# How Hierarchies of Concept Graphs Can Facilitate the Interpretation of RCA Lattices?

Sébastien Ferré, Peggy Cellier

# How Hierarchies of Concept Graphs Can Facilitate the Interpretation of RCA Lattices?*

Sébastien Ferré and Peggy Cellier

Univ Rennes, CNRS, INSA, IRISA
Campus de Beaulieu, 35042 Rennes cedex, France
ferre@irisa.fr, peggy.cellier@irisa.fr

**Abstract.** Relational Concept Analysis (RCA) has been introduced in order to allow concept analysis on multi-relational data. It significantly widens the field of application of Formal Concept Analysis (FCA), and it produces richer concept intents that are similar to concept definitions in Description Logics (DL). However, reading and interpreting RCA concept lattices is notoriously difficult. Nica *et al* have proposed to represent RCA intents by cpo-patterns in the special case of sequence structures. We propose an equivalent representation of a family of RCA concept lattices in the form of a hierarchy of concept graphs. Each concept belongs to one concept graph, and each concept graph exhibits the relationships between several concepts. A concept graph is generally transversal to several lattices, and therefore highlights the relationships between different types of objects. We show the benefits of our approach on several use cases from the RCA litterature.

**Keywords:** Formal Concept Analysis, Relational Concept Analysis, Data Mining, Concept Graph

## 1 Introduction

Many domains produce multi-relational data. For example, in the health domain, one can have patients taking drugs, drugs giving some symptoms and interacting with other drugs, doctors taking care of patients and prescribing drugs to patients, etc. In order to extract knowledge from that kind of data, many data mining techniques, such as Formal Concept Analysis (FCA) [8], require the flattening of multi-relational data but this results in loss of structural information, and a more difficult interpretation of discovered patterns. It is therefore desirable to have direct methods for multi-relational mining [2]. Several generalizations of FCA have been proposed to handle relational data: Power Context Families [10], Relational and Logical Concept Analysis [7], Relational Concept Analysis (RCA) [11], and Graph-FCA [5]. RCA has so far been the most frequently used approach with applications in health [11,9] or model driven engineering [4].

* This research is supported by ANR project PEGASE (ANR-16-CE23-0011-08).

An important issue for the effective use of RCA is the interpretation of its outputs. Indeed, RCA produces not one but several concept lattices, and the intent of each concept may depend on the intent of other relationally-related concepts, recursively. To the best of our knowledge, there has been only one proposal to automatically extract and graphically represent the relational patterns that are buried into concept intents: Nica's cpo-patterns [9]. However, it has several restrictions. First, it is defined only for sequential data. Second, it generates cpo-patterns only for the concepts of one chosen lattice. Third, it generates a cpo-pattern for each concept of the chosen lattice, missing potential factorizations between patterns and thus interesting information between patterns.

In this paper, we propose a novel and generic graphical representation of RCA outputs that emphasizes the relational patterns. We call it *hierarchy of concept graphs*. It has the following good properties. First, it makes no assumption on the context family, and can therefore handle all kinds of graph structures, not only sequences. Second, it is at the same time a *complete* and *non-redundant* representation of the family of concept lattices, and does not require to choose one concept lattice as starting point. Third, it offers a better balance in the display between generalization ordering (lattice edges), and relationships (relational attributes). Fourth, it clusters concepts into *concept graphs*, and hence produce a coarser-grained representation. Fifth, it can be efficiently computed from the concept lattices, in linear time.

Section 2 discusses related work. Section 3 shortly recalls the main definitions of RCA. Section 4 introduces our representation of RCA outputs as a hierarchy of concept graphs, illustrates it on a reference example of RCA, and discusses its properties. Section 5 evaluates our approach on a few use cases, and discusses the impact of representation choices. Section 6 concludes and draws perspectives.

## 2 Related Work

Several generalizations of FCA have been proposed to handle relational data. Power Context Families [10] has a formal context for each relation arity, i.e. a context of objects, a context of couples of objects, a context of triples of objects, etc. A concept lattice is computed for each context, independently of other contexts. The resulting concepts are used as a vocabulary of types and relations to build *concept graphs* that are similar to Conceptual Graphs [12,1]. Relational and Logical Concept Analysis [7] takes as input a power context family limited to unary and binary relations but extended to complex logical descriptions. It generates a single concept lattice where concept intents combine both unary and binary descriptors, and where the labeling of the concept lattice is extended with relationships between concepts. Relational Concept Analysis (RCA) [11] takes as input a power context family limited to unary and binary relations. In practice, the unary context is split in several unary contexts, one for each type of object. RCA generates a concept lattice for each type of object, where concept intents are sets of classical attributes and relational attributes. The latter represent relationships to other concepts in the concept lattice family.

