Stepwise functional refoundation of relational concept analysis

Jérôme Euzenat

RESEARCH REPORT
N° 9518
October 2023
Project-Team mOeX

ISSN 0249-6399 ISRN INRIA/RR--9518--FR+ENG

Stepwise functional refoundation of relational concept analysis

Jérôme Euzenat*

Project-Team mOeX

Research Report n° 9518 — version 2 — initial version October 2023 — revised version December 2023 — 73 pages

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

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Abstract: Relational concept analysis (RCA) is an extension of formal concept analysis dealing with several related contexts simultaneously. It has been designed for learning description logic theories from data and used within various applications. A puzzling observation about RCA is that it returns a single family of concept lattices although, when the data feature circular dependencies, other solutions may be considered acceptable. The semantics of RCA, provided in an operational way, does not shed light on this issue. In this report, we define these acceptable solutions as those families of concept lattices which belong to the space determined by the initial contexts (*well-formed*), which cannot scale new attributes (*saturated*), and which refer only to concepts of the family (*self-supported*). We adopt a functional view on the RCA process by defining the space of well-formed solutions and two functions on that space: one expansive and the other contractive. We show that the acceptable solutions are the common fixed points of both functions. This is achieved step-by-step by starting from a minimal version of RCA that considers only one single context defined on a space of contexts and a space of lattices. These spaces are then joined into a single space of context-lattice pairs, which is further extended to a space of indexed families of context-lattice pairs representing the objects manipulated by RCA. We show that RCA returns the least element of the set of acceptable solutions. In addition, it is possible to build dually an operation that generates its greatest element. The set of acceptable solutions is a complete sublattice of the interval between these two elements. Its structure and how the defined functions traverse it are studied in detail.

Key-words: Formal concept analysis – relational concept analysis – fixed point – fixed-point semantics – circular dependency

Refondation fonctionnelle progressive de l'analyse relationnelle de concepts

Résumé : L'analyse relationnelle de concepts (RCA) est une extension de l'analyse formelle de concepts qui permet de traiter plusieurs contextes liés simultanément. Elle a été conçue pour induire des théories en logiques de description à partir de données et est utilisée dans diverses applications. Une observation troublante est que la RCA retourne une unique famille de treillis de concepts bien que, lorsque les données entretiennent des dépendances circulaires, d'autres solutions semblent acceptables. La sémantique de l'analyse relationnelle de concepts, définie de manière opérationnelle, n'éclaire pas cette question. Dans ce rapport, les solutions acceptables sont définies comme les familles de treillis de concepts qui appartiennent à l'espace délimité par les contextes initiaux (*bien formées*), ne peuvent supporter de nouveaux attributs (*saturées*) et ne réfèrent qu'à des concepts de la famille (*auto-supportées*). Nous adoptons une approche fonctionnelle de la RCA en définissant l'espace des solutions bien formées et deux fonctions sur cet espace : l'une expansive et l'autre contractive. Nous montrons que les solutions acceptables sont les points fixes communs aux deux fonctions. Ce résultat est obtenu progressivement en partant d'une version minimale de RCA qui ne considère qu'un unique contexte et en définissant un espace de contextes et un espace de treillis correspondant. Ces espaces sont ensuite rassemblés en un espace de paires contexte-treillis qui est étendu en un espace de familles indexées de paires de contexte-treillis représentant les objets manipulés par la RCA. Ceci permet de montrer que l'algorithme de RCA retourne le plus petit élément de l'ensemble des solutions acceptables. De plus, il est possible de définir une opération duale qui retourne le plus grand élément. L'ensemble des solutions acceptables forme un sous-treillis complet de l'intervalle entre ces deux éléments. Nous étudions en détail la structure de celui-ci et, en particulier, comment les fonctions définies le parcourent.

Mots-clés : Analyse formelle de concepts – analyse relationnelle de concepts – point fixe – sémantique de point fixe – dépendance circulaire

Contents

1 Introduction

Formal concept analysis (FCA) is a well-defined and widely used operation for extracting concept lattices from binary data tables [Ganter and Wille, 1999]. It has received many extensions and has been put to work in a variety of applications [Ganter et al., 2005; Missaoui et al., 2022]. Relational concept analysis (RCA) is such an extension of FCA that generates several dependent concept lattices from related data tables [Rouane-Hacene et al., 2013a]. It has initially been designed to generate description logic terminologies (or ontologies) [Baader et al., 2003] from interrelated data. However, it can be used for other purposes such as generating link keys used to extract links from RDF data sets [Atencia et al., 2020].

Relational concept analysis can be used for instance for analysing the ecological and sanitary quality of watercourses [Ouzerdine et al., 2019]. For that purpose it connects contexts corresponding to such water courses to data collection points which themselves are connected to measures and to organisms collected in water that can be described by further attributes. These relations between objects help generating richer concept descriptions comprising relations between concepts. It is then possible to connect the abundance or scarcity of some species to the presence of some polluants (e.g. glyphosate).

The semantics of relational concept analysis has, so far, been provided in a rather operational way [Rouane-Hacene et al., 2013b]. It specifies that the RCA operation returns a family of concept lattices referring to each other that describe the input data and it shows that this result is unique. However, when there exist cycles in the dependencies between data, several families may satisfy these constraints. Hence, the question remains to understand which unique one is returned.

This question stemmed out of curiosity. It occurred to us through experimenting with relational concept analysis for extracting link keys. Although RCA was returning acceptable results, it was easy to identify other valid results that it did not return. When RCA is used for extracting description logic terminologies, it make sense to return minimal terminologies that may be extended. But different sets of link keys would return totally different sets of links. The problem also manifests itself in applications in which developers add artificial identifiers in their data in order to constrain the returned solution [Braud et al., 2018; Dolques et al., 2012].

Hence, relational concept analysis needs a more precise and process-independent semantics that defines what it returns. For that purpose, this report provides a structured description of the space on which relational concept analysis applies. It then defines acceptable solutions as those families of concept lattices which belong to the space determined by the initial contexts (*well-formed*), cannot scale new attributes (*saturated*), and refer only to concepts of the family (*self-supported*).

Relational concept analysis is then studied in a functional framework. It characterises the acceptable solutions as the fixed points of two functions, one expansive, which extends concept lattices as long as there are reasons to generate concepts distinguishing objects, and the other contractive, which reduces concept lattices as long as the attributes they are built on are not supported by remaining concepts. These functions are recursive and the acceptable solutions are those families of concept lattices which are fixed points for both (Proposition 60): there is no reason to either extend nor reduce them. The results provided by RCA are then proved to be the smallest acceptable solution, which is the least fixed point of the expansive function (Proposition 61). It also offers an alternative semantics based on the greatest element of this set, which is the greatest fixed point of the contractive function (Proposition 62). The structure of the set of fixed points is further characterised to support algorithmic developments.

This extends the results obtained for the RCA^0 restriction of RCA [Euzenat, 2021], which contains a single formal context, hence a single concept lattice, and no attribute, only relations. In spite of its simplicity, the main arguments of this work were already valid for RCA^0 which remains a good introduction to the problems faced. Here, we develop the full argument starting from RCA^0 and extending it step-by-step to apply to RCA.

In addition to these more general results, the current report provides better and more examples, a revised more general notation, wider related work, and insights on the structure of the set of acceptable solutions. It features many elementary properties and propositions which help keeping their proof manageable so the proofs are given immediately.

We first present the work on which this one builds (relational concept analysis) and relevant related work (Section 2). We then provide simple examples illustrating that RCA and RCA^0 may accept concept lattices which are not those provided by the RCA operation (Section 3). RCO^0 is then provided with a fixed-point semantics through an expansion function corresponding to the RCA algorithm (Section 4). We then discuss the notion of self-supported lattices (Section 5) which is characterised as the fixed points of another function that can be seen as dual to that used by RCA. So far, these characterisations have been provided in parallel on both formal contexts and concept lattices. Section 6 reconciles them by unifying both approaches in dependent pairs of contexts and lattices. This allows us to precisely characterise the space of acceptable solutions, fixed points of these complementary functions, by considering the composition of the corresponding closures. This is finally generalised to RCA globally by considering families of related context-lattices pairs and showing that the acceptable families of concept lattices as those which are fixed-points for both functions (Section 7). We characterise exactly the results of RCA as the smallest element of this set, we provide an alternative operation returning the greatest element and we study the structure of the set of acceptable solutions (Section 8).

2 Preliminaries and related work

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

Table 1 provides a list of symbols used here.

Sets and structures

G	set of objects ($g \in G$)	8
M	set of attribute ($M \subseteq D$, $m \in M$)	8
I	incidence relation ($I \subseteq M \times G$)	8
J	'ternary' element used in conceptual scaling ($J \subseteq G \times M \times W$)	10
R	set of relations ($R \subseteq G \times G'$, $r \in R$)	19
K	formal contexts ($K = \langle G, M, I\rangle$, $K \in \mathscr{K}$)	8
Ω	set of scaling operations ($\varsigma \in \Omega$)	12
$N(\cdot)$	concept names (given after extent, $N(L) \subseteq N(K) \subseteq 2^G$)	8
D	property language for expressing attributes (inspired from Pattern structures)	9
L	concept lattice ($L = \langle C, \preceq\rangle$, $L \in \mathscr{L}$)	8
C	set of formal concepts ($C \subseteq 2^{G \times M}$, $c \in C$)	8
T	context-lattice pairs ($T = \langle K, L\rangle$, $T \in \mathscr{T}$)	43
O	indexed context-lattice pairs ($O = \{T_x\}_{x \in X}$, $O \in \mathscr{O}$) for modelling RCA entirely	49
Σ	semantic structure grounding scaling ($\Sigma \in \mathscr{X}$, here $\Sigma \subseteq \{R\} \times \mathscr{L}$)	9

Functions

FCA	$: \mathscr{K} \to \mathscr{L}$ Formal concept analysis (extended to indexed families as FCA*)	8
κ	$: \mathscr{L} \to \mathscr{K}$ Context extraction operation (κ^*)	31
σ	$: \mathscr{K} \times \mathscr{L} \to \mathscr{K}$ Scaling operation (σ^*)	9
π	$: \mathscr{L} \to \mathscr{K}$ Purge function (π^*)	37
F	$: \mathscr{L} \to \mathscr{L}$ Concept lattice expansion	29
E	$: \mathscr{K} \to \mathscr{K}$ Context expansion	33
P	$: \mathscr{K} \to \mathscr{K}$ Context contraction	39
Q	$: \mathscr{L} \to \mathscr{L}$ Concept lattice contraction	38
T	$: \mathscr{K} \to \mathscr{T}$ Context-lattice pair constructor	43
EF	$: \mathscr{T} \to \mathscr{T}$ Context-lattice pair expansion (EF^*: $\mathscr{O} \to \mathscr{O}$)	45
PQ	$: \mathscr{T} \to \mathscr{T}$ Context-lattice pair contraction (PQ^*: $\mathscr{O} \to \mathscr{O}$)	47
$\underline{\text{RCA}}$	$: \mathscr{K}^* \to \mathscr{L}^*$ Relational concept analysis	16
$\overline{\overline{\text{RCA}}}$	$: \mathscr{K}^* \to \mathscr{L}^*$ Relational concept analysis with a greatest fixed-point semantics	58

Table 1: Some symbols used in this document (signatures are simplified omitting R and Ω).

2.1 Basics of formal concept analysis

This report relies only on the most basic results of formal concept analysis expressed as order-preserving functions.

Formal Concept Analysis (FCA) [Ganter and Wille, 1999] 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", also noted $m(g)$. 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 $B^{\downarrow} = B^{\downarrow\uparrow\downarrow}$.

A pair $\langle A, B\rangle \in 2^G \times 2^M$, 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 for the closure operators $\cdot^{\uparrow\downarrow}$ and $\cdot^{\downarrow\uparrow}$, respectively, i.e. $A^{\uparrow\downarrow} = A$ and $B^{\downarrow\uparrow} = B$.

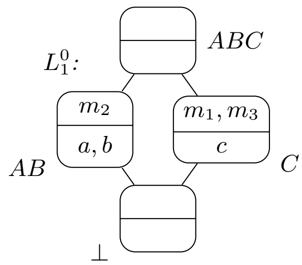
Concepts are partially ordered by $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle \Leftrightarrow A_1 \subseteq A_2$ or equivalently $B_2 \subseteq B_1$. With respect to this partial order, the set of all formal concepts is a complete lattice called the concept lattice of $\langle G, M, I\rangle$.

Example 1 (Formal concept analysis)**.** *Starting from a formal context* $K_1^0 = \langle G_1, M_1^0, I_1^0\rangle$ *with* $G_1 = \{a, b, c\}$*,* $M_1^0 = \{m_1, m_2, m_3\}$ *and* I_1^0 *as the incidence relation whose table is given below, the application of* FCA *results in the lattice made of the concepts* ABC*,* AB*,* C *and* $\bot$ *as:*

FCA(

K_1^0	m_1	m_2	m_3
a		×	
b		×	
c	×		×

) =



Concepts are named after their extent as throughout the report.

Formal concept analysis can be considered as a function that associates to a formal context $\langle G, M, I\rangle$ its concept lattice $\langle C, \leq\rangle = \text{FCA}(\langle G, M, I\rangle)$ This is illustrated by Example 1. By abuse of language, when a variable L denotes a concept lattice $\langle C, \leq\rangle$, L will also be used to denote C.

Given the finite set of objects G_x from which each lattice is built, the set of concepts that can be created from such contexts is finite and moreover each concept can be identified by its extent. Hence, $N(\langle G, M, I\rangle) = 2^G$ is the set of all concept names that may be used in any such concept lattice[1]. We will identify the concepts by such sets, but display their names as uppercase character strings; the extent of a so-named concept will be the set of objects in its name. In any specific lattice L, the subset $N(L)$ of $N(K)$ is the set of names of concepts in this lattice according to this convention as illustrated in Example 2.

[1]A similar remark is made in [Wajnberg, 2020, §4.1.2].

Example 2 (Concept names). *Consider the context K_1^0 of Example 1. The set of objects of K_1^0 being $G_1 = \{a, b, c\}$, the set of concept names that can be created for this in any concept lattice is $N(K_1^0) = \{ABC, AB, AC, BC, A, B, C, \bot\}$. In the specific lattice obtained in Example 1, the set of concept names is $N(L_1^0) = \{ABC, AB, C, \bot\}$.*

2.2 Extending formal concept analysis with scaling

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

2.2.1 Scaling: a generalisation

Scaling is one type of extension of type (a). It is a way to encode a more complex structure Σ into FCA. For that purpose, a scaling operation $\varsigma : \mathscr{X} \rightarrow 2^D$ generates Boolean attributes named after a language D from a structure $\Sigma \in \mathscr{X}$. $\mathscr{X}$ is left unspecified at that stage. In scaled contexts, the language D can be interpreted so that the incidence relation I is immediately derived from the attribute m following:

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

In FCA, $D = M$ and I is provided by its matrix:

$$I \models m(g) \text{ iff } \langle m, g \rangle \in I$$

Hence, adding attributes M' to a context under such a structure Σ consists of adding the attributes and extending the incidence relation according to this interpretation. It may be performed as:

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

and suppressing them as:

$$\mathrm{K}^{\Sigma}_{-M'}(\langle G, M, I \rangle) = \langle G, M \setminus M', I \setminus \{\langle g, m \rangle \in G \times M'\}\rangle$$

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

$$\sigma_{\varsigma}(K, \Sigma) = \mathrm{K}^{\Sigma}_{+\varsigma(\Sigma)}(K)$$

This unified view of scaling may be applied to many available scaling operations. We discuss these below.

name	language (D)	scale	Σ	condition ($m(g)$)
FCA	m	-	I	$\langle m, g\rangle \in I$
dichotomic	$n = v$	monocolumn	$\langle G, N, W, J\rangle$	$n(g) = v$
nominal	$n = w$	diagonal	$\langle G, N, W, J\rangle$	$n(g) = w \ \forall w \in W$
ordinal	$n \leq w$	triangular	$\langle G, N, W, J\rangle$	$n(g) \leq w \ \forall w \in W$
inter-ordinal	$n \leq w$, $n \geq w$	-	$\langle G, N, W, J\rangle$	$n(g) \leq w$ or $n(g) \geq w \ \forall w \in W$
contranominal	$n \neq w$	antidiagonal	$\langle G, N, W, J\rangle$	$n(g) \neq w \ \forall w \in W$

Table 2: Conceptual scaling operations (inspired from [Ganter and Wille, 1999]).

2.2.2 Conceptual scaling

Attributes found in data sets typically do not range in Booleans, but instead in numbers, intervals, strings, etc. Such data can be represented as a many-valued context $\Sigma = \langle G, M, W, J\rangle$, where G is a set of objects, M a set of attributes, W a set of values, and J a ternary relation defined on $G \times M \times W$. $\langle g, m, w\rangle \in J$ or simply $m(g) = w$ means that object g takes the value w for the attribute m. In addition, when $\langle g, m, w\rangle \in J$ and $\langle g, m, v\rangle \in J$ then $w = v$ [Ganter and Wille, 1999, §1.3]: in FCA, "many-valued" means that the range of an attribute may include more than two values, but for any object, the attribute can only have one of these values.

Conceptual scaling transforms such a many-valued context into a one-valued context. For instance, for nominal scaling, given a set W of values and a set N of properties taking these values, $D^{=}_{N,W} = \{n = w | n \in N \text{ and } w \in W\}$ splits the ranges of the multi-valued attributes in N into binary attributes. Attributes of $D^{=}_{N,W}$ are interpreted as:

$$\langle G, N, W, J\rangle \models gI(n = w) \text{ iff } \langle g, n, w\rangle \in J$$

There are other types of scalings and some of them are detailed in Table 2. The same can be built for ordinal scaling, e.g.

$$\langle G, N, W, J\rangle \models gI(n \ \leq w) \text{ iff } \langle g, n, v\rangle \in J \wedge v \leq w$$

These scaling operations only use a simple structure, i.e. $\Sigma = \langle G, N, W, J\rangle$ in which everything is stored in J and the attributes are expressed as predicates, e.g. $\cdot = v$ for nominal scaling or $\cdot \leq n$ for ordinal scaling. $\models$ is the evaluation of the predicate for the value, hence they can be called structural scaling.

Example 3 (Conceptual scaling). *Consider a many-valued context $\langle G, N, W, J\rangle$ with $G = \{$Alice, Bob, Carol$\}$, $N = \{$age, shoesize$\}$, $W = \mathbb{N}$ and J as below:*

J	*age*	*shoesize*
Alice	*12*	*32*
Bob	*14*	*36*
Carol	*14*	*34*

I	*age*$= 12$	*age*$= 14$	*shoesize*< 35
Alice	×		×
Bob		×	
Carol		×	×

It is possible to scale age *with nominal scaling and* shoesize *with (partial) ordinal scaling generating the formal context with* G*,* $M = \{\text{age}{=}12, \text{age}{=}14, \text{shoesize}{<}35\}$ *and* I *as above.*

2.2.3 Relational scaling

Relational scaling operations (σ_ς) considered in [Rouane-Hacene et al., 2013a] are based on a struture $\Sigma = \langle R, L\rangle$ made of a set of relations $R = \{r_y\}_{y\in Y}$ between two sets of objects, i.e. relations $r_y \subseteq G_x \times G_z$, and a family $L = \{L_x\}_{x\in X}$ of concept lattices whose extent is a subset of G_x (we note $R_{x,z} = \{r \in R | r \subseteq G_x \times G_z\}$).

For example, qualified existential scaling ($\exists$) adds attributes $\exists r.c$ for $r \in R_{x,z}$ and $c \in L_z$ whose satisfaction ($\models$) is defined by:

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

Example 4 (Relational scaling)**.** *Consider that the relation* q *is given by the table:*

q	a	b	c
d	×		
e		×	
f			×

and consider that the concept lattice L_1^0 *of Example 1 is the lattice corresponding to the context* K_1^0*. The concept in* L_1^0 *are* $N_1^0 = \{ABC, AB, C, \bot\}$ *(Example 2) hence scaling by* $\sigma_\exists$ *will only provide the attribute set* $\{\exists q.ABC, \exists q.AB, \exists q.C, \exists q.\bot\}$*. The description of the relation* q *allows to uncover the incidence relation for these (the incidence relation for* $\bot$ *is always empty so never displayed):*

	$\exists q.ABC$	$\exists q.AB$	$\exists q.C$
d	×	×	
e	×	×	
f	×		×

If, in addition, K_2^0 *had two other properties* n_1 *and* n_2 *whose incidence is given as* I_2^0 *displayed below, then the scaling operation would correspond to:*

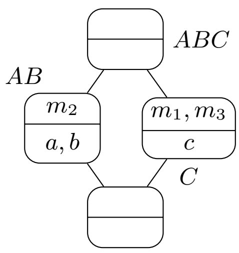
$\sigma_\exists($

K_2^0	n_1	n_2
d	×	
e	×	×
f		

,

q	a	b	c
d	×		
e		×	
f			×

,



$) =$

K_2^1	n_1	n_2	$\exists q.ABC$	$\exists q.AB$	$\exists q.C$
d	×		×	×	
e	×	×	×	×	
f			×		×

In principle, scaling is evaluated against $\Sigma = \langle R, L\rangle$. However, L is used in relational scaling to determine which objects belong a concept in L. Since this is available directly from the concept *names*, this is simplified by using $\Sigma = \langle R, N(L)\rangle$ instead. The set of attributes that can be scaled from ς on r of codomain G_z is $D_{\varsigma,r,N} = \{\varsigma r.c \mid c \in N\}$ with $N \subseteq N(K_z)$. This notation can be generalised so that, given a set of scaling operations Ω, a set of relations R over sets of objects in different contexts whose concept names are identified by $N = \{N_x\}_{x\in X}$ with $\forall x \in X$, $N_x \subseteq N(K_x)$:

$$D^x_{\Omega,R,N} = \bigcup_{\varsigma\in\Omega} \bigcup_{z\in X} \bigcup_{r\in R_{x,z}} D_{\varsigma,r,N_z}$$

Example 5 (Set of attributes). *In the context K^0_2 of Example 4, if there is only one relation q, whose codomain is K^0_1 of Example 1, and the existential scaling operator $\exists$ ($\Omega = \{\exists\}$), then the set of possible scalable attributes is:*

$$D^2_{\{\exists\},\{q\},\{N(K^0_1)\}} = \{\exists q.ABC, \exists q.AB, \exists q.AC, \exists q.BC, \exists q.A, \exists q.B, \exists q.C, \exists q.\bot\}$$

But using only those concepts from the lattice L^0_1 obtained in Example 1, this set is reduced to:

$$D^2_{\{\exists\},\{q\},\{N(L^0_1)\}} = \{\exists q.ABC, \exists q.AB, \exists q.C, \exists q.\bot\}$$

If K^0_2, whose objects are $\{d, e, f\}$, is also linked with relation s to itself (K^0_2) and the current set of concept names is $N(L^0_2) = \{DEF, DE, E\}$, then there would additionally scale the attributes: $D^2_{\{\exists\},\{s\},\{N(L^0_2)\}} = \{\exists s.DEF, \exists s.DE, \exists s.E\}$. If, in addition, the strict contains scaling operation ($\forall\exists C.r$, see Table 3) is used, then new properties would be:

$$\begin{aligned} D^2_{\{\exists,\forall\exists\},\{q,s\},\{N(L^0_1),N(L^0_2)\}} = \{&\exists q.ABC, \exists q.AB, \exists q.C, \exists q.\bot, \exists s.DEF, \exists s.DE, \exists s.E,\\ &\forall\exists ABC.q, \forall\exists AB.q, \forall\exists C.q, \forall\exists\bot.q,\\ &\forall\exists DEF.s, \forall\exists DE.s, \forall\exists E.s\} \end{aligned}$$

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 [Baader et al., 2003] (see Table 3). The set of attributes obtained from relational scaling may be large but remains finite. Cardinality constraints may entail infinite sets of concepts in theory, but in practice the set of meaningful concepts are bounded by $|G_z|$ which is finite.

In fact, RCA may be considered as a very general way to apply scaling across contexts. New operators may be provided [Braud et al., 2018; Wajnberg, 2020], such as those that we used for extracting link keys [Atencia et al., 2020].

2.2.4 Logical scaling

Logical scaling [Prediger, 1997] 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 queries (Q) and $\models$ is entailment or query evaluation. The scaling can be rewritten as:

$$\Sigma \models gIQ \text{ iff } \Sigma \models Q(g)$$

name	language (D)	Σ	condition ($m(g)$)
existential	$\exists r$	R	$r(g) \neq \varnothing$
universal (wide)	$\forall r.C$	R, L	$r(g) \subseteq extent(C)$
strict universal	$\forall\exists r.C$	R, L	$r(g) \neq \varnothing \wedge r(g) \subseteq extent(C)$
contains (wide)	$\forall C.r$	R, L	$extent(C) \subseteq r(g)$
strict contains	$\forall\exists C.r$	R, L	$extent(C) \neq \varnothing$ $\wedge extent(C) \subseteq r(g)$
qualified existential	$\exists r.C$	R, L	$r(g) \cap extent(C) \neq \varnothing$
qualified min cardinality	$\leq_n r.C$	R, L	$\lvert r(g) \cap extent(C)\rvert \leq n$
qualified max cardinality	$\geq_n r.C$	R, L	$\lvert r(g) \cap extent(C)\rvert \geq n$
$\forall$-condition	$\forall\langle r, r'\rangle_k$	$R \times R', L_{C\times C'}$	$r(g) =_k r'(g')$
$\exists$-condition	$\exists\langle r, r'\rangle_k$	$R \times R', L_{C\times C'}$	$r(g) \cap_k r'(g') \neq \varnothing$

Table 3: Relational scaling operations (inspired from [Braud et al., 2018; Rouane-Hacene et al., 2013a]) and additional link key condition scaling operations operators [Atencia et al., 2020].

Here, the nearly identical notation shows the relevance of this generalisation. We expressed it with respect to one individual g, so it applies to unary queries or formulas with one variable placeholder. However, it is possible to generalise this to contexts in which individuals in G are elements of the products of sets of individuals.

2.2.5 Relational scaling as logical scaling

The type of scaling used by RCA, relational scaling, can be though of as an extension of logical scaling based on description logic.

Relational scaling is based on a set of contexts $\{\langle G_x, M_x, I_x\rangle\}_{x\in X}$, the corresponding lattices $\{L_x\}_{x\in X} = \{\text{FCA}(\langle G_x, M_x, I_x\rangle)\}_{x\in X}$ and a set of relations $R = \{r_y\}_{y\in Y}$. This input can be encoded as sets of description logic axioms by:

$$
\begin{aligned}
|K_x| &= \{m(g) \mid m \in M_x \wedge g \in G_x \wedge gI_xm\} \\
|r_y| &= \{r_y(g, g') \mid g \in G_x \wedge g' \in G_z \wedge \langle g, g'\rangle \in r_y\} \\
\|L_x\| &= \{c \equiv \sqcap_{d\in intent(c)} d \mid c \in L_x\}
\end{aligned}
$$

The elements in $|\cdot|$ are part of an ABox and those in $\|\cdot\|$ are part of a TBox. They may be combined into a description logic knowledge base $\Sigma = \langle T_\Sigma, A_\Sigma\rangle$ such that:

$$
\begin{aligned}
T_\Sigma &= \bigcup_{x\in X} \|L_x\| \\
A_\Sigma &= \bigcup_{x\in X} |K_x| \cup \bigcup_{y\in Y} |r_y|
\end{aligned}
$$

In principle, it should be possible to deduce the extent of each concept and the order between concepts from this:

$$\Sigma \models c(g) \text{ iff } g \in \mathit{extent}(c)$$
$$\Sigma \models c \sqsubseteq c' \text{ iff } c \leq c'$$

The attributes provided by relational scaling are description logic concept descriptions, i.e. unary predicates. They can be interpreted with respect to the knowledge base associated to Σ:

$$\Sigma \models gI\ \forall\exists\ p.c \text{ iff } \Sigma \models (\forall p.c \sqcap \exists p)(g)$$

This way of interpreting relational scaling opens the door to introducing arbitrary description logic axioms within Σ and thus to use background knowledge.

2.3 Other extensions

There are other extensions providing formal concept analysis with more expressiveness without scaling (type *b* extension). Instead of scaling, they change the structure of the set of properties, staying within the scope of Galois lattices. We mention them here briefly.

Logical concept analysis Logical concept analysis is an extension of formal concept analysis in which the set of attributes is replaced by logical formulas attached to objects [Ferré and Ridoux, 2000].

$\langle 2^M, \subseteq, \cap, \cup\rangle$ is replaced by $\langle L, \models, \vee, \wedge\rangle$ in which L is a set of logic formulas. The extension is defined semantically, hence the formula may be thought of a the class of equivalent formulas. It could be redefined by using closed sets of formulas instead of single formulas and closed union instead of conjunction.

