Polyadic Relational Concept Analysis
Alexandre Bazin, Jessie Galasso-Carbonnel, Giacomo Kahn

To cite this version:
Alexandre Bazin, Jessie Galasso-Carbonnel, Giacomo Kahn. Polyadic Relational Concept Analysis. 2022. hal-03908993
Polyadic Relational Concept Analysis

Bazin, Alexandre¹, Galasso-Carbonnel, Jessie², and Kahn, Giacomo³

¹ LIRMM, CNRS, Université de Montpellier, FRANCE
alexandre.bazin@u-montpellier.fr
² DIRO, Université de Montréal, CANADA
jessie.galasso-carbonnel@umontreal.ca
³ Univ Lyon, Univ Lyon 2, Université Claude Bernard Lyon 1, Université Jean Monnet Saint-Etienne, INSA Lyon, DISP-UR4570, 69500 Bron, FRANCE
giacomo.kahn@univ-lyon2.fr

Abstract. Formal concept analysis is a mathematical framework based on lattice theory that aims at representing the information contained in binary object-attribute datasets (called formal contexts) in the form of a lattice of so-called formal concepts. Since its introduction, it has been extended to more complex types of data. In this paper, we are interested in two of those extensions: relational concept analysis and polyadic concept analysis that allow to process, respectively, relational data and $n$-ary relations. We present a framework for polyadic relational concept analysis that extends relational concept analysis to relational datasets that are made of $n$-ary relations. We show its basic properties and that it is a valid extension of relational concept analysis.

1 Introduction

Formal concept analysis (FCA [6]) is a mathematical framework based on lattice theory that aims at representing the information contained in binary object-attribute datasets (called formal contexts) in the form of a lattice of so-called formal concepts. Formal contexts being representations of the most basic type of data, direct applications of formal concept analysis on real, more complex data are limited but various extensions exist to exploit this powerful formalism. In this paper, we are interested in merging two of them : relational concept analysis and polyadic concept analysis.

Relational concept analysis (RCA [7]) is an extension of the formal concept analysis framework aimed at relational data that take the form of multiple formal contexts with relations between their objects. The idea is to represent through attributes the information contained in the relations, hereby enriching the descriptions of objects in the formal contexts. RCA has been used with great success in fields as diverse as software engineering, for variability modeling or feature location [2,8], or the analysis of ecological data about plants [12] or water quality [3].

Polyadic concept analysis (PCA [14]) is another extension of the formal analysis framework aimed at multidimensional data, i.e. $n$-ary relations between
elements of $n$ dimensions. For instance, objects described by attributes under conditions in the tridimensional case. PCA and its tridimensional version, triadic concept analysis [11], have been used for data exploration with multiple experts [12], or tri-clustering in machine learning [13]. RDF datasets are also important examples of multidimensional data and, with them, knowledge graphs. Such data rarely come alone and are generally part of a larger dataset containing relations. It would thus seem natural to combine RCA and PCA. However, as they are, these two frameworks are not compatible. In a triadic object-attribute-condition setting, if we kept representing the information contained in relations by adding elements to the set of attributes, under which condition would a newly created attribute describe an object? To solve this problem, we propose a new framework, Polyadic Relational Concept Analysis (PRCA), that works on sets of $n$-dimensional contexts with relations between their objects. PRCA is different from Graph-FCA [5], that also allows one to manipulate a set of $n$-ary relations, since a graph-FCA concept is a graph structure that is not a formal concept of the input formal context, while PRCA outputs concept $n$-lattices.

This paper presents the PRCA framework and some of its properties, including the fact that applying PRCA to a bidimensional formal context yields the same results as RCA, proving PRCA is a proper generalisation of RCA. Section 2 recalls the necessary background notions of formal concept analysis, relational concept analysis and polyadic concept analysis. Section 3 contains the definition of PRCA and an example of its application to a small toy dataset. Section 4 presents a proof of the fact that PRCA produces concepts that are in a one-to-one correspondence with concepts of an $n$-context of the same dimensionality as the initial context on which it is applied, bounding the size of the output and implying that PRCA is a proper generalisation of RCA.

2 Background Notions

2.1 Formal Concept Analysis

Formal concept analysis (FCA) is a mathematical framework that aims at using lattices to organise the units of knowledge that can be found in binary data.

