

Fixed-point semantics for barebone relational concept analysis^{*}

Jérôme Euzenat

Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, F-38000 Grenoble France
Jerome.Euzenat@inria.fr

Abstract. Relational concept analysis (RCA) extends formal concept analysis (FCA) by taking into account binary relations between formal contexts. It has been designed for inducing description logic TBoxes from ABoxes, but can be used more generally. It is especially useful when there exist circular dependencies between objects. In this case, it extracts a unique stable concept lattice family grounded on the initial formal contexts. However, other stable families may exist whose structure depends on the same relational context. These may be useful in applications that need to extract a richer structure than the minimal grounded one. This issue is first illustrated in a reduced version of RCA, which only retains the relational structure. We then redefine the semantics of RCA on this reduced version in terms of concept lattice families closed by a fixed-point operation induced by this relational structure. We show that these families admit a least and greatest fixed point and that the well-grounded RCA semantics is characterised by the least fixed point. We then study the structure of other fixed points and characterise the interesting lattices as the self-supported fixed points.

1 Motivation

Formal concept analysis (FCA [7]) is a useful tool for inducing a classification structure from data. Relational concept analysis (RCA [13]) is one of its extensions allowing to take advantage of relationships between objects to extract dependent concept lattices. One of its strong point is its ability to deal with circular dependencies between objects.

Although the result returned by RCA is solid and useful, it may not be the only possible result. The relational structure, when containing circuits, has the capability to induce richer lattice structures. Indeed, in the absence of information or of reason to separate objects, RCA classifies them within the same concept. On the contrary, in the absence of information or of reason to aggregate objects, it is possible to keep them in different concepts. A good compromise may sometimes reside in between these two extremes. As a data mining procedure, RCA can be useful in returning all possible structures and not necessarily the safest ones.

This is not really a problem in the target RCA application: extracting the core classes of a description logic ontology. However, this may be a problem for other applications.

^{*} in: Agnès Braud, Aleksey Buzmakov, Tom Hanika, Florence Le Ber (eds), Proc. 16th international conference on formal concept analysis (ICFCA), Strasbourg (FR), *Lecture notes in computer science* 12733:20-37, 2021 [doi:10.1007/978-3-030-77867-5_2](https://doi.org/10.1007/978-3-030-77867-5_2)

This work was initially motivated by one such application of RCA: We developed a link key candidate extraction algorithm on top of relational concept analysis [2]. Link keys are rules for identifying the same individuals from different data sources. In this context, the concepts of extracted lattices are link key candidates which will be selected on the basis of two independent measures [1]. As a data mining task, RCA is more useful if it generates all the possible link key candidates.

Hereafter, we illustrate the considered problem on RCA^0 , a minimal version of RCA. Although RCA^0 is simply a convenient way to illustrate the problem it requires solutions that will apply to RCA as a whole.

Understanding the nature of the problem and its relation with RCA lead to consider its semantics. The current semantics of RCA [14] focusses on the grounding of the process. We redefine this semantics on properties directly characterising the solutions.

We first consider the core function involved in the classical RCA algorithm and identify acceptable results as the fixed points of this function. We show, in the case of RCA^0 , that the classical RCA semantics corresponds to extracting its least fixed point.

We also provide a direct way to generate the greatest fixed point. However, although RCA extracts the minimal fixed point in its simplest form, this is not the case of the greatest fixed point: it would make reference to non-existent concepts. Hence we discuss the notion of self-supported concept lattice, so that the acceptable RCA results would be self-supported fixed points.

FCA is a domain of fixed points, hence it is easy to get lost among the various fixed points involved: (a) In description logics, on which RCA relies, the semantics of concepts is given by fixed points when circularities occur [11]; (b) FCA's goal is to compute fixed points: concepts are the result of a closure operator which is also a fixed point [4]; (c) finally, when confronted to cycles, the RCA concept lattice is the fixed point of the function that grows a lattice family from the previous one. The present work is concerned with the latter kind of fixed points.

In the remainder, we present the context as well as related work (§2). We illustrate the considered problem on a minimal example (§3). We then provide a fixed-point semantics for RCA based on a context-expansion function (§4) which allows to characterise the classical RCA semantics. However, this semantics being not fully satisfactory, we introduce the complementary notion of self-supported concept lattices (§5). We finally discuss concrete processing issues (§6).

2 Preliminaries and related work

We mix preliminaries with related works for reasons of space, but also because the paper directly builds on this related work.

2.1 Formal concept analysis

Formal Concept Analysis (FCA) [7] starts with a binary context $\langle G, M, I \rangle$ where G denotes a set of objects, M a set of attributes, and $I \subseteq G \times M$ a binary relation between G and M , called the incidence relation. The statement gIm is interpreted as “object g has

attribute m ". Two operators \cdot^\uparrow and \cdot^\downarrow define a Galois connection between the powersets $\langle 2^G, \subseteq \rangle$ and $\langle 2^M, \subseteq \rangle$, with $A \subseteq G$ and $B \subseteq M$:

$$A^\uparrow = \{m \in M \mid gIm \text{ for all } g \in A\}$$

$$B^\downarrow = \{g \in G \mid gIm \text{ for all } m \in B\}$$

The operators \cdot^\uparrow and \cdot^\downarrow are decreasing, i.e. if $A_1 \subseteq A_2$ then $A_2^\uparrow \subseteq A_1^\uparrow$ and if $B_1 \subseteq B_2$ then $B_2^\downarrow \subseteq B_1^\downarrow$. Intuitively, the less objects there are, the more attributes they share, and dually, the less attributes there are, the more objects have these attributes. It can be checked that $A \subseteq A^{\uparrow\downarrow}$ and that $B \subseteq B^{\downarrow\uparrow}$, that $A^\uparrow = A^{\uparrow\downarrow\uparrow}$ and that $B^\downarrow = B^{\downarrow\uparrow\downarrow}$.

