contexts than interaction with the larger set of all valid implications. In Section 6, we describe a data visualisation workflow designed to help the user identify "interesting" implications. Interestingness measures (see e.g. [20]) are beyond the scope of this chapter.

Diagonal context with $|M| = |G| = n$ has $\binom{n}{2}$ implications in basis.

Support The empirical evidence, or support, for an implication can be quantified as follows.

Definition 4. *The* **support** *of an implication* $\mathcal{L} \to \mathcal{R}$ *is the number* $|\mathcal{R}'|$ *of objects in the formal context which have all of the attributes in its consequent* $\mathcal{R}$.

Since the support of an implication is the cardinality of the extent of its consequent concept, implications having the same consequent have equal support. The support for each implication in our running example is listed in Table 2. The two implications having zero support are those associated with the infimum, which typically has an empty extent. These implications are qualitatively different from the remainder in that the usual interpretation, "Objects which have all attributes in $\mathcal{L}$ also have all attributes in $\mathcal{R} \setminus \mathcal{L}$," does not apply. However such empirically unsupported implications should not be entirely discounted, since they constitute predictions which may be subsequently falsified when new objects are added to the context, for example by a domain expert during attribute exploration [14].

Interaction primitives Users who are not yet fully conversant with implications, or with Formal Concept Analysis more generally, might be expected to benefit from the ability to explore a set of implications, compare selected implications, and examine their antecedents and abbreviated consequents. Coordination of interactive views of the context bigraph, lattice digraph and set of implications will assist such users to verify and interrogate implications and to relate them back to the data set from which they are derived. !Identify, examine and compare "interesting" implications from a potentially large set. !What about combining implications? !Introduce syntactic closure?

1. Query, sort and filter
2. Select multiple and compare
3. Select single and examine

1. `Overview`: Provide an overview(s) of all implications in a given set.
2. `Search`: Find all implications whose consequent contains a nominated set of attributes.
3. `Select`: Select a consequent.
4. `Refine`: Refine the selection from a consequent to an individual implication.
5. `Select`: Select an implication of interest.
6. `Examine`: Explore the antecedent and consequent of the selected implication.
7. `Compare`: For two implications with the same consequent, show the intersections of their antecedents and of their abbreviated consequents.
8. `Compare`: Find and compare two implications with comparable consequents.

Interaction primitive 5 differs from 3 in that the same consequent may be associated with multiple implications. Where this is the case, interaction primitive 4 is required to refine the selection to an individual implication. This could be achieved by combining 7 with user interaction to assign ambiguous attributes (brown vertices) to either the antecedent or consequent. Alternatively, it could be achieved through either a drop-down list which appears when a consequent is selected, or interaction with a coordinated view which tabulates the implications and highlights or shortlists those corresponding to the nominated consequent.

Interaction primitive 8 is a generalisation of 7, but with the precondition that the user has specified exactly two implications to compare. In what follows, we show that for implications in the canonical basis, two additional colours are required in order to distinguish four distinct sets.

Observation 3 *Let* $\mathcal{P}$ *and* $\mathcal{Q} \subset \mathcal{P}$ *be the antecedents of implications* $\mathcal{P} \to \mathcal{P}''$ *and* $\mathcal{Q} \to \mathcal{Q}''$ *in the canonical basis. Then the antecedent* $\mathcal{P} \supset \mathcal{Q}'' \supset \mathcal{Q}$ *of the former implication contains both the antecedent* $\mathcal{Q}$ *and the abbreviated consequent* $\mathcal{Q}'' \setminus \mathcal{Q}$ *of the latter, and the consequent* $\mathcal{Q}''$ *of the latter is disjoint from the abbreviated consequent* $\mathcal{P}'' \setminus \mathcal{P} \subset \mathcal{P}'' \setminus \mathcal{Q}''$ *of the former.*

When comparing the two implications in Observation 3, we need to visually distinguish between four disjoint sets: $\mathcal{Q}$, $\mathcal{Q}'' \setminus \mathcal{Q}$, $\mathcal{P} \setminus \mathcal{Q}''$ and $\mathcal{P}'' \setminus \mathcal{P}$. Members of $\mathcal{Q} \subset \mathcal{P}$ belong to the antecedents of both implications, and as per our colour convention their attribute concepts should be shown red. Members of $(\mathcal{Q}'' \setminus \mathcal{Q}) \subset \mathcal{P}$ are in the abbreviated consequent of the first and the antecedent of the second, and accordingly would be coloured brown. The alternative colouring ◓ would better signify that the attribute belongs to the abbreviated consequent of the super-concept and the antecedent of the sub-concept. Assigning any of these brown vertices to either the antecedent or abbreviated consequent sets would be sufficient to choose between the two concepts being compared; however, since they already have discrete consequents, this selection task is more readily achieved by selecting the corresponding consequent. Members of $\mathcal{P} \setminus \mathcal{Q}''$ and $\mathcal{P}'' \setminus \mathcal{P}$ are in the antecedent and abbreviated consequent, respectively, of the second implication only, and these attributes must be visually distinguished. We have previously used green to represent attributes which are in the antecedents of *both* implications being compared; green with reduced opacity could be used to indicate that this agreement does not exist for members of $\mathcal{P}'' \setminus \mathcal{P}$. The same lack of agreement on the elements of $\mathcal{P} \setminus \mathcal{Q}'' \subset \mathcal{P}$ would suggest red with reduced opacity for these vertices. The comparison of implications 3 and 6 using this colour scheme is illustrated in Figure 5. The antecedent $\mathcal{P}$ of the second implication is the union of the first three sets, since $\mathcal{P} = (\mathcal{P} \setminus \mathcal{Q}'') \cup (\mathcal{Q}'' \setminus \mathcal{Q}) \cup \mathcal{Q}$. Similarly, the consequent $\mathcal{P}''$ of the second implication is the union of all four sets.