**Definition 1 (Formal context).** A formal context is a triple $(\mathcal{O}, \mathcal{A}, \mathcal{I})$ in which $\mathcal{O}$ is a set of objects, $\mathcal{A}$ is a set of attributes and $\mathcal{I} \subseteq \mathcal{O} \times \mathcal{A}$ is a binary relation between objects and attributes.

Formal contexts (or just contexts) formalise binary datasets and can be represented by a crosstable, as illustrated in Figure 1 that has as objects plants, and attributes pests that are repelled by them.

**Definition 2 (Derivation operators).** Let $(\mathcal{O}, \mathcal{A}, \mathcal{I})$ be a formal context. The operators

\[ .^\prime : 2^\mathcal{O} \rightarrow 2^\mathcal{A} \]
Fig. 1: A formal context describing four plants as objects (Citrus, Alpinia, Pelargonium and Laphangium) through four attributes (s, m, t, ℓ) representing the pest that are repelled by each of the plants. This particular context, while not physically accurate, was inspired by project Knomana (https://agritrop.cirad.fr/591139/) [10].

\[ A' = \{ o \in O \mid \forall a \in A, (o, a) \in I \} \]

and

\[ O' = \{ a \in A \mid \forall o \in O, (o, a) \in I \} \]

are called the derivation operators of the formal context.

The two derivation operators of a formal context form a Galois connection and, as such, their compositions \(\cdot''\) are closure operators, i.e. \(X \subseteq X''\), \((X'')'' = X''\) and if \(X \subseteq Y\) then \(X'' \subseteq Y''\). Sets \(X\) such that \(X = X''\) are said to be closed.

**Definition 3 (Formal concept).** Let \(C = (O, A, I)\) be a formal context. A formal concept of \(C\) is a pair \((E, I)\) such that \(E \times I \subseteq I\) and both \(E\) and \(I\) are maximal for this property. The set \(E\) is called the extent and the set \(I\) the intent of the concept.

Formal concepts are by definition pairs of closed sets. They represent interesting units of knowledge, or “classes”, found in the data. They correspond to maximal rectangles of crosses in the crosstable representation of the formal context (up to permutation of rows and columns).

**Definition 4 (Formal concept, alternative definition).** Let \(C = (O, A, I)\) be a formal context. A formal concept of \(C\) is a pair \((E, I)\) such that \(E \times I \subseteq I\) and both \(E\) and \(I\) are maximal for this property.

The set of formal concepts of a formal context \(C\) is denoted \(T(C)\). Formal concepts can be ordered by the inclusion relations on their extents or intents: \((E_1, I_1) \leq (E_2, I_2) \iff E_1 \subseteq E_2\) or \(I_2 \subseteq I_1\). The pair \((T(C), \leq_O)\) is a complete lattice called the concept lattice of the formal context \(C\).
2.2 Relational Concept Analysis

Relational concept analysis (RCA [7]) is a generalisation of FCA to relational binary datasets.

Definition 5 (Relational context family (RCF)). A relational context family (RCF) is a pair (C, R) such that:

- C = \{C_i = (O_i, A_i, I_i)\} is a set of formal contexts (object-attribute relations)
- R = \{r_k \mid r_k \subseteq O_i \times O_j,\} is a set of relations between objects of source contexts C_i and those of target contexts C_j.

We use src(r) and tar(r) to denote, respectively, the source and target context of a relation r.

Relational context families formalise relational binary datasets. Both the contexts in C and the relations in R can be represented as binary crosstables but, for the sake of clarity, we shall only depict contexts as crosstable and relations as sets of pairs, as illustrated in Figure 2. Figure 3 depicts the concept lattices obtained from the Figure 2 RCF.

<table>
<thead>
<tr>
<th>Plants</th>
<th>Spodoptera (s)</th>
<th>Milax gagates (m)</th>
<th>Tetranychidae (t)</th>
<th>Locust (l)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Citrus (C)</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alpinia (A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pelargonium (P)</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laphangium (L)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Areas</th>
<th>Forests (F)</th>
<th>Desert (D)</th>
<th>Grassland (G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Eu)</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Africa (Af)</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Asia (As)</td>
<td>×</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>America (Am)</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

GrowsIn = \{(C, Eu), (C, Af), (A, Eu), (A, Am), (P, As), (L, Eu)\}

Fig. 2: A RCF containing two contexts (our plant context and a context about areas and ecosystems), and a relation between the plants and the areas, stating which plant grows in which area.