For $A \subseteq G$, $B \subseteq M$, a pair $\langle A, B \rangle$, such that $A^\uparrow = B$ and $B^\downarrow = A$, is called a formal concept, where A is the extent and B the intent of $\langle A, B \rangle$. Moreover, for a formal concept $\langle A, B \rangle$, A and B are closed sets for the closure operators $\cdot^{\uparrow\downarrow}$ and $\cdot^{\downarrow\uparrow}$, respectively, i.e. $A^{\uparrow\downarrow} = A$ and $B^{\downarrow\uparrow} = B$.

Concepts are partially ordered by $\langle A_1, B_1 \rangle \leq \langle A_2, B_2 \rangle \Leftrightarrow A_1 \subseteq A_2$ or equivalently $B_2 \subseteq B_1$. With respect to this partial order, the set of all formal concepts forms a complete lattice called the concept lattice of $\langle G, M, I \rangle$.

Formal concept analysis can be considered as a function that associates to a formal context $\langle G, M, I \rangle$ its concept lattice $\langle C, \leq \rangle = FCA(\langle G, M, I \rangle)$ (or $\mathfrak{B}(\langle G, M, I \rangle)$ [7]). By abuse of language, when a variable L denotes a concept lattice $\langle C, \leq \rangle$, L will also be used to denote C .

2.2 Extending FCA

Formal concept analysis is defined on relatively simple structures hence many extensions have been designed. They allow FCA to (a) deal with more complex input structure, and/or (b) generate more expressive and interpretable knowledge structures.

Scaling Scaling is one type of extension of type (a). A scaling operation $\zeta : \mathcal{X} \mapsto 2^D$ generates boolean attributes named after a language D from a structure $\Sigma \in \mathcal{X}$. In FCA, $D = M$ and I is provided by its matrix. In scaled contexts, this language can be interpreted so that the incidence relation I is immediately derived from the attribute m following:

$$gIm \text{ iff } \Sigma \models m(g)$$

Hence, adding attributes to a context under such a structure may be performed as:

$$K_{M'}^\Sigma(\langle G, M, I \rangle) = \langle G, M \cup M', I \cup \{(g, m) \in G \times M' \mid \Sigma \models m(g)\} \rangle$$

Applying a scaling operation ζ to a formal context K following a structure Σ can be thus decomposed into (i) determining the set of attributes $\zeta(\Sigma)$ to add, and (ii) extending the context with such attributes:

$$\sigma_\zeta(K, \Sigma) = K_{\zeta(\Sigma)}^\Sigma(K)$$

Many conceptual scaling operations have been discussed in [7] for dealing with non boolean variables in formal contexts. In general, Σ is void, D is expressed as predicates,

e.g. $\cdot = v$ for nominal scaling or $\cdot \leq n$ for ordinal scaling, and \models is the evaluation of the predicate for the value.

Logical scaling [12] has been introduced for more versatile languages such as description logics and SQL. It introduces query results within formal contexts. In this case, Σ is a logical theory or database tables, D the set of formulas of the logic or instantiated queries and \models is entailment or query evaluation.

Relational scaling operations considered in [13] are based on a structure $\Sigma = \langle R, C \rangle$ made of a family of relations $R = \{r_y\}_{y \in Y}$, i.e. relations $r_y \subseteq G_x \times G_z$ between two sets of objects, and a family $C = \{C_x\}_{x \in X}$ of sets of concepts whose extent is a subset of G_x . Its language $D_{\zeta, R, C}$ is the set of attribute descriptions involving ζ , R and C . For example, qualified existential scaling (\exists) adds attributes $\exists r.c$ for $r \in R$, $r \subseteq G_x \times G_z$, $c \in C_z$ and \models checks that

$$\langle R, L \rangle \models gI\exists r.c \text{ iff } \exists g'; \langle g, g' \rangle \in r \wedge g' \in \text{extent}(c)$$

Various relational scaling operations are used in RCA such as existential, strict and wide universal, min and max cardinality, which all follow the classical role restriction semantics of description logics [3].

Other extensions Pattern structures [6,9] provide a more structured attribute language without scaling. However, its use is not directly related to the problem of context dependencies considered here as the attributes do not refer to concepts.

On the contrary, other approaches [10,5] aim at extracting conceptual structures from n -ary relations without resorting to scaling. Their concepts have intents that can be thought of as conjunctive queries and extents as tuples of objects, i.e. answers to these queries. Hence, instead of being classes, i.e. monadic predicates, concepts correspond to general polyadic predicates. For that purpose, they rely on more expressive input, e.g. in Graph-FCA [5] the incidence relation is a hypergraph between objects, and produce a more expressive representation. A comparison of RCA and Graph-FCA is provided in [8]. Graph-FCA adopts a different approach than RCA but should, in principle, suffer from the same problem as the one illustrated here. However, intents would need to refer to concepts so created, i.e. named subqueries. This remains to be studied.

2.3 Relational concept analysis

Relational Concept Analysis (RCA) [13] extends FCA to the processing of relational datasets and allows inter-object relations to be materialised and incorporated into formal concept intents. RCA is a way to induce a description logic TBox from a simple ABox [3], using specific scaling operations. It may also be thought of as a general way to deal with circular references using different scaling operations.

RCA applies to a relational context $\langle K^0, R \rangle$, composed of a set of formal contexts $K^0 = \{\langle G_x, M_x^0, I_x^0 \rangle\}_{x \in X}$ and a set of binary relations $R = \{r_y\}_{y \in Y}$. A relation $r_y \subseteq G_x \times G_z$ connects two object sets, a domain G_x ($\text{dom}(r_y) = G_x, x \in X$) and a range G_z ($\text{ran}(r_y) = G_z, z \in X$).

RCA applies relational scaling operations from a set Ω to each $K_x^i \in K^i$ and all relations $r_y \subseteq G_x \times G_z$ from the set of concepts in corresponding $L_z = FCA(K_z^i)$.

For performing its operations, RCA thus relies on FCA and σ_ζ . More precisely it uses FCA^* and σ_Ω^* defined as:

$$FCA^*(\{\langle G_x, M_x, I_x \rangle\}_{x \in X}) = \{FCA(\langle G_x, M_x, I_x \rangle)\}_{x \in X}$$

$$\sigma_\Omega^*(\{\langle G_x, M_x, I_x \rangle\}_{x \in X}, R, \{L_x\}_{x \in X}) = \left\{ \bigoplus_{\substack{\zeta \in \Omega \\ r_y \in R \mid r \subseteq G_x \times G_z}} \sigma_\zeta(\langle G_x, M_x, I_x \rangle, r_y, L_z) \right\}_{x \in X}$$

such that $\bigoplus_{r_y \in R \mid r \subseteq G_x \times G_z}^{\zeta \in \Omega}$ scales, with all operations in Ω , the given context with all the relations starting from x (to any z).

