`FCA Tools Bundle` - a Tool that Enables Dyadic and Triadic Conceptual Navigation

Levente Lorand Kis, Christian Săcărea, and Diana Troancă

Babeş-Bolyai University Cluj-Napoca
kis_lori@yahoo.com, csacarea@cs.ubbcluj.ro, dianat@cs.ubbcluj.ro

Abstract. Formal Concept Analysis is a prominent field of applied mathematics handling collections of knowledge - formal concepts - which are derived from some basic data types, called formal contexts by using concept forming operators. One of the strengths of FCA is the elegant, intuitive and powerful graphical representation of landscapes of knowledge as concept lattices. Nevertheless, in case of triadic FCA (3FCA) for more than 20 years there was no automatic tool for graphical representation of triconcept sets. Moreover, the triangular representation of trilattices, used so far in 3FCA has several disadvantages. Besides the lack of clarity in representation, one major disadvantage is that not every trilattice has a triangular diagram representation. In this paper we focus on the problem of locally navigating in triconcept sets and propose a tool which implements this navigation paradigm. To the best of our knowledge this is the first tool which makes navigation in larger triconcepts sets possible, by flipping through a certain collection of concept lattices.

1 Introduction

The FCA community agrees upon the necessity of having powerful software tools for formal context handling and lattice representation, as well as for other FCA features which are close to the heart of everyone from this community. Moreover, these tools need to be also accessible also to users outside the FCA community and there is plenty of work which has been done in this area. Without being comprehensive, an overview of FCA software was compiled by Priss and can be found on her webpage[1]. We will not cite any of these tools, since citing all of them would expand the Bibliography section out of the scope of a workshop paper, and making a citation selection would mean to emphasize some tools and to disregard some others, which again is not the scope of this paper.

Nevertheless, we need to point out that all tools are handling the 'classical' dyadic case or some extensions like *patterns structures* or *relational FCA*, while the triadic case is neglected. We do not know any tool able to draw a trilattice diagram or to represent somehow triconcept sets. In our opinion, triadic FCA (3FCA) has a real potential for real life applications once the problem of knowledge representation has been solved in a satisfactory manner. Based on some

[1] `http://www.upriss.org.uk/fca/fcasoftware.html`

previous attempts of representing triconcept sets as graphs [9] or using a circular visualization tool [3], we focused on a local navigation paradigm for triconcept sets [8]. This paradigm is meant for exploring triconcept sets by choosing one triconcept (A_1, A_2, A_3) and locking one of its components (either the extent, or the intent, or the modus) and then using the dyadic projection $\mathbb{K}^{ij}_{Ak}$ of related *reachable* concepts. Brief details on this paradigm are included in Section 2.

In this paper, we present the current state of `FCA Tools Bundle`, a collection of tools for dyadic and triadic Formal Concept Analysis. The triadic part consists of the implementation of the above-mentioned local navigation paradigm, while the dyadic part contains lattice building and visualization tools. In Section 2 we briefly introduce some preliminaries regarding triadic contexts and the suggested navigation paradigm, while in Section 3 we motivate the implementation of the tool suite. Section 4 describes the features implemented by the tool, while Section 5 presents an example of navigating through a triadic dataset. In Section 6 we describe the architecture of the tool and the technologies as well as the external tools used in the implementation. Finally, Section 7 describes some of the future directions for the development of the tool.

2 Preliminaries

We focus in this section only on the local navigation paradigm, as it has been introduced in our previous paper [8]. For more, we refer to the standard literature [4, 7].

Let $\mathbb{K} = (K_1, K_2, K_3, Y)$ be a tricontext. There are two types of dyadic projections which can be obtained from $\mathbb{K}$. First, we can 'flatten' $\mathbb{K}$ by slicing it and putting all slices side by side: $\mathbb{K}^{(i)} := (K_i, K_j \times K_k, Y^{(i)})$, with $(a_i, (a_j, a_k)) \in Y^{(i)}$ if and only if $(a_i, a_j, a_k) \in Y$, where $a_i \in K_i, a_j \in K_j, a_k \in K_k$ and $\{i, j, k\} = \{1, 2, 3\}$ are pairwise different.

Another projection is given by 'locking' a subset $A_k \subseteq K_k$ and intersecting all those slices that correspond to elements of A_k. More specifically, for $\{i, j, k\} = \{1, 2, 3\}$ and $A_k \subseteq K_k$, we define $\mathbb{K}^{(ij)}_{A_k} := (K_i, K_j, Y^{(ij)}_{A_k})$, where $(a_i, a_j) \in Y^{(ij)}_{A_k}$ if and only if $(a_i, a_j, a_k) \in Y$ for all $a_k \in A_k$.

