A Reachability-based Navigation Paradigm for Triadic Concepts

Diana Troancă

Babeş-Bolyai University Cluj Napoca
dianat@cs.ubbcluj.ro

Abstract. Formal Concept Analysis offers a simple formalization for representing knowledge structures extracted from various data. Lately, the triadic case has become increasingly popular, given that data can often be interpreted in a triadic setting for further processing and analysis. However, visualization and navigation in triadic conceptual landscapes is not trivial and, so far, there are no tools implementing navigation in triconcept sets. This paper extends a navigation paradigm based on a reachability relation of triconcepts and on appropriately defined dyadic projections, and it offers a detailed description of different implementation methods. Moreover, we propose a visualization of the reachability clusters that gives an overview of the triconcepts' structure and can assist the local navigation.

1 Introduction

Nowadays, understanding big collections of data can have a great impact on advancing different scientific fields. Formal Concept Analysis (FCA) provides a powerful mathematical tool that addresses knowledge processing and knowledge representation [3]. The main advantage of FCA is the intuitive visualization and navigation methods offered by concept lattices. FCA was extended to the triadic case by Lehman and Wille in 1995 [6]. Since then, different theoretical aspects were studied and extended from the dyadic to the triadic case, and trilattices were proposed as a visualization method. However, this type of representation does not support an intuitive navigation method. Moreover, for slightly larger triadic data sets, the complexity of the representation makes any navigation attempt useless. Therefore, the problem of visualization and navigation in triadic conceptual spaces needs to be further analyzed and new approaches have to be found.

Previously, we have proposed two methods of navigating in triadic data. The first approach is based on narrowing down the space of formal concepts according to constraints added by the user. This membership-constraint-based approach was formally described and the theoretical aspects were studied in detail [7]. Moreover, the navigation paradigm was implemented, tested and evaluated [1, 9]. This approach, however, generates lists of elements, hence the users cannot visualize the underlying structure of the triconcept set.

The second approach has a local character and is based on appropriately defined dyadic projections. For this purpose, we defined the *reachability* relation and studied some of its properties, as well as the properties of other resulting structures, such as reachability *clusters* [8]. In our previous paper, we analyzed the theoretical aspects of the described paradigm and shortly sketched a navigation strategy without going into details about the implementation methods. This paper aims to extend the navigation paradigm and to offer a comprehensive description of strategy behind. Moreover, we show how the structure of the reachability clusters can be used for supporting the local navigation paradigm.

2 Preliminaries

This section introduces the basic notions of triadic formal concept analysis as well as some of the theoretical aspects of the reachability-based navigation. For a deeper understanding of formal concept analysis we refer the reader to the standard literature [3, 6], while a more detailed discussion on the properties of the reachability relation can be found in our previous paper [8].

The fundamental structures of triadic formal concept analysis are those of a triadic formal context and a triconcept.

Definition 1. *A* triadic formal context*, also referred to as a* tricontext*, is a quadruple* $\mathbb{K} = (K_1, K_2, K_3, Y)$ *consisting of three sets* K_1, K_2, K_3 *and a ternary relation* $Y \subseteq K_1 \times K_2 \times K_3$*. The elements of* K_1, K_2, K_3 *are called* (formal) object, attributes *and* conditions*. An element* $(g, m, b) \in Y$ *of the incidence relation is read* object g has attribute m under condition b*.*

Definition 2. *The* triadic concepts*, also called* triconcepts*, of a tricontext* $\mathbb{K} = (K_1, K_2, K_3, Y)$ *are exactly the triples* (A_1, A_2, A_3) *that satisfy* $A_1 \times A_2 \times A_3 \subseteq Y$ *and which are maximal w.r.t. component-wise set inclusion.*

The following definition shows how dyadic projections can be obtained from a triadic context.