The formal context is $\langle G, L, i\rangle$ with $i : G \rightarrow L$ a mapping. In this case, the two operators $\cdot^{\uparrow}$ and $\cdot^{\downarrow}$ define a Galois connection between $\langle 2^G, \subseteq\rangle$ and $\langle L, \models\rangle$ with $O \subseteq G$ and $\phi \in L$:

$$O^{\uparrow} = \bigvee_{o \in O} i(o)$$
$$\phi^{\downarrow} = \{o \in G \mid i(o) \models \phi\}$$

As for scaling, it is possible to rewrite:

$$\langle L, \models, \vee, \wedge\rangle \models oI\phi \text{ iff } i(o) \models \phi$$

One benefit of plunging the logic in FCA is the definition of contextualised entailment[2] as:

$$\phi \models^K \psi \text{ iff } \phi^{\downarrow} \subseteq \psi^{\downarrow}$$

This extension is very "structural": concepts are carrying theories which do not apply to the objects of the concepts.

[2]Contextualized deduction in [Ferré and Ridoux, 2000].

Generalised formal concept analysis Generalised formal concept analysis [Chaudron and Maille, 2000] attaches to each object o an element $T = \zeta(o)$ of a lattice $\langle \mathscr{L}, \sqsubseteq, \sqcap, \sqcup \rangle$ which replaces $\langle 2^M, \subseteq, \cap, \cup \rangle$ in formal concept analysis. Hence a general context is a triple $\langle G, \mathscr{L}, \zeta \rangle$, such that $\zeta : G \to \mathscr{L}$. The incidence relation is implicitly defined by:

$$\langle \mathscr{L}, \sqsubseteq, \sqcap, \sqcup \rangle \models gIT \text{ iff } \zeta(g) \sqsupseteq T$$

so that the two operators $\cdot^{\uparrow}$ and $\cdot^{\downarrow}$ define a Galois connection between $\langle 2^G, \subseteq \rangle$ and $\langle \mathscr{L}, \sqsubseteq \rangle$ with $O \subseteq G$ and $T \in \mathscr{L}$:

$$\begin{aligned} O^{\uparrow} &= \sqcap_{g \in O} \zeta(g) \\ T^{\downarrow} &= \{g \in G \mid T \sqsubseteq \zeta(g)\} \end{aligned}$$

This theory has been instantiated to the unification of existentially conjunction of first order atoms, represented as sets of atoms. $\sqsubseteq$ is syntactic subsumption, i.e. the fact that there exists a variable substitution on the subsumee such that it is included in the subsumer. In such a case, $\mathscr{L}$ is the set of such formulas reduced to a non redundant form and $\sqcap$ is antiunification. Because of the use of unification, this approach remains syntactic.

Pattern Structures It is also possible to avoid scaling and to directly work on complex data, using the formalism of "pattern structures" [Ganter and Kuznetsov, 2001; Kaytoue et al., 2011]. Pattern structures generalise FCA in a similar way as logical concept analysis. In this case, 2^M is replaced by elements of a meet-semilattice $\langle D, \sqcap \rangle$ [Ganter and Kuznetsov, 2001; Kuznetsov, 2009]. The formal context is now $\langle G, D, \delta \rangle$ with $\delta : G \to D$ a mapping.

In this case, the two operators $\cdot^{\uparrow}$ and $\cdot^{\downarrow}$ define a Galois connection between $\langle 2^G, \subseteq \rangle$ and $\langle D, \sqsubseteq \rangle$ with $A \subseteq G$ and $d \in D$:

$$\begin{aligned} A^{\uparrow} &= \sqcap_{g \in A} \delta(g) \\ d^{\downarrow} &= \{g \in G \mid d \sqsubseteq \delta(g)\} \end{aligned}$$

such that $c \sqsubseteq d \equiv c \sqcap d = c$.

This requires to define: (a) how to order its elements ($\sqsubseteq$), and (b) how to test that an object satisfies an attribute expression (gId for $d \in D$). This can be rewritten as for scaling:

$$\langle D, \sqsubseteq \rangle \models gId \text{ iff } d \sqsubseteq \delta(g)$$

Pattern structures [Ganter and Kuznetsov, 2001; Kuznetsov, 2009] provide a more structured attribute language without scaling.

However, these extensions are not directly affected by the problem of context dependencies considered here as the attributes do not refer to concepts.

Relational extensions On the contrary, other approaches [Ferré and Cellier, 2020; Kötters, 2013] 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 [Ferré and Cellier, 2020] the incidence relation is a hypergraph between objects, and produce a more expressive representation. A comparison of RCA and Graph-FCA is provided in [Keip et al., 2020]. Graph-FCA adopts a different approach from RCA but should, in principle, suffer from the same problem as the one considered here as soon as it contains circular dependencies: intents would need to refer to concepts so created, i.e. named subqueries. This remains to be studied.

Terminological base extraction Finally, description logic base mining [Baader and Distel, 2008; Guimarães et al., 2023] and relational concept analysis share the same purpose: inferring a TBox from an ABox (taken as an interpretation). However, RCA does this by introducing new named concepts based on FCA, where description logic base mining does not introduce new names but uses new concept descriptions inspired from Duquesnes-Guigne implication bases [Guigues and Duquenne, 1986]. Where, in Example 4 relational scaling would use attribute $\exists q.AB$, base mining would use the description $\exists q.\exists m_1.\top$. As soon as cycles occur in context dependencies, this naturally leads to cyclic concept definitions. This has been interpreted with the greatest fixed-point semantics in $\mathcal{EL}_{gfp}$. However, led by complexity considerations, work has focused on extracting minimal bases in $\mathcal{EL}$ through unravelling [Baader and Distel, 2008; Guimarães et al., 2023]. The problem raised in this paper is different but applies as well to description logic base mining as soon as it is taken as a knowledge induction task from data: circular dependencies may lead to different, equally well-behaving, bases that would be worth taking in consideration.

2.4 A very short introduction to RCA

Relational Concept Analysis (RCA) [Rouane-Hacene et al., 2013a] 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 [Baader et al., 2003], using specific scaling operations. It may also be though of as a general way to deal with circular references using different scaling operations.

2.4.1 Operations

RCA applies to a relational context[3] $\langle K^0, R\rangle$, composed of a family of formal contexts $K^0 = \{\langle G_x, M_x^0, I_x^0\rangle\}_{x\in X}$ indexed by a set X, and a set of binary relations $R = \{r_y\}_{y\in Y}$.

[3] We use the term 'relational context' instead of 'relational context family' reserving families to those sets indexed by X.

A relation $r_y \subseteq R_{x,z}$ 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 $K^t_x \in K^t$ and all relations $r_y \in R_{x,z}$ from the set of concepts in corresponding $L^t_z = \text{FCA}(K^t_z)$.

The classical RCA algorithm, that is called here RCA, thus relies on FCA and σ_ς. More precisely, it applies these in parallel on all contexts. Hence, FCA* and σ^*_Ω are defined as:

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

$$\sigma^*_\Omega(\{\langle G_x, M_x, I_x\rangle\}_{x\in X}, R, \{L_x\}_{x\in X}) = \left\{ \bigoplus_{\substack{\varsigma\in\Omega \\ r_y\in R_{x,z}}} \sigma_\varsigma(\langle G_x, M_x, I_x\rangle, r_y, L_z) \right\}_{x\in X}$$

such that $\oplus_{\varsigma\in\Omega, r_y\in R_{x,z}}$ scales, with all operations in Ω, the given context with all the relations starting from x (to any z).

2.4.2 Algorithm

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

$$K^{t+1} = \sigma^*_\Omega(K^t, R, \text{FCA}^*(K^t))$$

until reaching a fixed point, i.e. reaching n such that $K^{n+1} = K^n$. Then, $\underline{\text{RCA}}_\Omega(K^0, R) = \text{FCA}^*(K^n)$.

Thus, the RCA algorithm proceeds in the following way:

1. Initial formal contexts: $\{\langle G_x, M^0_x, I^0_x\rangle\}_{x\in X} \leftarrow \{\langle G_x, M_x, I_x\rangle\}_{x\in X}$.
2. $\{L^t_x\}_{x\in X} \leftarrow \text{FCA}^*(\{\langle G_x, M^t_x, I^t_x\rangle\}_{x\in X})$ (or, for each formal context, $\langle G_x, M^t_x, I^t_x\rangle$ the corresponding concept lattice $L^t_x = \text{FCA}(\langle G_x, M^t_x, I^t_x\rangle)$ is created using FCA).
3. $\{\langle G_x, M^{t+1}_x, I^{t+1}_x\rangle\}_{x\in X} \leftarrow \sigma^*_\Omega(\{\langle G_x, M^t_x, I^t_x\rangle\}_{x\in X}, R, \{L^t_x\}_{x\in X})$ (i.e. relational scaling is applied, for each relation r_y whose codomain lattice has new concepts, generating new contexts $\langle G_x, M^{t+1}_x, I^{t+1}_x\rangle$ including both plain and relational attributes in M^{t+1}_x).
4. If $\exists x \in X$ such that $M^{t+1}_x \neq M^t_x$ (scaling has occurred), go to Step 2.
5. Return: $\{L^t_x\}_{x\in X}$.

This is illustrated by Example 6.

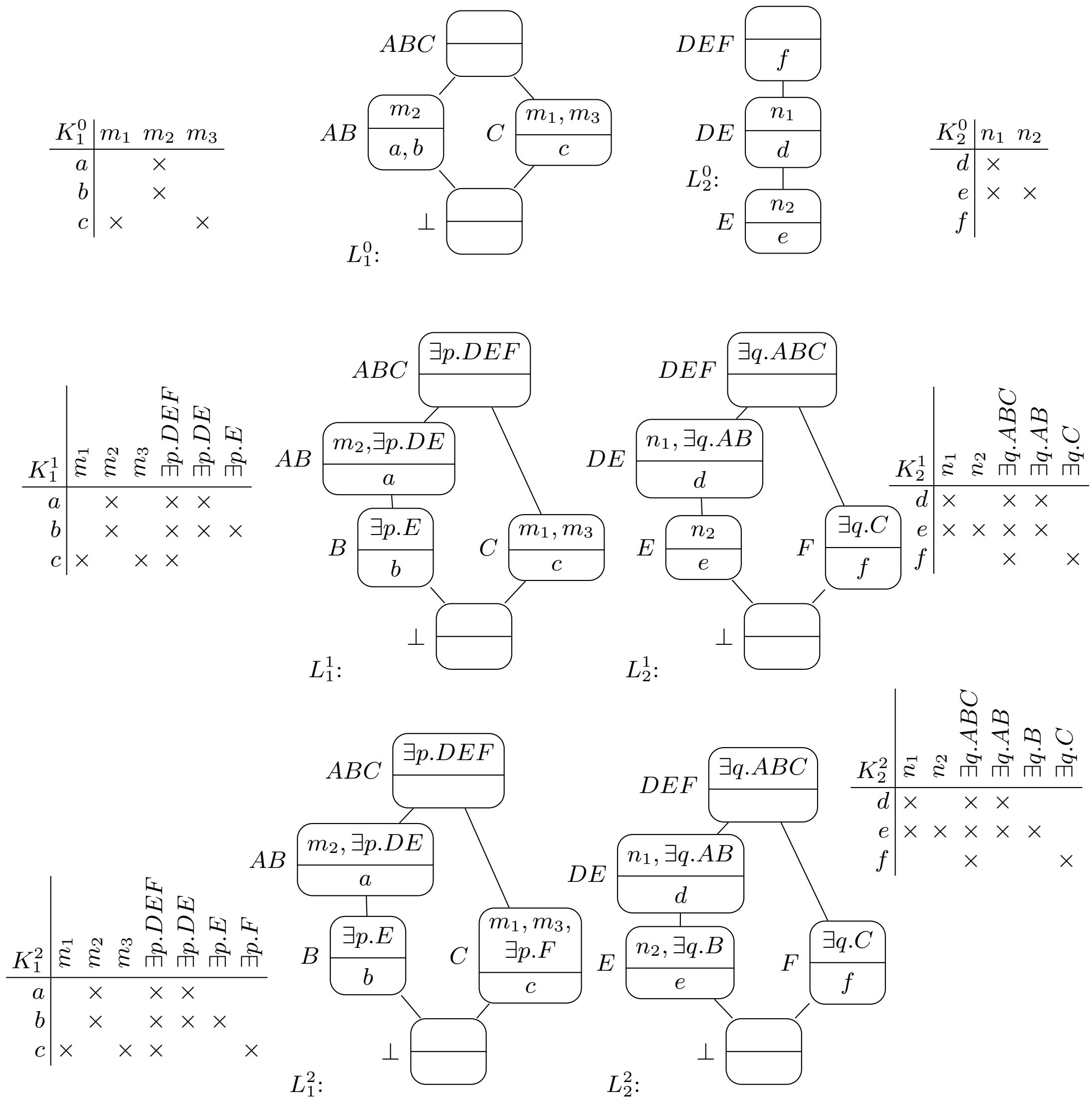
Example 6 (Relational concept analysis)**.** *Consider two relations p and q defined as:*

p	d	e	f
a	×		
b		×	
c			×

q	a	b	c
d	×		
e		×	
f			×

and applying on the contexts K^0_1 of Example 1 and K^0_2 of Example 4 (Figure 1).

Applying FCA* *to the two contexts K^0_1 and K^0_2, provides the very simple lattices L^0_1 and L^0_2 of Figure 1 with concepts ABC, AB, C and $\bot$, and DEF, DE and E,*



K_1^0	m_1	m_2	m_3
a		×	
b		×	
c	×		×

K_2^0	n_1	n_2
d	×	
e	×	×
f		

K_1^1	m_1	m_2	m_3	$\exists p.DEF$	$\exists p.DE$	$\exists p.E$
a		×		×	×	
b		×		×	×	×
c	×		×	×		

K_2^1	n_1	n_2	$\exists q.ABC$	$\exists q.AB$	$\exists q.C$
d	×		×	×	
e	×	×	×	×	
f			×		×

K_1^2	m_1	m_2	m_3	$\exists p.DEF$	$\exists p.DE$	$\exists p.E$	$\exists p.F$
a		×		×	×		
b		×		×	×	×	
c	×		×	×			×

K_2^2	n_1	n_2	$\exists q.ABC$	$\exists q.AB$	$\exists q.B$	$\exists q.C$
d	×		×	×		
e	×	×	×	×	×	
f			×			×

Figure 1: The three iterations of RCA from the initial contexts K_1^0 and K_2^0.

respectively. Applying scaling, as seen partially in Example 4, provides the contexts K_1^1 *and* K_2^1 *with new attributes* $\exists p.DEF$, $\exists p.DE$, $\exists p.E$ *and* $\exists q.ABC$, $\exists q.AB$, $\exists q.C$*. Applying* FCA* *to these contexts provides the lattices* L_1^1 *and* L_2^1 *with additional concepts* B *and* F*. These can in turn go through scaling and unveil new attributes* $\exists p.F$ *and* $\exists q.B$ *added to* K_1^1 *and* K_2^1 *to give* K_2^1 *and* K_2^2*.* FCA* *introduces the new attributes in the intent of existing concepts but does not introduce any new concept. Hence the process stops and* RCA *returns the family of concept lattices* $\{L_1^2, L_2^2\}$*.*

The result is thus quite different from the $\{L_1^0, L_2^0\}$ *that would have been returned by FCA alone.*

These operations are in general interpreted as generating a description logic T-box from a given A-box. This can be seen in Example 7.

Example 7 (Relational concept analysis and description logics)**.** *As an example, consider the following ABox:*

$$A^0_{12} = \{\top_1(a), \top_1(b), \top_1(c), m_1(c), m_2(a), m_2(b), m_3(c), p(a,d), p(b,e), p(c,f),\\ \top_2(d), \top_2(e), \top_2(f), n_1(d), n_1(e), n_2(e), q(d,a), q(e,b), q(f,c)\}$$

This can be encoded as the two formal context K^0_1 and K^0_2 (Figure 1) and the two relations p and q between these of Example 6.

From this, RCA generates the lattices L^2_1 and L^2_2 (Figure 1) which can be interpreted as the description logic T-boxes:

$$T^2_{12} = \{ABC \sqsubseteq \top_1 \sqcap \exists p.DEF, AB \sqsubseteq ABC \sqcap m_2 \sqcap \exists p.DE, B \sqsubseteq AB \sqcap \exists p.E,\\ C \sqsubseteq ABC \sqcap m_1 \sqcap m_2 \sqcap \exists p.F, AB \sqcap C \sqsubseteq \bot, DEF \sqsubseteq \top_2 \sqcap \exists q.ABC,\\ DE \sqsubseteq DEF \sqcap n_1 \sqcap \exists q.AB, E \sqsubseteq DE \sqcap n_2 \sqcap \exists q.B, F \sqsubseteq DEF \sqcap \exists q.C,\\ DE \sqcap F \sqsubseteq \bot\}$$

with the improved A-box:

$$A^2_{12} = \{AB(a), B(b), C(c), p(a,d), p(b,e), p(c,f),\\ DE(d), E(e), F(f), q(d,a), q(e,b), q(f,c)\}$$

2.4.3 Properties and semantics

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' (see Property 1, p. 27). 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 $\forall x \in X$, $M_x \subseteq M'_x$.

RCA always reaches a closed formal context family for reason of finiteness [Rouane-Hacene et al., 2013a] and the sequence $(K^t)^n_{t=0}$ is non-(intent-)contracting, i.e. $\forall t \geq 0, K^t \subseteq K^{t+1}$ [Rouane-Hacene et al., 2013b].

The RCA semantics characterises the set of concepts in resulting RCA lattices as all and only those grounded on the initial context family (K^0) based on relations (R) [Rouane-Hacene et al., 2013b]. It thus can be considered as a well-grounded semantics: an attribute is scaled and applied to an object at iteration $t+1$ only if its condition applies at stage t. Hence, everything is ultimately relying on K^0.

[Rouane-Hacene et al., 2013b] established 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 so-grounded concepts are in K^n).

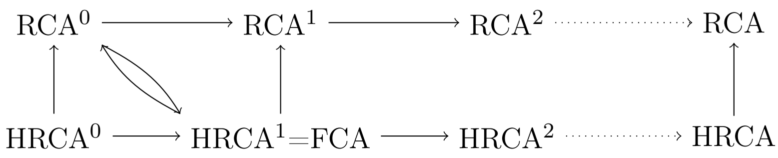


Figure 2: Relation between different restrictions of RCA (arrows mean: "can be rewritten into").

2.5 Dependencies and cycles

As can be seen, relations in RCA define a dependency graph between objects (of different or the same context). In turn, this graph of objects induces a dependency graph between concepts through the scaled attributes that refer to other concepts. It also induces a dependency graph between contexts: an edge exists between two contexts if one object of the former is related to one object of the latter.

This report is related to the circular dependencies, i.e. the circuits, that may exist within these graphs. We say that a set of relations R is *hierarchical* if its object dependency graph is not circular.

Circular dependencies create a problem when one wants to define the family of concept lattices that should be returned by relational concept analysis. As will be seen in Section 3, there may exists several such families

In order to explain some specific phenomena in a clearer way, it is possible to study them in various restrictions of RCA. We introduced some of these that are organised in Figure 2:

- RCA^n=RCA: RCA with n formal contexts
- RCA^1: RCA with one formal context
- RCA^0: RCA with one *empty* formal context
- HRCA: RCA with only hierarchical relations.
- HRCA^1=FCA: FCA
- FCA^0=HRCA^0: FCA with empty contexts is clearly not interesting

We first study the semantics of RCA within RCA^0, a special case of RCA. It is restricted in two ways:

- It contains only one formal context ($|X| = 1$),
- which has no attribute ($M^0_x = \varnothing$).

Additionally, we consider in the examples below only one single scaling operation: qualified existential scaling ($\Omega = \{\exists\}$). The results of the report are independent from this choice in examples.

Because RCA^0 is a restriction of RCA, we will use the same notation as defined above, thought it operates on simpler structures. Although RCA^0 seems very simple[4], FCA can be encoded into RCA^0. Indeed, given a formal context $\langle G, M, I\rangle$, for each

[4] An anonymous ICFCA 2021 reviewer complements the remarks of §2.3 noting that RCA^0 is also very related to Graph-FCA as they both have only one context and using existential scaling.

attribute $m \in M$ in the formal context, a relation $R_m \subseteq G \times G$ can be created such that $\forall g \in G, \langle g, g \rangle \in R_m$ if and only if gIm. Starting with $K^0 = \langle G, \varnothing, \varnothing \rangle$, it can be checked that $\sigma^*_\exists(K^0, R, \text{FCA}^*(K^0))$ will simply add to K^0 one attribute $\exists r_m.\top$ per $m \in M$ which exactly corresponds to m.

It is also possible to encode RCA^0 into FCA using the following trick: Given an RCA^0 relational context $\langle \{\langle G, \varnothing, \varnothing \rangle\}, \{R_p \subseteq G \times G\}_{p \in P} \rangle$, it can be encoded in a single FCA context:

- G remains the same;
- $M = \{p^o | p \in P \wedge o \in G\}$;
- $o' I p^o$ iff $\langle o', o \rangle \in R_p$.

As a result, all the information from the relational context has been preserved and FCA will return a result analogous to $\text{RCA}^0_{\{\exists\}}$.

Introducing RCA^0 is sufficient to hint at the problems and solutions that we want to illustrate, as will now be presented.

3 Examples

In order to illustrate the weakness of the RCA semantics, we first carry on the introductory Examples 1, 4–6 (§3.1). We then display it on more minimal examples that will be carried over the report: the minimal example used in [Euzenat, 2021] for RCA^0 (§3.2) and a somewhat equivalent example for RCA in general, i.e. involving more than one formal context (§3.3).

From such a simple basis, it is possible to consider more complex settings:

- By using more than two contexts;
- By using more than two relations between these contexts;
- By using more than two objects in each context;
- By using more than zero properties in the contexts.

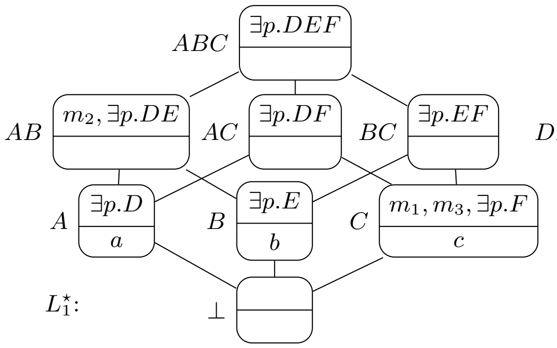
3.1 RCA may accept different concept lattice families

The simple Example 6 does not present a result in which each object is identified by a single class. Indeed, a has not more attributes than b. This result could also be obtained with far more objects a', a'', etc. sharing the attributes of a and b, or duplicating other objects.

However, the lattices $L_1^\star$ and $L_2^\star$ displayed in Figure 3 seem another good way to describe the data. Indeed, they are also valid concept lattices whose contexts extend K_1^0 and K_2^0 (Example 6). We will temporarily informally consider them as *acceptable*.

$K_1^\star$	m_1	m_2	$m3$	$\exists p.DEF$	$\exists p.DE$	$\exists p.DF$	$\exists p.EF$	$\exists p.D$	$\exists p.E$	$\exists p.F$
a		×		×	×	×		×		
b		×		×	×		×		×	
c	×		×	×		×	×			×

$K_2^\star$	n_1	n_2	$\exists q.ABC$	$\exists q.AB$	$\exists q.AC$	$\exists q.BC$	$\exists q.A$	$\exists q.B$	$\exists q.C$
d	×		×	×	×		×		
e	×	×	×	×		×		×	
f			×		×	×			×



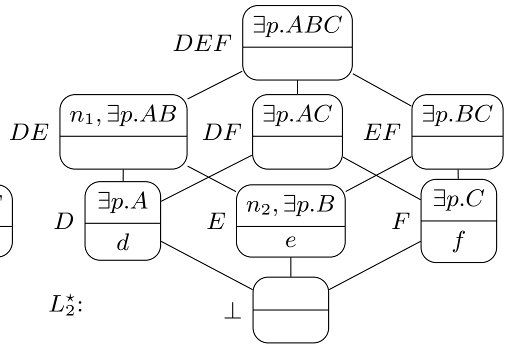


Figure 3: Alternative concept lattices for the example of Section 3.1 ($\star$ is simply a way to identify these objects).

The two corresponding T-Box and A-box would be:

$$\begin{aligned}
T_{12}^\star = \{&ABC \sqsubseteq \top_1 \sqcap \exists p.DEF, AB \sqsubseteq ABC \sqcap m_2 \sqcap \exists p.DE, B \sqsubseteq AB \sqcap \exists p.E,\\
&C \sqsubseteq ABC \sqcap m_1 \sqcap m_2 \sqcap \exists p.F, AB \sqcap C \sqsubseteq \bot, DEF \sqsubseteq \top_2 \sqcap \exists q.ABC,\\
&DE \sqsubseteq DEF \sqcap n_1 \sqcap \exists q.AB, E \sqsubseteq DE \sqcap n_2 \sqcap \exists q.B, F \sqsubseteq DEF \sqcap \exists q.C,\\
&DE \sqcap F \sqsubseteq \bot\}\\
A_{12}^\star = \{&A(a), B(b), C(c), p(a,d), p(b,e), p(c,f),\\
&D(d), E(e), F(f), q(d,a), q(e,b), q(f,c)\}
\end{aligned}$$

To be considered an acceptable solution, a family of concept lattices must have the following properties:

- They contain all attributes of K^0 and only extra attributes from $D_{\Omega,R,N(K^0)}$, i.e. that can be scaled from K^0 through R and Ω (well-formed).
- No additional attribute can be scaled from them (saturated).
- They only refer to concepts in each other (self-supported).

This is related to the two properties considered in Section 2.4.3: the notion of saturation is the same as completeness. Correctness however combines well-formedness and self-support as requiring that the support has to be found exclusively in K^0.

Hence the question: why the family of lattices of Figure 3 is not returned by RCA and what does RCA actually returns?

To help answering the question, we characterise it below in two minimal running examples.

3.2 Minimal RCA^0 example

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

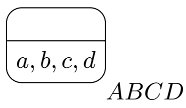
$$A_0^0 = \{\top_0(a), \top_0(b), \top_0(c), \top_0(d), r(a,b), r(b,a), r(c,d), r(d,c), r(a,a), r(b,b)\}$$

It can be encoded as an empty formal context (K_0^0) from which FCA will generate the concept lattice L_0^0 as follows:

FCA(

K_0^0
a
b
c
d

) = L_0^0:



Scaling with $\sigma_\exists$ and r provides the attribute $\exists r.ABCD$:

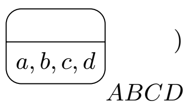
$\sigma_\exists$(

K_0^0
a
b
c
d

,

r	a	b	c	d
a	×	×		
b	×	×		
c				×
d			×	

, L_0^0:



) =

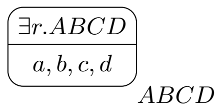
K_0^1	$\exists r.ABCD$
a	×
b	×
c	×
d	×

which run through FCA returns:

FCA(

K_0^1	$\exists r.ABCD$
a	×
b	×
c	×
d	×

) = L_0^1:



Since no new concept has been added, scaling would return K_0^1, hence L_0^1 is the result returned by RCA^0 (and RCA).

However, the concept lattices L_0' and $L_0^\star$ of Figure 4 are other valid lattices worth considering as acceptable solutions. As in classical RCA, each concept of these lattices is closed with respect to the specific formal context scaled by $\exists$ and r from the concepts of the lattice. Moreover, the lattices are self-supported in the sense that their attributes refer only to their concepts.

They correspond to different knowledge bases:

$$T_0^1 = \{ABCD \sqsubseteq \exists r.ABCD\}$$

$$A_0^1 = \{ABCD(a), ABCD(b), ABCD(c), ABCD(d), r(a,b), r(b,a), r(c,d), r(d,c), r(a,a), r(b,b)\}$$

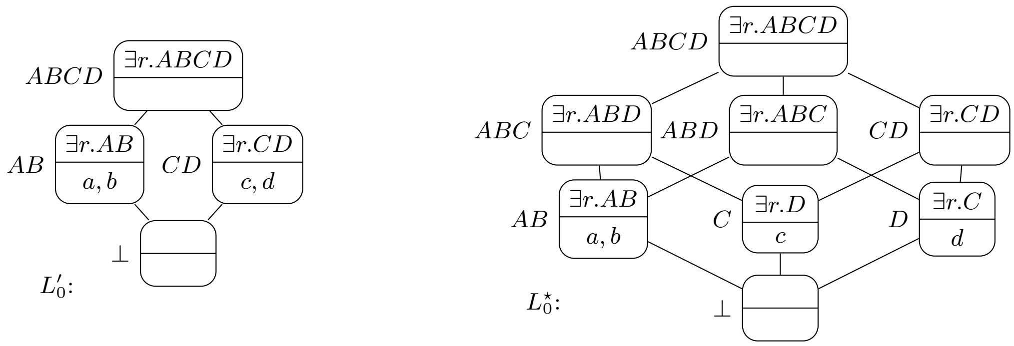


Figure 4: Alternative concept lattices (L'_0 and $L^\star_0$).

and

$$T'_0 = \{AB \sqsubseteq \top_0 \sqcap \exists r.AB, CD \sqsubseteq \top_0 \sqcap \exists r.CD, ABCD \sqsubseteq \exists r.ABCD\}$$
$$A'_0 = \{AB(a), AB(b), CD(c), CD(d), r(a,b), r(b,a), r(c,d), r(d,c), r(a,a), r(b,b)\}$$

and

$$\begin{aligned} T^\star_0 = \{&AB \sqsubseteq ABC \sqcap ABD \sqcap \exists r.AB, C \sqsubseteq ABC \sqcap CD \sqcap \exists r.D, D \sqsubseteq ABD \sqcap CD \sqcap \exists r.C,\\ &ABC \sqsubseteq ABCD \sqcap \exists r.ABD, ABD \sqsubseteq ABCD \sqcap \exists r.ABC, CD \sqsubseteq ABCD \sqcap \exists r.CD,\\ &ABCD \sqsubseteq \exists r.ABCD\} \end{aligned}$$
$$A^\star_0 = \{AB(a), AB(b), C(c), D(d), r(a,b), r(b,a), r(c,d), r(d,c), r(a,a), r(b,b)\}$$

The problem that there exists several acceptable candidate lattices applies to RCA as a whole because RCA^0 is included in RCA.