In our previous paper, we have studied some theoretical issues regarding triconcepts and these dyadic projections [8] . If (A_1, A_2, A_3) is a triconcept, then the dimensions $i \in \{1, 2, 3\}$ are called *i-perspectives.* Performing a dyadic projection after one of these three perspectives, one can observe that there is a one-to-one correspondence between the corresponding dyadic concepts and some triconcepts of the original tricontext $\mathbb{K}$. These triconcepts are called *directly reachable* using perspective i. The reachability relation is defined as the transitive closure of the direct reachability relation. This enables *local navigation* in triconcept sets, by iteratively choosing a perspective, then browsing the direct reachable triconcepts set by exploring the subsequent projected dyadic concept lattice and so on. This perspective-locking-and-unlocking procedure gives rise to interesting theoretical questions, partly solved in our previous paper [8]. Of course, the *local character* of the navigation comes with the cost of loosing overview. Nevertheless, navigating along the reachability relation is synonym to flipping through several concept lattices and investigate connections of triconcept sets which are not visible from the 'classical' trilattice diagram representation. Section 4 describes the practical implementation of this local navigation paradigm in `FCA Tools Bundle`.

3 Motivation

The main motivation of developing this tool comes from practical applications of FCA. Some data have an inherent triadic structure, folksonomies being one of the prominent examples, and for these data one can generate either a complete list of triconcepts or one can restrict only to the most frequent [6]. But even data without inherent triadic structure might be investigated from a triadic point of view, which increases the popularity of 3FCA and makes it suitable for more applications. These data can usually be interpreted as many-valued contexts which are then scaled using the `ToscanaJ Suite` [2]. Using a locally developed plugin called `Toscana2Trias`, one can build a triadic view on the data directly from `ToscanaJ`. Triconcepts can be easily computed but without a visualization tool there is no navigation support and hence the applicability of 3FCA remains uncertain.

Making FCA more popular outside of its natural community is another goal which motivates this development. The paradigm introduced in our previous paper seems to be a good starting point, since it enables navigation in triconcept sets by using the natural elegance and expressiveness of concept lattices [8].

Once the paradigm has been set up, several problems had to be overcome. For instance, the well-known node overlapping, which requires manual rearrangement of nodes. For this purpose, we implement a concept lattice generation method that integrates a collision detection functionality. This feature will enable to automatically check when nodes in the diagram are overlapping and rearrange the diagram in such a way that there are no more overlappings, i.e. collisions among the nodes.

Usability and accessibility is another goal of this tool development. Hence it has been developed as an open source project. Furthermore, it is made public as a web site which requires registration, but no other installation process which makes it easy to use.

Last but not least, revival of software development for FCA is in our opinion, a major objective for our community, to which we strongly adhere.

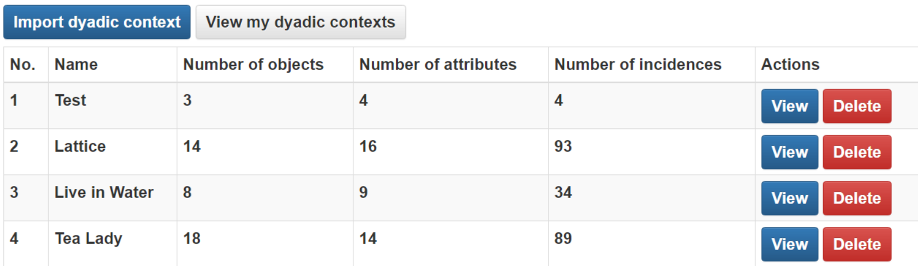
4 `FCA Tools Bundle` - Description and Features

The `FCA Tools Bundle`[2] currently implements features for dyadic and triadic formal concept analysis. The main purpose of the tool is to enable the user to

[2] `https://fca-tools-bundle.com`

visualize formal contexts of different dimensions in several formats (concept list, concept lattice) and to interact with them.

The tool offers some public contexts for users who want to test the functionalities (see Figure 1) and, in addition, there is an import functionality that allows users to add other dyadic or triadic contexts. The allowed formats for importing a formal context are `cxt` and `csv`. The `cxt` format is the standard format for dyadic contexts, while for triadic context it is a straightforward extension of the format. When using the second format, the `csv` file must contain only the tuples of the relation from which the context can be reconstructed.