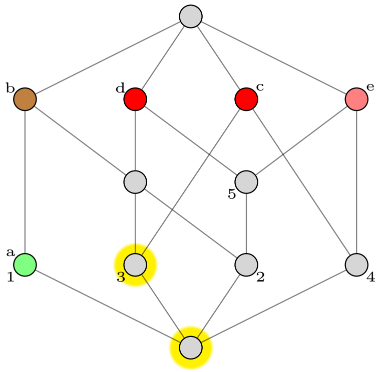


Fig. 5: Comparison of implications 3 and 6.

Dividing and conquering implications Consider a formal context which is Carve-divisible at the outermost level. Implications whose consequent is a concept within a parallel component other than a ghost infimum are also valid in the global context, since they can only involve attributes whose attribute concepts are either in that component or are the global supremum. With respect to the canonical basis, candidates $\mathcal{P}$ for pseudo-intents having $\mathcal{P}'' = M$ can be tested against each such $\mathcal{Q} \subset \mathcal{P}$ to ensure that $\mathcal{Q}'' \subset \mathcal{P} \subset M$. In the case of the CARVE example context, there are 8 implications in the canonical basis associated with the infimum and a further 9 associated with concepts in the 3 first-level containers. To confirm this, load `Carve example.cxt` into Concept Explorer and inspect the implications. The only consequent involving attributes from more than one CARVE component is the infimum, and this will typically have zero support. The remaining implications can be derived at lower computational cost from analysis of individual CARVE component. !Need to consider how this works for nested containers.

1. Interestingness measures less useful for differentiating between implications than between association rules because confidence $p(\mathcal{R}|\mathcal{L})$ is uniformly 100%. Maddouri and Gammoudi [20] claimed that each of the interestingness measures they studied could be expressed in terms of four probabilities: $p(\mathcal{L}\mathcal{R})$, $p(\mathcal{L}\bar{\mathcal{R}})$, $p(\bar{\mathcal{L}}\mathcal{R})$ and $p(\bar{\mathcal{L}}\bar{\mathcal{R}})$. Of these, the first is support normalised by $|G|$, and $p(\mathcal{L}\bar{\mathcal{R}}) = 0$ for implications. Departing from our previous notation, we have used $\mathcal{R}$ here to denote the *abbreviated* consequent.

3 Introduction to visual analytics

"Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces," which, inter alia, "seeks to marry techniques from information visualisation with techniques from computational transformation and analysis of data" [40]. We adopt a visual analytic approach to FCA by combining computational analysis with interactive visualisation.

3.1 Graph drawing

3.2 Data visualisation

3.3 Coordinated and multiple views

The discussion of implication in Section 2.4 would benefit from the discussion in this section.

3.4 Algorithmic analysis

3.5 Tight coupling

3.6 Scalability

Scalability is a key challenge for visual analytics. Algorithms must scale to large data sets, visualisations must make efficient and intelligible use of screen real-estate, and both must be responsive for interactive use. The number of formal concepts derived from a formal context is bounded above by an exponential function of the number of objects and attributes in that context. Consequently, three fundamental challenges confront those who wish to scale FCA to the interactive analysis of large data sets: the time required to enumerate the vertices, arcs and labels of the lattice digraph; the difficulty of meaningful and responsive user interaction with a large lattice digraph; and the discovery of insightful implications.

4 Layout, visualisation and interaction

This section provides a brief survey of techniques aimed at improving the scalability of FCA for more responsive visualisation and interaction.

4.1 Reducing digraph size

The most obvious approach to improving line diagram layout, visualisation and interaction, is to reduce the number of formal concepts. Querying and filtering the input context to remove objects and attributes which are not of interest will expedite concept enumeration and reduce the number of labels on the line diagram, but is not guaranteed to reduce the number of concepts [14]. Another means of achieving this objective is to impose a threshold on extent set cardinality, so that screen real-estate and user attention are reserved for formal concepts which represent suitably large subsets of the objects in the formal context. The partial order amongst these frequent closed item sets is referred to as an *iceberg* lattice. Algorithms exist [38] which exploit the monotonicity of the constraint on extent cardinality to expedite enumeration of the formal concepts.

4.2 Layout of line diagram

This section will be updated to reflect the work published in [30].

Standard algorithms [11, 39] and genetic variants (see e.g. [24]) exist for assigning the vertices of a DAG to layers and ordering them within each layer to improve aesthetic criteria such as edge crossings. In the present case of a lattice digraph, layer assignment is constrained by the maximum path length of a vertex from the source, and to the sink, vertex. The assignment of vertices to layers in the line diagram is typically under-constrained by the partial order, so that both layer assignment and horizontal order within a layer can be permuted when adjusting the graph layout to optimise aesthetic criteria. This graph layout problem has combinatorial complexity [11].