3.3 Minimal RCA example

As another example, consider the following ABox:

$A^0_{34} = \{\top_3(a), \top_3(b), \top_4(c), \top_4(d), p(a,c), p(b,d), q(c,a), q(b,d)\}$

This can be encoded as the two empty formal contexts K^0_3 and K^0_4 of Figure 6 and the two relations p and q of Figure 5.

p	c	d
a	×	
b		×

q	a	b
c	×	
d		×

Figure 5: Relations p and q for RCA.

Applying FCA to the two contexts K^0_3 and K^0_4 provides the very simple lattices L^0_3 and L^0_4 of Figure 6. From this, RCA generates new context K^1_3 and K^1_4 through scaling which provides new lattices L^1_3 and L^1_4 (Figure 6).

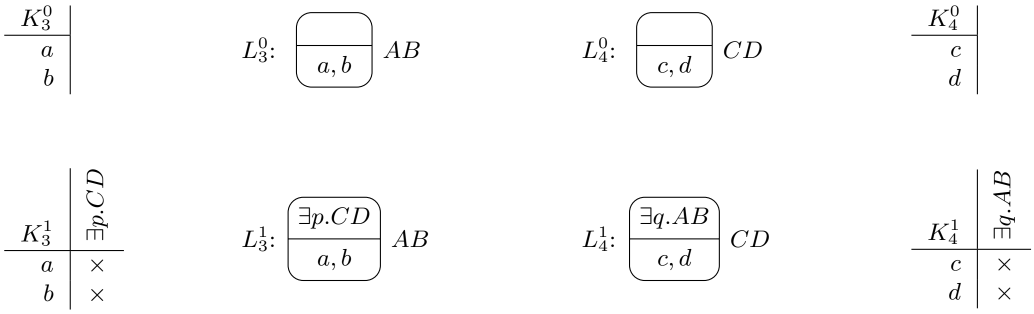


Figure 6: The two iterations of RCA from the initial contexts K^0_3 and K^0_4.

The lattices L^1_3 and L^1_4 of Figure 6 are those returned by RCA as applying scaling from them returns the same contexts K^1_3 and K^1_4.

However, there could be other acceptable solutions such as those displayed in Figure 7. They are all valid concept lattices whose contexts can be easily shown to extend the context K^0_3 and K^0_4 of Figure 6.

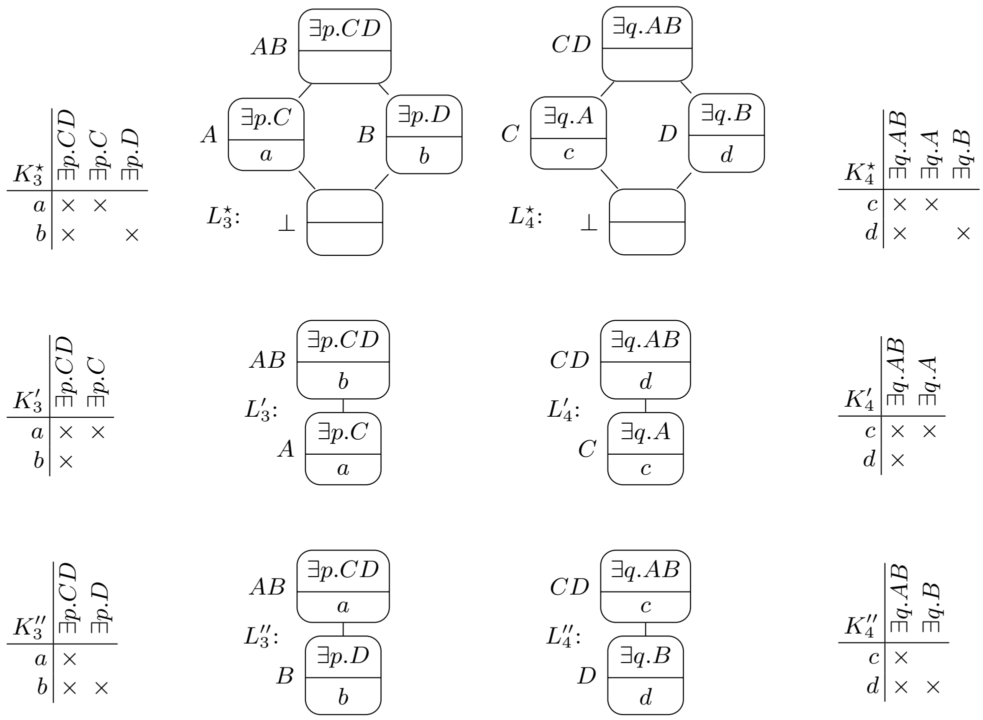


Figure 7: Alternative pairs of concept lattices covering the contexts of Figure 6.

On the contrary, Figure 8 displays a family of lattices $\{L^{\#}_3, L^{\#}_4\}$ which is not an acceptable solution. Although they contain all concepts of $\{L^0_3, L^0_4\}$ and no concept not in $\{L^\star_3, L^\star_4\}$, they would generate more attributes through scaling and applying RCA to their contexts $\{K_3^{\#}, K_4^{\#}\}$ would lead to $\{L_3^{\star}, L_4^{\star}\}$

Figure 8: A family of concept lattices $\{L_3^{\#}, L_4^{\#}\}$ which is not an acceptable solution.

Hence the question: Why does RCA return only one solution, and which one? Answering it requires to reconsider the RCA semantics. More precisely, it requires to define formally which families of concept lattices could be considered as acceptable solutions and which of them is returned by the RCA operator.

The following aims at defining acceptability formally.

4 A parallel context-lattice fixed-point semantics for RCA^0

In order to investigate the semantics of relational concept analysis, we adopt a functional standpoint in which RCA is defined as a function in a precisely defined space. In Sections 4.1 and 4.2, we provide two alternative, and equivalent, characterisations of that space, which provide the semantics for RCA^0. In Section 4.3, we relate the well-grounded RCA semantics to these two semantics by showing that RCA computes the least fixed point of these functions.

These results are further extended, by introducing self-supported lattices (Section 5) and combining them with saturated lattices (Section 6), to determine the set of acceptable concept lattices for RCA^0.

Although the results of this section aims at RCA^0, most of it applies to RCA, hence we use RCA^0 only when it matters.

Warning: a nest of fixed points RCA is a world of fixed points, hence it is easy to get lost among the various fixed points involved:

- In description logics, which RCA targets, the semantics of concepts is given by (least) fixed points when circularities occur [Nebel, 1990];
- FCA's goal is to compute fixed points: concepts are the result of a closure operator which is also a fixed point [Belohlávek, 2008];
- finally, the RCA result is the fixed point of the function that grows a lattice family from the previous one through scaling.

The present work is concerned with the fixed points of the latter function taking the others into account.

4.1 Semantics and properties: the context approach

We first study the semantics of RCA from the standpoint of the formal contexts. We define precisely the space of contexts in which RCA is computed and the functions underlying RCA in that space.

4.1.1 The lattice $\mathscr{K}$ of RCA^0 contexts

The contexts considered by RCA are formal context families scaled from the initial context using the scaling operations. They are determined by three elements given once and for all: $K^0 = \{\langle G_x, M_x^0, I_x^0\rangle\}_{x\in X}$, $R = \{r_y\}_{y\in Y}$, and Ω. This is even more specific for RCA^0 with $K^0 = \langle G, \varnothing, \varnothing\rangle$, but for most of this section we will ignore it.

Through these operations, only M_x^t and I_x^t change, the latter depending directly from the former (Property 1).

Property 1 (The incidence relation depends only on the relations)**.** *For a relational scaling operation ς and a relation $r \in R_{x,z}$, an attribute $m \in D_{\varsigma,r,N(K_z)}$ determines the incidence on objects of G_x.*

Proof. m is scaled from a scaling operation ς, a relation r and a concept C (possibly a cardinal n). From Table 3, it only depends on ς, r and the extent of C. However, ς and r are the same in all situations, they are not interpreted contextually. The concept C is identified by a name which denotes its extent. Hence, its extent does not depend on the context either. So whether an object of G_x satisfies this attribute or not depends solely on the attribute. □

The attribute language $D_{\Omega,R,N}$ that can be generated by scaling depends on the finite set of relations R, the scaling operations Ω and the set of possible concepts identified by their standardised names (§2.2.3). Given a set N of concepts that can be the codomain of relations in R, the set of contexts that can be obtained by scaling is

$$\mathscr{K}^N_{K^0,R,\Omega} = \{\mathrm{K}_{+M}^{\langle R,N(K^0)\rangle}(K^0) \mid M \subseteq D_{\Omega,R,N}\}$$

with $\mathrm{K}_{+M}^{\langle R,N(K^0)\rangle}(.)$ the operation defined in §2.2. For RCA, $N(K^0) = \bigcup_{x\in X} N(K_x^0)$ is the set of all concept names induced from all formal contexts in K^0. Similarly, for a set of indexed concept lattices $L = \{L_x\}_{x\in X}$, $N(L) = \bigcup_{x\in X} N(L_x)$.

In RCA^0, the set of class names is $N = N(K^0)$. Hence the attribute language $D_{\Omega,R,N(K^0)}$ is fully determined by the non-changing parts: G_x, the finite set of relations R and the scaling operations Ω.

$$\mathscr{K}_{K^0,R,\Omega} = \mathscr{K}^{N(K^0)}_{K^0,R,\Omega}$$

Below, when we write $\mathscr{K}^N$, the property applies for any $N \subseteq N(K^0)$, when we write $\mathscr{K}$ it holds for RCA^0, i.e. $N = N(K^0)$.

The contexts may be combined by meet and join:

Definition 1 (Meet and join of contexts)**.** *Given* $K, K' \in \mathscr{K}^N_{\langle G,M^0,I^0\rangle,R,\Omega}$ *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:*

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

The set of contexts is closed by meet and join.

Property 2 ([Euzenat, 2021])**.** $\forall K, K' \in \mathscr{K}^N_{K^0,R,\Omega}$*,* $K \wedge K' \in \mathscr{K}^N_{K^0,R,\Omega}$ *and* $K \vee K' \in \mathscr{K}^N_{K^0,R,\Omega}$*.*

Proof. Meet and join are defined from the union and intersection of subsets of $D_{\Omega,R,N(K^0)}$ (Definition 1). But $\mathscr{K}^N_{K^0,R,\Omega}$ is closed by union and intersection and the incidence relation is fully determined by the set of attributes (Property 1). Hence, meet and join of contexts in $\mathscr{K}^N_{K^0,R,\Omega}$ belong to $\mathscr{K}^N_{K^0,R,\Omega}$. □

Property 3 (Commutativity, associativity and absorption of $\vee$ and $\wedge$ on $\mathscr{K}$)**.** *For all* $K, K', K'' \in \mathscr{K}$*,*

$$K \vee K' = K' \vee K \quad \textit{and} \quad K \wedge K' = K' \wedge K \quad \text{(commutativity)}$$
$$(K \vee K') \vee K'' = K \vee (K' \vee K'') \quad \textit{and} \quad (K \wedge K') \wedge K'' = K \wedge (K' \wedge K'') \quad \text{(associativity)}$$
$$K \wedge (K \vee K') = K \quad \textit{and} \quad K \vee (K \wedge K') = K \quad \text{(absorption)}$$

Proof. Proofs are given for $\wedge$, those for $\vee$ follow the exact same pattern.

$$\begin{aligned}
K \wedge K' &= \langle G, M, I\rangle \wedge \langle G, M', I'\rangle \\
&= \langle G, M^0 \cup (M \cap M'), I^0 \cup (I \cap I')\rangle && \text{Definition 1} \\
&= \langle G, M^0 \cup (M' \cap M), I^0 \cup (I' \cap I)\rangle && \text{Commutativity of } \cap \\
&= \langle G, M', I'\rangle \wedge \langle G, M, I\rangle && \text{Definition 1} \\
&= K' \wedge K \\
(K \wedge K') \wedge K'' &= (\langle G, M, I\rangle \wedge \langle G, M', I'\rangle) \wedge \langle G, M'', I''\rangle \\
&= (\langle G, M^0 \cup (M \cap M'), I^0 \cup (I \cap I')\rangle) \\
&\quad \wedge \langle G, M'', I''\rangle) && \text{Definition 1} \\
&= \langle G, M^0 \cup ((M \cap M') \cap M''), I^0 \cup ((I \cap I') \cap I'')\rangle && \text{Definition 1} \\
&= \langle G, M^0 \cup (M \cap (M' \cap M'')), I^0 \cup (I \cap (I' \cap I''))\rangle && \text{Associativity of } \cap \\
&= \langle G, M^0 \cup M, I^0 \cup I\rangle \\
&\quad \wedge \langle G, M^0 \cup (M' \cap M''), I^0 \cup (I' \cap I'')\rangle && \text{Definition 1} \\
&= \langle G, M^0 \cup M, I^0 \cup I\rangle \\
&\quad \wedge (\langle G, M^0 \cup M', I^0 \cup I'\rangle \wedge \langle G, M^0 \cup M'', I^0 \cup I''\rangle) && \text{Definition 1} \\
&= K \wedge (K' \wedge K'') && \text{Definition 1}
\end{aligned}$$

$$
\begin{aligned}
K \vee (K \wedge K') &= \langle G, M, I\rangle \vee (\langle\langle G, M, I\rangle) \wedge \langle G, M', I'\rangle) \\
&= \langle G, M, I\rangle \vee \langle G, M^0 \cup (M \cap M'), I^0 \cup (I \cap I')\rangle && \text{Definition 1} \\
&= \langle G, M^0 \cup M \cup (M \cap M'), I^0 \cup I \cup (I \cap I')\rangle && \text{Definition 1} \\
&= \langle G, M^0 \cup M, I^0 \cup I\rangle && \text{Absorption of } \cup/\cap \\
&= \langle G, M, I\rangle \\
&= K && \square
\end{aligned}
$$

These operations are aligned with context inclusion (Property 4):

Property 4 ([Euzenat, 2021]). $\forall K, K' \in \mathscr{K}^N_{K^0,R,\Omega}, K \subseteq K'$ *iff* $K = K \wedge K'$

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

This provides the space of context with a complete lattice structure (Property 5):

Property 5 ([Euzenat, 2021]). $\langle \mathscr{K}^N_{K^0,R,\Omega}, \vee, \wedge\rangle$ *is a complete lattice.*

Proof. $\mathscr{K}^N_{K^0,R,\Omega}$ is closed by meet and join (Property 2). $\vee$ and $\wedge$ satisfy commutativity, associativity and the absorption laws (Property 3), so this is a lattice. It is complete because finite. $\square$

4.1.2 The context expansion function F

We reformulate RCA as based on a main single function, $F_{K^0,R,\Omega}$, the context expansion function[5] attached to a relational context $\langle K^0, R\rangle$ and a set Ω of scaling operations.

Definition 2 (Context expansion function [Euzenat, 2021]). *Given a relational context $\langle K^0, R\rangle$ and a set of relational scaling operations Ω, the function $F_{K^0,R,\Omega} : \mathscr{K}_{K^0,R,\Omega} \to \mathscr{K}_{K^0,R,\Omega}$ is defined by:*

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

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. $F_{K^0,R,\Omega}$ is only defined over $\mathscr{K}_{K^0,R,\Omega}$ because scaling is not restricted to an arbitrary N. From now on, we will abbreviate $\mathscr{K}_{K^0,R,\Omega}$ as $\mathscr{K}$ 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 $\mathscr{K}$:

Property 6 (F is internal to $\mathscr{K}$ [Euzenat, 2021]). $\forall K \in \mathscr{K}$*,* $F(K) \in \mathscr{K}$

Proof. Scaling only adds attributes from $D_{\Omega,R,N(K^0)}$. $\square$

[5]Named "complete relational extension" in [Rouane-Hacene et al., 2013b].

Property 7 (F is extensive and monotone [Euzenat, 2021])**.** *The function F, attached to a relational context and a set of scaling operations, satisfies:*

$$K \subseteq F(K) \qquad \text{(extensivity)}$$
$$K \subseteq K' \Rightarrow F(K) \subseteq F(K') \qquad \text{(monotony)}$$

Proof. extensivity holds because F can only add to each formal context in K new attributes scaled from $\text{FCA}(K)$. 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 $\text{FCA}(K)$ is included in that of $\text{FCA}(K')$, 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 $F(K) \subseteq F(K')$. □

Extensivity corresponds to the non-contracting property of the well-grounded semantics [Rouane-Hacene et al., 2013b] and monotony is also called order-preservation.

4.1.3 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 3 (fixed point [Euzenat, 2021])**.** *A formal context $K \in \mathscr{K}$ 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 $\mathscr{K}^N$ is a complete lattice and F is order-preserving (or monotone) on $\mathscr{K}$, then the Knaster-Tarski theorem applies:

Theorem 8 (Knaster-Tarski theorem [Tarski, 1955])**.** *Let $\mathscr{K}$ be a complete lattice and let $F : \mathscr{K} \to \mathscr{K}$ be an order-preserving function. Then the set of fixed points of F in $\mathscr{K}$ is also a complete lattice.*

In particular, this warrants that there exists least and greatest fixed points of F in $\mathscr{K}$ (called $\text{lfp}(F)$ and $\text{gfp}(F)$) which can be defined as:

$$\text{lfp}(F) = \bigwedge_{K \in \text{fp}(F)} K \text{ and } \text{gfp}(F) = \bigvee_{K \in \text{fp}(F)} K$$

4.2 Semantics and properties: the lattice approach

In formal concept analysis, there is a one-to-one correspondence between contexts and lattices. Hence the results of the previous section could in principle be derived through reasoning on lattices instead of contexts. In this section, we approach RCA from the lattice standpoint and we show, unsurprisingly, the close parallel with the context approach.

4.2.1 The lattice $\mathscr{L}$ of RCA^0 concept lattices

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

$$\mathscr{L}^N_{K^0,R,\Omega} = \{\text{FCA}(K^{\langle R,N(K^0)\rangle}_{+M}(K^0) \mid M \subseteq D_{\Omega,R,N}\}$$

In RCA^0, this time again the set of concept names is limited to those of the single context, $N(K^0)$:

$$\mathscr{L}_{K^0,R,\Omega} = \mathscr{L}^{N(K^0)}_{K^0,R,\Omega}$$

For each subset, the lattice obtained by FCA is necessarily syntactically different as its concepts refer to different attributes in their intents (at least one of them).

There is in fact a bijective correspondence between $\mathscr{L}_{K^0,R,\Omega}$ and $\mathscr{K}_{K^0,R,\Omega}$. On the one hand, for any formal context in $\mathscr{K}_{K^0,R,\Omega}$ corresponds only one lattice by FCA. On the other hand, and for any finite concept lattice in $\mathscr{L}_{K^0,R,\Omega}$ there exists an implicit context extraction function κ [Ganter and Wille, 1999, §1.2]: $\mathscr{L}_{K^0,R,\Omega} \to \mathscr{K}_{K^0,R,\Omega}$ such that FCA $\circ\, \kappa = Id_{\mathscr{K}}$ and $\kappa \circ$ FCA $= Id_{\mathscr{L}}$. This may be stated as:

Property 9. $K = \kappa(L)$ *iff* $L = \text{FCA}(K)$

Proof. It is necessary to prove that $K = \kappa(L)$ iff $L = \text{FCA}(K)$ or otherwise, that (a) $K = \kappa(\text{FCA}(K))$ and (b) $L = \text{FCA}(\kappa(L))$.

For (a), κ collects only those attributes which are in the intent of concepts in FCA(K). However, these have been created by using exclusively attributes in K, hence $K \supseteq \kappa(\text{FCA}(K))$. Moreover, the bottom concept in FCA, covers all attributes of the context (K), thus $K \subseteq \kappa(\text{FCA}(K))$. This makes that $K = \kappa(\text{FCA}(K))$.

For (b), it is sufficient to apply FCA to both sides of this last equation: $\text{FCA}(K) = \text{FCA}(\kappa(\text{FCA}(K)))$. But each lattice $L \in \mathscr{L}^N$ is obtained by applying FCA to a context $K \in \mathscr{K}^N$, hence $L = \text{FCA}(K)$. Thus, the equation can be rewritten $L = \text{FCA}(\kappa(L))$. □

$\kappa(L)$ can be induced by collecting the attributes present in L intents to build the unique M, from which the corresponding I is obtained.

It is directly generalised as

$$\kappa^*(\{L_x\}_{x\in X}) = \{\kappa(L_x)\}_{x\in X}$$

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 4 (Lattice homomorphism [Euzenat, 2021])**.** *A concept lattice homomorphism $h : \langle C, \leq\rangle \to \langle C', \leq'\rangle$ is a function which maps each concept $c \in C$ into a corresponding concept $h(c) \in C'$ such that:*

– *$\forall c \in C$, intent(c) $\subseteq$ intent($h(c)$) (or intent(c) $\sqsupseteq$ intent($h(c)$) if these are considered as description logic concept descriptions),*

– $\forall c \in C$, *extent*$(c) =$ *extent*$(h(c))$, *and*
– $\forall c, d \in C$, $c \leq d \Rightarrow h(c) \leq' h(c)$.

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 simply $=$. This owns to the fact that the homomorphism maps concepts of equal extent, hence, if they hold in both ways, there should be as many concepts in each lattice and these concepts will also have the same intent.

Property 10 (FCA is monotonous)**.** $K \subseteq K' \Rightarrow \text{FCA}(K) \preceq \text{FCA}(K')$

Proof. If $K \subseteq K'$, then each concept that can be built from K can be built from K'. The additional attributes in $M' \setminus M$ can only be used to separate further objects of existing concepts, introducing additional concepts. All concepts are preserved, possibly with a larger intent which preserve the homomorphism (Definition 4). Hence, $\text{FCA}(K) \preceq \text{FCA}(K')$. □

We can define $\wedge$ and $\vee$ on $\mathscr{L}^N_{K^0,R,\Omega}$.

Definition 5 (Meet and join of lattices)**.** *Given* $L, L' \in \mathscr{L}^N_{\langle G,M^0,I^0\rangle,R,\Omega}$,

$$L \vee L' = \text{FCA}(\kappa(L) \vee \kappa(L')) \quad \text{(join)}$$
$$L \wedge L' = \text{FCA}(\kappa(L) \wedge \kappa(L')) \quad \text{(meet)}$$

The set of lattices is also closed by meet and join:

Property 11. $\forall L, L' \in \mathscr{L}^N_{K^0,R,\Omega}$, $L \wedge L' \in \mathscr{L}^N_{K^0,R,\Omega}$ *and* $L \vee L' \in \mathscr{L}^N_{K^0,R,\Omega}$.

Proof. $\mathscr{L}^N_{K^0,R,\Omega}$ is closed by meet and join since $\mathscr{K}^N_{K^0,R,\Omega}$ is closed by meet and join (Property 2) and $\mathscr{L}^N_{K^0,R,\Omega}$ is the image of $\mathscr{K}^N_{K^0,R,\Omega}$ by FCA. □

Property 12. $\langle \mathscr{L}^N_{K^0,R,\Omega}, \vee, \wedge \rangle$ *is a complete lattice.*

Proof. $\mathscr{L}^N_{K^0,R,\Omega}$ is closed by meet and join (Property 11). $\vee$ and $\wedge$ satisfy commutativity, associativity and the absorption laws directly from the union and intersection on contexts (Property 3), so this is a lattice. It is complete because finite. □

Property 13. $\forall L, L' \in \mathscr{L}_{K^0,R,\Omega}, L \preceq L'$ *iff* $L = L \wedge L'$

Proof. First, for any $m \in D_{\Omega,R,N}$ belonging to both M and M', the pairs of the incidence relations I and I' for m are the same (Property 1).

⇒ $L \preceq L'$ means that $\forall c \in L$, $\exists h(c) \in L'$ such that *extent*$(c) =$ *extent*$(h(c))$ and *intent*$(c) \subseteq$ *intent*$(h(c))$. Given that $M = \bigcup_{c \in L}$ *intent*$(c) \setminus M^0$ and that $M' = \bigcup_{c \in L'}$ *intent*$(c) \setminus M^0$, then $M \subseteq M'$ (and $I \subseteq I'$ due to Property 1). Thus, $L = L \wedge L'$ because the contexts on which they are built (K^0_{+M} and $K^0_{+M \cap M'}$) are the same.

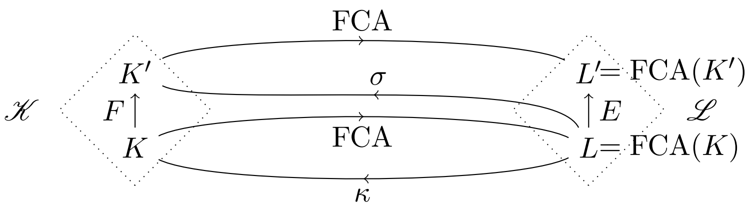


Figure 9: Relations between F and E through the alternation of FCA and σ_Ω (from [Euzenat, 2021]).

$\Leftarrow$ $L = L \wedge L'$ means that $M \subseteq M'$ (and then $I \subseteq I'$ according to Property 1). Hence, $\forall c \in L$, the attributes satisfied by $extent(h(c))$ in L' include those satisfied by $extent(c)$ in L and others belonging to $M' \setminus M$. Thus, $extent(h(c))$ is the extent of a concept in L' because it contains the only objects satisfying these attributes (it is closed). Consequently, $\exists h(c) \in L'$ such that $extent(c) = extent(h(c))$ and $intent(c) \subseteq intent(h(c))$ as $h(c)$ may satisfy additional attributes belonging to $M' \setminus M$, but it satisfies at least all those of $intent(c)$. So, $L \preceq L'$. □

4.2.2 The lattice expansion function E

As was done for contexts, it is possible to provide an expansion function for lattices. 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 operations.

Definition 6 (Lattice expansion function [Euzenat, 2021])**.** *Given a relational context $\langle K^0, R\rangle$ and a set of relational scaling operations Ω the function $E_{K^0,R,\Omega} : \mathscr{L}_{K^0,R,\Omega} \rightarrow \mathscr{L}_{K^0,R,\Omega}$ is defined by:*

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

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

The definition of E first applies scaling and then FCA, though F does the opposite. In consequence, E is the function corresponding to F in the sense that $E \circ \text{FCA} = \text{FCA} \circ F$ (see Figure 9).

E is an extensive and monotone internal operation for $\mathscr{L}$:

Property 14 (E is internal to $\mathscr{L}$)**.** *$\forall L \in \mathscr{L}$, $E(L) \in \mathscr{L}$*

Proof. Given $L \in \mathscr{L}$, $\kappa(L) \in \mathscr{K}$. $K = \sigma_\Omega(\kappa(L), R, L)$ adds attributes from $D_{\Omega,R,N(K^0)}$ to $\kappa(L)$, hence $K \in \mathscr{K}$. Consequently, $E(L) = \text{FCA}(K) \in \mathscr{L}$. □

Property 15 (E is monotone and extensive)**.** *The function E, attached to a relational context K^0, R and a set of scaling operations Ω, satisfies $\forall L, L' \in \mathscr{L}_{K^0,R,\Omega}$:*

$$L \preceq L' \Rightarrow E(L) \preceq E(L') \qquad \text{(monotony)}$$

$$L \preceq E(L) \qquad \text{(extensivity)}$$

Proof. monotony $L \preceq L'$ entails that all concepts of L are found in L' with a larger intent. Consequently, $N(L) \subseteq N(L')$ and $D_{\Omega,R,N(L)} \subseteq D_{\Omega,R,N(L')}$. This entails that $\sigma_\Omega(K, R, L')$ extends K with more attributes than $\sigma_\Omega(K, R, L)$. Hence $E(L) \preceq E(L')$ because $E(L)$ is the application of FCA to the same formal context, to which has been added attributes.

extensivity $L = \text{FCA}(K)$ for $K \in \mathscr{K}$, thus $K \subseteq \sigma_\Omega(K, R, L)$. $E(L) = \text{FCA}(\sigma_\Omega(K, R, L))$ so it will have at least all concepts generated by K (identified by extents) because σ only adds attributes, hence those allowing to generate a concept remain available and FCA can only generate more concepts. Thus, for each concept $c \in L$ there exists $h(c) \in E(L)$ (with $extent(c) = extent(h(c))$ and possibly with a larger intent, i.e. $intent(c) \subseteq intent(h(c))$), generated by the new scaled attrubutes. Hence, $L \preceq E(L)$. □

Monotony is also called order-preservation. It corresponds to the non-(intent-)contracting concept property of [Rouane-Hacene et al., 2013b].