RCA starts from the initial formal context family K^0 and thus iterates the application of the two operations:

$$K^{i+1} = \sigma_\Omega^*(K^i, R, FCA^*(K^i))$$

until reaching closure, i.e. reaching n such that $K^{n+1} = K^n$. Then, $RCA_\Omega(K^0, R) = FCA^*(K^n)$.

By abuse of notation, we note $\langle G, M, I \rangle \subseteq \langle G, M', I' \rangle$ whenever $M \subseteq M'$ and $I = I' \cap (G \times M)$. In this case, because I is the incidence relation between the same G and $M \subseteq M'$, the relation only depends on M and M' . This is generalised to formal context families $\{\langle G_x, M_x, I_x \rangle\}_{x \in X} \subseteq \{\langle G_x, M'_x, I'_x \rangle\}_{x \in X}$ whenever $\forall x \in X, M_x \subseteq M'_x$.

The RCA process always reaches a closed formal context family for reason of finiteness [13] and the sequence $(K^i)_{i=0}^n$ is non-contracting, i.e. $\forall i \geq 0, K^i \subseteq K^{i+1}$ [14]. The well-grounded semantics of RCA [14] further establishes that RCA indeed finds *the* K^n satisfying these constraints through correctness (the concepts of $FCA^*(K^n)$ are grounded in K^0 through R) and completeness (all such concepts are in K^n).

2.4 RCA⁰

To keep the paper short and simple, we restrict it to RCA^0 , a special case of RCA. It is restricted in two ways:

- It contains only one formal context ($|X| = 1$),
- which has no attributes ($M_x^0 = \emptyset$).

Additionally, we will consider below only qualified existential scaling ($\Omega = \{\exists\}$).

Because RCA^0 is a restriction of RCA, we will use the same notation as defined above, though it operates on simpler structures.

Although RCA^0 seems very simple, FCA can be encoded into RCA^0 . Introducing RCA^0 is sufficient to hint at the problem that we want to illustrate¹.

3 RCA may accept different concept lattice families: illustration

As an RCA^0 example, consider the following ABox:

¹ An anonymous reviewer complements the remarks of §2.2 noting that RCA^0 is also very related to Graph-FCA as they both have only one context and using existential scaling.

$$A = \{\top(a), \top(b), \top(c), \top(d), p(a,b), p(b,a), p(c,d), p(d,c), p(a,a), p(b,b)\}$$

This can be encoded as an empty formal context and the relation of Figure 1 (left). The empty context will generate the single lattice of Figure 1 (right) (names are assigned to concepts according to their extent).



Fig. 1. Relation (left) and initial concept lattice (right).

Scaling with \exists and p provides the attribute $\exists p.ABCD$ which generates the new context of Figure 2 (left), leading to the lattice of Figure 2 (right) which is the one returned by RCA.



Fig. 2. Scaled context (left) and final concept lattice L_1 (right).

However, the concept lattices of Figure 3 are other valid lattices worth considering.

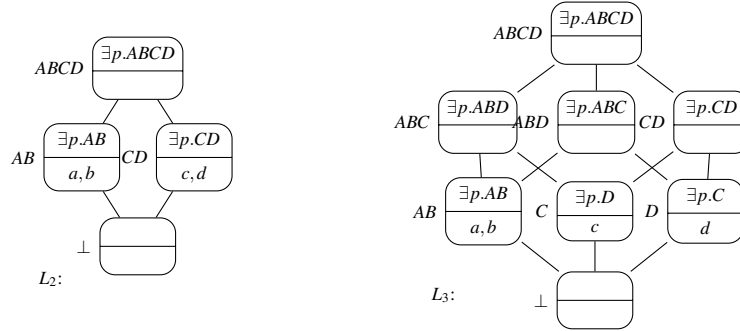


Fig. 3. Alternative concept lattices (L_2 and L_3).

They correspond to different knowledge bases:

$$T_1 = \{ABCD \sqsubseteq \exists p.ABCD\}$$

$$A_1 = \{ABCD(a), ABCD(b), ABCD(c), ABCD(d), p(a,b), p(b,a), p(c,d), p(d,c), p(a,a), p(b,b)\}$$

and

$$T_2 = \{AB \sqsubseteq \top \sqcap \exists p.AB, CD \sqsubseteq \top \sqcap \exists p.CD, ABCD \sqsubseteq \exists p.ABCD\}$$

$$A_2 = \{AB(a), AB(b), CD(c), CD(d), p(a, b), p(b, a), p(c, d), p(d, c), p(a, a), p(b, b)\}$$

and

$$T_3 = \{AB \sqsubseteq ABC \sqcap ABD \sqcap \exists p.AB, C \sqsubseteq ABC \sqcap CD \sqcap \exists p.D, D \sqsubseteq ABD \sqcap CD \sqcap \exists p.C,$$

$$ABC \sqsubseteq ABCD \sqcap \exists p.ABD, ABD \sqsubseteq ABCD \sqcap \exists p.ABC, CD \sqsubseteq ABCD \sqcap \exists p.CD,$$

$$ABCD \sqsubseteq \exists p.ABCD\}$$

$$A_3 = \{AB(a), AB(b), C(c), D(d), p(a, b), p(b, a), p(c, d), p(d, c), p(a, a), p(b, b)\}$$

In addition to extracting the TBox, these extend the ABox. However, in RCA and FCA, objects are also assigned to the created concepts. In this case, this assignment has consequences on the scaled attributes taken into account and hence the resulting lattice.

As in classical RCA, each concept of these lattices is closed with respect to the specific formal context scaled by \exists and p from the concepts of the lattice. Moreover, the lattices are self-supported in the sense that their attributes refer only to their concepts.