RCA aims at finding units of knowledge in the data, in the same fashion as FCA, by taking into consideration not only objects and their attributes but also their relations with other objects. In order to do so, information on the relations is converted into attributes, enriching the descriptions of objects.

We use r(o) to denote the image of an object o ∈ O_i by a relation r ⊆ O_i × O_j.
Definition 6 (Relational scaling of a context). Let \((C, R)\) be an RCF, \(C_i = (\mathcal{O}_i, \mathcal{A}_i, \mathcal{I}_i)\) a context in \(C\), \(R\) the set of relations in \(R\) for which \(C_i\) is the source and \(Q\) a set of scaling operators. In this paper, we will only consider \(Q \subseteq \{\exists, \forall\}\).

The formal context \(C^+_i = (\mathcal{O}_i, \mathcal{A}^+_i, \mathcal{I}^+_i)\) is defined such that

\[
\mathcal{A}^+_i = \mathcal{A}_i \cup \{qr.X \mid q \in Q, r \in R \text{ and } X \in T(\text{tar}(r))\}
\]

and

\[
\mathcal{I}^+_i = \mathcal{I}_i \cup \{(o, qr.X) \mid X = (E, I) \text{ and } E \text{ respects } q \text{ w.r.t. } r(o)\}
\]

We write \(C^+ = \{C^+_i \mid C_i \in C\}\).

The computation of \((C^+, R)\) requires the formal concepts of all the contexts in \(C\) that are targets of relations. For this reason, the RCA process uses two steps iteratively:

- Compute all the concepts in all the contexts in \(C\) that are targets of relations
- Construct \((C^+, R)\) by using the concepts to generate new attributes.

We use \(C^0 = C\) and \(C^m = C^{(m-1)+}\) to denote the state of the contexts at step 0 and \(m\), respectively. The process stops either when a user-defined number of iterations has been reached or when the concept lattices of every contexts do not change between \(C^m\) and \(C^{m+1}\), which happens at some point \([7]\).

Figs. 4 and 5 depict the result of the first (and final) step of the RCA process on the Figure 2 RCF. Four new relational attributes, corresponding to the four concepts in the lattice on the right-hand side of Fig 3, have been added to the Plants context. In the lattice, let us consider the concept \(\{A, P\}, \{t, \exists\text{Gr}C_2, \exists\text{Gr}C_3\}\). It expresses the fact that Alpinia (A) and Pelargonium (P) not only repel tetranychidae (t), but also grow in areas that correspond to \(C_2\) and \(C_3\), that is areas that have either forests, or grasslands and forests as ecosystems.
Fig. 4: The context $C^+$, where $C$ is the plants context depicted in Figure 2 and $Q = \{\exists\}$. The new attributes $\exists GrowsIn.C_i$ (abbreviated $\exists GrC_i$) come from the four concepts $C_0 = (\{Am\}, \{D, G, F\})$, $C_1 = (\{Am, Af, Eu\}, \{D, F\})$, $C_2 = (\{As, Am\}, \{G, F\})$ and $C_3 = (\{Am, Af, As, Eu\}, \{F\})$.

![Concept lattice of Plants](image)

Fig. 5: Concept lattice of Plants$^1$. Applying the same process for another step produces a concept lattice that is isomorphic to this one. Thus, the RCA process ends.

### 2.3 Polyadic Concept Analysis

Polyadic concept analysis (PCA) is the generalisation of FCA to datasets that take the form of an $n$-ary relation.

**Definition 7 ($n$-contexts).** An $n$-context is an $(n + 1)$-tuple $(D_1, \ldots, D_n, I)$ in which all the $D_i$ are sets called dimensions and $I \subseteq \prod_{i \in \{1, \ldots, n\}} D_i$ is an $n$-ary relation between dimensions.

Similarly to the bidimensional case, $n$-contexts formalise $n$-dimensional binary datasets. They can be represented as $n$-dimensional crosstables as illustrated in Figure 6.