4.2.3 Fixed points of E

Given E, it is possible to define its set of fixed points, i.e. the sets of concept lattices closed for E, as:

Definition 7 (fixed point)**.** *A concept lattice $L \in \mathscr{L}$ is a fixed point for a lattice expansion function E, if $E(L) \simeq L$. We call* fp(E) *the set of fixed points for E.*

We can define:

$$\text{lfp}(E) = \bigwedge_{L \in \text{fp}(E)} L \text{ and } \text{gfp}(E) = \bigvee_{L \in \text{fp}(E)} L$$

Since $\mathscr{L}^N$ is a complete lattice and E is order-preserving (or monotone) on $\mathscr{L}$, then we can apply the Knaster-Tarski theorem. This warrants that there exists least and greatest fixed points of E in $\mathscr{L}$.

4.3 Well-grounded and least fixed-point semantics

$\underline{\text{RCA}}$ may be redefined as

$$\underline{\text{RCA}}_\Omega(K^0, R) = \text{FCA}(F^\infty(K^0))$$

i.e. $\underline{\text{RCA}}$ iterates F from K^0 until reaching a fixed point, and ultimately applies FCA. Alternatively, $\underline{\text{RCA}}$ may be redefined as

$$\underline{\text{RCA}}_\Omega(K^0, R) = E^\infty(\text{FCA}(K^0))$$

i.e. $\underline{\text{RCA}}$ iterates E from FCA(K^0) until reaching a fixed point.

It seems thus that $\underline{\text{RCA}}$ returns a fixed point of E. Hence the question: which fixed point is returned by RCA's well-grounded semantics?

4.3.1 The RCA well-grounded semantics is the least fixed-point semantics

Since K^0 belongs to $\mathscr{K}$ and FCA(K^0) belongs to $\mathscr{L}$, then RCA is indeed based on E and F fixed points. These are the least fixed points.

Proposition 16 (The RCA algorithm on a RCA^0 context computes the least fixed point [Euzenat, 2021])**.** *Given F the context expansion function and E the lattice expansion function associated to K^0, R and Ω,*

$$\underline{\text{RCA}}_\Omega(K^0, R) = \text{FCA}(\text{lfp}(F_{K^0,R,\Omega}))$$

and

$$\underline{\text{RCA}}_\Omega(K^0, R) = \text{lfp}(E_{K^0,R,\Omega})$$

Proof. Concerning the first equation, $\underline{\text{RCA}}_\Omega(K^0, R) = \text{FCA}(F^n(K^0))$ for some n at which $F(F^n(K^0)) = F^n(K^0)$ [Rouane-Hacene et al., 2013a]. Let $K^\infty = F^n(K^0)$, $K^\infty \in \text{fp}(F)$ (Definition 3). $\forall K \in \text{fp}(F)$, $K \in \mathscr{K}$, thus $K^0 \subseteq K$ because all the contexts in $\mathscr{K}$ contain M^0. By monotony (Property 7), $K^\infty = F^n(K^0) \subseteq F^n(K) = K$, because K is a fixed point. Thus, K^∞ is a fixed point more specific than all fixed points: it is the least fixed point.

Concerning the second equation, $\underline{\text{RCA}}_\Omega(K^0, R) = E^n(K^0)$ for some n at which $E(E^n(K^0)) = E^n(K^0))$ [Rouane-Hacene et al., 2013a]. $E(K^0) \in \mathscr{L}$, hence (by Property 14), $E^n(K^0) \in \mathscr{L}$. Moreover, $E(E^n(K^0)) = E^n(K^0))$ thus $E^n(K^0)) \in \text{fp}(E)$. In addition, $\forall L \in \text{fp}(E)$, $E(K^0) \preceq L$ because the context from which L is created contains at least all attributes of K^0. But if $E^t(K^0) \preceq L$, then $E^{t+1}(K^0) \preceq L$ because by monotony (Property 15), $E^{t+1}(K^0) = E(E^t(K^0)) \preceq E(L)$ and E is idempotent on fixed points (by Definition 7). Thus, $\underline{\text{RCA}}_\Omega(K^0, R)$ is a fixed point more specific than all fixed points: it is the least fixed point. □

4.3.2 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 17 ([Euzenat, 2021])**.** $\text{gfp}(F_{K^0,R,\Omega}) = \text{K}^{\langle R,N(K^0)\rangle}_{+D_{\Omega,R,N(K^0)}}(K^0)$

Proof. This context is the greatest element of $\mathscr{K}$ as it contains all attributes of $D_{\Omega,R,N(K^0)}$. It is also a fixed point because F is extensive (Property 7) and internal (Property 6). □

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

This result is easy but very uncomfortable. The obtained lattice may contain many non-supported attributes as shown in Example 8. 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.

Example 8 (Greatest fixed point of F in RCA^0)**.** *In the example of Section 3.2, the attribute $\exists p.A$ belongs to $D_{\Omega,R,N(K^0)}$ though A does not belong to the maximal lattice $L_0^\star$, because it is not a closed concept for FCA. 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 (here A).*

On the contrary, there may be cases in which the greatest fixed point is the powerset lattice, i.e. in which all attributes are supported, and the least fixed point of F is directly FCA(K^0).

This problem is even more embarrassing if one wants to enumerate all acceptable solutions: many of the fixed points of E or F will feature such non-supported attributes.

5 Self-supported fixed points in RCA^0

In order to define acceptable solutions for RCA we introduce the notion of self-support. It specifies that a concept lattice is self-supported if the intents of its concepts only refer to concepts in this lattice. We describe a function Q which suppresses non supported attributes and whose closure yields self-supported lattices. We then identify the acceptable solutions as self-supported fixed points.

5.1 Self-supported lattices

Since both F and E are extensive functions, it is possible, starting from anywhere in $\mathscr{K}$ or $\mathscr{L}$, to consider attributes that do not refer to concepts and these attributes will be preserved. As a consequence, there may exist fixed points with these unwanted attributes and they are also found in the greatest fixed point. This is not the result that we expect: we need the results to be self-supported.

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 not be supported in a smaller lattice. This is a difficulty for enumerating these fixed points.

Example 9 (Non self-supported lattice in RCA^0)**.** *Figure 10 shows a context $K_0^{\#}$ and the associated concept lattice $L_0^{\#}$ that could be a solution for the example of Section 3.2 as it belongs to $\mathscr{L}_{\{\exists\},\{r\},K_0^0}$. However, the lattice is not self-supported because the concept AB uses the attribute $\exists r.B$ in $K_0^{\#}$ which refers to a concept (B) not present in $L_0^{\#}$.*

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

Definition 8 (Self-supported lattices [Euzenat, 2021])**.** *A concept lattice L is* self-supported *if $\forall c \in L, intent(c) \subseteq M^0 \cup D_{\Omega,R,N(L)}$.*

$K_0^{\#}$	$\exists r.B$	$\exists r.C$	$\exists r.AB$	$\exists r.ABD$
a	×		×	×
b	×		×	×
c				×
d		×		

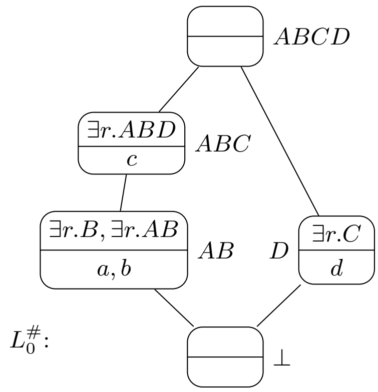


Figure 10: Non self-supported lattice $L_0^{\#}$.

The set of acceptable lattices that may be returned by RCA^0 can be circumbscribed as the self-supported fixed points of E. Such lattices are both saturated and self-supported well-formed elements of $\mathscr{L}$. Moreover, by construction of $\mathscr{K}$ and $\mathscr{L}$, they cover K^0, i.e. they contain all attributes in M^0.

This problem occurs in RCA^0 when the attributes only refer to the concepts of the lattice induced by this context [Euzenat, 2021]. However, as shown by Example 10, concerning RCA as a whole, the attributes are based on the lattices of other objects.

Example 10 (Non self-supported lattices in RCA)**.** *Figure 8 shows a familly of contexts* $\{K_3^{\#}, K_4^{\#}\}$ *and the associated family of concept lattices* $\{L_3^{\#}, L_4^{\#}\}$ *that could be a solution for the example of Section 3.3 as it belongs to* $\mathscr{O}_{\{\exists\},\{p,q\},\{K_3^0,K_4^0\}}$*. However, the lattices are not self-supported because the context* $K_3^{\#}$ *(and thus concept* A*) uses the attribute* $\exists p.C$ *which refers to a concept (*C*) not present in* $L_4^{\#}$ *and similarly for* $\exists q.B$ *in context* $K_4^{\#}$*.*

A lattice may then be supported by a family of lattices.

Definition 9 (Supported lattices)**.** *A concept lattice* L_z *is* supported *by a family of indexed lattices* $\{L_x\}_{x\in X}$ *in a relational context* $\langle\{\langle G_x, M_x^0, I_x^0\rangle\}_{x\in X}, R\rangle$ *if* $\forall c \in L_z$, $intent(c) \subseteq M_z^0 \cup D^z_{\Omega,R,\{N(L_x)\}_{x\in X}}$.

By extension, a family of indexed lattices is said self-supported if each lattice of the family is supported by the family.

The definition of self-supported lattices does not provide a direct way to transform a non self-supported lattice into a self-supported one. Simply suppressing non-supported attributes from intents could result in non concepts (with non-closed extents). 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 or purging function π which suppresses *from the induced context* ($\kappa(L)$) those attributes non supported *by the lattice*:

Definition 10 (Purging function [Euzenat, 2021]). *The function* $\pi_{K^0,R,\Omega} : \mathscr{L}^N \to \mathscr{K}^N$ *returns the context reduced to those attributes present in a lattice:*

$$\pi_{K^0,R,\Omega}(L) = \mathrm{K}^{\langle R,N\rangle}_{-D_{\Omega,R,N\setminus N(L)}}(\kappa(L))$$

The purging function and the following ones are defined over any N, as they only restrict the sets of possible contexts and do not expand them.

The purging function, like the scaling function, is only one step: it suppresses currently unsupported attributes, but this may lead to less concepts to be generated by FCA, and thus other non supported attributes. π and σ are not inverse functions: in particular, σ greatly depends on Ω and R to decide which attributes to scale, through π simply suppresses attributes non supported by the lattice(s), independently from Ω, which however determines the attribute language.

This can be generalised for RCA. As for σ_Ω and FCA, it is possible to introduce π (generalised to π^*):

$$\pi_{K^0,R,\Omega}(L_z, \{L_x\}_{x\in X}) = \mathrm{K}^{\langle R,\{L_x\}_{x\in X}\rangle}_{-D^z_{\Omega,R,N(K^0)\setminus N(\{L_x\}_{x\in X})}}(\kappa(L_z))$$
$$\pi^*_{K^0,R,\Omega}(\{L_x\}_{x\in X}) = \{\pi_{K^0,R,\Omega}(L_x, \{L_z\}_{z\in X})\}_{x\in X}$$

5.2 Contraction functions Q and P

Instead of dealing with expansion functions, it is possible to consider contraction functions for contexts or lattices based on π.

Definition 11 (Lattice contraction function). *The lattice contraction function* $Q : \mathscr{L}^N \to \mathscr{L}^N$ *is defined by*

$$Q(L) = \mathrm{FCA}(\pi_{K^0,R,\Omega}(L))$$

As previously, we will abbreviate $Q_{K^0,R,\Omega}$ as Q and $\pi_{K^0,R,\Omega}$ as π. Q is internal to the space of lattices.

Property 18 (Q is internal to $\mathscr{L}$). $\forall L \in \mathscr{L}^N$, $Q(L) \in \mathscr{L}^N$

Proof. $Q(L) = \mathrm{FCA}(\pi(L))$. $\pi(L)$ retracts attributes from $\kappa(L)$. By definition, $\kappa(L) \in \mathscr{K}^N$, hence $\pi(L) \in \mathscr{K}^N$. π never suppresses attributes from M^0 which are all supported (they do not depend on the existence of specific concepts in L). Consequently, $Q(L) = \mathrm{FCA}(\pi(L)) \in \mathscr{L}^N$. □

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

Property 19 (Q is anti-extensive and monotone [Euzenat, 2021]). *The function Q satisfies:*

$$Q(L) \preceq L \qquad \text{(anti-extensivity)}$$
$$L \preceq L' \Rightarrow Q(L) \preceq Q(L') \qquad \text{(monotony)}$$

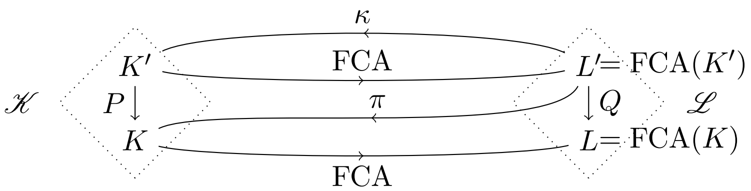


Figure 11: Relations between P and Q through the alternation of FCA and π (from [Euzenat, 2021]).

Proof. anti-extensivity $\pi(L) \subseteq \kappa(L)$ because π simply suppresses attributes from $\kappa(L)$. Hence, $\text{FCA}(\pi(L)) \preceq \text{FCA}(\kappa(L))$ because the latter contain all concepts of the former (identified by extent) possibly featuring the removed attributes (Property 10). Moreover, $\text{FCA}(\kappa(L)) = L$ by definition, thus $Q(L) = \text{FCA}(\pi(L)) \preceq \text{FCA}(\kappa(L)) = L$.

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 $N \setminus N(L) \supseteq N \setminus N(L')$ which entails $D_{\Omega,R,N\setminus N(L)} \supseteq D_{\Omega,R,N\setminus N(L')}$, which finally together leads to $M \setminus D_{\Omega,R,N\setminus N(L)} \subseteq M' \setminus D_{\Omega,R,N\setminus N(L')}$. Then, $\pi(L) \subseteq \pi(L')$ because a smaller context supported by a smaller lattice cannot result in a larger context. Hence, $Q(L) = \text{FCA}(\pi(L)) \preceq \text{FCA}(\pi(L')) = Q(L')$. □

Similarly on the context side, the P context contraction function may be introduced:

Definition 12 (Context contraction function)**.** *The context contraction function $P : \mathscr{K}^N \to \mathscr{K}^N$ is defined by*

$$P(K) = \pi_{K^0,R,\Omega}(\text{FCA}(K))$$

Figure 11 displays the relations between these two functions.

P is internal to $\mathscr{K}^N$:

Property 20 (P is internal to $\mathscr{K}$)**.** $\forall K \in \mathscr{K}^N$*,* $P(K) \in \mathscr{K}^N$

Proof. $P(K) = \pi(\text{FCA}(K))$. As discussed before (proof of Property 18), $\pi(\text{FCA}(K))$ retracts, from $\kappa(\text{FCA}(K))$, some attributes, non in M^0. By definition, $\kappa(\text{FCA}(K)) \in \mathscr{K}^N$, hence $P(K) = \pi(\text{FCA}(K)) \in \mathscr{K}^N$. □

Contrary to F and according to Q, P is anti-extensive and monotone:

Property 21 (P is anti-extensive and monotone)**.** *The function P satisfies:*

$$P(K) \subseteq K \qquad \text{(anti-extensivity)}$$
$$K \subseteq K' \Rightarrow P(K) \subseteq P(K') \qquad \text{(monotony)}$$

Proof. anti-extensivity If $P(K) \not\subseteq K$, this means that there exists a non empty set of attributes M' disjoint from M present in $P(K)$. Such attributes cannot have been brought by π since it only suppresses attributes. They should come from either FCA or κ. However, FCA does only include in intents attributes from M, and κ does only extracts attributes from the intents. Hence, $P(K) \subseteq K$.

monotony If $K \subseteq K'$, then $\text{FCA}(K) \preceq \text{FCA}(K')$ by monotony of FCA (Property 10). This means that there exist a lattice homomorphism between $\text{FCA}(K)$ and $\text{FCA}(K')$ for which the intent of all concepts of $\text{FCA}(K)$ is found in that of those of $\text{FCA}(K')$; moreover, all concepts of $\text{FCA}(K)$, as identified by their extent, are found in $\text{FCA}(K')$ (Definition 4). Hence, necessarily $\kappa(\text{FCA}(K)) \subseteq \kappa(\text{FCA}(K'))$ and the supporting concepts in $\text{FCA}(K)$ are still present in $\text{FCA}(K')$, so $P(K) = \pi(\text{FCA}(K)) \subseteq \pi(\text{FCA}(K')) = P(K')$. □

Like E and F, Q and P are not closure operators as they are not idempotent. However, with the same arguments as [Rouane-Hacene et al., 2013a], it can be argued that the repeated application of Q converges to a self-supported concept lattice.

Property 22 (Stability of Q [Euzenat, 2021])**.** $\forall L \in \mathscr{L}^N, \exists n; Q^n(L) = Q^{n+1}(L)$.

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 a fixed point has been reached), or it suppresses at least one attribute and then a strictly smaller context is reached. Since the number of scalable attributes is finite and attributes of M^0 are not purged, then the process will stop after a finite number of applications of Q. □

By convention, we note Q^∞ the closure operator[6] associated with Q and $\text{fp}(Q)$, the set of fixed points of Q.

5.3 Fixed points of Q

Like with E, it is possible to apply the Knaster-Tarski theorem to show that $\langle \text{fp}(Q), \preceq \rangle$ is a complete lattice.

The fixed points of Q are exactly those self-supported lattices in $\mathscr{L}$:

Property 23. *For any $L \in \mathscr{L}^N$, L is self-supported iff $L \in \text{fp}(Q)$.*

Proof. Any fixed point for Q is self-supported because if $Q(L) = L$, this is because π does not find any non-supported attribute in the lattice intents. This means that all of them are supported by L. Conversely, each self-supported lattice $L \in \mathscr{L}^N$ is such that $\pi(L) = \kappa(L)$ because all concepts of L only refer to attributes of L, so π does not suppress any attribute from the context. Thus, $Q(L) = \text{FCA}(\pi(L)) = \text{FCA}(\kappa(L)) = L$ (by construction of κ), hence $L \in \text{fp}(Q)$. □

To complete the description of Q, it is possible to establish that its least fixed point is $\text{FCA}(K^0)$.

Property 24. $\text{lfp}(Q) = \text{FCA}(K^0)$.

[6] Which could be named interior operator as well.

Proof. $\kappa(\text{FCA}(K^0)) = K^0$ hence $\pi(\text{FCA}(K^0)) = K^0$ because, it is not possible to suppress attributes from K^0 which being a formal context does not refer to any concept (and in RCA^0 this set of attributes is reduced to $\varnothing$). Thus, $Q(\text{FCA}(K^0)) = \text{FCA}(\pi(\text{FCA}(K^0)))$ $= \text{FCA}(K^0)$. Moreover, $\forall L \in \mathscr{L}^N$, $\text{FCA}(K^0) \preceq L$. Hence, $\text{FCA}(K^0)$ is a fixed point of Q and all other fixed points are greater. □

5.4 Relations between E and Q

We end up with two operations, E and Q, the former extensive and the latter anti-extensive. If we consider concept lattices from the standpoint of the extents, Q decreases the set of concepts of a lattice and E increases them.

An interesting property of the functions E and Q is that they preserve each other stability: E has the advantage of preserving self-supportivity (Property 25 replaces Proposition 3 of [Euzenat, 2021] due to Property 23):

Property 25 (E is internal to fp(Q) [Euzenat, 2021, Prop.3])**.** *$\forall L \in \text{fp}(Q)$, $E(L) \in \text{fp}(Q)$.*

Proof. If $L \in \text{fp}(Q)$, all attributes in intents of L are supported by concepts in L (Property 23). $L \preceq E(L)$, so these concepts are still in $E(L)$. Moreover, $E = \sigma_\Omega \circ \text{FCA}$ and σ_Ω first adds to $\kappa(L)$ attributes which are supported by L. Hence, the attributes in $\kappa(L)$ and those scaled by σ_Ω are still supported by $E(L)$. □

Property 26 (Q is internal to fp(E))**.** *$\forall L \in \text{fp}(E)$, $Q(L) \in \text{fp}(E)$*

Proof. If $L \in \text{fp}(E)$, this means that $E(L) = L$ and, in particular, that σ_Ω does not scale new attributes based on the concepts in L. $Q(L) \preceq L$, so that $Q(L)$ does not contain more concepts than L. $Q(L)$ having not more concepts than L, σ_Ω cannot scale new attributes either ($\sigma_\Omega(Q(L)) \subseteq \sigma_\Omega(L) = \varnothing$). Hence, $Q(L) \in \text{fp}(E)$. □

In addition, the closure operations associated with the two functions preserve the extrema of each other.

Property 27. *$Q^\infty(\text{gfp}(E)) = \text{gfp}(Q)$ and $E^\infty(\text{lfp}(Q)) = \text{lfp}(E)$*

Proof. $\forall L \in \mathscr{L}$, $L \preceq \text{gfp}(E)$ (from Proposition 17) and Q and thus Q^∞ is order preserving (Property 19), hence $Q^\infty(L) \preceq Q^\infty(\text{gfp}(E))$. Moreover, $Q^\infty(\text{gfp}(E)) \in \text{fp}(Q)$, thus $Q^\infty(\text{gfp}(E)) = \text{gfp}(Q)$.

Similarly, $\forall L \in \mathscr{L}$, $\text{lfp}(Q) \preceq L$ (Property 24) and E and thus E^∞ is order preserving (Property 15), hence $E^\infty(\text{lfp}(Q)) \preceq E^\infty(L)$. Moreover, $E^\infty(\text{lfp}(Q)) \in \text{fp}(E)$, thus $E^\infty(\text{lfp}(Q)) = \text{lfp}(E)$. □

The acceptable solutions for RCA are the self-supported fixed points of E, or said otherwise, the elements of $\text{fp}(E) \cap \text{fp}(Q)$.

These extrema are thus bounds within which to find these solutions (see also Figure 12):

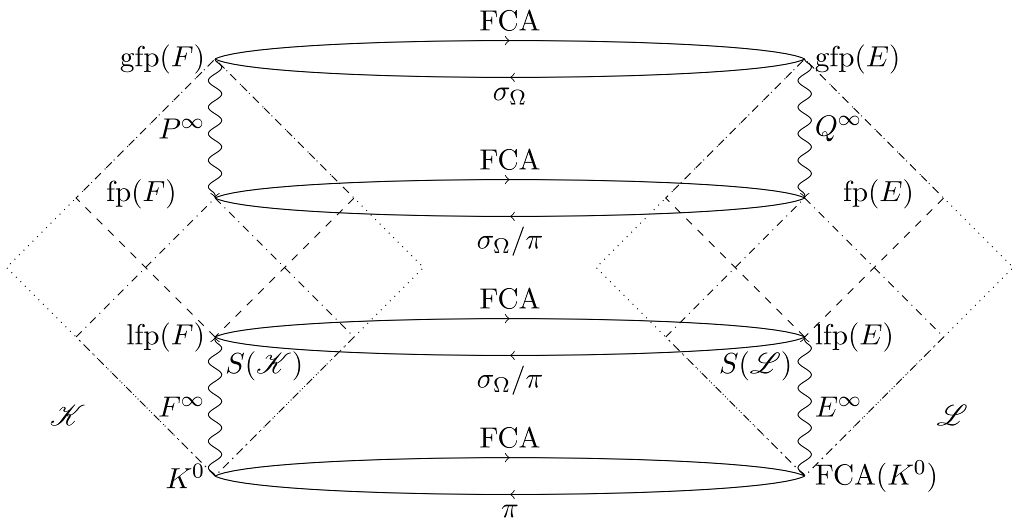


Figure 12: The $\mathscr{L}$ (resp. $\mathscr{K}$) lattice and effects of E and Q (resp. F and P) for characterising fp(E) and $S(\mathscr{L})$ (resp. fp(F) and $S(\mathscr{K})$) (from [Euzenat, 2021]).

Proposition 28 ([Euzenat, 2021])**.** $\forall L \in \text{fp}(E) \cap \text{fp}(Q), \text{lfp}(E) \preceq L \preceq \text{gfp}(Q)$

Proof. lfp(E) is the lower bound for fp(E). Assume that $\text{lfp}(E) \notin \text{fp}(Q)$, then there would exist $Q^\infty(\text{lfp}(E)) \in \text{fp}(Q)$ (by Property 22). By Property 26, $Q^\infty(\text{lfp}(E)) \in \text{fp}(E)$ and due to Property 19 (anti-extensivity), $Q^\infty(\text{lfp}(E)) \preceq \text{lfp}(E)$. This contradicts that lfp(E) is the lower bound for fp(E). Hence, $\text{lfp}(E) \in \text{fp}(E) \cap \text{fp}(Q)$ and is its infimum.

Similarly, gfp(Q) is the upper bound for fp(Q). Assume that $\text{gfp}(Q) \notin \text{fp}(E)$, then there would exist $E^\infty(\text{gfp}(Q)) \in \text{fp}(E)$ [Rouane-Hacene et al., 2013a]. By Property 25, $E^\infty(\text{gfp}(Q)) \in \text{fp}(Q)$ and due to Property 15 (extensivity), $\text{gfp}(Q) \preceq E^\infty(\text{gfp}(Q))$. This would mean that gfp(Q) is not the upper bound for fp(Q). Hence, $\text{gfp}(Q) \in \text{fp}(E) \cap \text{fp}(Q)$ and is its supremum. □

The elements of $\text{fp}(E) \cap \text{fp}(Q)$ thus belong to the interval sublattice [lfp(E) gfp(Q)]. However they do not cover it. The converse of Proposition 28 does not hold in general as shown by Example 11.

Example 11 (Non fixed points of [lfp(E) gfp(Q)] in RCA^0)**.** *The lattice $L_0^\#$ of Figure 10 (p. 37) can be checked to belong to the interval* $[\text{lfp}(E)\ \text{gfp}(Q)] = [L_0^1\ L_0^\star]$*, but it does not belong to* $\text{fp}(E) \cap \text{fp}(Q)$*: it is neither a fixed point for Q (not self-supported) because, as Example 9 shows, B does not belong to $L_0^\#$, nor for E (not saturated) because $\exists r.ABC$ does not belong to $K_0^\#$.*

The definitions and results of the two last sections have been restricted to RCA^0 for the sake of clarity. They will now be generalised.

6 Unified view of the RCA^0 space

Although $\mathscr{K}^N$ and $\mathscr{L}^N$ have been presented independently, it is useful to consider the two sets together as, in RCA, lattices in $\mathscr{L}^N$ are an intermediate result of the process which is used for computing the next context. Instead of dealing with two interrelated spaces independently, we tightly connect them. Doing so, we will consider objects which are pairs of formal contexts and associated concept lattices through FCA. They are called context-lattice pairs.

6.1 The lattice $\mathscr{T}$ of context-lattice pairs

From any context in $\mathscr{K}$, it is possible to generate a context-lattice pair using FCA. The T constructor does this.

Definition 13 (T constructor)**.** *Given a formal context $K \in \mathscr{K}^N$, $T : \mathscr{K}^N \to \mathscr{K}^N \times \mathscr{L}^N$ generates a context-lattice pair, such that:*

$$\mathrm{T}(K) = \langle K, \mathrm{FCA}(K)\rangle$$

We consider the set $\mathscr{T}^N_{K^0,R,\Omega}$ of pairs in $\mathscr{K}^N_{K^0,R,\Omega} \times \mathscr{L}^N_{K^0,R,\Omega}$ such that:

$$\mathscr{T}^N_{K^0,R,\Omega} = \{\langle K, L\rangle \in \mathscr{K}^N_{K^0,R,\Omega} \times \mathscr{L}^N_{K^0,R,\Omega} | L = \mathrm{FCA}(K)\}$$

This set is well defined because $\mathscr{K}^N_{K^0,R,\Omega}$ has already been defined and $\mathscr{L}^N_{K^0,R,\Omega}$ are precisely those lattices obtained by FCA from an element of $\mathscr{K}^N_{K^0,R,\Omega}$.