The problem applies to RCA as a whole as RCA^0 is included in RCA. Hence the question: Why does RCA returns only one lattice, and which one? Answering it requires to reconsider the RCA semantics.

4 Semantics and properties: a context approach

The alternative lattices presented in Section 3 are legitimate because, independently of the attributes, they rely exclusively on the structure of the relations between formal contexts. This structure is already used in the well-grounded RCA semantics, but they have not been fully exploited.

The answer will require to further define ‘legitimate’, in terms of fixed points of a specific function, and characterise the semantics of RCA as indeed grounded, in terms of these fixed points.

4.1 The lattice \mathcal{K} of RCA^0 contexts

We first define the space of formal context families considered by RCA. They are determined by three elements given once and for all: $K^0 = \{\langle G, M^0, I^0 \rangle\}_{x \in X}$, $R = \{r_y\}_{y \in Y}$, and Ω . This is even more specific for RCA^0 with $K^0 = \langle G, \emptyset, \emptyset \rangle$ and $\Omega = \{\zeta_{\exists}\}$, but for most of this section we will ignore it.

The contexts considered by RCA are formal context families obtained by the scaled initial context using the scaling operations. Given a finite set of objects G , the set of concepts that can be created from such contexts is finite and moreover each concept can be identified by its extent. Hence, we will consider that this induces a set of concept names $N(G) = 2^G$ valid for any such concept lattice; the extent of a so named concept will be the set of objects in its name. Given a finite set of relations R and scaling operations Ω , this determines the finite set $D_{\Omega, R, N(G)} = \bigcup_{r \in R}^{\zeta \in \Omega} D_{\zeta, r, N(G)}$ of possible scaled attributes in RCA^0 .

Hence, the formal contexts considered by RCA are those obtained by adding subsets of $D_{\Omega,R,N(G)}$:

$$\mathcal{K}_{\langle G, M^0, I^0 \rangle, R, \Omega} = \{K_M^{(R, N(G))}(\langle G, M^0, I^0 \rangle) \mid M \subseteq D_{\Omega, R, N(G)}\}$$

with $K_M^{(R, N(G))}(\cdot)$ the operation defined in §2.2.

Given $K, K' \in \mathcal{K}_{\langle G, M^0, I^0 \rangle, R, \Omega}$ such that $K = \langle G, M^0 \cup M, I^0 \cup I \rangle$ and $K' = \langle G, M^0 \cup M', I^0 \cup I' \rangle$, $K \vee K'$ and $K \wedge K'$ are defined as:

$$\begin{aligned} K \vee K' &= \langle G, M^0 \cup (M \cup M'), I^0 \cup (I \cup I') \rangle && \text{(join)} \\ K \wedge K' &= \langle G, M^0 \cup (M \cap M'), I^0 \cup (I \cap I') \rangle && \text{(meet)} \end{aligned}$$

It is clear that $\mathcal{K}_{K^0, R, \Omega}$ is closed by meet and join.

Property 1. $\langle \mathcal{K}_{K^0, R, \Omega}, \vee, \wedge \rangle$ is a complete lattice

Proof. \vee and \wedge satisfy commutativity, associativity and the absorption laws directly from the union and intersection on sets, so this is a lattice. It is complete because finite. \square

Property 2. $\forall K, K' \in \mathcal{K}_{K^0, R, \Omega}, K \subseteq K' \text{ iff } K = K \wedge K'$

Proof. This property also comes directly from its set theoretic counterpart application to M and M' : $K \subseteq K' \Leftrightarrow M \subseteq M' \Leftrightarrow M = M \cap M' \Leftrightarrow K = K \wedge K'$ \square

4.2 The context expansion function F

We reformulate RCA as based on a main single function, $F_{K^0, R, \Omega}$, the context expansion function attached to a relational context $\langle K^0, R \rangle$ and a set Ω of scaling operations.

Definition 1 (Context expansion function). *Given a relational context $\langle K^0, R \rangle$ and a set of relational scaling operations Ω , the function $F_{K^0, R, \Omega} : \mathcal{K}_{K^0, R, \Omega} \mapsto \mathcal{K}_{K^0, R, \Omega}$ is defined by:*

$$F_{K^0, R, \Omega}(K) = \sigma_{\Omega}^*(K, R, FCA^*(K))$$

The function expression is independent from K^0 , K^0 is used to restrict the domain of the function so that its elements cover K^0 . From now on, we will abbreviate $\mathcal{K}_{K^0, R, \Omega}$ as \mathcal{K} and $F_{K^0, R, \Omega}$ as F . This is legitimate because, for a given relational context, K^0 , R and Ω do not change. F is an extensive and monotone internal operation for \mathcal{K} :

Property 3. $\forall K \in \mathcal{K}, F(K) \in \mathcal{K}$

Proof. Scaling only adds attributes from $D_{\Omega, R, N(G)}$. \square

Property 4 (F is extensive and monotone). The function F attached to a relational context and a set of scaling operator satisfies:

$$\begin{aligned} K &\subseteq F(K) && \text{(extensivity)} \\ K \subseteq K' &\Rightarrow F(K) \subseteq F(K') && \text{(monotony)} \end{aligned}$$

Proof. **extensivity** holds because F eventually adds to each formal context in K new attributes scaled from $FCA(K)$. The set of attributes can thus not be smaller. **monotony** holds because $K \subseteq K'$ means that $M \subseteq M'$. This entails that the set of concepts of $FCA(K)$ is included in that of $FCA(K')$, hence the set of attributes A scaled from K is included in the set A' scaled from K' . Since, they are added to M and M' , then $M \cup A \subseteq M' \cup A'$, hence $F(K) \subseteq F(K')$. \square

Extensivity corresponds to the non-contracting property of the well-grounded semantics [14] and monotony is also called order-preservation.