4.3 Interactive visualisation

Usability testing of FCA applied to the management of email has demonstrated that users can successfully interpret line diagrams [13]. However, the combinatorial explosion of concepts with increasing size of the formal context poses challenges for the layout and visualisation of, as well as interaction with, the lattice digraph. On-demand construction and layout of the entire lattice digraph cannot be achieved in interactive timescales for large lattices, so that either prior or user-guided construction and layout is required to support responsive interaction[4]. For contexts of even moderate size, the potentially large number of resultant vertices and arcs compete for limited screen real estate and challenge user comprehension.

To help the user manage this problem of scale, interactive exploration, as opposed to static presentation, of the line diagram is essential. Information visualisation techniques such as pan and zoom, focus-plus-context, details-on-demand and structural navigation [6] can support user interaction with, and comprehension of, large graphs [16]. For example, geometric zooming or distortion of the line diagram [7, 22] can help allocate more screen real-estate to an area of interest. Alternatively, structural navigation of the lattice digraph can be facilitated by presentation of the immediate graph neighbourhood of the current vertex [7, 12, 42], possibly combined with an overview showing where that vertex resides in the full lattice. A third option is an interactive version of nested line diagrams [7, 41], in which each vertex serves as a container within which to display the line diagram for the same object set and (some of) the remaining attributes. In many applications, however, it is not clear *a priori* how best to group the attributes, or how to order the groups for nesting.

4.4 Discovering or imposing tree structure

A range of mature visualisation and interaction techniques exist for tree, as opposed to lattice, data structures [18, 34, 37]. Operating system interfaces for the structural navigation of directory hierarchies are ubiquitous, and user intuition is accordingly well established [7]. This intuition can be exploited for visualisation of the concept lattice digraph, provided that a tree structure can be discovered in, or imposed on, the graph.

Any spanning tree of the lattice digraph, which is rooted at the source or sink vertex, would arguably serve this purpose. Melo et al. [21] investigate various criteria by which a single parent can be chosen for each concept. Whereas any given vertex will typically lie on multiple directed paths from the source [to the sink] of the lattice digraph, the corresponding path in a spanning tree is unique. To make it easier to purposefully locate a vertex of interest, or more likely that such a vertex might be encountered during less goal-directed user exploration, each vertex, along with the sub-lattice of which it is the supremum, can be replicated on demand under each of its parent vertices [7, 23].

Another approach to imposing tree structure on a graph to facilitate user interaction is hierarchical clustering or partitioning of its vertex set [16]. Hierarchical clustering involves the recursive application of a graph clustering algorithm to the clusters (sub-graphs) it identifies. Graph clustering involves optimising some measure of cluster quality, such as modularity, which takes into account factors such as the number, or total weight, of intra- versus inter-cluster links. Whilst

---

[4] User-guided construction of the lattice digraph is addressed in Section 5.2.

the global optimisation of modularity is NP complete, sub-optimal solutions can be computed for large graphs in responsive timeframes [8].

A range of techniques and tools exist for browsing hierarchically clustered graphs. Amongst these are structural zooming on inclusion layouts [34], and the GrouseFlocks environment [2] which supports the use and modification of multiple hierarchical clusterings on the same graph.

4.5 Demand for enhanced tool support

There is a clear trend in operating system interfaces towards tagging and querying rather than navigation of a hierarchical file system. Users typically require assistance in recalling or constructing a set of tags with which to retrieve a suitably small set of objects which contains the object(s) of interest. This trend will drive a demand for well-designed user interfaces through which, like trees before them, multiple-inheritance hierarchies become intuitive with use. The conceptually simple generalisation of a tree to allow a vertex to have multiple parents poses significant challenges for user navigation. More generally, scalable visualisation and interaction of multiple-inheritance hierarchies, and in particular of the digraph produced by FCA, remains an open challenge.

4.6 Implications

Discussion on using the lattice digraph as a substrate for the visualisation of implications. A view of the lattice digraph with the consequents of implications highlighted as digraph vertices is coordinated with alternate visualisations. The latter are discussed in Section 6.

Move some of the discussion from Section 2.4 to here?

5 Three FCA prototypes

This section briefly presents three software prototypes which the Defence Science and Technology Group has developed to address aspects of this scalability challenge.

5.1 Hierarchical parallel decomposition

Carve [26, 30] supports interactive analysis of large formal contexts by discovering and exploiting hierarchical structure which we have identified in bibliographic contexts, and which Bhatti et al. [5] found in software systems. That hierarchical structure is used to expedite and enhance both the layout of, and user interaction with, the concept lattice. The Carve algorithm [31] discovers a hierarchical decomposition of amenable contexts and of the corresponding lattice digraph. It produces a tree, representing both a partial parallel decomposition of the lattice digraph [11] and a corresponding decomposition of the context bigraph, along with the digraph itself. The decomposition tree for the example context in Figure 6a is shown in Figure 6b. Carve uses this tree both to divide and conquer the computational problem of laying out the digraph as a line diagram, and as a coordinated view to facilitate user interaction with the context bigraph and lattice digraph.