Formal contexts are 2-contexts.
Plants | s | m | t | ℓ | s | m | t | ℓ
---|---|---|---|---|---|---|---|---
Citrus (C) | × | × | | | | | × | ×
Alpinia (A) | × | × | | | | | × | ×
Pelagonium (P) | × | | | | | | × | ×
Laphangium (L) | × | | | | | | × | ×

Root | | | | | | | | | Leaf

Fig. 6: A 3-context representing plants (Citrus, Alpinia, Pelagonium and Laphangium) that repel pests (s, m, t and ℓ), depending on their parts (leaf or root).

Definition 8 (n-concepts). Let \( C = (D_1, \ldots, D_n, I) \) be an n-context. An n-concept of \( C \) is an n-tuple \((X_1, \ldots, X_n)\) such that \( \prod_{i \in \{1, \ldots, n\}} X_i \subseteq I \) and there is no \( k \in \{1, \ldots, n\} \) and \( x_k \in D_k \setminus X_k \) for which \((X_1, \ldots, X_k \cup \{x_k\}, \ldots, X_n)\) respects this property.

As in the bidimensional case, n-concepts are units of knowledge in the data. They correspond to maximal n-dimensional boxes of crosses in the n-context. We use \( T(C) \) to denote the set of n-concepts of an n-context \( C \). For a concept \( C = (X_1, \ldots, X_n) \), we use \( D_i(C) = X_i \) to denote the \( i \)th component of the tuple.

The 3-concepts from Figure 6 are:

\[
\begin{align*}
&\{(L), \{\text{Root}, \text{Leaf}\}, \{t, \ell\}\} & &\{(A, L), \{\text{Root}, \text{Leaf}\}, \{t\}\} & &\{(A, P, L), \{\text{Leaf}\}, \{t\}\} \\
&\{(C, A, P, L), \{\text{Root}, \text{Leaf}\}, \emptyset\} & &\{(A, P), \{\text{Leaf}\}, \{m, t\}\} & &\{(C, P), \{\text{Root}, \text{Leaf}\}, \{m\}\} \\
&\{(C, A, P), \{\text{Leaf}\}, \{m\}\} & &\{(C), \{\text{Root}, \text{Leaf}\}, \{s, m\}\} & &\{(C, P), \{\text{Root}\}, \{s, m\}\} \cup \{(C, A, P, L), \emptyset, \{s, m, t, \ell\}\}. \\
&\{(A, L), \{\text{Root}\}, \{t, \ell\}\} & &\{(C), \{\text{Root}, \text{Leaf}\}, \{s, m, t, \ell\}\} & &\{(C, A, P, L), \emptyset, \{s, m, t, \ell\}\}. \\
\end{align*}
\]

The main theorem of PCA [14] states that, for any n-context \( C \), \( (T(C), \subseteq D_1, \ldots, \subseteq D_n) \) is a complete n-lattice called the concept n-lattice of \( C \).

3 Polyadic Relational Concept Analysis

Relational concept analysis does not work with multidimensional contexts because of the way the information on relations is integrated into the contexts as attributes. Let us consider a 3-context describing objects through the attributes they possess under some conditions. The objects of this 3-contexts are in relation with objects of another context. Using the RCA approach, one would create attributes from the concepts of the other context. However, where should these attributes be placed? If added to the set of attributes, under which conditions are the objects described by them?

To solve this problem, we propose a new formalism, called Polyadic Relational Concept Analysis (PRCA), in which the new “attributes” constructed from a (quantifier,relation) pair at a given step form a new dimension in the n-context. More formally:

Definition 9 (Polyadic relational context family). Let a polyadic relational context family (PRCF) be a pair \((C, R)\) in which
- $C = \{C_i = (\mathcal{D}_i^1, \ldots, \mathcal{D}_i^n, T)\}$ is a set of $n$-contexts with various numbers of dimensions
- $R = \{r_k \mid r_k \subseteq \mathcal{D}_i^1 \times \mathcal{D}_i^1\}$ is a set of relations between the elements of the first dimensions (hereafter called objects without loss of generality) of source $n$-contexts $C_i$ and those of target $n$-contexts $C_j$.

Just as in RCA, we use $src(r)$ and $tar(r)$ to denote, respectively, the source and target $n$-context of a relation $r$.