Alternatively, using Property 9, it can be defined from κ:

$$\mathscr{T}^N_{K^0,R,\Omega} = \{\langle K, L\rangle \in \mathscr{K}^N_{K^0,R,\Omega} \times \mathscr{L}^N_{K^0,R,\Omega} | K = \kappa(L)\}$$

As before, we use $\mathscr{T}_{K^0,R,\Omega} = \mathscr{T}^{N(K^0)}_{K^0,R,\Omega}$ and, for any $\langle K, L\rangle \in \mathscr{T}^N_{K^0,R,\Omega}$ we note:

$$k(\langle K, L\rangle) = K$$
$$l(\langle K, L\rangle) = L$$

It is possible to define the meet and join:

Definition 14 (Meet and join of context-lattice pairs)**.** *Given T, $T' \in \mathscr{T}^N_{K^0,R,\Omega}$ $T \vee T'$ and $T \wedge T'$ are defined as:*

$$T \vee T' = \mathrm{T}(k(T) \vee k(T')) \qquad \text{(join)}$$
$$T \wedge T' = \mathrm{T}(k(T) \wedge k(T')) \qquad \text{(meet)}$$

As this definition makes clear, the operations of $\mathscr{T}^N$ only depend on the context part. But the usual relations with the meet and join on the contexts and lattices are preserved:

Property 29.

$$T \vee T' = \langle k(T) \vee k(T'), l(T) \vee l(T') \rangle$$
$$T \wedge T' = \langle k(T) \wedge k(T'), l(T) \wedge l(T') \rangle$$

Proof. This is a simple consequence on the definition of conjunctions and disjunction on context-lattice pairs (Definition 13) and lattices (Definition 5) as:

$$L \vee L' = \mathrm{FCA}(K \vee K') \qquad \text{(join)}$$
$$L \wedge L' = \mathrm{FCA}(K \wedge K') \qquad \text{(meet)}$$

□

The set of context-lattice pairs is once again closed by meet and join:

Property 30. $\forall T, T' \in \mathscr{T}^N_{K^0,R,\Omega}$, $T \wedge T' \in \mathscr{T}^N_{K^0,R,\Omega}$ *and* $T \vee T' \in \mathscr{T}^N_{K^0,R,\Omega}$.

Proof. $\mathscr{O}_{K^0,R,\Omega}$ is closed by meet and join because meet and join is the piecewise meet or join of context-lattice pairs (Definition 19) and for each $x \in X$, $\mathscr{T}^{N(K^0)}_{K^0_x,R,\Omega}$ is closed by meet and join (Property 30). □

Property 31 (Commutativity, associativity and absorption of $\vee$ and $\wedge$ on $\mathscr{T}$)**.** *For all* $T, T', T'' \in \mathscr{T}$*,*

$$T \vee T' = T' \vee T \quad \textit{and} \quad T \wedge T' = T' \wedge T \qquad \text{(commutativity)}$$
$$(T \vee T') \vee T'' = T \vee (T' \vee T'') \quad \textit{and} \quad (T \wedge T') \wedge T'' = T \wedge (T' \wedge T'') \qquad \text{(associativity)}$$
$$T \wedge (T \vee T') = T \quad \textit{and} \quad T \vee (T \wedge T') = T \qquad \text{(absorption)}$$

Proof. Proofs are given for $\wedge$, those for $\vee$ follow the exact same pattern.

$$\begin{aligned}
T \wedge T' &= \mathrm{T}(k(T) \wedge k(T')) && \text{Definition 14}\\
&= \mathrm{T}(k(T') \wedge k(T)) && \text{Property 3}\\
&= T' \wedge T && \text{Definition 14}\\
(T \wedge T') \wedge T'' &= \mathrm{T}(k(T) \wedge k(T')) \wedge T'' && \text{Definition 14}\\
&= \mathrm{T}(k(\mathrm{T}(k(T) \wedge k(T'))) \wedge k(T'')) && \text{Definition 14}\\
&= \mathrm{T}((k(T) \wedge k(T')) \wedge k(T'')) && \text{Definition 13}\\
&= \mathrm{T}(k(T) \wedge (k(T') \wedge k(T''))) && \text{Property 3}\\
&= \mathrm{T}(k(T) \wedge k(\mathrm{T}((k(T') \wedge k(T''))))) && \text{Definition 13}\\
&= T \wedge \mathrm{T}((k(T') \wedge k(T''))) && \text{Definition 14}\\
&= T \wedge (T' \wedge T'') && \text{Definition 14}\\
T \vee (T \wedge T') &= T \vee \mathrm{T}(k(T) \wedge k(T')) && \text{Definition 14}\\
&= \mathrm{T}(k(T) \vee k(\mathrm{T}(k(T) \wedge k(T')))) && \text{Definition 14}\\
&= \mathrm{T}(k(T) \vee (k(T) \wedge k(T'))) && \text{Definition 13}\\
&= \mathrm{T}(k(T)) && \text{Property 3}\\
&= T && \text{Definition 13}
\end{aligned}$$

□

We also define the order between two context-lattice pairs by combining the orders on contexts and lattices:

Definition 15 (Order)**.** *Given T, $T' \in \mathscr{T}^N_{K^0,R,\Omega}$,*

$$T \preceq T' \text{ if } k(T) \subseteq k(T') \text{ and } l(T) \preceq l(T')$$

Figure 13 presents the relations between $\mathscr{K}^N$, $\mathscr{L}^N$ and $\mathscr{T}^N$ and their respective orders. Since FCA is monotonous (from the proof of Property 10), $T \preceq T'$ iff $k(T) \preceq k(T')$. Like before, we note $T \simeq T'$ if $T \preceq T'$ and $T' \preceq T$, and again $\simeq$ is $=$.

This can be applied to the T constructor.

Property 32. $\forall K, K' \in \mathscr{K}^N_{K^0,R,\Omega}$*, if* $K \subseteq K'$ *then* $\mathrm{T}(K) \preceq \mathrm{T}(K')$

Proof. This is due to monotony of FCA (Property 10). $k(\mathrm{T}(K)) = K \subseteq K' = k(\mathrm{T}(K'))$ means that $l(\mathrm{T}(K)) = \mathrm{FCA}(K) \preceq \mathrm{FCA}(K') = l(\mathrm{T}(K'))$. Thus, $\mathrm{T}(K) \preceq \mathrm{T}(K')$. □

This definition complies with that of meet and join.

Property 33. $\forall T, T' \in \mathscr{T}^N_{K^0,R,\Omega}, T \preceq T'$ *iff* $T = T \wedge T'$

Proof. By Property 4, we have that $k(T) \preceq k(T')$ iff $k(T) = k(T) \wedge k(T')$ and, by Property 13, that $l(T) \preceq l(T')$ iff $l(T) = l(T) \wedge l(T')$, consequently $T \preceq T'$ iff $T = T \wedge T'$ by Property 29. □

This makes $\mathscr{T}^N_{K^0,R,\Omega}$ a complete lattice (Property 34).

Property 34. $\langle \mathscr{T}^N_{K^0,R,\Omega}, \vee, \wedge \rangle$ *is a complete lattice.*

Proof. $\mathscr{T}^N_{K^0,R,\Omega}$ is closed by meet and join (Property 30). $\vee$ and $\wedge$ satisfy commutativity, associativity and the absorption laws (Property 31), so this is a lattice. It is complete because finite. □

6.2 The expansion function EF

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

Definition 16 (Expansion function)**.** *Given a relational context $\langle K^0, R \rangle$ and a set of relational scaling operations Ω, the function $EF_{K^0,R,\Omega} : \mathscr{T}_{K^0,R,\Omega} \to \mathscr{T}_{K^0,R,\Omega}$ is defined by:*

$$EF_{K^0,R,\Omega}(T) = \mathrm{T}(\sigma_\Omega(k(T), R, l(T)))$$

This function is an extension of the previous E and F:

Property 35 (EF extends F and E)**.** $EF(T) = \langle F(k(T)), E(l(T)) \rangle$

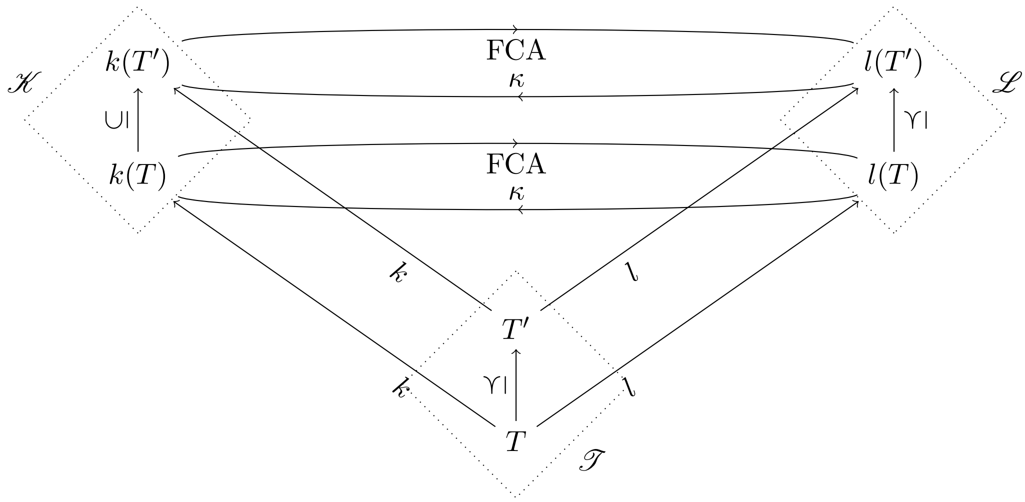


Figure 13: Relations between $\mathscr{T}$, $\mathscr{K}$ and $\mathscr{L}$.

Proof. This is the consequence of $F(K) = \sigma_\Omega(K, R, \text{FCA}(K))$ (Definition 2) and $l(T) = \text{FCA}(k(T))$, so on the one hand, $F(k(T)) = \sigma_\Omega(k(T), R, l(T))$. On the other hand, $E(L) = \text{FCA}(\sigma_\Omega(\kappa(L), R, L))$ (Definition 6) and $\kappa(l(T)) = k(T)$ (Property 9). Hence, $EF(T) = \langle \sigma_\Omega(k(T), R, l(T)), \text{FCA}(\sigma_\Omega(k(T), R, l(T)))\rangle = \langle F(k(T)), E(l(T))\rangle$. □

As previously, we will abbreviate $\mathscr{T}_{K^0,R,\Omega}$ as $\mathscr{T}$ and $EF_{K^0,R,\Omega}$ as EF.

EF is an extensive and monotone internal operation for $\mathscr{T}$:

Property 36 (EF is internal to $\mathscr{T}$)**.** *$\forall T \in \mathscr{T}$, $EF(T) \in \mathscr{T}$*

Proof. $EF(T) \in \mathscr{K}_{K^0,R,\Omega} \times \mathscr{L}_{K^0,R,\Omega}$ because $T \in \mathscr{K}_{K^0,R,\Omega} \times \mathscr{L}_{K^0,R,\Omega}$ and E and F are internal to $\mathscr{K}_{K^0,R,\Omega}$ (Property 6) and $\mathscr{L}_{K^0,R,\Omega}$ (Property 14), respectively. Moreover, $EF(T) = T(F(k(T))) = \langle F(k(T)), \text{FCA}(F(k(T)))\rangle$ (Definition 16 and Property 35), hence $l(EF(T)) = \text{FCA}(k(EF(T)))$. □

Property 37 (EF is extensive and monotone)**.** *The function EF, attached to a relational context and a set of scaling operations, satisfies:*

$$T \preceq EF(T) \quad \text{(extensivity)}$$

$$T \preceq T' \Rightarrow EF(T) \preceq EF(T') \quad \text{(monotony)}$$

Proof. extensivity holds because $T \preceq EF(T)$ if and only if $k(T) \preceq k(EF(T))$. However, $k(EF(T)) = F(k(T))$ (Property 35) and $K \preceq F(K)$ (Property 7). monotony relies on the monotony of F (Property 7) and E (Property 15): $T \preceq T'$ if and only if $k(T) \preceq k'(T)$ and $l(T) \preceq l'(T)$, but this entail $F(k(T)) \preceq F(k'(T))$ (Property 7) and $E(l(T)) \preceq E(l'(T))$ (Property 15), and so $EF(T) \preceq EF(T')$. □

6.3 The contraction function PQ

It is also possible to consider a single contraction function, $PQ_{K^0,R,\Omega}$, attached to a relational context $\langle K^0, R\rangle$ and a set Ω of scaling operations.

The context-lattice pairs $\langle K, L\rangle$ may contain many unsupported attributes. Unsupported attributes are those which refer to classes non existing in the lattice. Indeed, $\exists r.c$ may be part of the attributes of K is well-defined by the incidence relation, but c does not belong to L.

PQ may be defined from π (§4.2.1) and T.

Definition 17 (Contraction function)**.** *Given a relational context $\langle K^0, R\rangle$ and a set of relational scaling operations Ω, the function $PQ_{K^0,R,\Omega} : \mathscr{T}^N_{K^0,R,\Omega} \to \mathscr{T}^N_{K^0,R,\Omega}$ is defined by:*

$$PQ_{K^0,R,\Omega}(T) = \mathrm{T}(\pi_{K^0,R,\Omega}(l(T)))$$

As previously, we will abbreviate $PQ_{K^0,R,\Omega}$ as PQ. This function is also an extension of the previous Q and P:

Property 38 (PQ extends P and Q)**.** $PQ(T) = \langle P(k(T)), Q(l(T))\rangle$

Proof. $\forall T \in \mathscr{T}^N$, $l(T) = \mathrm{FCA}(k(T))$, hence $\pi(l(T)) = \pi(\mathrm{FCA}(k(T)))$ and $P(K) = \pi(\mathrm{FCA}(K))$ (Definition 12), thus $\pi(l(T)) = P(k(T))$. Moreover, $Q(L) = \mathrm{FCA}(\pi(L))$ (Definition 11), thus $\mathrm{FCA}(\pi(l(T))) = Q(l(T))$. So, $\mathrm{T}(\pi(l(T))) = \langle\pi(l(T)), \mathrm{FCA}(\pi(l(T)))\rangle$ $= \langle P(k(T)), Q(l(T))\rangle$ □

PQ is an anti-extensive and monotone internal operation for $\mathscr{T}^N$:

Property 39 (PQ is internal to $\mathscr{T}^N$)**.** $\forall T \in \mathscr{T}^N$, $PQ(T) \in \mathscr{T}^N$

Proof. This follows from P and Q being internal to $\mathscr{K}^N_{K^0,R,\Omega}$ and $\mathscr{L}^N_{K^0,R,\Omega}$, respectively (Property 20 and 18) and Property 38. □

Property 40 (PQ is anti-extensive and monotone)**.** *The function PQ, attached to a relational context and a set of scaling operations, satisfies:*

$$PQ(T) \preceq T \qquad \text{(anti-extensivity)}$$
$$T \preceq T' \Rightarrow PQ(T) \preceq PQ(T') \qquad \text{(monotony)}$$

Proof. anti-extensivity holds because $PQ(T) \preceq T$ if and only if $P(k(T)) = k(PQ(T)) \preceq k(T)$ (Property 38) and $P(K) \preceq K$ (Property 21). monotony relies on the monotony of P and Q: if $T \preceq T'$, then $k(T) \subseteq k(T')$ and $l(T) \preceq l(T')$, hence $P(k(T)) \subseteq P(k(T'))$ (Property 21) and $Q(l(T)) \preceq Q(l(T'))$ (Property 19), hence $PQ(T) \preceq PQ(T')$. □

Joining together contexts and lattices was a preliminary step to consider families of such pairs to represent the behaviour of relational concept analysis as a whole. This is done hereafter.

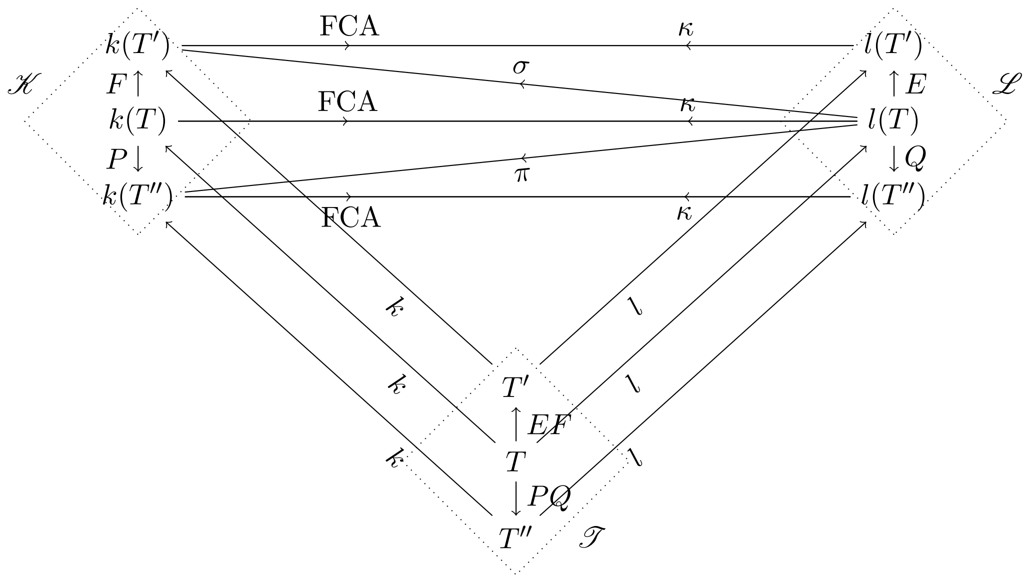


Figure 14: Relations between EF, PQ, E, F, P and Q.

7 Generalisation to full RCA

So far, we have only considered one context independently from the others. We now consider RCA in its entirety.

RCA deals with families contexts. Its elements are thus simple vectors of the pairs generated by each context. These vectors will be considered as sets indexed by X. All provided definitions can be applied to indexed families of context-lattice pairs, the order between them will be the product of the piece-wise orders. All operations remain monotone and extensive (or anti-extensive) as soon as the selected scaling operations are.

The only important change is the notion of self-supported lattices that has to be replaced by supported lattices for individual lattices. Indeed, if $R = \varnothing$ or only contains endorelations, it is sufficient to work on the families as free product of pairs. However, in RCA this product is constrained by the relations in R which may provide support for otherwise unsupported concepts, invalidating those solutions which do not consider such concepts. Hence, here the product must be constrained by R.

In the following we will thus redefine the objects on which RCA operates (§7.1), and the expansion (§7.2) and contraction functions (§7.3). We will then consider the fixed points of these functions and their relations (§7.4).

7.1 The lattice $\mathscr{O}$ of families of context-lattice pairs

The input of RCA is given by a family of formal contexts: $K^0 = \{K^0_x\}_{x\in X}$, a set of relations R between the objects of these contexts, and a set of relational scaling operations Ω.

From this, it is possible to characterise the space $\mathscr{O}_{K^0,R,\Omega}$ associated with RCA by combining the pairs associated with each context.

Definition 18 ($\mathscr{O}_{K^0,R,\Omega}$)**.** *Given an indexed family of contexts $K^0 = \{\langle G_x, M_x^0, I_x^0\rangle\}_{x\in X}$, a set of relations R between the objects of these contexts, and a set of relational scaling operations Ω. The space $\mathscr{O}_{K^0,R,\Omega}$ of indexed families of context-lattice pairs is:*

$$\mathscr{O}_{K^0,R,\Omega} = \bigtimes_{x\in X} \mathscr{T}^{N(K^0)}_{K_x^0,R,\Omega}$$

represented as an indexed set.

As usual, $\mathscr{O}_{K^0,R,\Omega}$ will simply be referred to as $\mathscr{O}$.

This is well defined because the set of all possible context extents across all contexts is determined by the set of objects in the context. This, permits us to name unambiguously all the concepts in the family of lattices. As before, $N(K^0) = \bigcup_{x\in X} N(K_x^0)$. In turn, since $N(K^0)$, R and Ω do not change and I is determined by $\{M_x\}_{x\in X}$ (Property 1), this determines finitely all attributes that can occur in a scaled RCA context.

There is one difference with the RCA^0 setting: the scaled attributes depend on R that makes the connection from one context to another, e.g. from $\mathscr{T}_x$ to $\mathscr{T}_z$. But since it is possible to name concepts in the lattices generated by the scaled relations according to their finite elements in G_z, then the set of scalable attributes in M_x is finite and can be established as $D^x_{\Omega,R,N(K^0)}$ from the beginning.

Finally, $l(O)$ is determined directly from $k(O)$: $l(O) = FCA^*(k(O))$.

The previous notations can be extended:

$$\mathrm{T}(\{K_x\}_{x\in X}) = \{\mathrm{T}(K_x)\}_{x\in X} = \{\langle K_x, \mathrm{FCA}(K_x)\rangle\}_{x\in X}$$

For any $\{T_x\}_{x\in X} \in \mathscr{O}_{K^0,R,\Omega}$:

$$\begin{aligned}
k(\{T_x\}_{x\in X}) &= \{k(T_x)\}_{x\in X}\\
l(\{T_x\}_{x\in X}) &= \{l(T_x)\}_{x\in X}\\
k_z(\{T_x\}_{x\in X}) &= k(T_z)\\
l_z(\{T_x\}_{x\in X}) &= l(T_z)
\end{aligned}$$

We can define $\wedge$ and $\vee$ on $\mathscr{O}_{K^0,R,\Omega}$.

Definition 19 (Meet and join of families of context-lattice pairs)**.** *Given $O = \{T_x\}_{x\in X}$, $O' = \{T'_x\}_{x\in X} \in \mathscr{O}_{K^0,R,\Omega}$, $O \vee O'$ and $O \wedge O'$ are defined as:*

$$O \vee O' = \{T_x \vee T'_x\}_{x\in X} \qquad \text{(join)}$$
$$O \wedge O' = \{T_x \wedge T'_x\}_{x\in X} \qquad \text{(meet)}$$

The set of families of context-lattice pairs is closed by meet and join:

Property 41. $\forall O, O' \in \mathscr{O}^N_{K^0,R,\Omega}$*,* $O \wedge O' \in \mathscr{O}^N_{K^0,R,\Omega}$ *and* $O \vee O' \in \mathscr{O}^N_{K^0,R,\Omega}$*.*

Proof. $\mathscr{O}_{K^0,R,\Omega}$ is closed by meet and join because for each $x \in X$, $\mathscr{T}^{N(K^0)}_{K^0_x,R,\Omega}$ is closed by meet and join (Property 30). □

Meet and join also satisfies commutativity, associativity and absorption law.

Property 42 (Commutativity, associativity and absorption of $\vee$ and $\wedge$ on $\mathscr{O}$)**.** *For all $O, O', O'' \in \mathscr{O}$,*

$$\begin{array}{lcll} O \vee O' = O' \vee O & \textit{and} & O \wedge O' = O' \wedge O & \text{(commutativity)} \\ (O \vee O') \vee O'' = O \vee (O' \vee O'') & \textit{and} & (O \wedge O') \wedge O'' = O \wedge (O' \wedge O'') & \text{(associativity)} \\ O \wedge (O \vee O') = O & \textit{and} & O \vee (O \wedge O') = O & \text{(absorption)} \end{array}$$

Proof. As usual, proofs are given for $\wedge$, those for $\vee$ follow the exact same pattern.

$$\begin{aligned} O \wedge O' &= \{T_x \wedge T'_x\}_{x\in X} && \text{Definition 19} \\ &= \{T'_x \wedge T_x\}_{x\in X} && \text{Property 31} \\ &= O' \wedge O && \text{Definition 19} \\ (O \wedge O') \wedge O'' &= \{T_x \wedge T'_x\}_{x\in X} \wedge O'' && \text{Definition 19} \\ &= \{(T_x \wedge T'_x) \wedge T''_x\}_{x\in X} && \text{Definition 19} \\ &= \{T_x \wedge (T'_x \wedge T''_x)\}_{x\in X} && \text{Property 31} \\ &= O \wedge \{T'_x \wedge T''_x\}_{x\in X} && \text{Definition 19} \\ &= O \wedge (O' \wedge O'') && \text{Definition 19} \\ O \vee (O \wedge O') &= O \vee \{T_x \wedge T'_x\}_{x\in X} && \text{Definition 19} \\ &= \{T_x \vee (T_x \wedge T'_x)\}_{x\in X} && \text{Definition 19} \\ &= \{T_x\}_{x\in X} && \text{Property 31} \\ &= O && \text{Definition 18} \end{aligned}$$

□

We also define the order between two objects by combining the previous definitions.

Definition 20 (Order)**.** *Given $O = \{T_x\}_{x\in X}$, $O' = \{T'_x\}_{x\in X} \in \mathscr{O}_{K^0,R,\Omega}$,*

$$O \preceq O' \textit{ if } \forall x \in X, T_x \preceq T'_x$$

Like before, we note $O \simeq O'$ if $O \preceq O'$ and $O' \preceq O$, and again $\simeq$ is $=$.

Property 32 can be generalised: the order between families of context-lattice pairs may be reduced to the order between contexts (and ultimately the order between their sets of attributes).

Property 43. $\forall O, O' \in \mathscr{O}_{K^0,R,\Omega}$, *if $k(O) \subseteq k(O')$ then $O \preceq O'$*

Proof. $k(O) \subseteq k(O')$ means that $\forall x \in X$, $k_x(O) \subseteq k_x(O')$ which is equivalent to $\langle k_x(O), l_x(O)\rangle \preceq \langle k_x(O'), l_x(O')\rangle$ (Property 32) and hence $O \preceq O'$ (Definition 20). □

This order is compatible with meet and join.

Property 44. $\forall O, O' \in \mathscr{O}_{K^0,R,\Omega}, O \preceq O'$ *iff* $O = O \wedge O'$

Proof. By Property 33, we have that $T_x \preceq T'_x$ iff $T_x = T_x \wedge T'_x$ and this $\forall x \in X$, consequently $O \preceq O'$ iff $O = O \wedge O'$. □

Finally, the set $\mathscr{O}_{K^0,R,\Omega}$ of families of context-lattice pairs is a complete lattice.

Property 45. $\langle \mathscr{O}_{K^0,R,\Omega}, \vee, \wedge \rangle$ *is a complete lattice.*

Proof. $\mathscr{O}_{K^0,R,\Omega}$ is closed by meet and join (Property 41). $\vee$ and $\wedge$ satisfy commutativity, associativity and the absorption laws (Property 42), so this is a lattice. It is complete because finite. □

7.2 The expansion function EF^*

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

Definition 21 (Expansion function)**.** *Given a relational context* $\langle K^0, R \rangle$ *and a set of relational scaling operations* Ω*, the expansion function* $EF^*_{K^0,R,\Omega} : \mathscr{O}_{K^0,R,\Omega} \to \mathscr{O}_{K^0,R,\Omega}$ *is defined by:*

$$EF^*_{K^0,R,\Omega}(O) = \mathrm{T}(\sigma^*_\Omega(k(O), R, l(O)))$$

Beware of the $l(O)$ as these operations depend on the whole set of lattices.

This function is thus not anymore in direct connection with the previous EF but it extends it:

Property 46. $\{EF_{K^0_x,R,\Omega}(T_x)\}_{x \in X} \preceq EF^*_{K^0,R,\Omega}(\{T_x\}_{x \in X})$

Proof. For any $x \in X$, $l_x(O) \in l(O)$, hence $\sigma_\Omega(k_x(O), R, l_x(O)) \subseteq \sigma_\Omega(k_x(O), R, l(O))$ because there are less concepts to scale with: only those in $l_x(O)$.
Thus, $\mathrm{T}(\sigma_\Omega(k_x(O), R, l_x(O))) \preceq \mathrm{T}(\sigma_\Omega(k_x(O), R, l(O)))$ (Property 32).
Hence, $\{EF_{K^0,R,\Omega}(T_x)\}_{x \in X} = \{\mathrm{T}(\sigma_\Omega(k_x(O), R, l_x(O)))\}_{x \in X} \preceq \{\mathrm{T}(\sigma_\Omega(k_x(O), R, l(O)))\}_{x \in X}$ $= EF^*_{K^0,R,\Omega}(O)$. □

As previously, we will abbreviate $\mathscr{O}_{K^0,R,\Omega}$ as $\mathscr{O}$ and $EF^*_{K^0,R,\Omega}$ as EF^*.