Consider replacing with context bigraph and lattice digraph terminology

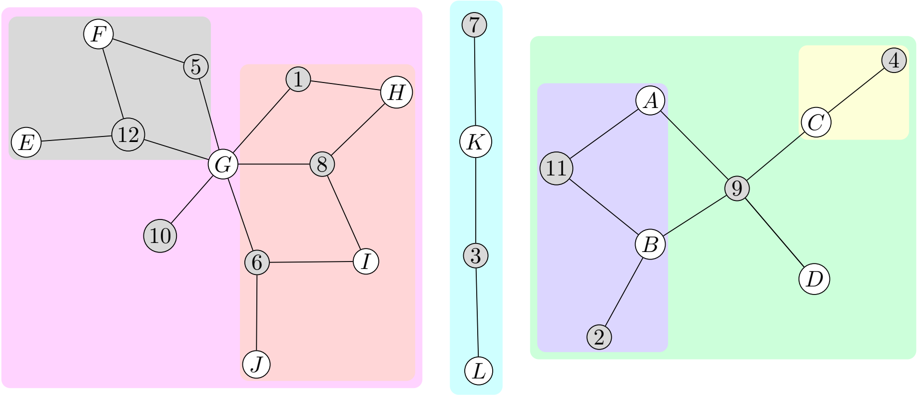
Each vertex of the tree returned by the Carve algorithm corresponds to a sub-context identified during hierarchical decomposition of the formal context, and to the lattice digraph for that sub-context. This tree can be drawn using an inclusion layout, in which each vertex of the tree is represented as a container within which the containers representing descendant tree vertices are nested. In Figures 6a and 6c, these nested containers are shown as coloured boxes whose colour is that of the corresponding vertex in the decomposition tree in Figure 6b. Each leaf vertex of this tree serves in Figure 6c as a container for the line diagram of the corresponding trivial or otherwise indivisible sub-lattice digraph. The use of these containers to enclose the corresponding bigraph and digraph vertices gives rise to the acronym Carve, which stands for Context Analysis through Recursive Vertex Enclosure.

The sink [source] vertex of the sub-lattice digraph corresponds to a concept in the global context (G, M, I) iff it has an attribute [object] label. Sink [source] vertices which are concepts are shown in Figure 6c as circles with black [white] fill, while those which are not are represented as point junctions of the arcs from their lower [to their upper] neighbours. Such junctions can be seen, for example, at the top and bottom of the salmon-coloured container in Figure 6c. Each sink [source] vertex is connected by an arc to its counterpart in the parent container. In the case where the former vertex is not a concept, it serves as a collection [distribution] point for a “trunk” arc to [from] its counterpart. These trunk arcs reduce clutter by condensing multiple arcs into a single line.

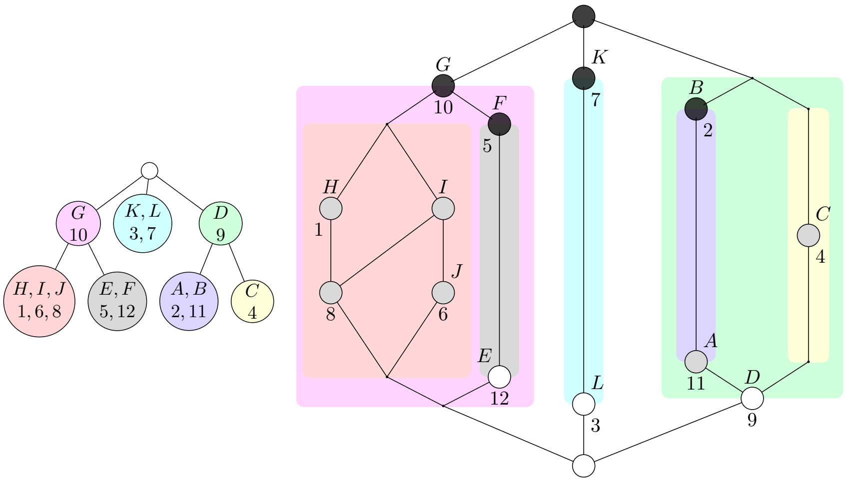
The Carve software prototype uses coordinated and multiple views to present the decomposition tree, context bigraph and lattice digraph. The tree view uses a layout typical of file system browsers, which is more space-efficient than that shown in Figure 6b. Selecting a node of the decomposition tree updates the context bigraph and lattice digraph views to show only the corresponding sub-context. These two sub-graphs can be laid out simultaneously to spatially cluster bicliques in the context bigraph and improve the intelligibility of the lattice digraph [30]. By interacting with the decomposition tree, the user can drill down to sub-contexts of interest, for which the context bigraph and lattice digraph can be significantly smaller than for the global context.

5.2 User-guided FCA

The DAnCE prototype [27, 29] improves the scalability of FCA for interactive use by allowing the user to steer the analysis towards areas of interest, and to halt construction of the lattice digraph as soon as their analytic objectives are satisfied. The resultant lattice digraph will be more task-focused, and, depending on the application, may be considerably smaller, than the lattice digraph for the original context.