Definition 3. *Every triadic context* (K_1, K_2, K_3, Y) *gives rise to the following dyadic contexts:*

$\mathbb{K}^{(1)} := (K_1, K_2 \times K_3, Y^{(1)})$ *with* $gY^{(1)}(m, b) :\Leftrightarrow (g, m, b) \in Y$*,*
$\mathbb{K}^{(2)} := (K_2, K_1 \times K_3, Y^{(2)})$ *with* $mY^{(2)}(g, b) :\Leftrightarrow (g, m, b) \in Y$*, and*
$\mathbb{K}^{(3)} := (K_3, K_1 \times K_2, Y^{(3)})$ *with* $bY^{(3)}(g, m) :\Leftrightarrow (g, m, b) \in Y$*.*

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* $a_k \in A_k$*.*

Intuitively, the contexts $\mathbb{K}^{(i)}$ represent "flattened" versions of the triadic context, obtained by putting the "slices" of (K_1, K_2, K_3, Y) side by side. Moreover, $\mathbb{K}^{(ij)}_{A_k}$ corresponds to the intersection of all those slices that correspond to elements of A_k.

Next we introduce the notion of *reachability* relation and *reachability cluster*.

Definition 4. *For (A_1, A_2, A_3) and (B_1, B_2, B_3) triadic concepts, we say that (B_1, B_2, B_3) is* directly reachable *from (A_1, A_2, A_3)* using perspective (1) *and we write $(A_1, A_2, A_3) \prec_1 (B_1, B_2, B_3)$ if and only if $(B_2, B_3) \in \mathfrak{B}(\mathbb{K}^{(23)}_{A_1})$. Analogously, we can define direct reachability using perspectives* (2) *and* (3).

We say that (B_1, B_2, B_3) is directly reachable *from (A_1, A_2, A_3) if it is directly reachable using at least one of the three perspectives, that is, formally $(A_1, A_2, A_3) \prec (B_1, B_2, B_3) :\Leftrightarrow [(A_1, A_2, A_3) \prec_1 (B_1, B_2, B_3)] \vee [(A_1, A_2, A_3) \prec_2 (B_1, B_2, B_3)] \vee [(A_1, A_2, A_3) \prec_3 (B_1, B_2, B_3)]$.*

Definition 5. *We define the* reachability relation *between two triconcepts as being the transitive closure of the direct reachability relation $\prec$. We denote this relation by $\lhd$.*

For checking whether triconcept (B_1, B_2, B_3) is directly reachable from triconcept (A_1, A_2, A_3), we have proposed the following algorithm.

Algorithm 1: Procedure directlyReachable((A_1, A_2, A_3), (B_1, B_2, B_3))

```
If A_1 = B_1 or A_2 = B_2 or A_3 = B_3 then
        Return true

If A_1 ⊂ B_1 then
    P_e = K^(23)_{A_1}
    If (B_2)'_{P_e} = B_3 and (B_3)'_{P_e} = B_2 then
        Return true

If A_2 ⊂ B_2 then
    P_i = K^(13)_{A_2}
    If (B_1)'_{P_e} = B_3 and (B_3)'_{P_e} = B_1 then
        Return true

If A_3 ⊂ B_3 then
    P_m = K^(12)_{A_3}
    If (B_1)'_{P_e} = B_2 and (B_2)'_{P_e} = B_1 then
        Return true

Return false
```

The derivations used in the description of the algorithm are the simple dyadic derivations and the index was added just to highlight that each dyadic derivation corresponds to a different dyadic context. For example, $(B_2)'_{P_e} = B_3$ uses the dyadic derivation operator of the context P_e.

When studying the properties of the reachability relation, the notion of reachability cluster arises. Intuitively, a reachability cluster is a maximal set of mutually reachable triconcepts. Formally, a reachability cluster is defined as follows.

Definition 6. *The equivalence class of a triconcept* (A_1, A_2, A_3) *with respect to the preorder* $\lhd$ *on* $\mathfrak{T}(\mathbb{K})$ *will be called a* reachability cluster *and denoted by* $[(A_1, A_2, A_3)]$.

Next, we consider the dyadic context of reachable triconcepts $\mathbb{K}_{\lhd}$.

Definition 7.
Let $\mathbb{K} = (K_1, K_2, K_3, Y)$ *be a triadic context. Then we denote with* $\mathbb{K}_{\lhd} = (\mathfrak{T}(\mathbb{K}), \mathfrak{T}(\mathbb{K}), \lhd)$ *the formal context of triconcepts with the reachability relation.*

While analyzing the correlation between reachability clusters and the dyadic concepts $(M, N) \in \mathfrak{B}(\mathbb{K}_{\lhd})$ of the reachability context, we have shown that there is a one-to-one relation between the clusters and the non-empty intersections of the extent and intent from the dyadic concepts $M \cap N$. These results give rise to a display method of all reachability clusters that can support the navigation among triconcepts, as detailed in the next sections.