Import dyadic context | View my dyadic contexts

No.	Name	Number of objects	Number of attributes	Number of incidences	Actions
1	Test	3	4	4	View Delete
2	Lattice	14	16	93	View Delete
3	Live in Water	8	9	34	View Delete
4	Tea Lady	18	14	89	View Delete

Fig. 1: List of dyadic contexts

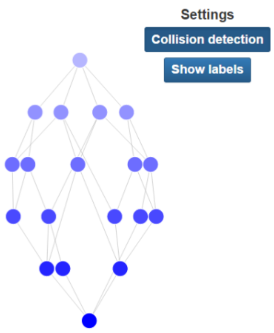


Fig. 2: Concept Lattice of a Dyadic Context

The key features offered by `FCA Tools Bundle` for a dyadic context are to visualize the details of the context, i.e. object set, attribute set, incidences and concept set and to compute and visualize the concept lattice of the context represented in Figure 2.

As mentioned previously, another useful feature offered by the tool is the possibility to avoid overlappings in the concept lattice using a collision detection functionality. After computing the concept lattice, the y coordinate of the nodes is locked, but they are allowed to move freely in their respective layer. Therefore, the user can move the nodes on the x-axis and rearrange them as they find suitable.

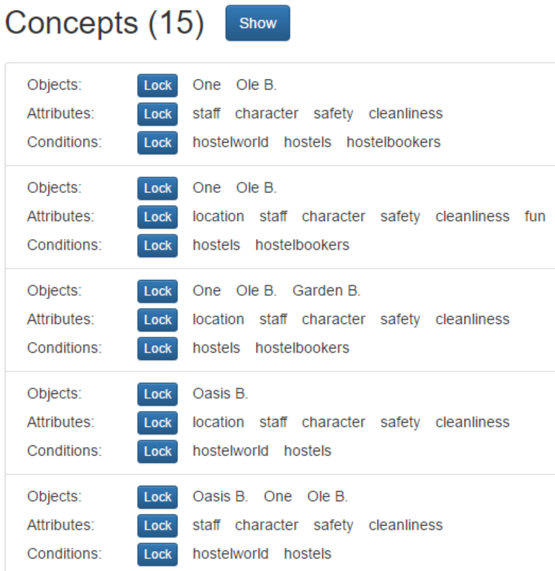


Fig. 3: Triconcepts of the hostels context

For a triadic context, one can first visualize the concept list. Figure 3 shows a list of triconcepts from the hostel example given by Glodeanu [5]. In addition, an important functionality, which was not implemented by any previous tool, is a local visualization of parts of the triadic context which enables the navigation paradigm in the triconcept set. One can choose a set of elements (of the same type, i.e. objects, attributes, or conditions) to project on and visualize the concept lattice of the dyadic projection. This functionality can be used for the navigation paradigm based on dyadic projections ([8]) and a proof of concept will be presented in Section 5.

5 Example

In a previous paper we defined a local navigation paradigm for triadic contexts based on appropriately defined dyadic projections. In this section we show how `FCA Tools Bundle` can be used to navigate in a triadic context using this paradigm. For this purpose we consider the hostels context, which was previously defined by Glodeanu [5]. Figure 4 depicts the layered cross-table representation of the hostel context.

As mentioned previously, some of the triconcepts of the hostel context are represented in Figure 3. We choose the triconcept

$$T_1 = (\{One, OleB., GardenB.\}, \{location, staff, character, safety, cleanliness\},$$
$$\{hostels, hostelbookers\})$$

as a starting point and lock the third perspective, i.e. the set of conditions $\{hostels, hostelbookers\}$. The corresponding dyadic projection is depicted in Fig-

hostelworld	character	safety	location	staff	fun	cleanliness
Nuevo S.			×			
Samay		×	×	×		×
Oasis B.	×	×	×	×		×
One	×	×		×		×
Ole B.	×	×		×		×
Garden B.			×	×		×

hostels	character	safety	location	staff	fun	cleanliness
Nuevo S.			×	×		
Samay		×	×	×		×
Oasis B.	×	×	×	×	×	×
One	×	×	×	×	×	×
Ole B.	×	×	×	×	×	×
Garden B.	×	×	×	×	×	×

hostelbookers	character	safety	location	staff	fun	cleanliness
Nuevo S.			×	×		
Samay	×	×	×	×		×
Oasis B.		×	×	×	×	×
One	×	×	×	×	×	×
Ole B.	×	×	×	×	×	×
Garden B.	×	×	×	×		×