Graph-FCA [5] takes as input a power context family without restriction on arities. It generates a set of graph patterns where each node represents a unary concept, each pair of nodes represents a binary concept, etc. For each concept arity, the set of all concepts forms a concept lattice.

The above shows that there are two kinds of representations of the results: concept lattices and concept graphs. They complement each other: concept lattices emphasizes the generalization ordering between concepts, while concept graphs emphasize the relationship patterns between objects in data. In RCA, the native representation is made of concept lattices, and the relationship patterns are only indirectly accessible through relational attributes. Recently, Nica *et al* [9] have proposed a solution to combine concept lattices with graph patterns. However it is not a general solution for RCA because of the limitations already discussed in the introduction.

## 3 Relational Concept Analysis (RCA)

We here recall the definitions of *context family* and *lattice family* in RCA. We therefore focus on the input and output of RCA, and we ignore the methodology and algorithms that are used to compute the concept lattices from the context family. Indeed we are here concerned with the graphical representation of RCA lattices rather than on their computation. A detailed presentation of RCA is available in previous papers, in particular [11]. The input data of RCA is called a *relational context family* (RCF). In words, it is a collection of formal contexts, one for each kind of objects, together with a collection of binary relations going from the objects of one context to the objects of the same or another context.

**Definition 1.** *A* Relational Context Family *(RCF) is a pair* $(\boldsymbol{K}, \boldsymbol{R})$ *where:*

- $\boldsymbol{K} = \{\mathcal{K}_i\}_{i=1..n}$ *is a set of contexts* $\mathcal{K}_i = (O_i, A_i, I_i)$*, and*
- $\boldsymbol{R} = \{r_k\}_{k=1..m}$ *is a set of relations* $r_k$ *where* $r_k \subseteq dom(r_k) \times ran(r_k)$*, and* $dom(r_k), ran(r_k) \in \{O_i\}_{i=1..n}$ *are respectively the* domain *and* range *of* $r_k$*.*

As a running example, we reuse the RCF defined in [11] about pharmacovigilance of AIDS patients and drugs: $(\{\mathcal{K}_p, \mathcal{K}_d\}, \{takes, itb, iw\})$. Context $\mathcal{K}_p$ describes 4 patients in terms of age, gender, and observed Adverse Drug Reactions (ADR) (14 attributes). Context $\mathcal{K}_d$ describes 6 drugs in terms of active molecule, and expected ADR (16 attributes). Relation *takes* relates patients to the drugs they have taken. Relation *itb* ("is taken by") is the inverse of *takes*. Relation *iw* ("interacts with") relates couples of drugs that interact with each other (it is symmetric).

Given an RCF, the output of RCA is a collection of concept lattices, one for each context of the RCF. The relations of the RCF are taken into account by repeatedly applying a mechanism of relational scaling on each context and its concept lattice, until convergence is reached. This leads to the introduction of *relational attributes* that express relational constraints, and contribute to the formation of concepts (see [11] for more details).