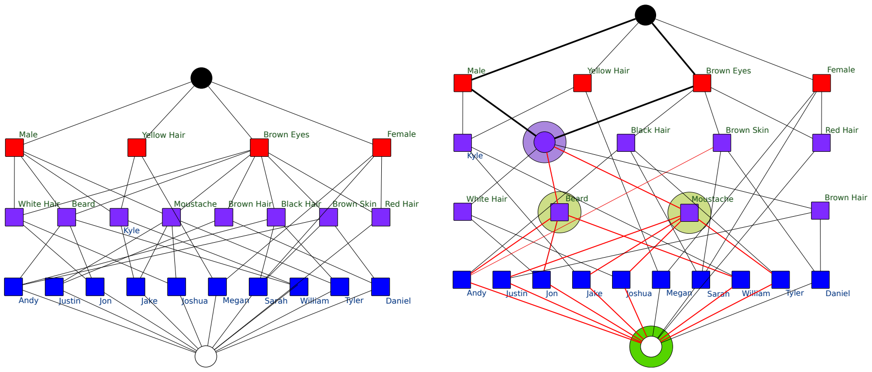
(a) Context bigraph



(b) Decomposition tree (c) Modified line diagram

Fig. 6: Example context bigraph, the corresponding decomposition tree, and the modified line diagram with discovered sub-lattices within nested containers.

We have developed the Dynamic Analysis for Conceptual Exploration (DAnCE) prototype to explore this possibility, modifying a top-down algorithm for concept enumeration to respond to user control and guidance [29]. Instead of autonomous enumeration of all concepts followed by batch-mode construction of the lattice digraph and corresponding line diagram, DAnCE allows the user interactive control over the process of concept enumeration and provides dynamic, incremental update of the line diagram. In addition to being able to start, stop, restart and step through the process of concept enumeration, the user can: select a concept of interest and prioritise the enumeration of comparable concepts which are below it in the lattice; or select multiple concepts and prioritise the generation



(a) Initial diagram (b) Multiple selection

Fig. 7: The initial line diagram and the result of multiple selection.

of the concepts which correspond to the set intersections of their intents and extents. Each vertex and arc is displayed either as soon as it is discovered, or in a batch-mode update of the line diagram after a specified number of steps of the enumeration algorithm.

Visualisation challenges faced by DANCE include: ensuring intelligible layout of the partially-constructed diagram; maintaining the user's mental model while vertices and arcs are added; and ensuring that the labelling scheme described in Section 2.3 applies when some lattice digraph vertices and edges have yet to be discovered. DANCE maintains the complete line diagram for the partial order amongst the concepts generated to date, ensuring its consistent interpretation as new concepts are added. Figure 7 shows mock-ups of this line diagram for an example formal context consisting of people and their physical attributes. Figure 7a depicts the state of the line diagram first presented to the user. By this stage, all of the attribute and object concepts have been generated by the concept enumeration algorithm, labelled, and laid out to establish the framework for insertion of subsequent concepts. Algorithms exist for efficient generation of the requisite attribute-object concept (AOC) poset [4] and for horizontally ordering the atoms and co-atoms to reduce the number of edge crossings in the lattice digraph [30]. Establishing this framework *ab initio* minimises subsequent disruption of the user's mental model as new concepts are inserted into the line diagram, while the presence and labelling of all attribute and object concepts ensures that intent and extent membership can be read from the outset. The problem of efficiently generating only the remaining – abstract – concepts remains open.

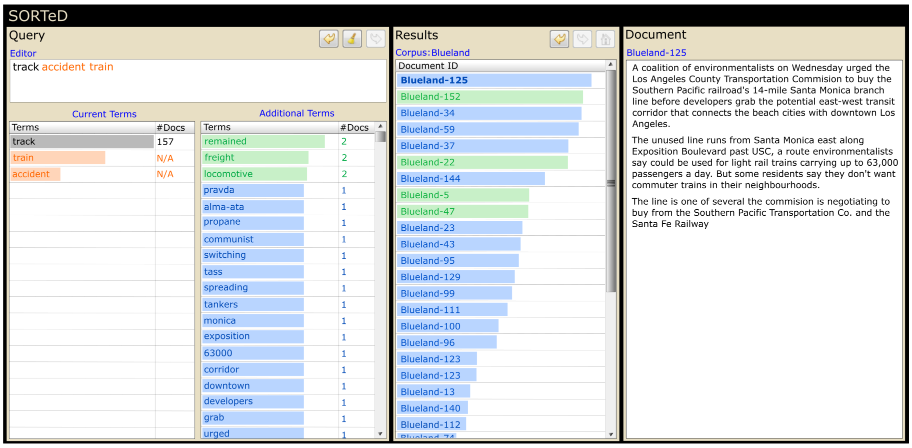
Figure 7b shows the result of the user selecting in this diagram the attribute concepts for "Beard" and "Moustache", which are highlighted in response with small grey halos. This multiple selection triggers the calculation of the extent and intent intersections for the selected concepts. The former corresponds to the infimum, which is accordingly highlighted with a green halo; the latter corresponds to a new concept, which is consequently inserted into the line diagram and highlighted with a purple halo. Since this extent intersection has path length 2 from the supremum, a new row has been inserted to accommodate concepts now with path length 3, and lower neighbours demoted to it. The extent intersection is inserted into row 2 at ordinal position 2 of 5; this position is based on the horizontal barycentre of its associated layer 1 ancestors (co-atoms) and layer 5 descendants (atoms), which are predominantly to the left of the centreline.

The technique described here for user-guided FCA could be applied within the leaf-node containers of CARVE in cases where the corresponding sub-context remains large. This can occur in contexts which are especially large to start with, or are not particularly amenable to the divide-and-conquer technique employed by CARVE. Whereas CARVE currently performs batch-mode construction and layout of the sub-context digraph during leaf-node traversal, this could be deferred to support user-guided construction.