Fig. 4: Hostels tricontext

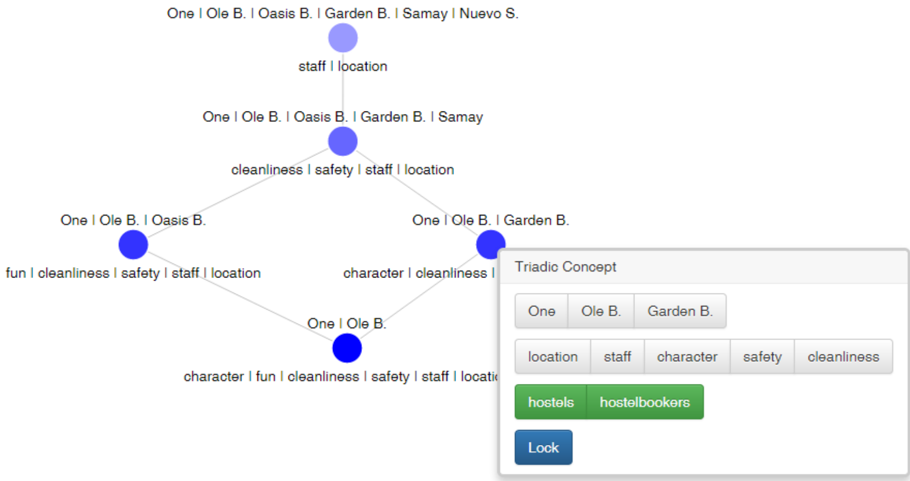


Fig. 5: Dyadic projection on the condition set $\{hostels, hostelbookers\}$

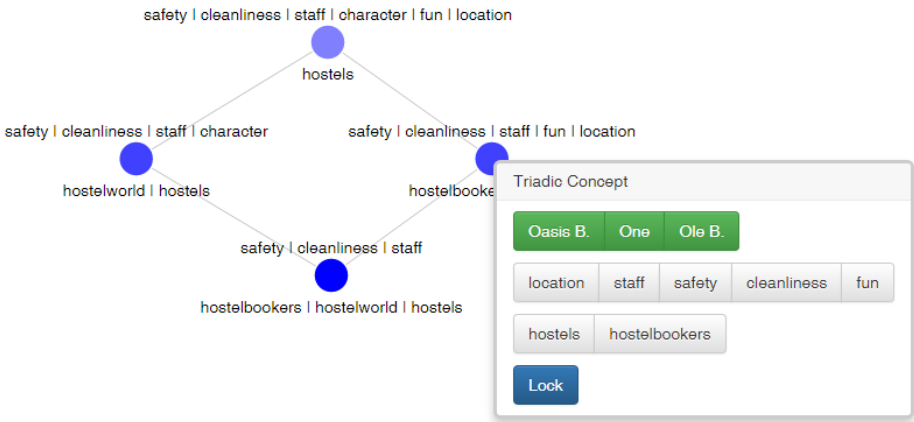


Fig. 6: Dyadic projection on the object set $\{OasisB., One, OleB.\}$

ure 5. By right clicking a dyadic concept in the projection the user can see the associated triadic concept. Here, the triconcept T_1 corresponds to the right-most dyadic concept. Herefrom, one can choose a different triconcept, for example the left-most concept

$$T_2 = (\{One, OleB., OasisB.\}, \{fun, cleanliness, safety, staff, location\} \\ \{hostels, hostelbookers\}),$$

and project on the object set. The corresponding concept lattice, after this second lock step, is depicted in Figure 6.

6 Architecture of FCA Tools Bundle

FCA Tools Bundle is a platform that intends to implement FCA features for the dyadic and triadic case. It has a log-in system which can offer a personalized experience to the user. Therefore, a formal context can be public, i.e. visible to every other user, or private, i.e. visible only to the user who defined it.

There are multiple technologies used in implementing FCA Tools Bundle as well as some external FCA tools that are integrated for using some of the previously implemented FCA algorithms. For computing dyadic concepts a slightly modified version of InClose2 ([1]) is used. In the triadic case, the Trias algorithm is used for computing the triconcepts [6].