We use the polyadic relational context family that consists of the 3-context $Plants$ from Figure 2 together with the context $Areas$ and the relation $GrowsIn$ from Figure 3 as a running example.

Similarly to RCA, we define the relational scaling of an $n$-context in order to take into account information on the relations involving its objects. We use the . notation to denote the fusion of two tuples. For example, $(x_1, \ldots, x_i, \ldots, x_n) = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n).x_i$.

**Definition 10 (Relation scaling of an $n$-context).** Let $(C, R)$ be a PRCF, $C_i = (\mathcal{D}_i^1, \ldots, \mathcal{D}_i^n, T)$ an $n$-context in $C$, $R$ the set of relations in $R$ for which $C_i$ is the source and $Q$ a set of scaling operators. We still only consider $Q \subseteq \{\exists, \forall\}$. The formal context $C_i^+ = \{\mathcal{D}_1^i, \mathcal{D}_2^i, \ldots, \mathcal{D}_{n+k}^i, T^i\}$ is defined such that

$$\mathcal{D}_k^i = \mathcal{D}_k^i \cup \{\emptyset\}, \forall j \in \{2, \ldots, n\}$$

$$\mathcal{D}_{n+k}^i = \{\emptyset\} \cup \{q_k r_k \mid (q_k, r_k) \in Q \times R \text{ and } X \in T(tar(r_k))\}$$

with $Q \times R = \{(q_1, r_1), \ldots, (q_m, r_m)\}$,

$$Z = \{(d_1, \ldots, d_n, q_1 r_1 X_1, \ldots, q_m r_m X_m) \mid (d_1, \ldots, d_n) \in I_i$$

and $\forall k \in \{1, \ldots, m\}$,

$$\begin{cases} r(d_1) \subseteq D_1(X_k) \text{ if } q_k = \forall \\ r(d_1) \cap D_1(X_k) \neq \emptyset \text{ if } q_k = \exists \end{cases}$$

and

$$I_i^+ = Z \cup \{x.(\bigcup_{i=1}^{m} I_i) \mid \{i_1, \ldots, i_k\} \subseteq \{2, \ldots, n + m\}$$

and $\exists y \in \prod_{i=1}^{k} D_{1i}, x.y \in Z$.)

We write $C^+ = \{C_i^+ \mid C_i \in C\}$.

Less formally, the relational scaling of the $n$-context consists in creating a new dimension for each quantifier and relation pair that contains elements constructed from concepts of the target context. Then, the whole descriptions of objects are projected on the elements of the new dimensions iff the image of the object by the relation satisfies the constraint of the quantifier w.r.t. the concept used to create the element. An additional element $\emptyset$ is added to each dimension except the first and all the “crosses” of $I$ are projected on this new element.
Hence, $\circ_i$ can be interpreted as a disjunction over all the elements of $D_i$ and is used to link the dimensions in case a set of objects is not described by any element $D_i$.

Figure 7 shows the 4-context resulting from a step of this process for the polyadic RCF described above. The new dimension corresponding to the relational attributes is depicted on the bottom of the context. The layer of the neutral element $\circ_4$ contains the starting 3-context. Then, the layers that correspond to relational attributes are filled as described above, using the $\exists$ quantifier. For instance, as $(A, Am)$ is an element of the relation $GrowsIn$, and $Am$ is in the extent of $Areas$’ concept $C_0$, the description of object $A$ is copied on the new layer corresponding to $\exists Gr C_0$. The lines that correspond to Root and Leaf are copied as they are in the starting context, and the part that corresponds to $\circ_3$ gets their union. Since only the object $A$ is related via $GrowsIn$ to an element from $C_0$’s extent, only object $A$’s description is present in this layer. The next layer corresponds to a relation with $C_1$’s extent. Objects $C$, $A$, and $L$ from $Plants$ are related, in $GrowsIn$ and with the quantifier $\exists$, to this concept. Thus, their descriptions are copied in the layers corresponding to Root and Leaf, and the union of their descriptions in $\circ_3$.