EF^* is an extensive and monotone internal operation for $\mathscr{O}$:

Property 47 (EF^* is internal to $\mathscr{O}$)**.** $\forall O \in \mathscr{O}$, $EF^*(O) \in \mathscr{O}$

Proof. $\forall x \in X$, $\sigma_\Omega(k_x(O), R, l(O)) = F(k_x(O)) \in \mathscr{K}^{N(K^0)}_{K^0_x,R,\Omega}$ because σ_Ω only scales properties in $D_{\Omega,R,N(K^0)}$. Thus, $EF(T) = \mathrm{T}(\sigma_\Omega(k_x(O), R, l(O))) \in \mathscr{T}^{N(K^0)}_{K^0_x,R,\Omega}$. hence, $EF^*(O) = \{\mathrm{T}(\sigma_\Omega(k_x(O), R, l(O)))\}_{x \in X} \in \mathscr{O}$ (Definition 18). □

Property 48 (EF^* is extensive and monotone)**.** *The function* EF^* *attached to a relational context and a set of scaling operations satisfies:*

$$O \preceq EF^*(O) \qquad \text{(extensivity)}$$
$$O \preceq O' \Rightarrow EF^*(O) \preceq EF^*(O') \qquad \text{(monotony)}$$

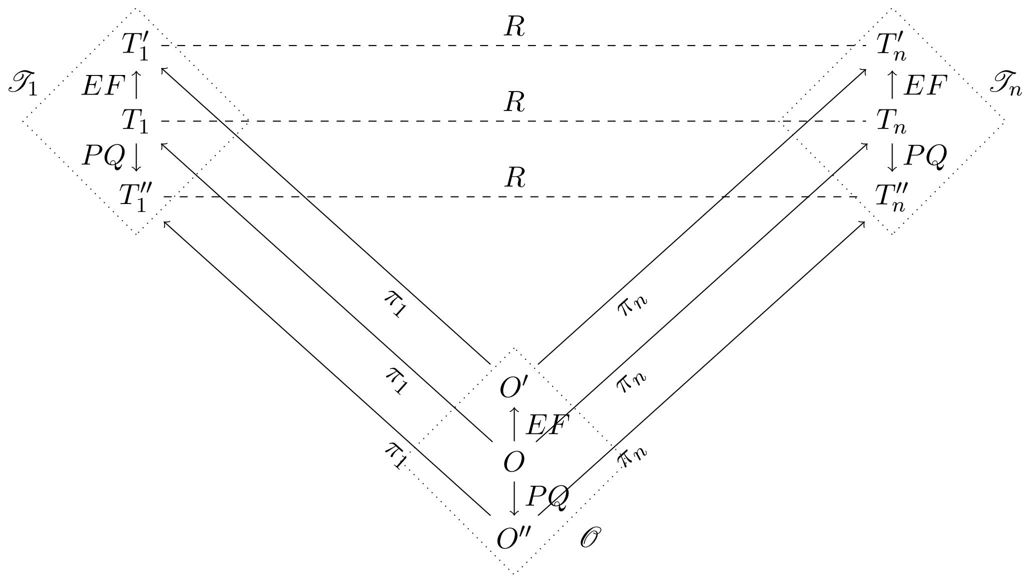


Figure 15: Relations between $\mathscr{O}$, $\mathscr{T}_1$ and $\mathscr{T}_n$.

Proof. extensivity holds because EF^* can only add to $k(O)$ attributes scaled from $l(O)$, hence $\forall x \in X, k_x(O) \subseteq k_x(EF^*(O))$, and by monotony of FCA (Property 10), $\forall x \in X, l_x(O) \preceq l_x(EF^*(O))$. Hence, $O \preceq EF^*(O)$. monotony holds because $O \preceq O'$ means that $\forall x \in X$, $l_x(O) \preceq l_x(O')$ and $k_x(O) \subseteq k_x(O')$. The former entails that $N(l(O)) \subseteq N(l(O'))$ and consequently, that $D^x_{\Omega,R,N(l(O))} \subseteq D^x_{\Omega,R,N(l(O'))}$. A smaller context $(k_x(O))$ is extended by a smaller set of attributes $(D^x_{\Omega,R,N(l(O))})$, thus $k_x(EF^*(O)) \subseteq k_x(EF^*(O'))$. Moreover, by monotony of FCA (Property 10), $k_x(EF^*(O)) \subseteq k_x(EF^*(O'))$ entails $l_x(EF^*(O)) \preceq l_x(EF^*(O'))$. Hence, $EF^*(O) \preceq EF^*(O')$. □

7.3 The contraction function PQ^*

Similarly to $EF^*_{K^0,R,\Omega}$, it is possible to define $PQ^*_{K^0,R,\Omega}$ the contraction function attached to a relational context $\langle K^0, R\rangle$ and a set Ω of scaling operations.

Definition 22 (Contraction function)**.** *Given a relational context $\langle K^0, R\rangle$ and a set of relational scaling operations Ω, the contraction function $PQ^*_{K^0,R,\Omega} : \mathscr{O}_{K^0,R,\Omega} \to \mathscr{O}_{K^0,R,\Omega}$ is defined by:*

$$PQ^*_{K^0,R,\Omega}(O) = \mathrm{T}(\pi^*_{K^0,R,\Omega}(l(O)))$$

As for EF^*, π^* uses the lattices of the whole family $(l(O))$.

This function is thus not anymore in direct connection with the previous PQ but it extends it:

Property 49. $\{PQ_{K^0_x,R,\Omega}(T_x)\}_{x\in X} \preceq PQ^*_{K^0,R,\Omega}(\{T_x\}_{x\in X})$

Proof. For any $x \in X$, $l_x(O) \in l(O)$, hence $\pi(k_x(O), l_x(O)) \subseteq \pi(k_x(O), l(O))$ because there are more attributes to preserve from concepts in $l(O)$.
Thus, $\mathrm{T}(\pi(k_x(O), l_x(O))) \preceq \mathrm{T}(\pi(k_x(O), l(O)))$ (Property 32).
Hence, $\{PQ_{K^0,R,\Omega}(T_x)\}_{x\in X} = \{\mathrm{T}(\pi(k_x(O), l_x(O)))\}_{x\in X} \preceq \{\mathrm{T}(\pi(k_x(O), l(O)))\}_{x\in X} = PQ^*_{K^0,R,\Omega}(O)$. □

As previously, we will abbreviate $PQ^*_{K^0,R,\Omega}$ as PQ^*.

PQ^* is an anti-extensive and monotone internal operation for $\mathscr{O}$:

Property 50 (PQ^* is internal to $\mathscr{O}$)**.** *$\forall O \in \mathscr{O}$, $PQ^*(O) \in \mathscr{O}$*

Proof. $PQ^*(O) = \mathrm{T}(\pi^*(l(O))) = \mathrm{T}(\{\pi(l_x(O), l(O))\}_{x\in X}) = \{\mathrm{T}(\pi(l_x(O), l(O)))\}_{x\in X}$. Hence, $PQ^*(O) \in \mathscr{O}$ if $\pi(l_x(O), l(O)) \in \mathscr{K}^{N(K^0)}_{K^0_x,R,\Omega}$ (Definition 18). This is the case because $\mathscr{K}^{N(K^0)}_{K^0_x,R,\Omega}$ contains all contexts extending K^0_x with attributes from $D^x_{\Omega,R,N(K^0)}$, that $k_x(O) \in \mathscr{K}^{N(K^0)}_{K^0_x,R,\Omega}$ and that π only suppresses attributes from $k_x(O)$ preserving those of K^0_x. □

Property 51 (PQ^* is anti-extensive and monotone)**.** *The function PQ^* attached to a relational context and a set of scaling operations satisfies:*

$$PQ^*(O) \preceq O \qquad \text{(anti-extensivity)}$$
$$O \preceq O' \Rightarrow PQ^*(O) \preceq PQ^*(O') \qquad \text{(monotony)}$$

Proof. anti-extensivity holds because PQ^* can only suppress from $k(O)$ attributes not supported by $l(O)$, hence $\forall x \in X, k_x(PQ^*(O)) \subseteq k_x(O)$, and by monotony of FCA (Property 10), $\forall x \in X, l_x(PQ^*(O)) \preceq l_x(O)$. Hence, $PQ^*(O) \preceq O$. monotony holds because $O \preceq O'$ means that $\forall x \in X, k_x(O) \subseteq k_x(O')$ and $l_x(O) \preceq l_x(O')$. This entails that $N(l(O)) \subseteq N(l(O'))$ and thus, $\forall x \in X$, $D^x_{\Omega,R,N(l(O))} \subseteq D^x_{\Omega,R,N(l(O'))}$. Because $PQ^*(O)$ suppresses from $k_x(O)$ attributes not in $M^0_x \cup D^x_{\Omega,R,N(l(O))}$, this implies that $k_x(PQ^*(O)) \subseteq k_x(PQ^*(O'))$ and by monotony of FCA (Property 10) that $l_x(PQ^*(O)) \preceq l_x(PQ^*(O'))$. Hence, $PQ^*(O) \preceq PQ^*(O')$. □

7.4 The fixed points of EF^* and PQ^*

Given EF^* and PQ^*, it is possible to define their sets of fixed points, i.e. the sets of families of context-lattice pairs closed for EF^* and PQ^*, as:

Definition 23 (fixed points)**.** *A family of context-lattice pairs $O \in \mathscr{O}$ is a fixed point for a function ϕ, if $\phi(O) \simeq O$. We call $\mathrm{fp}(\phi)$ the set of fixed points for ϕ.*

This characterises $\mathrm{fp}(EF^*)$ and $\mathrm{fp}(PQ^*)$.

This may be directly expressed

Property 52. *$O \in \mathrm{fp}(EF^*)$ iff $\sigma^*_\Omega(k(O), R, l(O)) = k(O)$*

Proof. $O = \mathrm{T}(k(O))$ and $EF^*(O) = \mathrm{T}(\sigma^*_\Omega(k(O), R, l(O)))$. ($\Leftarrow$) If $\sigma^*_\Omega(k(O), R, l(O)) = k(O)$, then $EF^*(O) = O$ and thus is a fixed point of EF^*. ($\Rightarrow$) If $\sigma^*_\Omega(k(O), R, l(O)) \neq k(O)$, then $EF^*(O) \neq O$, so it is not a fixed point. □

Property 53. *$O \in \mathrm{fp}(PQ^*)$ iff $\pi^*(l(O)) = k(O)$*

Proof. $O = \mathrm{T}(k(O))$ and $PQ^*(O) = \mathrm{T}(\pi^*(l(O)))$. ($\Leftarrow$) If $\pi^*(l(O)) = k(O)$, then $PQ^*(O) = O$ and thus is a fixed point of PQ^*. ($\Rightarrow$) If $\pi^*(l(O)) \neq k(O)$, then $PQ^*(O) \neq O$, so it is not a fixed point. □

Since $\mathscr{O}$ is a complete lattice (Property 45) and EF^* and PQ^* are order-preserving (or monotone) on $\mathscr{O}$ (Properties 48 and 51), then we can apply the Knaster-Tarski theorem (Theorem 8).

This warrants that there exists least and greatest fixed points of EF^* and PQ^* in $\mathscr{O}$. For such a function ϕ, operating on the finite set $\mathscr{O}$, we can define their least and greatest fixed points:

$$\mathrm{lfp}(\phi) = \bigwedge_{O \in \mathrm{fp}(\phi)} O \text{ and } \mathrm{gfp}(\phi) = \bigvee_{O \in \mathrm{fp}(\phi)} O$$

They may be further characterised for our two functions.

Property 24 (apparently not in [Euzenat, 2021]) can be generalised as:

Property 54 (least fixed point of PQ^*)**.**

$$\mathrm{lfp}(PQ^*) = \mathrm{T}(K^0)$$

Proof. $\forall x \in X$, $\kappa(\mathrm{FCA}(K^0_x)) = K^0_x$ (Property 9) hence $\mathrm{T}(\pi(K^0_x, \mathrm{FCA}^*(K^0))) = \mathrm{T}(K^0_x)$ because, it is not possible to suppress attributes from K^0_x which being a simple formal context does not refer to any concept. Thus, $PQ^*(\mathrm{T}(K^0)) = \mathrm{T}(\{\pi(K^0_x, \mathrm{FCA}^*(K^0))\}_{x \in X}) = \mathrm{T}(K^0)$. Moreover, $\forall O \in \mathscr{O}$, $\mathrm{T}(K^0) \preceq O$. Hence, $\mathrm{T}(K^0)$ is a fixed point of PQ^* and all other fixed points are greater. □

Proposition 17 [Euzenat, 2021] can be generalised as:

Property 55 (greatest fixed point of EF^*)**.**

$$\mathrm{gfp}(EF^*_{K^0,R,\Omega}) = \mathrm{T}(\{\mathrm{K}^{\langle R, N(K^0_x)\rangle}_{+D^x_{\Omega,R,N(K^0)}}(K^0_x)\}_{x \in X})$$

Proof. This family of context-lattice pairs is the greatest element of $\mathscr{O}$ as its context contains all attributes of $M^0 \cup D_{\Omega,R,N(K^0)}$ and due to Property 43. It is also a fixed point because EF^* is extensive (Property 48) and internal (Property 47). □

The EF^* and PQ^* converge after a finite number of applications.

Property 56 (Stability of EF^* and PQ^*). *$\forall O \in \mathscr{O}$,*

$$\exists n; EF^{*n}(O) = EF^{*n+1}(O)$$

and

$$\exists n; PQ^{*n}(O) = PQ^{*n+1}(O)$$

Proof. EF^* can only increase the contexts when there are new concepts in lattices and increase the lattices when contexts grows. However, the set of attributes that can increase contexts, and the set of concepts that can be in lattices, is finite. Hence, at each step either an attribute is added or n has been reached such that the family of context-lattice pairs is the same. This is the same argument as that of [Rouane-Hacene et al., 2013a].

Conversely, PQ^* can only decrease the contexts and reduce lattices. Since these are finite (and the decrease does not affect the attributes of K^0), there exists a n at which the decrease stops. □

The finite application of EF^* and PQ^* as many times as necessary are closure functions denoted by $EF^{*\infty}$ and $PQ^{*\infty}$, respectively.

Property 57 ($EF^{*\infty}$ and $PQ^{*\infty}$ are closures).

Proof. Since EF^* is extensive and monotone (Property 48), $EF^{*\infty}$ is also extensive and monotone by transitivity. In order to be a closure operator it has to be idempotent. This is the case, because $\forall O \in \mathscr{O}$, $EF^{*\infty}(O) = EF^{*n}(O) = EF^{*n+1}(O) = EF^*(EF^{*n}(O))$. Since $EF^{*n}(O) = EF^*(EF^{*n}(O))$, EF^* can be applied n times, yielding $EF^{*\infty}(O) = EF^{*n}(O) = EF^{*n}(EF^{*n}(O)) = EF^{*\infty}(EF^{*\infty}(O))$.

The same can be obtained from PQ^*, albeit anti-extensive (Property 51). □

In addition, they are extrema of their set of fixed points.

Property 58 (Closure functions are smallest subsuming and greatest subsumed fixed points). *$\forall O \in \mathscr{O}$,*

$$EF^{*\infty}(O) = \min_{\preceq}(\text{fp}(EF^*) \cap \{O'|O \preceq O'\})$$
$$PQ^{*\infty}(O) = \max_{\preceq}(\text{fp}(PQ^*) \cap \{O'|O' \preceq O\})$$

Proof. $EF^{*\infty}(O) \in \text{fp}(EF^*)$ and $PQ^{*\infty}(O) \in \text{fp}(PQ^*)$ as they satisfy Definition 23. Moreover, $EF^{*\infty}(O) \in \{O'|O \preceq O'\}$ and $PQ^{*\infty}(O) \in \{O'|O' \preceq O\}$ as EF^* and PQ^* are respectively extensive and anti-extensive and monotonous (Property 48 and 51). There cannot be $O' \in \text{fp}(EF^*) \cap \{O'|O \preceq O'\}$ such that $O' \prec EF^{*\infty}(O)$ because otherwise $k(O') \subseteq k(EF^{*\infty}(O))$ and $k(O) \subset k(O')$. In other terms, O' contains all attributes of O but not all attributes of $EF^{*\infty}(O)$. But, $EF^{*\infty}$ only adds scalable attributes and $k(EF^{*\infty}(O))$ contains only attributes scalable from O. Hence, O' is not closed for EF^* ($O' \notin \text{fp}(EF^*)$).

The same holds for $PQ^{*\infty}(O)$, there cannot be $O' \in \text{fp}(PQ^*) \cap \{O'|O' \preceq O\}$ such that $PQ^{*\infty}(O) \prec O'$ because otherwise $k(O') \subseteq k(O)$. In other terms, O' contains not all attributes of O but all attributes of $PQ^{*\infty}(O)$. But, $PQ^{*\infty}$ only suppresses non supported attributes and $k(PQ^{*\infty}(O))$ contains only attributes supported from O. Hence, O' is not closed for PQ^* ($O' \notin \text{fp}(PQ^*)$). □

The respective relations of these various objects can be summarised by the following property:

Property 59. $\forall O \in \mathscr{O}$,

$$\text{lfp}(PQ^*) \preceq PQ^{*\infty}(O) \preceq PQ^*(O) \preceq O \preceq EF^*(O) \preceq EF^{*\infty}(O) \preceq \text{gfp}(EF^*)$$

Proof. All the inner equations are consequences of the extensivity of EF^* (Property 48) and anti-extensivity of PQ^* (Property 51). The outer ones owe to the fact that the two closure operations are fixed points (Property 58), thus they are subsumed by, resp. subsuming, their greatest, resp. least, fixed point. □

What we called acceptable solutions in Section 3 can now be rephrased in Definition 24.

Definition 24 (Acceptable family of context-lattice pairs)**.** *Given a family of contexts K^0, a set of scaling operators Ω and a set of relations R, a family of context-lattice pairs O is acceptable if*

- $O \in \mathscr{O}_{K^0,R,\Omega}$ *(well-formedness)*
- $\forall x \in X$, $k_x(O) = \sigma_\Omega(k_x(O), R, l(O))$ *(saturation)*
  alt. $\nexists \varsigma r.C \in D_{K^0,R,\Omega} \setminus k_x(O)$ *such that* $\varsigma r.C \in \sigma_\Omega(k_x(O), R, l(O))$
- $\forall x \in X$, $k_x(O)$ *is supported by* $l(O)$ *(self-support)*
  alt. $\forall \varsigma r.C \in k_x(O)$, $\varsigma \in \Omega$, $r \in R_{x,z}$ *and* $C \in l_z(O)$

This can be characterised as those families of context-lattice pairs fixed points of both EF^* and PQ^*.

Proposition 60 (Acceptable solutions are fixed points of both EF^* and PQ^*)**.** *Given Ω, R and K^0, a family of context-lattice pairs O is acceptable iff $O \in \mathscr{O}_{K^0,R,\Omega}$ and $O \in \text{fp}(EF^*) \cap \text{fp}(PQ^*)$.*

Proof. O is well-formed as it belongs to $\mathscr{O}_{K^0,R,\Omega}$. $O \in \text{fp}(EF^*)$ means that $O = EF^*_{K^0,R,\Omega}(O)$, that is $O = T(\sigma^*_\Omega(k(O), R, l(O)))$ (Definition 21). This also means that $k(O) = \sigma^*_\Omega(k(O), R, l(O))$ and $\forall x \in X, k_x(O) = \sigma_\Omega(k_x(O), R, l(O))$ (Definition 2).

$O \in \text{fp}(PQ^*)$ means that $O = PQ^*_{K^0,R,\Omega}(O)$, that is $O = T(\pi^*(l(O)))$ (Definition 22). This also means that $k(O) = \pi^*(l(O))$ and thus $k_x(O) = \pi(l_x(O), l(O)) = \text{K}^{\langle R,l(O)\rangle}_{-D_{\Omega,R,N(K^0)\setminus N(l(O))}}(k_x(O))$ (Definition 10), i.e. all attributes non supported by $l(O)$ are suppressed from $k_x(O)$. Hence, O is self-supported (or $k_x(O)$ is supported by $l(O)$), and vice versa. Thus, O is acceptable (Definition 24). □

Hence, the set of acceptable solutions is $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$.

Example 12 (Acceptable solutions). *In the example of Section 3.3, it can be checked that the given solutions belong to the expected fixed points:*

$$EF^*(\{\langle K_3^1, L_3^1\rangle, \langle K_4^1, L_4^1\rangle\}) = \{\langle K_3^1, L_3^1\rangle, \langle K_4^1, L_4^1\rangle\} = PQ^*(\{\langle K_3^1, L_3^1\rangle, \langle K_4^1, L_4^1\rangle\})$$
$$EF^*(\{\langle K_3^\star, L_3^\star\rangle, \langle K_4^\star, L_4^\star\rangle\}) = \{\langle K_3^\star, L_3^\star\rangle, \langle K_4^\star, L_4^\star\rangle\} = PQ^*(\{\langle K_3^\star, L_3^\star\rangle, \langle K_4^\star, L_4^\star\rangle\})$$
$$EF^*(\{\langle K_3', L_3'\rangle, \langle K_4', L_4'\rangle\}) = \{\langle K_3', L_3'\rangle, \langle K_4', L_4'\rangle\} = PQ^*(\{\langle K_3', L_3'\rangle, \langle K_4', L_4'\rangle\})$$

and

$$EF^*(\{\langle K_3'', L_3''\rangle, \langle K_4'', L_4''\rangle\}) = \{\langle K_3'', L_3''\rangle, \langle K_4'', L_4''\rangle\} = PQ^*(\{\langle K_3'', L_3''\rangle, \langle K_4'', L_4''\rangle\})$$

and none of the other elements of $\mathscr{O}$ as displayed in Figure 17.

In lattice theory, saturation and self-support would have been easily called closedness. The terms saturation and self-support have been chosen in order to be clearer.

8 The fixed-point semantics of RCA

Now that the acceptable solutions have been characterised structurally and functionally, we can define the semantics of RCA. RCA returns the smallest acceptable solution. It is also the least fixed point of the EF^* function (§8.1).

It is also possible to be interested by operators that generate the greatest acceptable solution, which is also the greatest fixed point of PQ^* (§8.2). It is also worth considering obtaining the whole set $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$. Section 8.3 investigate the structure of $[\text{fp}(EF^*),\ \text{fp}(PQ^*)]$ and its relation with $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$ towards that goal. It provides various results that may be exploited to develop efficient algorithms.

8.1 Classical RCA computes EF^*'s least fixed point

RCA as it has been defined in Section 2.4 may be redefined as

$$\underline{\text{RCA}}_\Omega(K^0, R) = l(EF^{*\infty}_{K^0,R,\Omega}(\text{T}(K^0)))$$

i.e. RCA iterates EF^* from $\text{T}(K^0)$ until reaching a fixed point, and ultimately the corresponding lattices are returned.

It seems thus that RCA returns a fixed point of EF^*. Hence the question: which fixed point is returned by RCA's well-grounded semantics? These are the least fixed points.

Proposition 61 (The RCA algorithm computes the least fixed point of EF^*). *Given EF^* the expansion function associated to K^0, R and Ω,*

$$\underline{\text{RCA}}_\Omega(K^0, R) = l(\text{lfp}(EF^*_{K^0,R,\Omega}))$$

Proof. $\mathrm{T}(K^0) \in \mathscr{O}$, hence $EF^{*\infty}(\mathrm{T}(K^0)) \in \mathscr{O}$ (by Property 47). Moreover, $EF^{*\infty}(\mathrm{T}(K^0)) = \min_{\preceq}(\mathrm{fp}(EF^*) \cap \{O' \mid \mathrm{T}(K^0) \preceq O'\})$ (Property 58). But $\forall O' \in \mathscr{O}$, $\mathrm{T}(K^0) \preceq O'$, hence $EF^{*\infty}(\mathrm{T}(K^0)) = \min_{\preceq}(\mathrm{fp}(EF^*))$. Thus, $EF^{*\infty}_{K^0,R,\Omega}(\mathrm{T}(K^0))$ is a fixed point more specific than all fixed points: it is the least fixed point.

$\underline{\mathrm{RCA}}_\Omega(K^0, R) = l(EF^{*\infty}_{K^0,R,\Omega}(\mathrm{T}(K^0)))$ returns the lattice associated with the least fixed point of $EF^*_{K^0,R,\Omega}$. □

8.2 Greatest fixed-point (of PQ^*) semantics

It is possible to define $\overline{\mathrm{RCA}}$ as returning the greatest acceptable solution. The greatest fixed point of EF^* (Property 55) is not necessarily an acceptable solution because it may not be self-supported. Said otherwise, it does not belong to $\mathrm{fp}(EF^*) \cap \mathrm{fp}(PQ^*)$ because it is not a fixed point for $\mathrm{fp}(PQ^*)$.

Alternatively, a greatest fixed-point semantics may be defined as::

$$\overline{\mathrm{RCA}}_\Omega(K^0, R) = l(PQ^{*\infty}_{K^0,R,\Omega}(\mathrm{T}(\{\mathrm{K}^{\langle R,N(K^0_x)\rangle}_{+D^x_{\Omega,R,N(K^0)}}(K^0_x)\}_{x\in X})))$$

and it can be characterised analogously as the greatest fixed point of $PQ^*_{K^0,R,\Omega}$.

Proposition 62 ($\overline{\mathrm{RCA}}$ determines the greatest fixed point of PQ^*)**.** *Given PQ^* the contraction function associated to K^0, R and Ω,*

$$\overline{\mathrm{RCA}}_\Omega(K^0, R) = l(\mathrm{gfp}(PQ^*_{K^0,R,\Omega}))$$

Proof. $O^\infty = \mathrm{T}(\{\mathrm{K}^{\langle R,N(K^0_x)\rangle}_{+D^x_{\Omega,R,N(K^0)}}(K^0_x)\}_{x\in X}) \in \mathscr{O}$, hence $PQ^{*\infty}_{K^0,R,\Omega}(O^\infty) \in \mathscr{O}$ (by Property 50). Moreover, $PQ^{*\infty}(O^\infty) = \max_{\preceq}(\mathrm{fp}(PQ^*) \cap \{O' | O' \preceq O^\infty\})$ (Property 58). But $\forall O' \in \mathscr{O}$, $O' \preceq O^\infty$, hence $PQ^{*\infty}(O^\infty) = \max_{\preceq}(\mathrm{fp}(PQ^*))$ Thus, $PQ^{*\infty}_{K^0,R,\Omega}(O^\infty)$ is a fixed point more general than all fixed points: it is the greatest fixed point.

$\overline{\mathrm{RCA}}_\Omega(K^0, R) = l(PQ^{*\infty}_{K^0,R,\Omega}(O^\infty))$ returns the lattice associated with the greatest fixed point of $PQ^*_{K^0,R,\Omega}$. □

In order to find $\mathrm{gfp}(PQ^*)$, the process starts with the largest family of context-lattice pairs $\mathrm{T}(\{\mathrm{K}^{\langle R,N(K^0_x)\rangle}_{+D^x_{\Omega,R,N(K^0)}}(K^0_x)\}_{x\in X})$ and iterates the application of PQ^*, i.e. the two operations π^* and FCA^*, until reaching a fixed point, i.e. reaching n such that $O^{n+1} = O^n$.

Thus, the $\overline{\mathrm{RCA}}$ algorithm proceeds in the following way:

1. Initial formal contexts: $\{\langle G_x, M^0_x, I^0_x\rangle\}_{x\in X} \leftarrow \{\mathrm{K}^{\langle R,N(K^0_x)\rangle}_{+D^x_{\Omega,R,N(K^0)}}(K^0_x)\}_{x\in X}$
2. $\{L^t_x\}_{x\in X} \leftarrow \mathrm{FCA}^*(\{\langle G_x, M^t_x, I^t_x\rangle\}_{x\in X})$ (or, for each formal context, $\langle G_x, M^t_x, I^t_x\rangle$ the corresponding concept lattice $L^t_x = \mathrm{FCA}(\langle G_x, M^t_x, I^t_x\rangle)$ is created using FCA).
3. $\{\langle G_x, M^{t+1}_x, I^{t+1}_x\rangle\}_{x\in X} \leftarrow \pi^*(\{L^t_x\}_{x\in X})$ (i.e. suppressing form K^t_x each attribute in L^t_x referring through a relation $r_y \in R_{x,z}$ to a concept c_z not appearing in L^t_z).
4. If $\exists x \in X; M^{t+1}_x \neq M^t_x$ (purging has occurred), go to Step 2.
5. Return: $\{L^t_x\}_{x\in X}$.

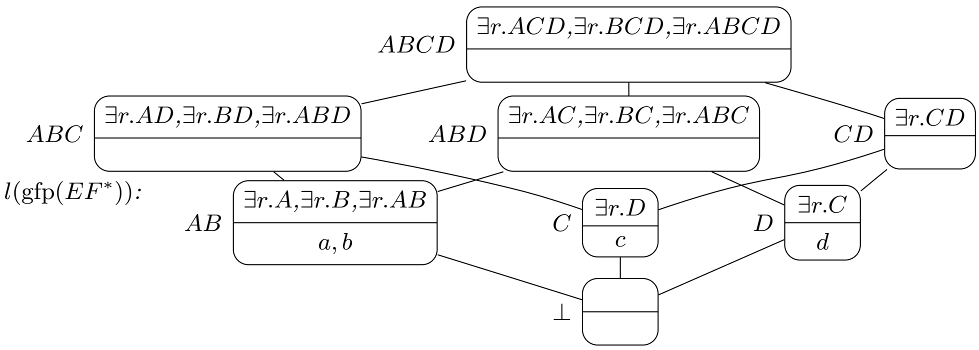
This algorithm is the dual of the RCA procedure.

Example 13 shows how this is processed in RCA^0.