Considering that one of our motivations was to improve visualization methods for concept sets, special attention was given to the concept lattice representation. For this purpose, we used a force-directed approach that tries to position the nodes at somewhat equal distance and with as few intersections among the edges as possible. First, a layer is assigned to each node, which determines its final position on the y-axis of the lattice diagram, i.e. it cannot be changed by dragging the node up or down. For this purpose, we use a concept lattice drawer algorithm proposed by Roland Puntaier[3]. Afterwards the forces rearrange the nodes in the concept lattice as follows. Each node has a repulsion force that pushes other nodes away, while simultaneously each line between the nodes acts as a spring-like force and attracts pairs of nodes towards each other. This approach seems to produce one of the best outputs for the concept lattice, however, it can still be the case that there are overlappings of nodes. For that reason, we implement a custom collision detection algorithm adapted to the x-axis of the lattice representation, since rearranging the nodes on this axis will be sufficient for avoiding overlappings of the nodes. This algorithm follows the ideas described by Mike Bostock[4] and uses the quadtree structure implemented in the D3JS library.

7 Conclusions and Future Work

In this paper, we presented FCA Tools Bundle, a platform that offers, for now, features of visualization and navigation for dyadic and triadic FCA. We have improved concept lattices generation using a detection collision algorithm, in order

[3] https://gist.github.com/rpuntaie/37a380a84f9843b5dd17
[4] https://bl.ocks.org/mbostock/3231298

to avoid manually arranging the concept lattice for concept visibility. Moreover, we have shown how concept lattices can be used for a triadic navigation paradigm based on appropriately defined dyadic projections.

The development of the tool is ongoing and there are multiple functionalities that we plan to implement in the future, some of which are shortly described in this section. We intend to implement exploration procedures for dyadic as well as triadic FCA, based on implication computation. Moreover, temporal FCA is another branch that lacks user-friendly tools, hence implementing lifetracks and temporal views is another work which needs to be done [10]. Another important case that we plan to consider, since so far it was only implemented in the `ToscanaJ Suite`, are many-valued contexts. Furthermore, we will relate FCA to pattern structures and implement corresponding algorithms.

In conclusion, we believe that the presented version of `FCA Tools Bundle` brings an important contribution to the collection of FCA tools, by implementing functionalities of visualization and navigation in triadic concept sets, which, to the best of our knowledge, are not present in any other tool.

References

1. Andrews, S.: In-Close2, a High Performance Formal Concept Miner. In: Andrews, S., Polovina, S., Hill, R., Akhgar, B. (eds.) Proc. of ICCS 2011. LNCS, vol. 6828, pp. 50–62. Springer (2011)
2. Becker, P., Correia, J.H.: The ToscanaJ Suite for Implementing Conceptual Information Systems. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis, Foundations and Applications. LNCS, vol. 3626, pp. 324–348. Springer (2005)
3. Dragoş, S., Haliţă, D., Săcărea, C., Troancă, D.: Applying Triadic FCA in Studying Web Usage Behaviors. In: Proc. of KSEM 2014, pp. 73–80. Springer (2014)
4. Ganter, B., Wille, R.: Formal Concept Analysis - Mathematical Foundations. Springer (1999)
5. Glodeanu, C.V.: Tri-ordinal Factor Analysis. In: Cellier, P., Distel, F., Ganter, B. (eds.) Proc. of ICFCA 2013. LNCS, vol. 7880, pp. 125–140. Springer (2013)
6. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: TRIAS - An Algorithm for Mining Iceberg Tri-Lattices. In: Clifton, C.W., Zhong, N., Liu, J., Wah, B.W., Wu, X. (eds.) Proc. of ICDM 2006. pp. 907–911. IEEE (2006)
7. Lehmann, F., Wille, R.: A Triadic Approach to Formal Concept Analysis. In: Ellis, G., Levinson, R., Rich, W., Sowa, J.F. (eds.) Proc. of ICCS 1995. LNCS, vol. 954, pp. 32–43. Springer (1995)
8. Rudolph, S., Săcărea, C., Troancă, D.: Towards a Navigation Paradigm for Triadic Concepts. In: Baixeries, J., Săcărea, C., Ojeda-Aciego, M. (eds.) Proc. of ICFCA 2015. LNCS, vol. 9113, pp. 232–248. Springer (2015)
9. Sacarea, C.: Investigating Oncological Databases Using Conceptual Landscapes. In: Hernandez, N., Jäschke, R., Croitoru, M. (eds.) Proc. of ICCS 2014. LNCS, vol. 8577, pp. 299–304. Springer (2014)
10. Wolff, K.E.: Temporal Concept Analysis. In: Proc. of ICCS 2001 - International Workshop on Concept Lattices-based KDD. pp. 91–107 (2001)