4.3 Fixed points of F

Given F , it is possible to define its sets of fixed points, i.e. the sets of formal contexts closed for F , as:

Definition 2 (fixed point). *A formal context $K \in \mathcal{K}$ is a fixed point for a context expansion function F , if $F(K) = K$. We call $\text{fp}(F)$ the set of fixed points for F .*

Since \mathcal{K} is a complete lattice and F is order-preserving (or monotone) on \mathcal{K} , then the Knaster-Tarski theorem applies:

Theorem 1 (Knaster-Tarski theorem [15]). *Let \mathcal{K} be a complete lattice and let $F : \mathcal{K} \mapsto \mathcal{K}$ be an order-preserving function. Then the set of fixed points of F in L is also a complete lattice.*

In particular, this warrants that there exists least and greatest fixed points of F in \mathcal{K} (called $\text{lfp}(F)$ and $\text{gfp}(F)$).

In FCA, and subsequently in RCA without circular dependencies, the images by FCA^* of all fixed points of F are isomorphic. They will have the same concept lattice as seen from their intent. Even with RCA and circular dependencies (between the objects or between the contexts), this is often the case. These are these concept lattices that the RCA algorithm computes. But the example of Section 3 shows that, even in RCA^0 , there may be several fixed points for F with non isomorphic lattices. Hence the question: which fixed point is returned by RCA's well-grounded semantics [14]?

4.4 The well-grounded semantics of RCA is the least fixed-point semantics

RCA may be redefined as

$$RCA_{\Omega}(K^0, R) = FCA^*(F^{\infty}(K^0))$$

RCA iterates F from K^0 until closure, and ultimately applies FCA^* . Since K^0 belongs to \mathcal{K} , then it computes a fixed point of F . This is the least fixed point.

Proposition 1 (The RCA algorithm computes the least fixed point). *Given F the context expansion function associated to K^0 , R and Ω ,*

$$RCA_{\Omega}(K^0, R) = FCA^*(\text{lfp}(F_{K^0, R, \Omega}))$$

Proof. $RCA_{\Omega}(K^0, R) = FCA^*(F^n(K^0))$ for some n at which $F(F^n(K^0)) = F^n(K^0)$ [13]. Let $K^{\infty} = F^n(K^0)$, $K^{\infty} \in fp(F)$ (Definition 2). $\forall K \in fp(F)$, $K \in \mathcal{K}$, thus $K^0 \subseteq K$ because all the contexts in \mathcal{K} contain M^0 . By monotony (Property 4), $K^{\infty} = F^n(K^0) \subseteq F^n(K) = K$, because K is a fixed point. Thus, K^{∞} is a fixed point more specific than all fixed points: it is the least fixed point. \square

4.5 Computing the greatest fixed point

A natural question is how to obtain the greatest fixed point. In fact, under this approach this is (theoretically) surprisingly easy.

Proposition 2. $gfp(F_{(G, M^0, I^0), R, \Omega}) = K_{D_{\Omega, R, N(G)}}^{(R, N(G))}(\langle G, M^0, I^0 \rangle)$

Proof. This context is the greatest element of \mathcal{K} as it contains all attributes of $D_{\Omega, R, N(G)}$. It is also a fixed point because F is extensive and internal. \square

The lattice corresponding to the greatest fixed point will be $L = FCA^*(gfp(F_{K^0, R, \Omega}))$.

This result is easy but very uncomfortable. The obtained lattice may contain many useless attributes. Indeed, $\exists r.c$ is well defined by the incidence relation, but it is of no use to RCA if c does not belong to L .

In the example of Section 3, the attribute $\exists p.A$ belongs to $D_{\Omega, R, N(G)}$ though A does not belong to the maximal lattice L_3 , because it is not closed. The fact that both a and b satisfy this attribute makes that it will find its place in the intent of AB . If one considers the lattice in isolation, this is perfectly valid because the scaled context is well-defined: $\exists p.A$ is just an attribute among others satisfied by a and b . However, if the lattice is transformed in a description logic TBox, this is not correct to refer to an undefined class. This is not the result that we expected: we need the results to be self-supported.

This problem is even more embarrassing if one wants to enumerate all fixed points, which are as many solutions to the RCA problem: many of these will feature such non-supported attributes.

5 Self-supported fixed points

We first quickly approach this problem from the concept lattice standpoint, it is better understood with both contexts and lattices together². We then define self-supported concept lattices and consider their interaction with fixed points.

5.1 The lattice \mathcal{L} of RCA^0 lattices and the lattice expansion function E

From $\mathcal{K}_{K^0, R, \Omega}$, one can define $\mathcal{L}_{K^0, R, \Omega}$ as the finite set of images of $\mathcal{K}_{K^0, R, \Omega}$ by FCA. These are concept lattices obtained by applying FCA on K^0 extended with a subset of $D_{\Omega, R, N(G)}$:

² Instead of developing both \mathcal{K} and \mathcal{L} independently and maintaining an equivalence between them, it would have been possible to use a more FCA-like structure associating the corresponding contexts and lattices.

$$\mathcal{L}_{(G,M^0,I^0),R,\Omega} = \{FCA(\langle G, M^0 \cup M, I^0 \cup I \rangle) \mid M \subseteq D_{\Omega,R,N(G)}\}$$

We define a specific type of homomorphisms between two concept lattices when concepts are simply mapped into concepts with the same extent and possibly increased intent.

Definition 3 (Lattice homomorphism). A concept lattice homomorphism $h: \langle C, \leq \rangle \mapsto \langle C', \leq' \rangle$ is a function which maps each concept $c \in C$ into a corresponding concept $h(c) \in C'$ such that:

- $\forall c \in C, \text{intent}(c) \subseteq \text{intent}(h(c)),$ and
- $\forall c \in C, \text{extent}(c) = \text{extent}(h(c)),$ and
- $\forall c, d \in C, c \leq d \Rightarrow h(c) \leq' h(d).$

We note $L \preceq L'$ if there exists a homomorphism from L to L' . In principle, $L \simeq L'$ if $L \preceq L'$ and $L' \preceq L$, but here, \simeq is $=$. The order between concept lattices is straightforwardly extended to families of concept lattices such that: $\{L_x\}_{x \in X} \preceq \{L'_x\}_{x \in X}$ iff $\forall x \in X, L_x \preceq L'_x$.

There exists an implicit function $\kappa: \mathcal{L}_{K^0,R,\Omega} \mapsto \mathcal{H}_{K^0,R,\Omega}$ such that $\forall L \in \mathcal{L}_{K^0,R,\Omega}, L = FCA(\kappa(L))$. Since \simeq is the same as $=$ which identifies lattices containing concept having exactly the same intent and extent. $\kappa(L)$ can be induced by collecting the attributes present in L intents to build the unique M , from which the corresponding I is obtained [7].