3 Navigation Strategy

Considering the theoretical aspects highlighted in Section 2 and described in more detail in our previous paper ([8]), we propose a strategy for navigating among triconcepts inside a reachability cluster as well as between clusters. In addition, as guidance during the navigation, we suggest the use of the cluster structure that shows an overview of the reachable triconcepts. Intuitively, a step of the navigation paradigm consists of moving from one triconcept to another directly reachable triconcept. Hence, by following a navigation path of several steps one can explore the triadic conceptual knowledge landscape.

One problem that has not been solved yet in an efficient manner is how to choose a starting point. Currently, for this purpose, we use a preprocessing step that computes all triconcepts, for example using `Trias` [4]. Once we have the triconcept set, one can choose a triconcept as a starting point and navigate by choosing one perspective, i.e. one of the three dimensions. However, we highlight the fact that this preprocessing step is not necessary for the rest of the navigation and, if a starting point is chosen using a different method, this time consuming step can be eliminated. The local navigation paradigm is described by the following steps:

- choose a triconcept $T = (A_1, A_2, A_3)$ and a perspective (i) with $i \in \{1, 2, 3\}$
- compute the derived context $\mathbb{K}^{(jk)}_{A_i}$ of the triadic relation with $j, k \in \{1, 2, 3\} \setminus \{i\}$ s.t. $j < k$
- generate the concept lattice of $\mathbb{K}^{(jk)}_{A_i}$
- attach as labels to the dyadic concepts in the lattice the corresponding triconcepts by adding the third component
- choose one of the triconcepts that are represented by the nodes in the dyadic lattice as a next step

For computing the derived context $\mathbb{K}_{A_i}^{(jk)}$, one must select from the triadic relation the pairs of elements $(a_j, a_k) \in A_j \times A_k$ which are in relation with all elements from A_i, i.e. $(a_i, a_j, a_k) \in Y, \forall a_i \in A_i$, assuming, without loss of generality, that $i < j < k$. Afterwards, the third step consists of generating the concept lattice of the derived context. This can be done using one of the existing FCA tools [2]. In the next step, for a dyadic concept (B_j, B_k) of the derived context $\mathbb{K}_{A_i}^{(jk)}$ we must identify the corresponding triconcept $(B_i, B_j, B_k) \in \mathfrak{T}(\mathbb{K})$. Theoretically, this can be done by using the corresponding derivation operator to compute the third component of a triconcept. However, considering that we have already computed all triconcepts, it is more efficient to select the triconcept having the two components B_j and B_k (which will be unique given the maximality condition) from the triconcept set.

The previously described navigation and visualization paradigm has a local character which is an advantage when considering large contexts. However, one disadvantage of the navigation is that not every triconcept can be reached from every other triconcept, although this seems to be the case in most practical scenarios [8]. Therefore, we believe that it is useful to have an overall view of the navigation clusters' structure in order to understand whereto one can navigate. With that purpose, we obtain the lattice structure of the reachability context $\mathbb{K}_{\lhd} = (\mathfrak{T}(\mathbb{K}), \mathfrak{T}(\mathbb{K}), \lhd)$ as follows:

- compute the direct reachability relation between triconcepts
- compute the transitive closure of the direct reachability relation
- represent the concept lattice of clusters

For implementing the first step, we can use Algorithm 1 that outputs whether a triconcept is directly reachable from another triconcept. For the second step, the transitive closure of the direct reachability relation $\prec$ must be computed in order to obtain the reachability relation $\lhd$. This can be done by using one of the existing algorithms that compute the transitive closure of a relation, such as `Warshall` algorithm ([11]), `Warren` algorithm ([10]), etc.

After obtaining that, the reachability context $\mathbb{K}_{\lhd}$ can be formed and we can compute the clusters and the partial order on the cluster set, using the concept lattice of $\mathbb{K}_{\lhd}$. We have shown that each reachability cluster is uniquely identified by the intersection of extent and intent of exactly one dyadic concept of $\mathbb{K}_{\lhd}$. Hence, we can compute the concept lattice of $\mathbb{K}_{\lhd}$ and label each node with the intersection of extent and intent, i.e. the corresponding cluster, and no label when the intersection is empty. In the obtained lattice one can visualize the partial order between the clusters. An example illustrating the described procedure for obtaining the cluster structure is presented in Section 4.