5.3 Structural navigation

The SORTeD prototype [27, 28] supports the retrieval of documents (objects) from a corpus based on queries over the terms (attributes) they contain. User queries are constrained to term combinations which occur in the corpus, and are generalised by removing, or specialised by adding, terms to navigate to comparable concepts. Unlike previous interfaces for structural navigation of the lattice digraph [7, 12, 42], those comparable concepts are not constrained to be neighbours of the current concept. Depicted in Figure 8, the user interface mockup offers valid terms to add or remove from the query. Terms not specified by the user, but which are in the closure of the set of user-specified query terms, are referred to as *closure terms*, and are shown orange in Figure 8. The computational challenge is to compute the set of all concepts reachable from the query concept by the addition or removal of a user-specified query term to facilitate interactive use. Despite structurally navigating the lattice digraph through a keyhole view, the user may be unaware of the lattice digraph's existence, relying instead on intuition established through long-term use of conventional information retrieval interfaces.

In SORTeD, FCA provides a mechanism for literal search over the corpus, with the user interface assisting the construction and interactive refinement of conjunctive Boolean queries. The search results are ranked using Latent Semantic Analysis (LSA) according to their cosine similarity to the search terms [19], and the search terms are ranked according to their cosine similarity to the result set. The latter ranking is less conventional, indicating the comparative relevance of the search terms to the result set, from which the user may judge whether the result set is likely to satisfy their requirements. It is these two "semantic" rankings which give rise to the acronym SORTED, which stands for Semantically-Ordered Ranking of Terms and Documents.



track accident train

Current Terms

Terms	#Docs
track	157
train	N/A
accident	N/A

Additional Terms

Terms	#Docs
remained	2
freight	2
locomotive	2
pravda	1
alma-ata	1
propane	1
communist	1
switching	1
tass	1
spreading	1
tankers	1
monica	1
exposition	1
63000	1
corridor	1
downtown	1
developers	1
grab	1
urged	1

Document ID
Blueland-125
Blueland-152
Blueland-34
Blueland-59
Blueland-37
Blueland-22
Blueland-144
Blueland-5
Blueland-47
Blueland-23
Blueland-43
Blueland-95
Blueland-129
Blueland-99
Blueland-111
Blueland-100
Blueland-96
Blueland-123
Blueland-123
Blueland-13
Blueland-140
Blueland-112

A coalition of environmentalists on Wednesday urged the Los Angeles County Transportation Commision to buy the Southern Pacific railroad's 14-mile Santa Monica branch line before developers grab the potential east-west transit corridor that connects the beach cities with downtown Los Angeles.

The unused line runs from Santa Monica east along Exposition Boulevard past USC, a route environmentalists say could be used for light rail trains carrying up to 63,000 passengers a day. But some residents say they don't want commuter trains in their neighbourhoods.

The line is one of several the commision is negotiating to buy from the Southern Pacific Transportation Co. and the Santa Fe Railway

Fig. 8: SORTeD interface for information retrieval combining FCA and LSA.

In addition to offering *conjunctive* search terms – those which co-occur with the existing search terms – to assist the user to refine the query, the interface also offers, ranks and visually distinguishes *disjunctive* terms – those which are only semantically related to the result set. Conjunctive search terms are shown green in the `Additional Terms` list in Figure 8, while disjunctive search terms are shown blue. Similarly, the `Results` list shows in green the identifiers of documents matching the current conjunctive Boolean query, and in blue the identifiers of those which do not match. In this example, document `Blueland-125` is not a literal match to the conjunctive Boolean query, but is the highest-ranking semantic match.

Selecting a disjunctive term currently initiates a literal query in which the selected term is substituted for the existing set of query terms. A technique has subsequently been described whereby the query is instead edited to include the disjunctive term [28]. The semi-automated editing algorithm searches the lattice digraph constructed from the document corpus for an intent which: contains the disjunctive term; and preserves as many user-specified query terms, and introduces as few new terms, as possible. If more than one intent scores equally against these criteria, the user is asked to choose which of them better reflects their information needs. These choices could be computed for all disjunctive terms while the user assesses the result of their most recent query.

Any closure terms identified during this interactive query refinement process constitute the abbreviated consequent of an implication whose antecedent is the current set of user-specified query terms. Thus while querying the document corpus, the user may also be discovering implications amongst the terms present in the corpus.

6 Discovering insightful implications

This section is based on [9].

6.1 Visualisation of implications

6.2 Our Approach to Data Visualisation for Implications

The Attribute Plot

The Implication Plot

The Rules Data Table

7 Scaling visual analytic FCA to big data volumes

This section discusses future prospects for scaling visual analytic FCA to the data volumes encompassed by the term “big data”.