We define $E_{K^0,R,\Omega}$, the lattice expansion function attached to a relational context $\langle K^0, R \rangle$ and a set Ω of scaling operators.

Definition 4 (Lattice expansion function). Given a relational context $\langle K^0, R \rangle$ and a set of relational scaling operations Ω the function $E_{K^0,R,\Omega}: \mathcal{L}_{K^0,R,\Omega} \mapsto \mathcal{L}_{K^0,R,\Omega}$ is defined by:

$$E_{K^0,R,\Omega}(L) = FCA^*(\sigma_\Omega^*(\kappa(L), R, L))$$

Here again, K^0 is only used to constrain the domain of the function, not its expression. From now on, we will abbreviate $\mathcal{L}_{K^0,R,\Omega}$ as \mathcal{L} and $E_{K^0,R,\Omega}$ as E .

Instead of considering that $RCA(K^0) = FCA^*(F^\infty(K^0))$, it is possible to consider that $RCA(K^0) = E^\infty(FCA^*(K^0))$. Hence, RCA may be redefined as

$$RCA_\Omega(K^0, R) = E^\infty(FCA^*(K^0))$$

RCA iterates E from $FCA^*(K^0)$ until closure. The definition of E amounts to first scaling and then applying FCA, though F does the opposite (see Figure 4).

In consequence, E is the function corresponding to F in the sense that $E = FCA \circ F \circ \kappa$ and $FCA^* \circ E = F \circ FCA^*$ (see Figure 4). Actually, the results obtained for \mathcal{H} and F , hold exactly for \mathcal{L} and E :

- $\langle \mathcal{L}, \preceq \rangle$ is a complete lattice;
- E is an internal, monotone and extensive operation of \mathcal{L} ;
- $RCA_\Omega(K^0, R) = \text{Ifp}(E_{K^0,R,\Omega})$.

E inherits exactly all properties of F : the desirable ones and the problematic ones. So, apparently no progress has been made.

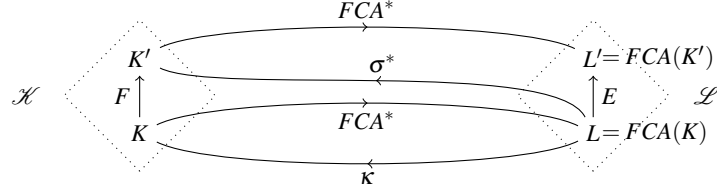


Fig. 4. Relations between F and E through the alternation of FCA^* and σ_Ω^* .

5.2 Self-supported lattices

The problem is that both F and E are extensive functions. Hence, it is possible, starting from anywhere in \mathcal{H} or \mathcal{L} , to consider attributes that do not refer to concepts and these attributes will be preserved. As a consequence, there are fixed points with these unwanted attributes and they are also found in the greatest fixed point.

One may consider identifying such attributes from the greatest fixed point and forbidding them. However, these meaningless attributes are contextual: one supported attribute in the greatest fixed point, may be non supported in a smaller lattice. This is a base difficulty for enumerating these fixed points.

Instead, we consider only self-supported lattices, i.e. lattices whose intents only refer to their own concepts.

Definition 5 (Self-supported lattices). Let \mathcal{L} a set of concept lattices, its set of self-supported lattices is

$$S(\mathcal{L}) = \{L \in \mathcal{L}_{K^0, R, \Omega} \mid \forall c \in L, \text{intent}(c) \subseteq D_{\Omega, R, L}\}$$

The set of interesting lattices that may be returned by RCA^0 can be circumscribed as $\text{fp}(E) \cap S(\mathcal{L})$ as these are stable and self-supported. Moreover, by construction of \mathcal{H} and \mathcal{L} , they cover K^0 .

E has the advantage of preserving self-supportivity.

Proposition 3 (E is internal to $S(\mathcal{L})$). $\forall L \in S(\mathcal{L}), E(L) \in S(\mathcal{L})$.

Proof. If $L \in S(\mathcal{L})$, all attributes in intents of L are supported by concepts in L . $E = FCA^* \circ \sigma_\Omega^*$. σ_Ω^* first adds to $\kappa(L)$ attributes which are supported by L . $L \preceq E(L)$, so these concepts are still in $E(L)$. Hence, the attributes in $\kappa(L)$ and those scaled by σ_Ω^* are still supported by $E(L)$. \square

But the definition of S does not provide a direct way to transform a non self-supported lattice into a self-supported one: the suppression of non self-supported attributes from intents could result in non-concepts (with non closed-extent). One possible way to solve this problem consists of extracting only the attributes currently in the lattice and to apply FCA^* to the resulting context.

For that purpose, we introduce a filtering function $\pi : \mathcal{L} \mapsto \mathcal{H}$ which suppresses from the induced context ($\kappa(L)$) those attributes non supported by the lattice:

$$\pi(L) = \langle G, M \setminus D_{\Omega, R, N(G) \setminus L}, I \setminus \{\langle g, m \rangle \mid m \in D_{\Omega, R, N(G) \setminus L}\} \rangle$$

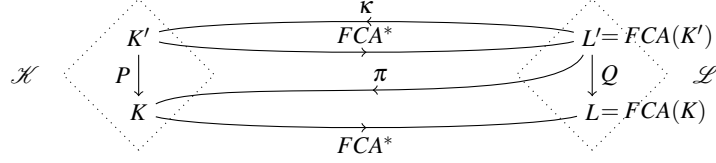


Fig. 5. Relations between P and Q through the alternation of FCA^* and π .

such that $\kappa(L) = \langle G, M, I \rangle$.

One can define $Q : \mathcal{L} \mapsto \mathcal{L}$, such that

$$Q(L) = FCA^*(\pi(L))$$

or $P : \mathcal{H} \mapsto \mathcal{H}$, such that $P(K) = \pi(FCA^*(K))$, see Figure 5.