Alternatively, we can make use of the directed graph having the triconcepts as vertices and the edges given by the direct reachability relation. Here, we can identify the reachability clusters, as well as deduce the transitive closure of the direct reachability relation as described in Proposition 1.

Proposition 1. *Let $\mathbb{K}$ be a tricontext and G the graph with $\mathfrak{T}(\mathbb{K})$ as vertices and the edges given by the direct reachability relation. Then, the reachability clusters* *of $\mathbb{K}$ are identified by the strongly connected components of G. Furthermore, for triconcepts $T_1, T_2 \in \mathfrak{T}(\mathbb{K})$, we have that $T_1 \lhd T_2$ if, in graph G, there is a directed path from T_1 to T_2.*

Furthermore, the same graph G can be used to deduce the partial order relation between the clusters as described in Proposition 2.

Proposition 2. *Let $\mathbb{K}$ be a tricontext and G the graph with $\mathfrak{T}(\mathbb{K})$ as vertices and the edges given by the direct reachability relation. If $T_1 \in C_1$ and $T_2 \in C_2$ are two triconcepts from different clusters s.t. in G there is a path from T_1 to T_2, then we have that $C_1 \leq C_2$. Observe that, considering T_1 and T_2 belong to different clusters, in the case that there is a path from T_1 to T_2, we cannot have a path from T_2 to T_1.*

However, a disadvantage of the graph-based approach is that it does not output the lattice representation of the clusters, hence an FCA tool has to be used if we want to obtain a visualization of the cluster structure.

An important aspect to keep in mind during the navigation is that not every triconcept is reachable from any other triconcept. In fact, when navigating from a triconcept to another belonging to a different cluster, one cannot navigate back by using the standard navigation method described. For this purpose, we propose the use of a navigation history, allowing the user to go back to a certain triconcept in the previous navigation path.

In conclusion, to support exploration through the dataset, we suggest using both the local visualization method for triconcepts and the visualization of the cluster structure as follows. We compute the lattice showing the cluster structure at the beginning of the navigation and make it available to the user throughout the whole exploration process. Then, at each step of the navigation, when visualizing the concept lattice of the possible next steps, i.e. the directly reachable triconcepts, we highlight all the triconcepts belonging to the same cluster as the current triconcept. In that way, the user can easily choose to navigate within the same cluster or to a different one. Furthermore, looking at the cluster structure in parallel, the user can navigate more easily towards a potential goal of the navigation.

4 Example

In practice, experiments that we ran on real datasets showed that tricontexts had one single reachability cluster comprising all triconcepts. This leads us to believe, that, in general, real datasets have a high correlation in the data and therefore, all triconcepts are contained in the same reachability cluster. However, theoretically it is possible that a tricontext comprises several reachability cluster. In this section, we present a small example of an artificial context containing more than two reachability clusters and exemplify how the structure of the reachability clusters can be obtained.

Let us consider the following triadic context $\mathbb{K} = (G, M, B, Y)$, with the object set $G = \{g_1, g_2, g_3\}$, the attribute set $M = \{m_1, m_2, m_3\}$ and the condition set $B = \{b_1, b_2, b_3\}$.

b1	m1	m2	m3
g1	×		
g2			
g3			

b2	m1	m2	m3
g1			
g2			
g3		×	

b3	m1	m2	m3
g1	×	×	
g2			
g3			

The triconcepts of this context are the following:

$T_1 = (\{g_3\}, \{m_2\}, \{b_2\})$
$T_2 = (\{g_1\}, \{m_1\}, \{b_1.b_3\})$
$T_3 = (\{g_1\}, \{m_1.m_2\}, \{b_3\})$
$T_4 = (\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\}, \emptyset)$
$T_5 = (\{g_1, g_2, g_3\}, \emptyset, \{b_1, b_2, b_3\})$
$T_6 = (\emptyset, \{m_1, m_2, m_3\}, \{b_1, b_2, b_3\})$

The triconcepts are partitioned in clusters the following way:

$C_1 = \{(\{g_3\}, \{m_2\}, \{b_2\})\}$
$C_2 = \{(\{g_1\}, \{m_1\}, \{b_1.b_3\}), (\{g_1\}, \{m_1.m_2\}, \{b_3\})\}$
$C_3 = \{(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\}, \emptyset), (\{g_1, g_2, g_3\}, \emptyset, \{b_1, b_2, b_3\}),$
$(\emptyset, \{m_1, m_2, m_3\}, \{b_1, b_2, b_3\})\}$

Then, we obtain the dyadic context of reachability $\mathbb{K}_\triangleleft$ as depicted in Figure 1.