8 Conclusions and future work

References

1. Aho, A., Garey, M., Ullman, J.: The transitive reduction of a directed graph. SIAM Journal on Computing **1**(2), 131–137 (June 1972), `http://dx.doi.org/10.1137/0201008`
2. Archambault, D., Munzner, T., Auber, D.: Grouseflocks: Steerable exploration of graph hierarchy space. IEEE Transactions on Visualization and Computer Graphics **14**(4), 900–913 (July/August 2008)
3. Armstrong, W.W.: Dependency structures of database relationships. Information Processing pp. 580–583 (1974), `https://web.archive.org/web/20180126091352if_/https://ipfs.io/ipfs/QmWYWTGUZyTm2iRFTZY2pTr2x1vWkDiJr2CBp2PGVpSVSv`
4. Berry, A., Gutierrez, A., Huchard, M., Napoli, A., Sigayret, A.: Hermes: a simple and efficient algorithm for building the AOC-poset of a binary relation. Annals of Mathematics and Artificial Intelligence **72**(1), 45–71 (Oct 2014). https://doi.org/10.1007/s10472-014-9418-6
5. Bhatti, M.U., Anquetil, N., Huchard, M., Ducasse, S.: A catalog of patterns for concept lattice interpretation in software reengineering. In: Zhang, D., Reformat, M., Gokhale, S., Maldonado, J.C. (eds.) Proceedings of the 24th International Conference on Software Engineering & Knowledge Engineering (SEKE’2012). pp. 118–124. Knowledge Systems Institute Graduate School, Hotel Sofitel, Redwood City, San Francisco Bay, USA (July 1-3 2012), `http://www.ksi.edu/seke/Proceedings/seke/SEKE2012_Proceedings.pdf`

6. Card, S., Mackinlay, J., Shneiderman, B.: Readings in information visualization: Using vision to think. Morgan Kaufmann (1999)
7. Carpineto, C., Romano, G.: Exploiting the potential of concept lattices for information retrieval with CREDO. Journal of Universal Computing **10**(8), 985–1013 (2004). https://doi.org/10.3217/jucs-010-08-0985
8. Clémençon, S., Arazoza, H.D., Rossi, F., Tran, V.C.: Hierarchical clustering for graph visualization. In: Proceedings of XIXth European Symposium on Artificial Neural Networks (ESANN 2011). pp. 227–232. Bruges, Belgium (April 2011)
9. Cordero, P., Enciso, M., Mora, A., Gomez González, P.: An approach to visualize implications. In: Cristea, D., Le Ber, F., Missaoui, R., Kwuida, L., Sertkaya, B. (eds.) Supplementary Proceedings of ICFCA 2019 Conference and Workshops (2019), `http://ceur-ws.org/Vol-2378/longBDE2.pdf`
10. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge University Press, England, second edn. (2002)
11. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.: Graph drawing: Algorithms for the visualization of graphs. Prentice Hall, Upper Saddle River, NJ, USA (1999)
12. Ducrou, J., Eklund, P.: Browsing and searching MPEG-7 images using Formal Concept Analysis. In: Proceedings of the 24th IASTED International Conference on Artificial Intelligence and Applications (AIA'06). pp. 317–322. ACTA Press, Innsbruck, Austria (2006)
13. Eklund, P.W., Ducrou, J., Brawn, P.: Concept lattices for information visualization: Can novices read line diagrams? In: Eklund, P. (ed.) Proceedings of the 2nd International Conference on Formal Concept Analysis (FCA'04). Lecture Notes in Computer Science, vol. 2961, pp. 57–73. Springer-Verlag, Sydney, Australia (February 23-26 2004). https://doi.org/10.1007/978-3-540-24651-0_7
14. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin, Heidelberg (1999). https://doi.org/10.1007/978-3-642-59830-2
15. Guigues, J.l., Duquenne, V.: Familles minimales d'implications informatives d'un tableau de données binaires. Mathématiques et Sciences Humaines **95**, 5–18 (1986)
16. Herman, I., Melançon, G., Marshall, M.S.: Graph visualization and navigation in information visualization: a survey. IEEE Transactions on Visualization and Computer Graphics **6**, 24–43 (2000). https://doi.org/10.1109/2945.841119
17. Ignatov, D.I., Nourine, L. (eds.): Proceedings of The Fourteenth International Conference on Concept Lattices and Their Applications (CLA 2018), vol. 2123. CEUR-WS.org (2018), `http://ceur-ws.org/Vol-2123/`
18. Lamping, J., Rao, R., Pirolli, P.: A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 401–408. CHI '95, ACM Press/Addison-Wesley Publishing Co., New York,

NY, USA (1995). https://doi.org/10.1145/223904.223956, `http://dx.doi.org/10.1145/223904.223956`