Contrary to E , Q is anti-extensive and monotone:

Proposition 4 (Q is anti-extensive and monotone). *The function Q satisfies:*

$$Q(L) \preceq L \quad (\text{anti-extensivity})$$

$$L \preceq L' \Rightarrow Q(L) \preceq Q(L') \quad (\text{monotony})$$

Proof. **anti-extensivity** $\pi(L) \subseteq \kappa(L)$ because π simply suppresses attributes from $\kappa(L)$.

Hence, $FCA^*(\pi(L)) \preceq FCA^*(\kappa(L))$ because the latter contain all concepts of the former (identified by extent) eventually featuring the removed attributes. Moreover, $FCA^*(\kappa(L)) = L$ by definition, thus $Q(L) = FCA^*(\pi(L)) \preceq FCA^*(\kappa(L)) = L$.

monotony If $L \preceq L'$, then $\kappa(L) \subseteq \kappa(L')$, otherwise FCA^* would not generate a smaller lattice. In addition, $L \preceq L'$ entails $N(G) \setminus L \supseteq N(G) \setminus L'$ which entails $D_{\Omega, R, N(G) \setminus L} \supseteq D_{\Omega, R, N(G) \setminus L'}$, which finally together leads to $M \setminus D_{\Omega, R, N(G) \setminus L} \subseteq M' \setminus D_{\Omega, R, N(G) \setminus L'}$. Then, $\pi(L) \subseteq \pi(L')$ because a smaller context supported by a smaller lattice cannot result in a larger context. Hence, $Q(L) = FCA^*(\pi(L)) \preceq FCA^*(\pi(L')) = Q(L')$. \square

It would be possible to redefine $S(\mathcal{L})$ as $fp(Q)$. Like with E , it is possible to apply the Knaster-Tarski theorem to show that $\langle fp(Q), \preceq \rangle$ is a complete lattice.

But like E , Q is not a closure operator as it is not idempotent. However, with the same arguments as [13], it can be argued that the repeated application of Q converges to a self-supported concept lattice.

Proposition 5. $\forall L \in \mathcal{L}, \exists n; Q^n(L) = Q^{n+1}(L)$ and $Q^n(L) \in S(\mathcal{L})$.

Proof. First, L is a finite concept lattice. Moreover, $Q(L) \preceq L$, hence it not possible to build an infinite chain of non converging application of Q since at each iteration, either π suppresses no attribute (and then closure has been reached), or it suppresses at least one attribute and then a strictly smaller context is reached. Ultimately, the least fixed point $lfp(Q) = FCA^*(K^0)$ is reached. It is a fixed point because $\kappa(FCA^*(K^0)) = K^0$ contains no scaled attribute and thus is self-supported. When closure is reached, this is because π does not find any non-supported attribute in the lattice intents. This means that all of them are supported by the lattice. \square

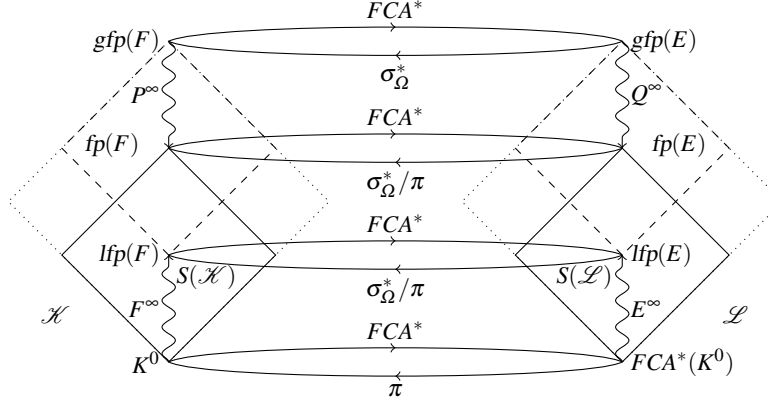


Fig. 6. The \mathcal{L} (resp. \mathcal{H}) lattice and effects of E and Q (resp. F and P) for characterising $fp(E)$ and $S(\mathcal{L})$ (resp. $fp(F)$ and $S(\mathcal{H})$).

By convention, we note Q^∞ the closure function associated with Q .

We end up with two operations, E and Q , the former extensive and the latter anti-extensive, that may be transformed into closure operators. These functions are instrumental to provide the infimum and supremum of our desired lattices (see also Figure 6):

Proposition 6. *$lfp(E)$ and $Q^\infty(gfp(E))$ are respectively the infimum and supremum of $fp(E) \cap S(\mathcal{L})$ for \preceq .*

Proof. $lfp(E)$ is the lower bound for $fp(E) \cap S(\mathcal{L})$ because it is the lower bound for $fp(E)$. It is the infimum of $fp(E) \cap S(\mathcal{L})$ for \preceq because $FCA^*(K^0) \in S(\mathcal{L})$ and by Proposition 3 this property is preserved by E and since $lfp(E) = E^\infty(FCA^*(K^0))$, it belongs to $S(\mathcal{L})$.

$Q^\infty(gfp(E))$ is the upper bound for $fp(E) \cap S(\mathcal{L})$ because $gfp(E)$ contains all possible closed concepts that can be built from $D_{\Omega, R, N(G)}$. Hence, those attributes not belonging to $\pi(gfp(E))$ cannot belong to any self-supported lattice. By Proposition 5, $Q^\infty(gfp(E)) \in S(\mathcal{L})$. If $Q^\infty(gfp(E)) \notin fp(E)$, this entails $Q^\infty(gfp(E)) \prec E(Q^\infty(gfp(E)))$ and moreover that $\exists n; E^n(Q^\infty(gfp(E))) \in fp(E)$ (because E is extensive and the space is finite). But, by Proposition 3, E preserves self-supportiveness. Thus, $E^n(Q^\infty(gfp(E))) \in fp(E) \cap S(\mathcal{L})$ and $Q^\infty(gfp(E)) \prec E^n(Q^\infty(gfp(E)))$, which is contradictory with the fact that $Q^\infty(gfp(E))$ is an upper-bound for all fixed points. Thus, $Q^\infty(gfp(E))$ is the supremum of $fp(E) \cap S(\mathcal{L})$ for \preceq . \square

6 Discussion

Our initial goal was to define which concept lattices could be considered as the result of RCA on a relational context. RCA provides a practical algorithm (based on F or E and FCA^*) to find out the smallest of these: $lfp(E)$. We have characterised the greatest one: $gfp(Q)$ or $Q^\infty(gfp(E))$.