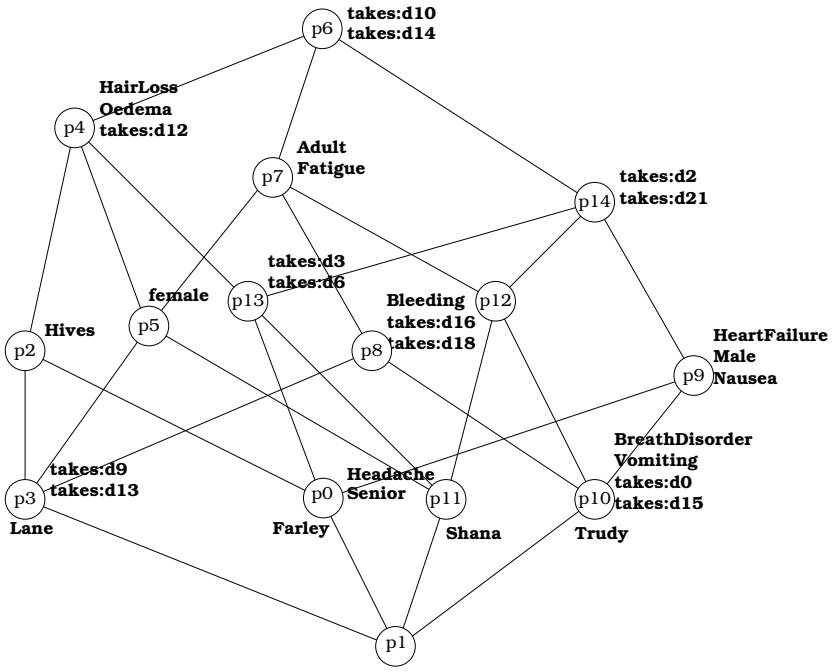


**Fig. 1.** The relational concept lattice of patients $\mathcal{L}_p$ (reduced labelling)

**Definition 2.** *Let* $(\boldsymbol{K}, \boldsymbol{R})$ *be a RCF. The* Relational Concept Lattice Family *(RCLF) is a set of concept lattices* $\boldsymbol{L} = \{\mathcal{L}_i\}_{i=1..n}$*, one for each context* $\mathcal{K}_i$*. Each concept* $c_i$ *in* $\mathcal{L}_i$ *is a pair* $(X, Y)$ *where:*

- $X \subseteq O_i$ *is the extent of the concept, and*
- $Y$ *is the intent of the concept, and contains attributes in* $A_i$ *and* relational attributes *in the form* $\rho\ r : c_j$ *where* $\rho \in \{\exists, \forall\exists, ...\}$ *is a scaling operator,* $r \in \boldsymbol{R}$*,* $dom(r) = O_i$*,* $ran(r) = O_j$*, and* $c_j \in \mathcal{L}_j$*.*

In this paper, we only consider existential scaling ($\rho = \exists$) even though our approach is applicable to other scaling operators. Figure 1 shows the concept lattice $\mathcal{L}_p$ of patients, and Figure 2 the concept lattice $\mathcal{L}_d$ of drugs. Both are represented with reduced labelling, i.e. each object/attribute appears only once. Object labels are placed below the concept, while attribute labels are placed above and on the right of the concept. Each relational attribute $\exists r : c_j$ is displayed as $r : c_j$.

The reading and interpretation of RCA lattices is notoriously difficult. The main reason is probably that reading the intent of a concept requires not only to traverse the lattice upward, as in FCA, but also to follow relationships to other concepts through the relational attributes. For example, concept $p7$ groups the adult patients who have fatigue, and take a drug in concept $d10$ and a drug in concept $d14$. Concept $d10$ groups the drugs for which diarrhea is expected, and

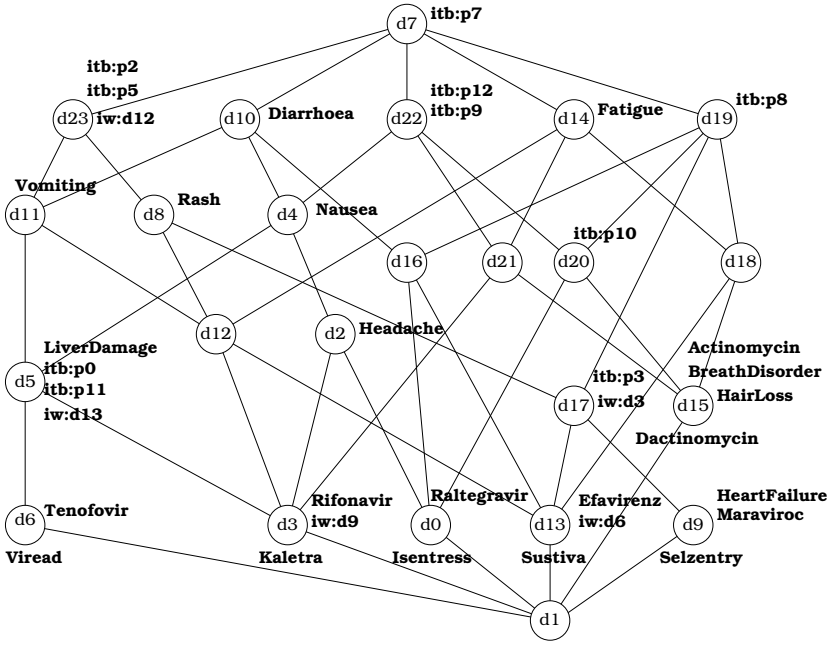


**Fig. 2.** The relational concept lattice of drugs $\mathcal{L}_d$ (reduced labelling)

which are taken by patients in concept $p7$. Loops in the exploration of the intent, like in that example, lead to circular definitions of concepts, and contribute to the difficulty of interpretation.