19. Landauer, T., McNamara, D., Dennis, S., Kintsch, W. (eds.): Handbook of Latent Semantic Analysis. Taylor & Francis (2013). https://doi.org/10.4324/9780203936399
20. Maddouri, M., Gammoudi, J.: On semantic properties of interestingness measures for extracting rules from data. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) Adaptive and Natural Computing Algorithms. pp. 148–158. Springer, Berlin, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71618-1_17
21. Melo, C., Bezerianos, A., Le-Grand, B., Aufaure, M.A.: Cubix: A visual analytics tool for Formal Concept Analysis. In: IHM '11: 23rd French Speaking Conference on Human-Computer Interaction. ACM, New York, NY, USA (2011), `http://ihm2011.unice.fr/06_Ressources/Actes_Complementaires_IHM11.pdf`
22. Melo, C., Le-Grand, B., Marie-Aude, A., Bezerianos, A.: Extracting and visualising tree-like structures from concept lattices. In: Proceedings of the 15th International Conference on Information Visualisation. pp. 261–266. IEEE Computer Society, Washington, DC, USA (2011). https://doi.org/10.1109/IV.2011.46
23. Nauer, E., Toussaint, Y.: CreChainDo: An iterative and interactive web information retrieval system based on lattices. International Journal of General Systems **38**(4), 363–378 (May 2009)
24. Owais, S., Gajdoš, P., Snášel, V.: Usage of genetic algorithm for lattice drawing. In: Bělohlávek, R., Snášel, V. (eds.) Concept Lattices and Applications 2005. pp. 82–91 (2005)
25. Papoulis, A.: Probability, Random Variables, and Stochastic Processes. McGraw-Hill Series in Electrical Engineering, McGraw-Hill, second edn. (1984)
26. Pattison, T., Weber, D., Ceglar, A.: Enhancing layout and interaction in Formal Concept Analysis. In: 2014 IEEE Pacific Visualization Symposium (PacificVis). pp. 248–252. IEEE (March 2014). https://doi.org/10.1109/PacificVis.2014.21
27. Pattison, T.: Interactive visualisation of formal concept lattices. In: Burton, J., Stapleton, G., Klein, K. (eds.) Joint Proceedings of the 4th International Workshop on Euler Diagrams (ED 2014) and the 1st International Workshop on Graph Visualization in Practice (GViP 2014). CEUR Workshop Proceedings, vol. 1244, pp. 78–79. CEUR-WS.org, Melbourne, Australia (Jul 28 – Aug 1 2014), `http://ceur-ws.org/Vol-1244/GViP-paper6.pdf`
28. Pattison, T.: Interactive query refinement using Formal Concept Analysis. In: Ignatov and Nourine [17], `http://ceur-ws.org/Vol-2123/paper17.pdf`
29. Pattison, T., Ceglar, A.: Interaction challenges for the dynamic construction of partially-ordered sets. In: Bertet, K., Rudolph, S. (eds.) Proceedings of the Eleventh International Conference on Concept Lattices and Their Applications. pp. 23–34. CEUR Workshop Proceedings, CEUR-

WS.org, Košice, Slovakia (October 2014), `http://ceur-ws.org/Vol-1252/cla2014_submission_2.pdf`

30. Pattison, T., Ceglar, A.: Simultaneous, polynomial-time layout of context bigraph and lattice digraph. In: Cristea, D., Le Ber, F., Sertkaya, B. (eds.) Proceedings of the 15th International Conference on Formal Concept Analysis. LNCS: Artificial Intelligence, vol. 11511, pp. 223–240 (2019). https://doi.org/10.1007/978-3-030-21462-3
31. Pattison, T., Ceglar, A., Weber, D.: Efficient Formal Concept Analysis through recursive context partitioning. In: Ignatov and Nourine [17], pp. 219–230, `http://ceur-ws.org/Vol-2123/`
32. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Formal Concept Analysis in knowledge discovery: A survey. In: Croitoru, M., Ferré, S., Lukose, D. (eds.) Conceptual Structures: From Information to Intelligence, pp. 139–153. LNCS: Artificial Intelligence, Springer Verlag (2010), `http://books.google.com.au/books?id=7TZ_4KOr_SUC`
33. Priss, U.: Formal Concept Analysis in Information Science. Annual Review of Information Science and Technology **40**, 521–543 (2006). https://doi.org/10.1002/aris.1440400120
34. Pulo, K., Eades, P., Takatsuko., M.: Smooth structural zooming of h-v inclusion tree layouts. In: Proceedings of International Conference on Coordinated & Multiple Views in Exploratory Visualization (2003)
35. Russell, S., Norvig, P. (eds.): Artificial Intelligence: A Modern Approach. Artificial Intelligence, Prentice Hall, second edn. (2003)
36. Sertkaya, B.: A survey on how description logic ontologies benefit from FCA. In: Kryszkiewicz, M., Obiedkov, S.A. (eds.) Proceedings of the 7th International Conference on Concept Lattices and Their Applications. CEUR Workshop Proceedings, vol. 672. CEUR-WS.org, Sevilla, Spain (October 19-21 2010), `http://ceur-ws.org/Vol-672/paper2.pdf`
37. Shneiderman, B.: Tree visualization with treemaps: a 2-D space-filling approach. ACM Transactions on Graphics **11**(1), 92–99 (Jan 1992)
38. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing Iceberg concept lattices with Titanic. Data and Knowledge Engineering **42**(2), 189–222 (2002)
39. Sugiyama, K., Tagawa, S., Toda., M.: Methods for visual understanding of hierarchical system structures. IEEE Transactions on Systems, Man, and Cybernetics **11**(2), 109–125 (1981)
40. Thomas, J.J., Cook, K.A. (eds.): Illuminating the Path: The Research and Development Agenda for Visual Analytics. IEEE Press (2005), `http://vis.pnnl.gov/pdf/RD_Agenda_VisualAnalytics.pdf`
41. Wille, R.: Restructuring lattice theory: An approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered sets, vol. 23, pp. 445–470. Reidel Publishing, Dordrecht–Boston (1982)
42. Wray, T., Eklund, P.: Exploring the information space of cultural collections. In: Valtchev, P., Jäschke, R. (eds.) Formal Concept Analysis: 9th International Conference, ICFCA 2011. Springer Verlag, Nicosia, Cyprus (May 2011)