4 Properties of PRCA’s output

The PRCA process adds a new dimension to the context for every (quantifier, relation) pair, at each step. This can quickly result in an overwhelming number of dimensions. Even though we do not have tight upper bounds for the number of $n$-concepts in an $n$-context at the time of writing, it is generally admitted that this number quickly increases as dimensions are added which could render PRCA unusable on real data. In this section, we show that PRCA is indeed usable as most of the concepts it produces are in a one-to-one correspondence with the concepts of a context of the same dimensionality as the initial data. To do this, we prove the following Theorem 1.

**Theorem 1.** Let $C = (D_1, \ldots, D_n, \mathcal{I})$ be an $n$-context.

Let $C^\square = (D_1, \ldots, D_n, D_{n+1}, \mathcal{I}^\square)$ be an $(n+1)$-context such that

$$(x_1, \ldots, x_n, x_{n+1}) \in \mathcal{I}^\square \Rightarrow (x_1, y_2, \ldots, y_n, x_{n+1}) \in \mathcal{I}^\square, \forall (x_1, y_2, \ldots, y_n) \in \mathcal{I}.$$

Let $C^\triangle = (D_1, D_2 \times D_{n+1}, \ldots, D_n \times D_{n+1}, \mathcal{I}^\triangle)$ be an $n$-context such that

$$(x_1, (x_2, y_2), \ldots, (x_n, y_n)) \in \mathcal{I}^\triangle \Leftrightarrow (x_1, x_2, \ldots, x_n, y_i) \in \mathcal{I}^\square, \forall i \in \{2, \ldots, n\}.$$

We have that, if $C_{n+1} \neq \emptyset$,

$$(C_1, \ldots, C_{n+1}) \in \mathcal{T}(C^\square) \Leftrightarrow (C_1, C_2 \times C_{n+1}, \ldots, C_n \times C_{n+1}) \in \mathcal{T}(C^\triangle).$$
Fig. 7: Context $C^+$ extended with the relational attributes built from the concepts of Areas.
**Proof.** First, we show $\Rightarrow$. Let $(C_1, \ldots, C_{n+1})$ be a concept in $T(C^\Box)$. We have that $C_1 \times \ldots \times C_n \subseteq T^\Box$ so we know that $(x_1, (x_2, y_2), \ldots, (x_n, y_n)) \in T^\Delta$ for all $y_i$ such that $(x_1, x_2, \ldots, x_n, y_i) \in C_1 \times \ldots \times C_n$. If $(C_1, C_2 \times C_{n+1}, \ldots, C_n \times C_{n+1})$ is not a concept in $T(C^\Delta)$, then there is a $c_1 \in C_1$ such that $\{c_1\} \times (C_2 \times C_{n+1}) \times \ldots \times (C_n \times C_{n+1}) \subseteq T^\Delta$ or there is a $c_i \in (C_i \times C_{n+1})$ such that $C_1 \times \cdot \cdot \cdot \times (C_i \times C_{n+1}) \times \cdot \cdot \cdot \times (C_n \times C_{n+1}) \subseteq T^\Delta$. In both cases, it implies that $(C_1, \ldots, C_{n+1})$ is not a concept in $T(C^\Box)$ as it is not a maximal box. Hence, $(C_1, C_2 \times C_{n+1}, \ldots, C_n \times C_{n+1}) \in T(C^\Box)$.

Now, we show $\Leftarrow$. Let $(C_1, (C_2 \times C_{n+1}), \ldots, (C_n \times C_{n+1}))$ be a concept in $T(C^\Delta)$. We have that $C_1 \times (C_2 \times C_{n+1}) \times \ldots \times (C_n \times C_{n+1}) \subseteq T^\Delta$ so $C_1 \times C_2 \times \cdot \cdot \cdot \times C_{n+1} \subseteq T^\Box$. If $(C_1, \ldots, C_{n+1})$ is not a concept in $T(C^\Box)$, then there is a $c_i \in C_i$ such that $C_1 \times \cdot \cdot \cdot \times \{c_i\} \times \cdot \cdot \cdot \times C_{n+1} \subseteq T^\Box$ for all $i \in \{1, \ldots, n+1\}$. Then, $(C_1, (C_2 \times C_{n+1}), \ldots, (C_n \times C_{n+1}))$ is not a concept in $T(C^\Delta)$ as it is not a maximal box. Hence, $(C_1, \ldots, C_{n+1}) \in T(C^\Box)$.