$\mathbb{K}_\triangleleft$	T_1	T_2	T_3	T_4	T_5	T_6
T_1	×	×	×	×	×	×
T_2		×	×	×	×	×
T_3		×	×	×	×	×
T_4				×	×	×
T_5				×	×	×
T_6				×	×	×

Fig. 1: Dyadic context of reachability $\mathbb{K}_\triangleleft$

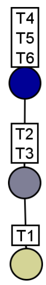

Fig. 2: Concept lattice of $\mathbb{K}_\triangleleft$

Moreover, when computing the concept lattice of $\mathbb{K}_\triangleleft$, we obtain a representation of the reachability clusters as depicted in Figure 2. In this lattice, the bottom node corresponds to cluster C_1, the node in the middle corresponds to cluster C_2 and the upper node corresponds to cluster C_3. The partial order between the clusters can be read from the lattice the following way: if one can navigate from cluster C_1 to cluster C_2 going upwards, then $C_1 \leq C_2$, so we have that $C_1 \leq C_2 \leq C_3$.

During the navigation, taking this cluster structure into consideration, one can deduce, for example, that starting from triconcept T_4 you can never reach any of the triconcepts T_1, T_2 or T_3. Hence, the structure of the clusters can be of use also when choosing a starting point for the navigation.

5 Conclusions and Future Work

In this paper, we have extended a navigation paradigm for triadic datasets that helps the user to get an overview of the data and to understand its underlying structure. To this end we described the steps of the navigation among triconcepts based on local dyadic projections and we have shown how the structure of the reachability clusters can be obtained using different methods. Furthermore, we have highlighted the fact that the local navigation paradigm can benefit from the lattice representation of the clusters' structure by offering the user an overview of the underlying data structure. The described navigation paradigm was implemented in a tool suite called `FCA Tools Bundle` that is described in more detail in an additional paper [5].

In our future work, we plan to combine the reachability-based navigation with the membership-constraint-based navigation into a new improved paradigm in order to solve the problem of choosing a starting point. The reachability-based navigation can offer the visualization support, while the constraints added by the user can be taken into consideration by highlighting the concepts in the local visualization that satisfy the constraints.

References

1. Dragoş, S., Haliţă, D., Troancă, D.: Investigating trend-setters in e-learning systems using polyadic formal concept analysis and answer set programming. In: Proc. of AI4KM 2016, co-located with IJCAI. pp. 42–48 (2016)
2. Ganter, B., Stumme, G., Wille, R. (eds.): Formal Concept Analysis, Foundations and Applications, LNCS, vol. 3626. Springer (2005)
3. Ganter, B., Wille, R.: Formal Concept Analysis - Mathematical Foundations. Springer (1999)
4. 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)
5. Kis, L.L., Săcărea, C., Troancă, D.: FCA Tools Bundle - a Tool that Enables Dyadic and Triadic Conceptual Navigation. In: Proc. of FCA4AI 2015, co-located with IJCAI 2015 (2015)
6. 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. Lecture Notes in Computer Science, vol. 954, pp. 32–43. Springer (1995)
7. Rudolph, S., Săcărea, C., Troancă, D.: Membership constraints in formal concept analysis. In: Yang, Q., Wooldridge, M. (eds.) Proc. of IJCAI 2015. pp. 3186–3192. AAAI Press (2015)
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. Rudolph, S., Săcărea, C., Troancă, D.: Conceptual navigation for polyadic formal concept analysis. In: Proc. of AI4KM 2016, co-located with IJCAI. pp. 35–41 (2016)
10. Warren, H.S.: A modification of Warshall's algorithm for the transitive closure of binary relations. Commun. ACM 18(4), 218–220 (1975)
11. Warshall, S.: A Theorem on Boolean Matrices. J. ACM 9(1), 11–12 (1962)