We end up with two functions, complementary in their structure, one expanding the context, the other contracting it. In the perspective of enumerating all self-supported fixed points, it is tempting to either start from $lfp(E)$ and use E or start from $gfp(Q)$ and use Q . Unfortunately, these starting points being fixed points for these very functions, this leads nowhere. It is necessary to escape the fixed points. For instance, starting from $lfp(E)$, one could add non-supported attributes until they become supported. Performing this attribute by attribute is not very smart. Figure 3 shows possible fixed points: L_2 and L_3 . They require to add 2 or 6 attributes to L_1 . A smarter strategy would consist of analysing the sets of attributes that support each others, through the induction of concepts, and adding these one by one to $lfp(E)$ or suppressing them from $gfp(Q)$. There is a known bound to this set since none of the attributes not in the intents of $gfp(Q)$ can be added, and none of those in $lfp(E)$ can be suppressed. Finally, these sets may entertain dependencies (adding one set of attributes would immediately support another). This may be dealt with by computing such dependencies or by applying the required closure operator (E^∞ or Q^∞) after each addition.

Such a procedure seems to be achievable with RCA^0 , it will be more difficult to set up with RCA due to dependencies across lattices.

7 Conclusions

Motivated by the requirement to extract more concepts with relational concept analysis, we gave a new, fixed-point based, semantics for RCA^0 . The main contribution of this work is the formulation of the RCA semantics in terms of fixed points of the function (F or E) at the core of RCA^0 . Then it is shown that the well-grounded semantics of RCA corresponds to the least fixed-point semantics.

We also identified as self-supported fixed points those other fixed points of interest. The least fixed point being the smaller of these. This led to develop another function (P or Q) which, together with FCA^* , allows extracting the greatest of them as an alternative to RCA .

This result does not mean that RCA is wrong. In FCA , conceptual scaling has been considered as a human-driven analysis tool: a knowledgeable person could provide attribute in this language for describing better the data to be analysed. In RCA , scaling is used as an extraction tool, with the drawback to potentially generate many attributes. By only extracting the least fixed point, RCA avoids generating too many of them.

In the context of extracting a $TBox$ for a particular $ABox$, extracting the least fixed point is adequate since it may be relatively complex and it is a good starting point. But for other applications, such as link key candidate extraction, it is very important to have all possible fixed points because external measures are used for selecting the best one (which has no reason to be either the least or the greatest one).

The definitions and results of Sections 4 and 5 have been restricted to RCA^0 for the sake of clarity. Although this remains to be proved, they should hold for RCA as a whole. Indeed, all definitions can be applied to families of contexts and lattices, the order between them being the product order induced by the piece-wise conjunction. All operations remain monotone and extensive (or anti-extensive) as soon as the selected scaling operations are. This is enough to preserve the results.

Acknowledgements

This work has been partially funded by the ANR Elker project (ANR-17-CE23-0007-01). The author thanks Philippe Besnard for pointing to the Knaster-Tarski theorem.

References

1. Atencia, M., David, J., Euzenat, J.: Data interlinking through robust linkkey extraction. In: Proc. 21st european conference on artificial intelligence (ECAI), Praha (CZ). pp. 15–20 (2014)
2. Atencia, M., David, J., Euzenat, J., Napoli, A., Vizzini, J.: Link key candidate extraction with relational concept analysis. *Discrete applied mathematics* **273**, 2–20 (2020)
3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The description logic handbook: theory, implementations and applications. Cambridge University Press (2003)
4. Belohlávek, R.: Introduction to formal concept analysis. Tech. rep., Univerzita Palackého, Olomouc (CZ) (2008)
5. Ferré, S., Cellier, P.: Graph-FCA: an extension of formal concept analysis to knowledge graphs. *Discrete applied mathematics* **273**, 81–102 (2020)
6. Ganter, B., Kuznetsov, S.: Pattern structures and their projections. In: Proc. 9th International conference on conceptual structures (ICCS). Lecture Notes in Computer Science, vol. 2120, pp. 129–142 (2001)
7. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer, Heidelberg (DE) (1999)
8. Keip, P., Ferré, S., Gutierrez, A., Huchard, M., Silvie, P., Martin, P.: Practical comparison of FCA extensions to model indeterminate value of ternary data. In: Proc. 15th International Conference on Concept Lattices and Their Applications (CLA), Tallinn (EE). CEUR Workshop Proceedings, vol. 2668, pp. 197–208 (2020)
9. Kuznetsov, S.: Pattern structures for analyzing complex data. In: Proc. International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing (RSFDGrC). Lecture notes in computer science, vol. 5908, pp. 33–44 (2009)
10. Kötters, J.: Concept lattices of a relational structure. In: Proc. 21th International Conference on Conceptual Structures (ICCS). Lecture Notes in Computer Science, vol. 7735, pp. 301–310 (2013)
11. Nebel, B.: Reasoning and revision in hybrid representation systems. Lecture Notes in Artificial Intelligence 422, Springer Verlag, Berlin (DE) (1990)
12. Prediger, S.: Logical scaling in formal concept analysis. In: Proc. 5th International Conference on Conceptual Structures (ICCS), Seattle (WA US). Lecture Notes in Computer Science, vol. 1257, pp. 332–341 (1997)
13. Rouane Hacene, M., Huchard, M., Napoli, A., Valtchev, P.: Relational concept analysis: mining concept lattices from multi-relational data. *Annals of Mathematics and Artificial Intelligence* **67**(1), 81–108 (2013)
14. Rouane Hacene, M., Huchard, M., Napoli, A., Valtchev, P.: Soundness and completeness of relational concept analysis. In: Proc. 11th International Conference on Formal Concept Analysis (ICFCA). Lecture Notes in Computer science, vol. 7880, pp. 228–243. Springer (2013)
15. Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. *Pacific journal of mathematics* **5**(2), 285–309 (1955)