Example 13 (Greatest-fixed point semantics). *Consider the example of Section 3.2.* $\text{gfp}(EF^*) = \text{T}(\{\text{K}^{\langle R,N(K^0_0)\rangle}_{+D^x_{\Omega,R,N(K^0_0)}}(K^0_0)\})$ *(Property 55) is presented below:*

$k(\text{gfp}(EF^*))$	$\exists r.A$	$\exists r.B$	$\exists r.C$	$\exists r.D$	$\exists r.AB$	$\exists r.AC$	$\exists r.AD$	$\exists r.BC$	$\exists r.BD$	$\exists r.CD$	$\exists r.ABC$	$\exists r.ABD$	$\exists r.ACD$	$\exists r.BCD$	$\exists r.ABCD$
a	×	×			×	×	×	×	×		×	×	×	×	×
b	×	×			×	×	×	×	×		×	×	×	×	×
c				×			×		×	×		×	×	×	×
d			×			×		×		×	×		×	×	×

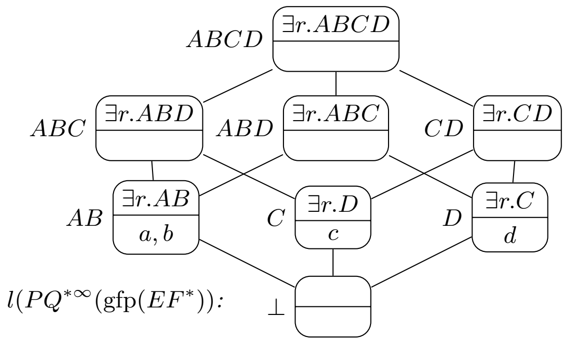
It leads to the following lattice:



It can be checked that it is a fixed point for EF^**: no additional attribute can be scaled.*

On the contrary, PQ^* *can be applied to* $\text{gfp}(EF^*)$ *leading to the following result:*

$k(PQ^{*\infty}(\text{gfp}(EF^*)))$	$\exists r.C$	$\exists r.D$	$\exists r.AB$	$\exists r.CD$	$\exists r.ABC$	$\exists r.ABD$	$\exists r.ABCD$
a			×		×	×	×
b			×		×	×	×
c		×		×		×	×
d	×			×	×		×



In this case, $PQ^*(\text{gfp}(EF^*)) = PQ^{*\infty}(\text{gfp}(EF^*))$*, this is not necessarily true as some concepts may be supported by attributes which may be retracted from the lattice due to lack of support. When full RCA is considered, this may span from context to context.*

It may be interesting, for some applications to know that there is only one acceptable solution. This can easily be characterised by:

Property 63. $\text{lfp}(EF^*_{K^0,R,\Omega}) = \text{gfp}(PQ^*_{K^0,R,\Omega})$ *iff* $|\text{fp}(EF^*_{K^0,R,\Omega}) \cap \text{fp}(PQ^*_{K^0,R,\Omega})| = 1$

The proof of this proposition is given here, but it makes reference to results relying on a further investigation on the structure of fixed points which is the object Section 8.3. Of course, none of these results rely on Proposition 63.

Proof. $\Rightarrow$) Since all solutions are within the interval between both fixed points (Property 67), if these are equal then the interval contains only one object, so are the set of solutions.
$\Leftarrow$) If there is only one solution, since both $\text{lfp}(EF^*_{K^0,R,\Omega})$ and $\text{gfp}(PQ^*_{K^0,R,\Omega})$ are among them ($\text{lfp}(EF^*_{K^0,R,\Omega}) \in \text{fp}(PQ^*_{K^0,R,\Omega})$ and $\text{gfp}(PQ^*_{K^0,R,\Omega}) \in \text{fp}(EF^*_{K^0,R,\Omega})$ are consequences of Properties 64, 65 and 66), then they are equal. □

This can be tested using $\underline{\text{RCA}}$ and $\overline{\text{RCA}}$.

FCA can be described as RCA with $R = \varnothing$. In this case, $D_{\Omega,R,N(K^0)} = \varnothing$. Thus, $\mathscr{O} = \{T(K^0)\} = \{\langle K^0, FCA(K^0)\rangle\}$ and $\text{fp}(EF^*) = \text{fp}(PQ^*) = \{T(K^0)\}$. Hence,

$$\underline{\text{RCA}}_\Omega(K^0, \varnothing) = \overline{\text{RCA}}_\Omega(K^0, \varnothing) = \text{FCA}(K^0)$$

8.3 The structure of fixed points

Besides obtaining the least fixed points of EF^* ($\underline{\text{RCA}}_\Omega$) or the greatest fixed point of PQ^* ($\overline{\text{RCA}}_\Omega$), an interesting problem is to obtain all acceptable solutions, i.e. those families of context-lattice pairs belonging to the fixed points of both functions ($\text{fp}(EF^*) \cap \text{fp}(PQ^*)$). The interval between $\text{lfp}(EF^*_{K^0,R,\Omega})$ and $\text{gfp}(PQ^*_{K^0,R,\Omega})$ may be thought of as an approximation of the situation described by the initial context K^0. For some purposes, this may be sufficient. However, it may also be interesting to navigate within the set $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$ of fixed points or to compute it.

A naive algorithm for this consists in enumerating all elements of the interval and testing if they are fixed points. This would not be very efficient. Figure 17 shows that, in our simple Example 3.3, among the 16 elements in the interval only 4 belong to $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$.

One way to try to improve on this situation is to understand the structure of the set of fixed points and its relation with the two functions and their closures. Figure 16 illustrates the structure of $\mathscr{O}$ and how $EF^{*\infty}$ and $PQ^{*\infty}$ and their composition traverse this structure.

An interesting property of the functions EF^* and PQ^* is that they preserve each other stability:

Property 64 (EF^* is internal to $\text{fp}(PQ^*)$)**.** $\forall O \in \text{fp}(PQ^*)$, $EF^*(O) \in \text{fp}(PQ^*)$.

Proof. If $O \in \text{fp}(PQ^*)$, all attributes in intents of $l(O)$ are supported by concepts in $k(O)$ (by extension of Property 23). $O \preceq EF^*(O)$, so these concepts are still in $k(EF^*(O))$. Moreover, EF^* only adds to $k(O)$ attributes which are supported by $l(O)$ (they only refer to concepts in $l(O)$. Hence, the attributes in $k(EF^*(O))$ and those scaled by σ_Ω are still supported by $l(EF^*(O))$. □

Property 65 (PQ^* is internal to $\text{fp}(EF^*)$)**.** *$\forall O \in \text{fp}(EF^*)$, $PQ^*(O) \in \text{fp}(EF^*)$*

Proof. If $O \in \text{fp}(EF^*)$, this means that $EF^*(O) = O$ and, in particular, that σ_Ω does not scale new attributes based on the concepts in $k(O)$. $PQ^*(O) \preceq O$, so that $k(PQ^*(O))$ does not contain more concepts than $k(O)$. Then σ_Ω cannot scale new attributes either ($\sigma_\Omega(k(PQ^*(O))) \subseteq \sigma_\Omega(k(O)) = \varnothing$). Hence, $PQ^*(O) \in \text{fp}(EF^*)$. □

In addition, the closure operations associated with the two functions preserve their extrema.

Property 66. *$PQ^{*\infty}(\text{gfp}(EF^*)) = \text{gfp}(PQ^*)$ and $EF^{*\infty}(\text{lfp}(PQ^*)) = \text{lfp}(EF^*)$*

Proof. $\forall O \in \mathscr{O}$, $O \preceq \text{gfp}(EF^*)$ (from Property 59), and $PQ^{*\infty}$ is order preserving (Property 57), hence $PQ^{*\infty}(O) \preceq PQ^{*\infty}(\text{gfp}(EF^*))$. Thus, $\forall O \in \text{fp}(PQ^*)$, $O \preceq PQ^{*\infty}(\text{gfp}(EF^*))$. Moreover, $PQ^{*\infty}(\text{gfp}(EF^*)) \in \text{fp}(PQ^*)$, thus $PQ^{*\infty}(\text{gfp}(EF^*)) = \text{gfp}(PQ^*)$.

Similarly, $\forall O \in \mathscr{O}$, $\text{lfp}(PQ^*) \preceq O$ (Property 59), and $EF^{*\infty}$ is order preserving (Property 57), hence $EF^{*\infty}(\text{lfp}(PQ^*)) \preceq EF^{*\infty}(O)$. Thus, $\forall O \in \text{fp}(EF^*)$, $EF^{*\infty}(\text{lfp}(PQ^*)) \preceq O$. Moreover, $EF^{*\infty}(\text{lfp}(PQ^*)) \in \text{fp}(EF^*)$, thus $EF^{*\infty}(\text{lfp}(PQ^*)) = \text{lfp}(EF^*)$. □

The acceptable solutions for RCA are the elements of $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$. These extrema are thus bounds within which to find them (see also Figure 16):

Proposition 67. *$\forall O \in \text{fp}(EF^*) \cap \text{fp}(PQ^*), \text{lfp}(EF^*) \preceq O \preceq \text{gfp}(PQ^*)$*

Proof. $\text{lfp}(EF^*)$ is the lower bound for $\text{fp}(EF^*)$. Assume that $\text{lfp}(EF^*) \notin \text{fp}(PQ^*)$, then there would exist $PQ^{*\infty}(\text{lfp}(EF^*)) \in \text{fp}(PQ^*)$ (by Property 56). By Property 65, $PQ^{*\infty}(\text{lfp}(EF^*)) \in \text{fp}(EF^*)$ and due to Property 57 (anti-extensivity), $PQ^{*\infty}(\text{lfp}(EF^*)) \preceq \text{lfp}(EF^*)$. This contradicts that $\text{lfp}(EF^*)$ is the lower bound for $\text{fp}(EF^*)$. Hence, $\text{lfp}(EF^*) \in \text{fp}(EF^*) \cap \text{fp}(PQ^*)$ and is its infimum.

Similarly, $\text{gfp}(PQ^*)$ is the upper bound for $\text{fp}(PQ^*)$. Assume that $\text{gfp}(PQ^*) \notin \text{fp}(EF^*)$, then there would exist $EF^{*\infty}(\text{gfp}(PQ^*)) \in \text{fp}(EF^*)$ (by Property 56). By Property 64, $EF^{*\infty}(\text{gfp}(PQ^*)) \in \text{fp}(PQ^*)$ and due to Property 57 (extensivity), $\text{gfp}(PQ^*) \preceq EF^{*\infty}(\text{gfp}(PQ^*))$. This would mean that $\text{gfp}(PQ^*)$ is not the upper bound for $\text{fp}(PQ^*)$. Hence, $\text{gfp}(PQ^*) \in \text{fp}(EF^*) \cap \text{fp}(PQ^*)$ and is its supremum. □

The elements of $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$ thus belong to the interval $[\text{lfp}(EF^*)\ \text{gfp}(PQ^*)]$. However they do not cover it: the converse of Proposition 67 does not hold in general as shown in the counter-Exex:neq.

Example 14 (Non coverage in RCA)**.** *In the example of Section 3.3,* $\text{lfp}(EF^*) = \{\langle K_1^1, L_1^1\rangle, \langle K_2^1, L_2^1\rangle\}$ *and* $\text{gfp}(PQ^*) = \{\langle K_1^\star, L_1^\star\rangle, \langle K_2^\star, L_2^\star\rangle\}$. *The family* $\{\langle K_1^\#, L_1^\#\rangle, \langle K_2^\#, L_2^\#\rangle\}$ *of Figure 8 belongs to* $[\text{lfp}(EF^*)\ \text{gfp}(PQ^*)]$*, but not to* $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$ *as mentioned in Example 10. Figure 17 shows that 12 out of 16 elements of the interval are in this situation.*

The layout of Figures 16 and 18 do not help understanding the situation, but Figure 17 illustrates the presence of non acceptable objects within the interval.

In order to find the elements of $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$, the closure of EF^* and PQ^*, $EF^{*\infty}$ and $PQ^{*\infty}$, can be used as functions which maps elements of $\mathscr{O}$ into families of context-lattice pairs in $\text{fp}(EF^*)$ and $\text{fp}(PQ^*)$, respectively. Moreover, Properties 65 and 64 entails that $PQ^{*\infty} \circ EF^{*\infty}$ and $EF^{*\infty} \circ PQ^{*\infty}$ map any element of $\mathscr{O}$ into an acceptable family of context-lattice pairs in $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$. Hence, the set of acceptable solutions are those elements in the image of $\mathscr{O}$ by the composition of these two closure operations, in any order.

Property 68. $\text{Im}(PQ^{*\infty} \circ EF^{*\infty}) = \text{fp}(EF^*) \cap \text{fp}(PQ^*) = \text{Im}(EF^{*\infty} \circ PQ^{*\infty})$

Proof. We show it for $PQ^{*\infty} \circ EF^{*\infty}$, the other part is dual:

$\subseteq$ By definition, $\text{Im}(PQ^{*\infty} \circ EF^{*\infty}) \subseteq \text{Im}(PQ^{*\infty}) = \text{fp}(PQ^*)$. Moreover, $\text{Im}(EF^{*\infty}) = \text{fp}(EF^*)$, but by Property 65, if $O \in \text{fp}(EF^*)$, then $PQ^{*\infty}(O) \in \text{fp}(EF^*)$. Hence, $\text{Im}(PQ^{*\infty} \circ EF^{*\infty}) \subseteq \text{fp}(EF^*) \cap \text{fp}(PQ^*)$.

$\supseteq$ $\forall O \in \text{fp}(PQ^*) \cap \text{fp}(EF^*)$, $O \in \text{fp}(EF^*)$, thus $EF^{*\infty}(O) = O$ and $O \in \text{fp}(PQ^*)$, thus $PQ^{*\infty}(O) = O$. Hence, $O = PQ^{*\infty}(EF^{*\infty}(O)) = PQ^{*\infty} \circ EF^{*\infty}(O) \in \text{Im}(PQ^{*\infty} \circ EF^{*\infty})$ and consequently $\text{fp}(EF^*) \cap \text{fp}(PQ^*) \subseteq \text{Im}(PQ^{*\infty} \circ EF^{*\infty})$. □

In addition, these functions are monotonous and idempotent.

Property 69. *$PQ^{*\infty} \circ EF^{*\infty}$ (resp. $EF^{*\infty} \circ PQ^{*\infty}$) is order-preserving and idempotent:*

$$\forall O, O' \in \mathscr{O}, O \preceq O' \Rightarrow PQ^{*\infty} \circ EF^{*\infty}(O) \preceq PQ^{*\infty} \circ EF^{*\infty}(O') \quad \text{(monotony)}$$

$$PQ^{*\infty} \circ EF^{*\infty} \circ PQ^{*\infty} \circ EF^{*\infty}(O) = PQ^{*\infty} \circ EF^{*\infty}(O) \quad \text{(idempotence)}$$

Proof. We prove it for $PQ^{*\infty} \circ EF^{*\infty}$, the $EF^{*\infty} \circ PQ^{*\infty}$ case is strictly dual.

monotony is obtained as the combination of order-preservation of the two functions: $O \preceq O'$, hence $EF^{*\infty}(O) \preceq EF^{*\infty}(O')$, and thus $PQ^{*\infty} \circ EF^{*\infty}(O) \preceq PQ^{*\infty} \circ EF^{*\infty}(O')$ (Property 51) (applying Property 57 twice).

idempotence is obtained from Property 68: $\forall O \in \mathscr{O}$, $PQ^{*\infty} \circ EF^{*\infty}(O) \in \text{fp}(EF^*) \cap \text{fp}(PQ^*)$, hence $PQ^{*\infty} \circ EF^{*\infty}(O) = O$ and $PQ^{*\infty} \circ EF^{*\infty} \circ PQ^{*\infty} \circ EF^{*\infty}(O) = O$, thus $PQ^{*\infty} \circ EF^{*\infty} \circ PQ^{*\infty} \circ EF^{*\infty}(O) = PQ^{*\infty} \circ EF^{*\infty}(O)$. □

The monotony of these functions entails that $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$ is a complete lattice:

Proposition 70. *$\langle \text{fp}(EF^*) \cap \text{fp}(PQ^*), \preceq \rangle$ is a complete sublattice of $\langle \mathscr{O}, \preceq \rangle$.*

Proof. $\text{fp}(EF^*) \cap \text{fp}(PQ^*) = \text{Im}(PQ^{*\infty} \circ EF^{*\infty})$ (Property 68) and $\text{Im}(PQ^{*\infty} \circ EF^{*\infty}) = \text{fp}(PQ^{*\infty} \circ EF^{*\infty})$ due to idempotence (Property 69), hence the Knaster-Tarski theorem can be applied based on Property 69 (monotony), concluding that it is a complete lattice. It is included in $\mathscr{O}$, thus this is a sublattice of $\langle \mathscr{O}, \preceq \rangle$. □

This is illustrated by Example 15.

However, these functions are not necessarily extensive nor anti-extensive (see Figure 18 and Example 17, p. 67). Hence, they would not be closure operators.

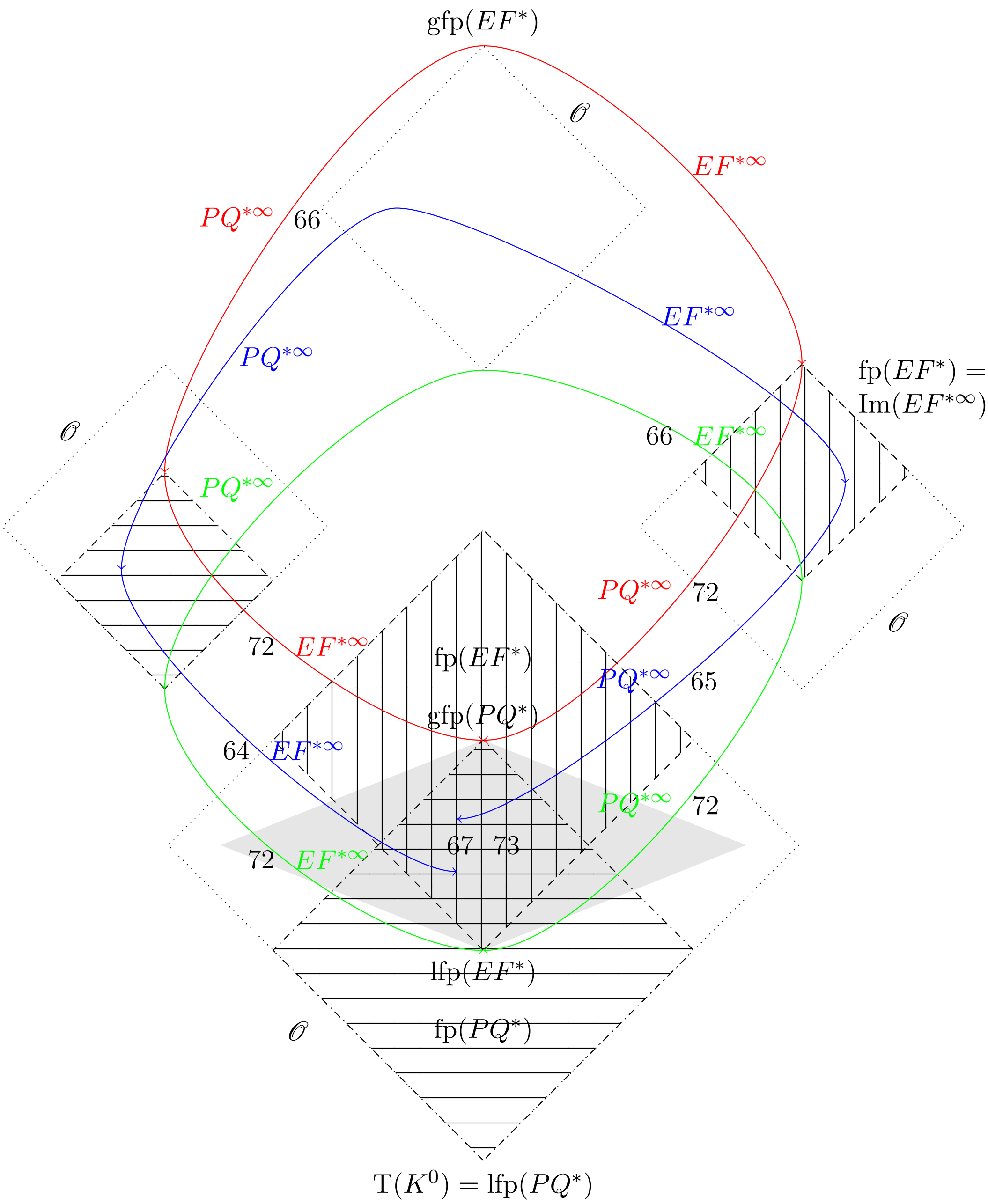


Figure 16: Illustration of Properties 64, 65, 66, 67, 68, 72 and 73. The figure displays four times $\mathscr{O}$ and the images of gfp(EF^*) (red), a random family of context-lattice pairs (blue) and lfp(PQ^*) = T(K^0) (green) through $PQ^{*\infty}$ (left) and $EF^{*\infty}$ (right). fp(EF^*) is drawn in vertical lines; fp(PQ^*) in horizontal lines and the grey area depicts the interval [lfp(EF^*) gfp(PQ^*)].

Example 15 (Interval lattice)**.** *Figure 17 shows all elements of* $[\text{lfp}(EF^*)\ \text{gfp}(PQ^*)]$ *for the example of Section 3.3. It can be observed that* $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$ *is a proper sublattice of* $\mathscr{O}$*. Actually only 4 out of 16 possible objects in the interval are acceptable.*

In the figure, direct edges corresponding to EF^* *or* PQ^**, from lattice pairs of level 2 and 4, are drawn in solid or dashed, respectively. All the objects, of level 3, which are not comparable with the two intermediate fixed points map to the extrema of the interval and thus are not displayed.*

For any family of context-lattice pairs within the fixed points, i.e. $\text{fp}(EF^*)$ *or* $\text{fp}(PQ^*)$ (the vertically or horizontally stripped area of Figure 18), the two functions are equal.

Property 71. $\forall O \in \text{fp}(EF^*) \cup \text{fp}(PQ^*), PQ^{*\infty} \circ EF^{*\infty}(O) = EF^{*\infty} \circ PQ^{*\infty}(O)$

Proof. For any lattice O belonging to $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$, $PQ^*(O) = EF^*(O) = O$, hence $PQ^{*\infty} \circ EF^{*\infty}(O) = EF^{*\infty} \circ PQ^{*\infty}(O) = O$. Similarly, for any lattice O belonging to $\text{fp}(EF^*)$, then $EF^{*\infty}(O) = O$, so $PQ^{*\infty} \circ EF^{*\infty}(O) = PQ^{*\infty}(O)$. However, by Property 64, since $O \in \text{fp}(EF^*)$, $PQ^{*\infty}(O) \in \text{fp}(EF^*)$. This means that $EF^{*\infty} \circ PQ^{*\infty}(O) = PQ^{*\infty}(O)$ as well. The same can be proved for $O \in \text{fp}(PQ^*)$ with Property 65. □

What is actually shown by the proofs of Property 71 is that:

$$\begin{array}{l}\text{if } O \in \text{fp}(EF^*) \text{ then } PQ^{*\infty} \circ EF^{*\infty}(O) = EF^{*\infty} \circ PQ^{*\infty}(O) = PQ^{*\infty}(O)\\ \text{if } O \in \text{fp}(PQ^*) \text{ then } PQ^{*\infty} \circ EF^{*\infty}(O) = EF^{*\infty} \circ PQ^{*\infty}(O) = EF^{*\infty}(O)\end{array}$$

In particular, this applies to the bounds of $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$:

Property 72.

$$EF^{*\infty} \circ PQ^{*\infty}(\text{gfp}(EF^*)) = PQ^{*\infty} \circ EF^{*\infty}(\text{gfp}(EF^*)) = \text{gfp}(PQ^*)$$

and

$$PQ^{*\infty} \circ EF^{*\infty}(\text{lfp}(PQ^*)) = EF^{*\infty} \circ PQ^{*\infty}(\text{lfp}(PQ^*)) = \text{lfp}(EF^*)$$

Proof. The first part of these equations are consequences of Property 71, since $\text{gfp}(EF^*)$ and $\text{lfp}(PQ^*)$ belong to $\text{fp}(EF^*)$ and $\text{fp}(PQ^*)$, respectively. The second part is due to $PQ^{*\infty} \circ EF^{*\infty}(\text{gfp}(EF^*)) = PQ^{*\infty}(\text{gfp}(EF^*))$ and $EF^{*\infty} \circ PQ^{*\infty}(\text{lfp}(PQ^*)) = EF^{*\infty}(\text{lfp}(PQ^*))$ for the same reason that $\text{gfp}(EF^*) \in \text{fp}(EF^*)$ and $\text{lfp}(PQ^*) \in \text{fp}(PQ^*)$, respectively. Property 66 shows that the second terms correspond to $\text{gfp}(PQ^*)$ and $\text{lfp}(EF^*)$, respectively. □

Examples 16 and 17 show that $EF^{*\infty} \circ PQ^{*\infty}(O^\#) \prec O^\# \prec PQ^{*\infty} \circ EF^{*\infty}(O^\#)$ hence that the equality does not hold in general.

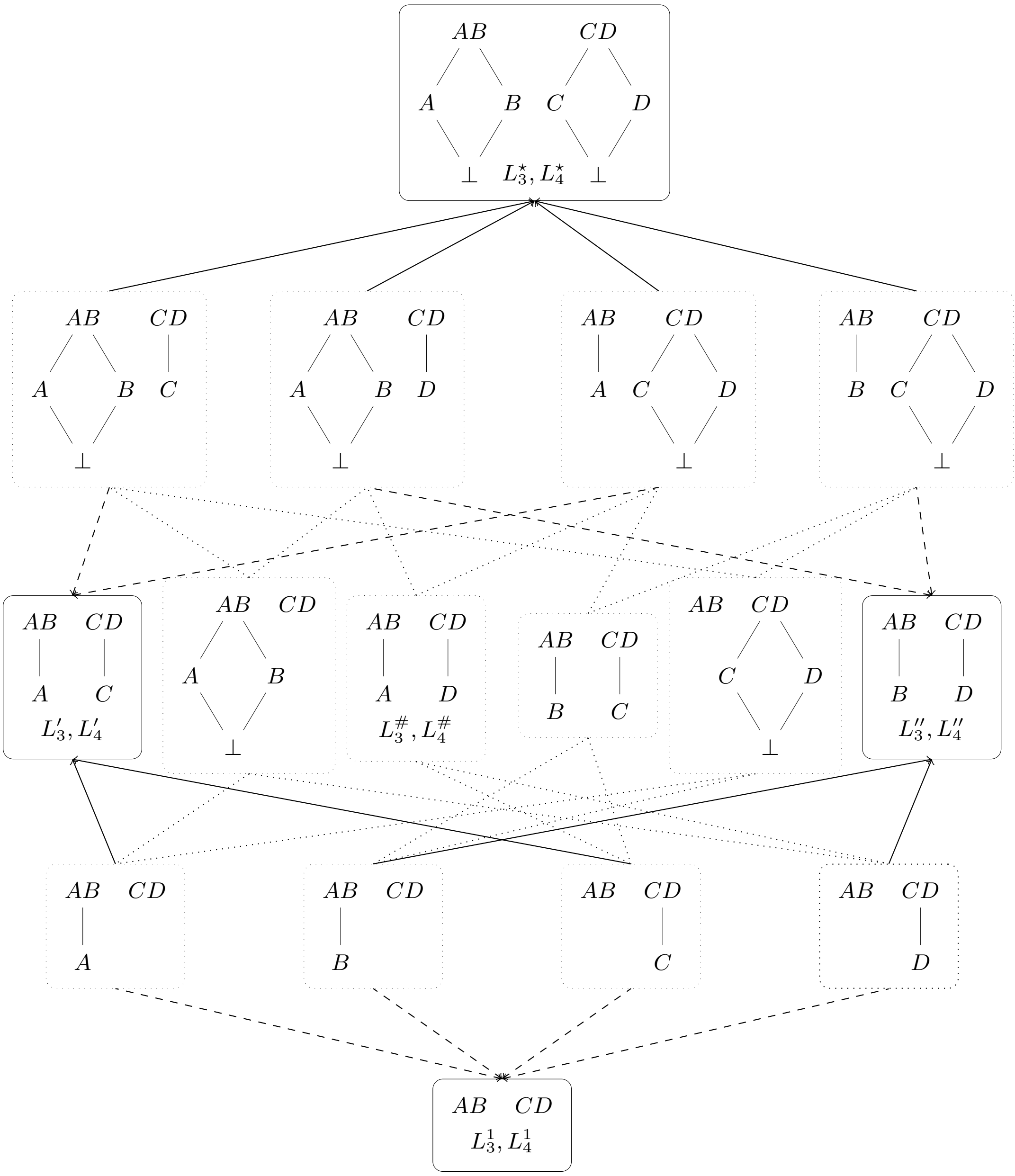


Figure 17: All the lattices belonging to [lfp(EF^*) gfp(PQ^*)] in the example of Section 3.3. Those in fp(EF^*) ∩ fp(PQ^*) are within solid boxes. As usual, only direct edges are displayed. Solid arrows show direct most specific subsumers corresponding to EF^* and dashed arrows show direct more general subsumees corresponding to PQ^*.