The relational scaling of an $n$-context with a single (quantifier, relation) pair, as described in Definition 10, is an instance of the $(n+1)$-context $C^\Box$ of the theorem. Hence, this theorem tells us that the number of concepts produced by PRCA is bounded by the maximal number of concepts in a context of the same dimensionality as the initial context (plus some number of $n$-contexts with empty components in the dimensions created by the process). Most importantly, it means that PRCA produces what amounts to the elements of a concept lattice when applied to a bidimensional context.

### 4.1 PRCA as a generalisation of RCA

In this section, we show that the concepts produced by PRCA when applied to a dyadic context are the same as the ones produced by RCA (Proposition 1). This validates PRCA as a generalisation of RCA.

**Proposition 1.** Let $C = (D_1, D_2, I)$ be a dyadic context in a PRCF, $T(C_{RCA}^+)\Box$ the set of concepts of the context after an iteration of the RCA process and $T(C_{PRCA}^+)\Box$ the set of concepts of the context after an iteration of the PRCA process. We have that

$$(X, Y, Z_1, \ldots, Z_n) \in T(C_{PRCA}^+) \Leftrightarrow (X, (Y \{\triangledown_2\}) \cup \bigcup_{i=1}^{n}(Z_i \{\triangledown_{i+2}\})) \in T(C_{RCA}^+)
$$

if $Y, Z_1, \ldots, Z_n \neq \emptyset$.

**Proof.** First, we show $\Rightarrow$. If $(X, Y, Z_1, \ldots, Z_n)$ is a concept in $T(C_{PRCA}^+)\Box$, then for all $x \in X$ and $(y, z_1, \ldots, z_n) \in Y \times Z_1 \times \ldots \times Z_n$, we have that $(x, y, z_1, \ldots, z_n)$ is in the incidence relation $I_{PRCA}^+$ of $I_{PRCA}$. From the construction of $I_{PRCA}^+$, we have that $(x, y) \in I$, and, for all $z_1 \neq \triangledown_{i+2}$, either

$$\begin{align*}
    r(x) \cap A \neq \emptyset \text{ if } z_i = \exists r(A, B) \\
    r(x) \subseteq A \text{ if } z_i = \forall r(A, B).
\end{align*}$$

(r(x) \cap A \neq \emptyset \text{ if } z_i = \exists r(A, B) \text{ or } r(x) \subseteq A \text{ if } z_i = \forall r(A, B).$$

}
Thus, \((x, z_i)\) is in \(I_{PRCA}^+\). As a consequence, if \((X, \{Y \setminus \{\varnothing_2\}\} \cup \bigcup_{i=1}^n Z_i \setminus \{\varnothing_{i+2}\})\) is not a concept in \(T(C_{PRCA}^+)\), then either there is a \(v\) in \(D_1 \setminus X\) such that \((v, y)\) is in \(I_{RC}^+\) for all \(y\) in \((Y \setminus \{\varnothing_2\}) \cup \bigcup_{i=1}^n Z_i \setminus \{\varnothing_{i+2}\}\), or there is a \(w\) in \((D_2 \setminus \{\varnothing_2\}) \cup \bigcup_{i=1}^n (D_{i+1} \setminus \{\varnothing_{i+2}\})\) \(\setminus (Y \cup \bigcup_{i=1}^n Z_i)\) such that \((x, w)\) is in \(I_{RC}^+\) for all \(x \in X\). By construction of \(I_{PRCA}^+\), we have that both those cases imply that \((X, Y, Z_1, \ldots, Z_n)\) is not a concept in \(T(C_{PRCA}^+)\), which contradicts our hypothesis. Hence, \((X, Y, Z_1, \ldots, Z_n)\) is a concept in \(T(C_{RC}^+)\).