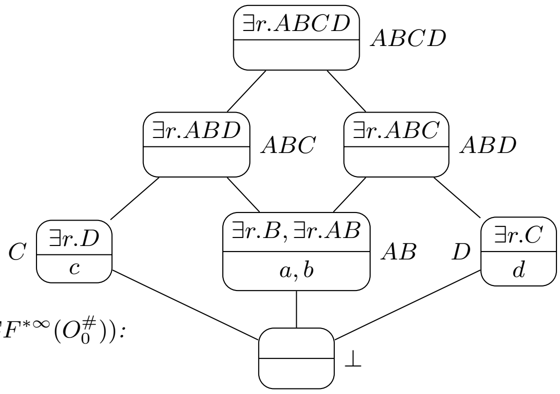
Example 16 (Counterexample to equality in RCA^0)**.** *Consider $O_0^{\#} = \langle K_0^{\#}, L_0^{\#}\rangle$ of Figure 10 (p. 37) in the context of Example 3.2 (p. 23). $O_0^{\#} \in \mathscr{O}$ because all attributes belong to $M_0^0 \cup D_{\Omega,R,N(K^0)}$ and $L_0^{\#} = \text{FCA}^*(K_0^{\#})$.*

In fact, $O_0^{\#} \in [\text{lfp}(EF^*)\ \text{gfp}(PQ^*)] = [\langle K_0^1, L_0^1\rangle\ \langle K_0^\star, L_0^\star\rangle]$, but $O_0^{\#} \notin \text{fp}(EF^*) \cap \text{fp}(PQ^*)$ as explained in Example 11.

Indeed, applying $EF^{*\infty}$ returns $EF^{*\infty}(O_0^{\#})$:

$k(EF^{*\infty}(O_0^{\#}))$	$\exists r.B$	$\exists r.C$	$\exists r.D$	$\exists r.AB$	$\exists r.ABC$	$\exists r.ABD$	$\exists r.ABCD$
a	×			×	×	×	×
b	×			×	×	×	×
c						×	×
d		×	×		×		×

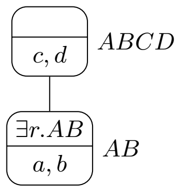
$l(EF^{*\infty}(O_0^{\#}))$:



Applying $PQ^{*\infty}$ returns $PQ^{*\infty}(O_0^{\#})$:

$k(PQ^{*\infty}(O_0^{\#}))$	$\exists r.AB$
a	×
b	×
c	
d	

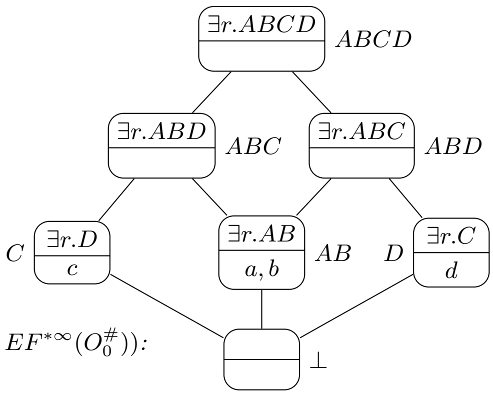
$l(PQ^{*\infty}(O_0^{\#}))$:



In fact, none of $EF^{*\infty}(O_0^{\#})$ nor $PQ^{*\infty}(O_0^{\#})$ are either fixed point for PQ^* and EF^*, respectively. Indeed, $PQ^{*\infty} \circ EF^{*\infty}(O_0^{\#})$ is:

$k(PQ^{*\infty} \circ EF^{*\infty}(O_0^{\#}))$	$\exists r.C$	$\exists r.D$	$\exists r.AB$	$\exists r.ABC$	$\exists r.ABD$	$\exists r.ABCD$
a			×	×	×	×
b			×	×	×	×
c					×	×
d	×	×		×		×

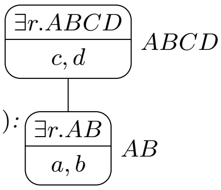
$l(PQ^{*\infty} \circ EF^{*\infty}(O_0^{\#}))$:



and $EF^{*\infty} \circ PQ^{*\infty}(O_0^{\#})$:

$k(EF^{*\infty} \circ PQ^{*\infty}(O_0^{\#}))$	$\exists r.AB$	$\exists r.ABCD$
a	×	×
b	×	×
c		×
d		×

$l(EF^{*\infty} \circ PQ^{*\infty}(O_0^{\#}))$:



Now, $PQ^{*\infty} \circ EF^{*\infty}(O_0^{\#})$ and $EF^{*\infty} \circ PQ^{*\infty}(O_0^{\#})$ belong to $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$. Yet they are not isomorphic. In fact, $EF^{*\infty} \circ PQ^{*\infty}(O_0^{\#}) \prec PQ^{*\infty} \circ EF^{*\infty}(O_0^{\#})$ This is the result of σ which may add needed support (ABD from ABC) and π which may suppress unsupported concepts (ABC missing ABD).

Example 17 (Counterexample to equality in RCA)**.** *Consider Example 3.3 (p. 24), $\text{lfp}(EF^*) = O_{12}^1 = \{\langle L_1^1, K_1^1\rangle, \langle L_2^1, K_2^1\rangle\}$ and $\text{gfp}(PQ^*) = O_{12}^\star = \{\langle L_1^\star, K_1^\star\rangle, \langle L_2^\star, K_2^\star\rangle\}$. $O_{12}^{\#} = \{\langle L_1^{\#}, K_1^{\#}\rangle, \langle L_2^{\#}, K_2^{\#}\}$ belongs to $[\text{lfp}(EF^*)\ \text{gfp}(PQ^*)]$ but not to $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$. It happens that $EF^*(O_{12}^{\#}) = O_{12}^\star$ and $PQ^*(O_{12}^{\#}) = O_{12}^1$, hence $PQ^{*\infty} \circ EF^{*\infty}(O_{12}^{\#}) = PQ^* \circ EF^*(O_{12}^{\#}) = O_{12}^\star$ and $EF^{*\infty} \circ PQ^{*\infty}(O_{12}^{\#}) = EF^* \circ PQ^*(O_{12}^{\#}) = O_{12}^1$. These two objects are not isomorphic. What can be said, in this case, is that $EF^{*\infty} \circ PQ^{*\infty}(O_{12}^{\#}) \prec PQ^{*\infty} \circ EF^{*\infty}(O_{12}^{\#})$. This is the result of σ which may add needed support (for C and B from A and C) and π which may suppress unsupported concepts (A missing C and D missing B).*

It is not necessary that the results of the closure be the bounds of the interval as is shown for any object of the second and fourth lines of the lattice of Figure 17. Example 16 illustrates this even better.

It may be that, as illustrated by Example 17, when $PQ^{*\infty}$ is first applied, it suppresses non-supported attributes which cannot be recovered by scaling. Conversely, $EF^{*\infty}$ applied first may scale attributes which support previously non-supported attributes (ABC in Example 17). These will not be suppressed any more.

Property 73 shows that, in addition, there is still a homomorphism between the two resulting objects.

Property 73. $\forall O \in \mathscr{O}, EF^{*\infty} \circ PQ^{*\infty}(O) \preceq PQ^{*\infty} \circ EF^{*\infty}(O)$

Proof. $PQ^{*\infty}(O) \preceq O$ by Property 59. But $EF^{*\infty}$ is monotonous (Property 57), hence $EF^{*\infty} \circ PQ^{*\infty}(O) \preceq EF^{*\infty}(O)$. $PQ^{*\infty}$ is also monotonous (Property 57), thus $PQ^{*\infty} \circ EF^{*\infty} \circ PQ^{*\infty}(O) \preceq PQ^{*\infty} \circ EF^{*\infty}(O)$. However, $PQ^{*\infty} \circ EF^{*\infty}(O) \in \text{fp}(EF^*) \cup \text{fp}(PQ^*)$ so $PQ^{*\infty} \circ EF^{*\infty}(O) = EF^{*\infty} \circ PQ^{*\infty}(O)$ (Property 71). Thus, $PQ^{*\infty} \circ EF^{*\infty} \circ PQ^{*\infty}(O) = EF^{*\infty} \circ PQ^{*\infty} \circ PQ^{*\infty}(O) = EF^{*\infty} \circ PQ^{*\infty}(O)$. This means that $EF^{*\infty} \circ PQ^{*\infty}(O) \preceq PQ^{*\infty} \circ EF^{*\infty}(O)$. □

Alternative proof. The same reasoning can be held from $O \preceq EF^{*\infty}(O)$ (Property 59) and $EF^{*\infty}$ and $PQ^{*\infty}$ being monotonous (Property 57). Hence, $PQ^{*\infty}(O) \preceq PQ^{*\infty} \circ EF^{*\infty}(O)$ and $EF^{*\infty} \circ PQ^{*\infty}(O) \preceq EF^{*\infty} \circ PQ^{*\infty} \circ EF^{*\infty}(O)$. But, $PQ^{*\infty} \circ EF^{*\infty}(O) \in$

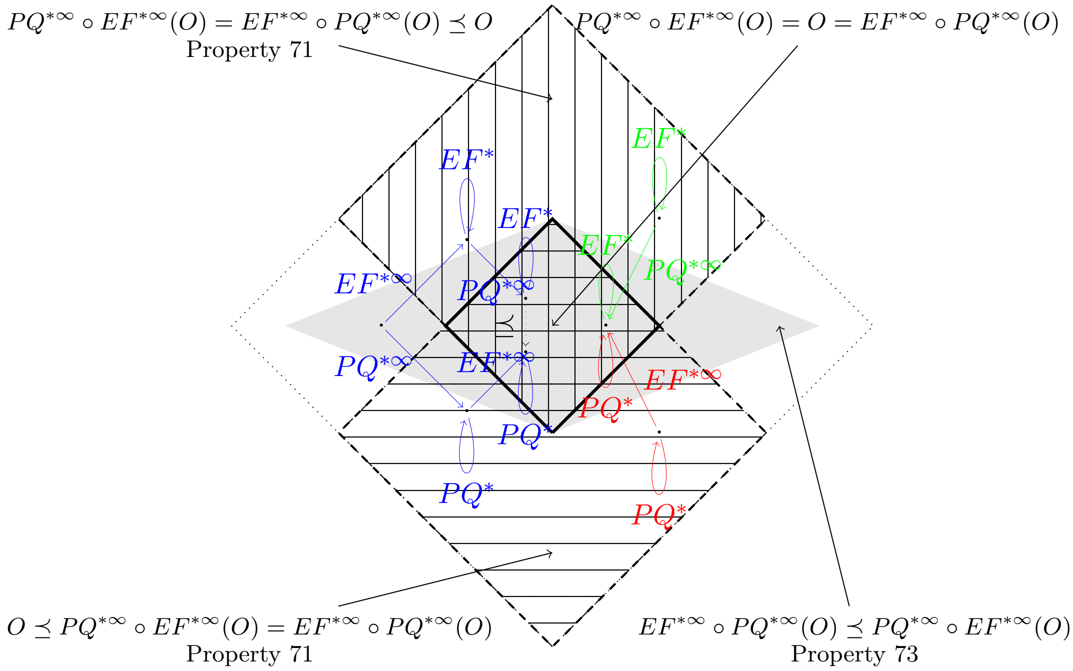


Figure 18: Illustration of the position of $PQ^{*\infty} \circ EF^{*\infty}(O)$ and $EF^{*\infty} \circ PQ^{*\infty}(O)$ depending on O's origin (in dotted, $\mathscr{O}$, in dashed $\text{fp}(EF^*) \cup \text{fp}(PQ^*)$, in plain $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$). Colours corresponds to that of Figure 16: green starting from $\text{fp}(EF^*)$, red starting from $\text{fp}(PQ^*)$, blue starting outside of them.

$\text{fp}(EF^*) \cup \text{fp}(PQ^*)$, so $PQ^{*\infty} \circ EF^{*\infty}(O) = EF^{*\infty} \circ PQ^{*\infty}(O)$ (Property 71). Thus $EF^{*\infty} \circ PQ^{*\infty} \circ EF^{*\infty}(O) = PQ^{*\infty} \circ EF^{*\infty} \circ EF^{*\infty}(O) = PQ^{*\infty} \circ EF^{*\infty}(O)$. Hence, $EF^{*\infty} \circ PQ^{*\infty}(O) \preceq PQ^{*\infty} \circ EF^{*\infty}(O)$. □

The alternative proof is given here to show that starting from the EF^* or PQ^* give the same result.

It is thus unclear what to do with $EF^{*\infty} \circ PQ^{*\infty}$ and $PQ^{*\infty} \circ EF^{*\infty}$ in general. For instance, if one needs an operation to map elements of $\mathscr{O}$ to $\text{fp}(EF^*) \cap \text{fp}(PQ^*)$, which one is preferable? There may be an interest in studying the interval $[EF^{*\infty} \circ PQ^{*\infty}(O)\ PQ^{*\infty} \circ EF^{*\infty}(O)]$. Does it contain only fixed points or no fixed points? Are these the image of other lattices? This question can be answered if O can be compared to these bounds (Proposition 74): the intermediate families are *not* fixed points.

Proposition 74. *$\forall O \in \mathscr{O} \setminus (\text{fp}(EF^*) \cap \text{fp}(PQ^*))$:*
- *if $O \preceq PQ^{*\infty} \circ EF^{*\infty}(O)$, then $\forall O' \in [O\ PQ^{*\infty} \circ EF^{*\infty}(O)[$, $O' \notin \text{fp}(EF^*) \cap \text{fp}(PQ^*)$*
- *if $EF^{*\infty} \circ PQ^{*\infty}(O) \preceq O$, then $\forall O' \in]EF^{*\infty} \circ PQ^{*\infty}(O)\ O]$, $O' \notin \text{fp}(EF^*) \cap \text{fp}(PQ^*)$*

Proof. Considering the first item of the proposition, $O \preceq PQ^{*\infty} \circ EF^{*\infty}(O)$ can only occur if $EF^{*\infty}(O) \in \text{fp}(PQ^*)$, i.e. $PQ^{*\infty} \circ EF^{*\infty}(O) = EF^{*\infty}(O)$. Indeed, if this were not the case, then $PQ^{*\infty}$ would suppress attributes from $EF^{*\infty}(O)$. However, since $O \preceq PQ^{*\infty} \circ EF^{*\infty}(O)$, these could not be attributes from O, but only attributes added by $EF^{*\infty}$. But since $EF^{*\infty}$ only adds attributes if they are supported and it starts with attributes from O, this is not possible. Thus, if $O' \in [O\ PQ^{*\infty} \circ EF^{*\infty}(O)[$, then $O' \in [O\ EF^{*\infty}(O)[$. However, O' cannot be a fixed point for EF^* because it contains all attributes of O which would scale to generate all those of $EF^{*\infty}(O)$. Hence $O' \notin \mathrm{fp}(EF^*) \cap \mathrm{fp}(PQ^*)$.

The second item has a similar proof: $EF^{*\infty} \circ PQ^{*\infty}(O) \preceq O$ can only occur if $PQ^{*\infty}(O) \in \mathrm{fp}(EF^*)$, i.e. $EF^{*\infty} \circ PQ^{*\infty}(O) = PQ^{*\infty}(O)$. Indeed, if this were not the case, then $EF^{*\infty}$ would generate attributes from $PQ^{*\infty}(O)$. However, since $EF^{*\infty} \circ PQ^{*\infty}(O) \preceq O$, these could only be attributes of O which were suppressed by $PQ^{*\infty}$ due to lack of support. But this is not possible because if they lacked support in O, there is not more support for them in $PQ^{*\infty}(O)$, which only reduces O. Thus, if $O' \in]EF^{*\infty} \circ PQ^{*\infty}(O)\ O]$, then $O' \in]PQ^{*\infty}(O)\ O]$. However, O' cannot be a fixed point for PQ^* because it contains less attributes than O: if these attributes lacked supports in O, they would still lack it in O'. Hence, $O' \notin \mathrm{fp}(EF^*) \cap \mathrm{fp}(PQ^*)$. □

This result cannot be generalised to the interval $]EF^{*\infty} \circ PQ^{*\infty}(O)\ PQ^{*\infty} \circ EF^{*\infty}(O)[$ as shown by Example 18 and Example 19.

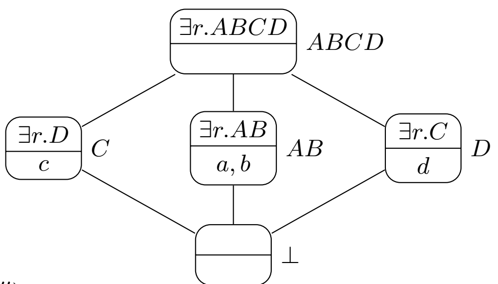
Example 18 (The subinterval may contain fixed points)**.** *In Example 17 (p. 67), $\langle L_3^\#, L_4^\# \rangle$ is not a fixed point for either EF^* or PQ^*. $PQ^{*\infty} \circ EF^{*\infty}(\langle L_3^\#, L_4^\# \rangle) = EF^{*\infty}(\langle L_3^\#, L_4^\# \rangle) = \langle L_3^\star, L_4^\star \rangle$ and $EF^{*\infty} \circ PQ^{*\infty}(\langle L_3^\#, L_4^\# \rangle) = PQ^{*\infty}(\langle L_3^\#, L_4^\# \rangle) = \langle L_3^1, L_4^1 \rangle$. $\langle L_3', L_4' \rangle \in \mathrm{fp}(EF^*) \cap \mathrm{fp}(PQ^*)$ and $\langle L_3', L_4' \rangle \in]EF^{*\infty} \circ PQ^{*\infty}(\langle L_3^\#, L_4^\# \rangle)\ PQ^{*\infty} \circ EF^{*\infty}(\langle L_3^\#, L_4^\# \rangle)[=]\langle L_3^1, L_4^1 \rangle\ \langle L_3^\#, L_4^\# \rangle[$ as can be observed in Figure 17.*

Example 19 (The subinterval may contain fixed points)**.** *The family of context-lattice pairs $O_0^\#$ of Example 16 (p. 66), is such that: (a) $EF^{*\infty}(O_0^\#) \neq PQ^{*\infty} \circ EF^{*\infty}(O_0^\#)$, (b) $PQ^{*\infty}(O_0^\#) \neq EF^{*\infty} \circ PQ^{*\infty}(O_0^\#)$, and (c) $EF^{*\infty} \circ PQ^{*\infty}(O_0^\#) \prec PQ^{*\infty} \circ EF^{*\infty}(O_0^\#)$.*

Note that none of $EF^{*\infty} \circ PQ^{*\infty}(O_0^\#)$ nor $PQ^{*\infty} \circ EF^{*\infty}(O_0^\#)$ are the bounds of $\mathrm{fp}(EF^*) \cap \mathrm{fp}(PQ^*)$ (O_0^1 and $O_0^\star$) contrary to Example 18.

Moreover, consider O_0'' defined as follows:

$k(O_0'')$	$\exists r.C$	$\exists r.D$	$\exists r.AB$	$\exists r.ABCD$
a			×	×
b			×	×
c		×		×
d	×			×



$l(O_0'')$

$O_0'' \in]EF^{*\infty} \circ PQ^{*\infty}(O_0^\#), PQ^{*\infty} \circ EF^{*\infty}(O_0^\#)[$ and $O_0'' \in \mathrm{fp}(EF^*) \cap \mathrm{fp}(PQ^*)$.

This counter-example is not sensible to conditions (a) and (b) above: (a) can be relaxed, by simply adding $\exists r.ABCD$ to $K_0^\#$, (b) can be relaxed by suppressing $\exists r.B$ from $K_0^\#$, and both can be relaxed together. In each of these cases, $EF^{*\infty} \circ PQ^{*\infty}(O_0^\#)$ and $PQ^{*\infty} \circ EF^{*\infty}(O_0^\#)$ will not be changed, preventing to discard the presence of acceptable solutions (O_0'') in the interval.

Proposition 74 can however be useful algorithmically. Indeed, if one considers the pairs of lattices $\langle L_3^\#, L_4^\# \rangle$ of Figure 17 or another non acceptable pair of lattices on the same line, then this result invalidates two pairs of lattices on the second and fourth line without testing them.

9 Conclusion

We addressed the questions of which family of lattices was returned by relational concept analysis and, more generally, which such families could be considered acceptable.

This report provides an answer to these questions by characterising the acceptable families of context-lattice pairs that describe a particular initial family of contexts as those families which are well-formed, saturated and self-supported. It identifies the results returned by relational concept analysis as the smallest element of this set. It also defines an alternative operation providing its greatest elements. We went further by characterising the structure of this set and those regularities that allows to navigate in it.

To that extent the report defined the set of well-formed objects $\mathscr{O}$, a function EF^*, generalising RCA, expanding a family, and a function PQ^* contracting a family. The fixed points of these functions characterise the saturated families and the self-supported families respectively. Hence, the acceptable solutions are those element of the intersection of the fixed points of such functions ($\text{fp}(EF^*) \cap \text{fp}(PQ^*)$).

These results rely fundamentally on the finiteness of the structure and monotony of the operations. Dealing with infinite structures would jeopardise the construction of D, however as soon as it preserves the termination of the application of the operations, this should not be a problem. Non-monotonic operations could be induced by non monotonic scaling operations in Ω. Such operations would prevent relational concept analysis to work properly and require fully different mechanisms.

In FCA, conceptual scaling is considered as a human-driven analysis tool: a knowledgeable person could provide attributes to be scaled 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. This is useful when generating a description logic TBox because all concepts are well-defined and necessary, but other contexts may benefit from exploiting other solutions.

Beyond the minimal common acceptable lattices returned by $\underline{RCA}$ and the most detailed ones that $\overline{RCA}$ returns, algorithms may be developed for returning all acceptable solutions [Atencia et al., 2021]. However, our work does not provide an "efficient" way to obtain all elements of this set. The characterisation of the structure of the space of acceptable solutions aims at contributing to this goal.

This work also opens perspectives for helping users to identify the acceptable solution that they prefer. Beyond generating all solutions, which may not be a convenient way to work, another option is to offer users the opportunity to guide the navigation among them. The structure of admissible solutions and the associated functions may be fruitfully exploited in order to help users finding an acceptable solution featuring the concepts and attributes they want and not unnecessary ones.

Finally, the position of relational concept analysis with respect to formal concept analysis and Galois connections would be worth investigating. On the one hand, this work shows that, contrary to other extensions that use scaling to encode a problem within FCA, RCA cannot be encoded in FCA. Indeed, RCA admits various fixed points contrary to FCA. RCA is not just a product or sequence of FCA, but relations between contexts introduce constraints between them leading to the possibility of alternative fixed points. Hence, an encoding would not be direct, so that it provides RCA solutions directly. On the other hand, other generalisations of FCA get closer to general Galois connections by extending the structure of attributes. The open question is whether RCA is another instance of a Galois connection extending FCA or if these two need a common generalisation.

Acknowledgements

This work has been partially funded by the ANR Elker project (ANR-17-CE23-0007-01). The author thanks Petko Valtchev for comments and suggestions on an earlier version of this work, Marianne Huchard for taking the pain to describe the trick used for forcing discrimination, Amedeo Napoli for explaining me RCA (and so many other things), Jérôme David for reviewing the text, and Philippe Besnard for pointing to the Knaster-Tarski theorem.

10 References

Atencia, M., David, J., Euzenat, J., Napoli, A., & Vizzini, J. (2020). Link key candidate extraction with relational concept analysis. *Discrete applied mathematics*, *273*, 2–20 (cit. on pp. 5, 12, 13).

Atencia, M., David, J., Euzenat, J., Napoli, A., & Vizzini, J. (2021). *Relational concept analysis for circular link key extraction* (Deliverable No. 1.2) (https://moex.inria.fr/files/reports/elker-1.2.pdf). Elker. (Cit. on p. 70).

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. (Eds.). (2003). *The description logic handbook: theory, implementations and applications*. Cambridge University Press. (Cit. on pp. 5, 12, 16).

Baader, F., & Distel, F. (2008). A finite basis for the set of $\mathcal{EL}$-implications holding in a finite model. *Proc. 6th International Conference on Formal Concept Analysis (ICFCA), Montréal (CA)*, *4933*, 46–61 (cit. on p. 16).

Belohlávek, R. (2008). *Introduction to formal concept analysis* (tech. rep.) (http://belohlavek.inf.upol.cz/vyuka/IntroFCA.pdf). Univerzita Palackého, Olomouc (CZ). (Cit. on p. 26).

Braud, A., Dolques, X., Huchard, M., & Le Ber, F. (2018). Generalization effect of quantifiers in a classification based on relational concept analysis. *Knowledge-based systems*, *160*, 119–135 (cit. on pp. 5, 12, 13).

Chaudron, L., & Maille, N. (2000). Generalized formal concept analysis. *Proc. 8th International Conference on Conceptual Structures (ICCS), Darmstadt (DE)*, *1867*, 357–370 (cit. on p. 15).

Dolques, X., Huchard, M., Nebut, C., & Reitz, P. (2012). Fixing generalization defects in UML use case diagrams. *Fundamenta informaticae*, *115*(4), 327–356 (cit. on p. 5).

Euzenat, J. (2021). Fixed-point semantics for barebone relational concept analysis. *Proc. 16th international conference on formal concept analysis (ICFCA), Strasbourg (FR)*, *12773*, 20–37 (cit. on pp. 6, 21, 28–31, 33, 35–42, 54).

Ferré, S., & Cellier, P. (2020). Graph-FCA: an extension of formal concept analysis to knowledge graphs. *Discrete applied mathematics*, *273*, 81–102 (cit. on p. 16).

Ferré, S., & Ridoux, O. (2000). A logical generalization of formal concept analysis. *Proc. 8th International Conference on Conceptual Structures (ICCS), Darmstadt (DE)*, *1867*, 371–384 (cit. on p. 14).

Ganter, B., & Kuznetsov, S. (2001). Pattern structures and their projections. *Proc. 9th International conference on conceptual structures (ICCS), Stanford (CA US)*, *2120*, 129–142 (cit. on p. 15).

Ganter, B., Stumme, G., & Wille, R. (Eds.). (2005). *Formal concept analysis: foundations and applications*. Springer. (Cit. on p. 5).

Ganter, B., & Wille, R. (1999). *Formal concept analysis: mathematical foundations*. Springer. (Cit. on pp. 5, 7, 10, 31).

Guigues, J.-L., & Duquenne, V. (1986). Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et sciences humaines*, *95*, 5–18 (cit. on p. 16).

Guimarães, R., Ozaki, A., Persia, C., & Sertkaya, B. (2023). Mining $\mathcal{EL}^{\bot}$ bases with adaptable role depth. *Journal of artificial intelligence research*, *76*, 119–135 (cit. on p. 16).

Kaytoue, M., Kuznetsov, S., Napoli, A., & Duplessis, S. (2011). Mining gene expression data with pattern structures in formal concept analysis. *Information science*, *181*(10), 1989–2001 (cit. on p. 15).

Keip, P., Ferré, S., Gutierrez, A., Huchard, M., Silvie, P., & Martin, P. (2020). Practical comparison of FCA extensions to model indeterminate value of ternary data. *Proc. 15th International Conference on Concept Lattices and Their Applications (CLA), Tallinn (EE)*, *2668*, 197–208 (cit. on p. 16).

Kötters, J. (2013). Concept lattices of a relational structure. *Proc. 20th International Conference on Conceptual Structures (ICCS), Mumbay (IN)*, *7735*, 301–310 (cit. on p. 16).

Kuznetsov, S. (2009). Pattern structures for analyzing complex data. *Proc. 12th International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing (RSFDGrC), Delhi (IN)*, *5908*, 33–44 (cit. on p. 15).

Missaoui, R., Kwuida, L., & Abdessalem, T. (Eds.). (2022). *Complex data analytics with formal concept analysis*. Springer. (Cit. on p. 5).

Nebel, B. (1990). *Reasoning and revision in hybrid representation systems* (Vol. 422). Springer. (Cit. on p. 26).

Ouzerdine, A., Braud, A., Dolques, X., Huchard, M., & Le Ber, F. (2019). Adjusting the exploration flow in relational concept analysis – an experience on a watercourse quality dataset. In R. Jaziri, A. Martin, M.-C. Rousset, L. Boudjeloud-Assala & F. Guillet (Eds.), *Advances in knowledge discovery and management* (pp. 175–198, Vol. 1004). Springer. (Cit. on p. 5).

Prediger, S. (1997). Logical scaling in formal concept analysis. *Proc. 5th International Conference on Conceptual Structures (ICCS), Seattle (WA US)*, *1257*, 332–341 (cit. on p. 12).

Rouane-Hacene, M., Huchard, M., Napoli, A., & Valtchev, P. (2013a). Relational concept analysis: mining concept lattices from multi-relational data. *Annals of Mathematics and Artificial Intelligence*, *67*(1), 81–108 (cit. on pp. 5, 11, 13, 16, 19, 35, 40, 42, 55).

Rouane-Hacene, M., Huchard, M., Napoli, A., & Valtchev, P. (2013b). Soundness and completeness of relational concept analysis. *Proc. 11h International Conference on Formal Concept Analysis (ICFCA)*, *7880*, 228–243 (cit. on pp. 5, 19, 29, 30, 34).

Tarski, A. (1955). A lattice-theoretical fixpoint theorem and its applications. *Pacific journal of mathematics*, *5*(2), 285–309 (cit. on p. 30).

Wajnberg, M. (2020). *Analyse relationnelle de concepts: une méthode polyvalente pour l'extraction de connaissance* [Doctoral dissertation, Université du Québec à Montréal & Université de Lorraine] [https://hal.archives-ouvertes.fr/tel-03042085]. (Cit. on pp. 8, 12).

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée

655 avenue de l'Europe Montbonnot

38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-0803