Now, we show \(\Rightarrow\). If \((X, Y, Z_1, \ldots, Z_n)\) is a concept in \(T(C_{RC}^+)\), then for all \(x \in X\) and \(y \in Y\), \((x, y)\) is in \(I\). Additionally, for all \(z\) in \(\bigcup_{i=1}^n Z_i \setminus \{\varnothing_{i+2}\}\), either

\[
\begin{cases}
    r(x) \cap A \neq \emptyset \text{ if } z = \exists r(A, B) \\
    r(x) \subseteq A \text{ if } z = \forall r(A, B)
\end{cases}
\]

Hence, by construction of \(I_{PRCA}^+\), \(X \times Y \times Z_1 \times \cdots \times Z_n \subseteq I_{PRCA}^+\). Then, if \((X, Y, Z_1, \ldots, Z_n)\) is not a concept in \(T(C_{PRCA}^+)\), it means either that there is a \(v\) in \(D_1 \setminus X\) such that \(\{v\} \times Y \times Z_1 \times \cdots \times Z_n \subseteq I_{PRCA}^+\), that there is a \(w\) in \(D_2 \setminus Y\) such that \(X \times \{w\} \times Z_1 \times \cdots \times Z_n \subseteq I_{PRCA}^+\), or that there is a \(z_i\) in \(D_{i+2} \setminus Z_i\) such that \(X \times Y \times Z_1 \times \cdots \times \{z_i\} \times \cdots \times Z_n \subseteq I_{PRCA}^+\). All three cases imply that \((X, Y, Z_1, \ldots, Z_n)\) is not a concept in \(T(C_{RC}^+)\), which contradicts our hypothesis. Thus, \((X, Y, Z_1, \ldots, Z_n)\) is a concept in \(T(C_{PRCA}^+)\).

\[\square\]

Proposition 1 shows that the knowledge generated by applying PRCA on a dyadic context or RCA is fundamentally the same, albeit through a reframing of the relational attributes as new dimensions.

We now provide an example of the application of PRCA, as described in Section 3, on the dyadic context depicted in Figure 2. The first step gives the same concept lattices as in Figure 3. The context Plants is then extended with a new dimension that contains the following elements

\[\{\varnothing_3, \exists \text{GrowsIn}C_0, \exists \text{GrowsIn}C_1, \exists \text{GrowsIn}C_2, \exists \text{GrowsIn}C_3\},\]

while its second dimension is extended with \(\varnothing_2\).

The incidence relation is built from the Plants context, by reporting the description of the objects on the corresponding layers: first in the layer corresponding to \(\varnothing_3\) and then in the layers that correspond to concepts related to each object via the quantifier.

Let us list the concepts from the 3-context in Figure 8.
many new elements are created so dimensions can be big. As the number of concepts grows quickly with the size of the context, the size of the output can become overwhelming. In addition to that, efficient graphical representations of n-lattices are still open questions so human interpretation of the output of PRCA can be difficult. It would be interesting, as a future work, to study algorithms for eventual ternary or polyadic contexts, and can take into account the generalisation of such algorithms is blurrier than in traditional lattices so the generalisation of such algorithms is not straightforward.

Let us consider the concept \( \{A, P\}\{C, t\}, \{C, P, L\} \) from Figure 8. It provides the same information as the equivalent RCA concept \( \{A, P\}, \{t, C, P, L\} \) (see Figure 5). More generally, the RCA concepts can be obtained from the PRCA concepts by removing the elements \( C \) and merging the last two components. Thus, one can check that the mapping described in Proposition 4 holds.

5 Discussion and Conclusion

Our proposed framework extends RCA to polyadic contexts. As such, it inherits the drawbacks of both relational and polyadic concept analysis. At each step, many new elements are created so dimensions can be big. As the number of n-concepts grows quickly with the size of the n-context, the size of the output can become overwhelming. In addition to that, efficient graphical representations of n-lattices are still open questions so human interpretation of the output of PRCA can be difficult. It would be interesting, as a future work, to study algorithms for the interactive exploration of the n-lattice in line with those that exist for the dyadic case [11]. However, the notion of neighbours of an n-concept in an n-lattice is blurrier than in traditional lattices so the generalisation of such algorithms is not straightforward.

The use cases for our methods are as wide as are RCA’s. It can be used in place of RCA in dyadic relational datasets, and can take into account the eventual ternary or n-ary context in those datasets. These applications can range from plants datasets [13] to families of RDF datasets.
Lastly, allowing $n$-ary relations in contexts is a step forward but, in our framework, relations between objects of different contexts are still binary. As these relations can also be $n$-ary in real datasets (or just in real life), it would be interesting to improve the framework to take such relations into consideration.

Acknowledgements

Part of this work was supported by the French National Research Agency (ANR) through the project SmartFCA (ANR-21-CE23-0023).

References