## 4 Hierarchies of Concept Graphs

Our objective is to facilitate the reading of the intent of RCA concepts, in order to facilitate their interpretation. The first idea is to display the different lattices side-by-side, and to materialize each relational attribute $r : c_2$ on a concept $c_1$ as a *relational edge*, i.e. a labeled and directed edge $c_1 \xrightarrow{r} c_2$. However, the graphical representation becomes denser and even less readable; and its structure is dominated by the lattice structures at the cost of elongated relational edges. A better balance between lattice edges and relational edges is desirable. The second idea is to identify *relational structures* as subsets of interrelated concepts from different lattices, and to use them as building blocks in the graphical representation. We propose to define those relational structures as the Strongly Connected Components (SCC) [3] of the *dependency graph* between concepts. The intuition behind that dependency graph is that the intent of a concept depends on its ancestors in the lattice, and also on the target concepts of relational attributes.

*Dependency Graph.* We first define the notion of concept dependency and then the dependency graph of a RCLF.

**Definition 3.** *Let* $\boldsymbol{L}$ *be a RCLF and* $c_1$, $c_2$ *be two concepts in* $\boldsymbol{L}$*. Concept* $c_1$ *depends on concept* $c_2$*, denoted by* $c_1 \rightarrow c_2$*, if:*

- $c_2$ *is a parent concept of* $c_1$ $(c_1 \leq c_2)$*,*
- *or* $c_1$ *is labeled by a relational attribute* $\rho\, r : c_2$*.*

**Definition 4.** *Let* $\boldsymbol{L}$ *be a RCLF. The* dependency graph *of* $\boldsymbol{L}$ *is the directed graph* $G_{\boldsymbol{L}} = (V, E)$ *where:*

- $V$ *is the set of all concepts of all lattices in* $\boldsymbol{L}$ *except bottom concepts,*
- *and* $E = \{(c_1, c_2) \mid c_1, c_2 \in V \text{ and } c_1 \rightarrow c_2\}$*.*

*SCCs as Concept Graphs.* From there, a SCC of $G_{\boldsymbol{L}}$ is a maximal set of concepts (possibly from several lattices) where each concept has a dependency path to all other concepts in the SCC for the definition of its intent. SCCs are used to define *concept graphs*, which are the building blocks of our graphical representation.

**Definition 5.** *A* concept graph *is the subgraph of the lattice family (enriched with relational edges) that is induced by a SCC of* $G_{\boldsymbol{L}}$*. It therefore mixes concepts from several lattices, and both lattice edges and relational edges.*

In Figure 3, each rounded box contains a concept graph (G1-G6). Nodes are concepts from the two RCA lattices in Figures 1 and 2 (same id, same labels). Relational edges (arrows) replace the relational attributes to graphically represent the relational dependencies. Lattice edges that cross concept graph boundaries are displayed as dotted lines to keep the graph light, and to emphasize the concept graphs over the global lattice structures. It is notable in this example that no relational edge crosses concept graph boundaries. This is because in the context family each relation either has an inverse relation (e.g., *takes* and *itb*) or is a symmetric relation (e.g., *iw*).

Furthermore, it is known that the SCCs of a graph form a directed acyclic graph, where $SCC_1 \rightarrow SCC_2$ if any concept in $SCC_1$ depends on any concept in $SCC_2$. The concept graphs of a RCLF can therefore be organized into a *hierarchy of concept graphs*. For example, concept graph G2 is a child of concept graph G1 in Figure 3 because several concepts in G2 have lattice edges (dotted lines) to concepts in G1: e.g., from p4 to p6, or from d12 to d14. Those hierarchical relationships can be seen as a complex version of the lattice edges, combining several lattice edges across different lattices.

*Interpretation.* We here give a short interpretation of the hierarchy of concept graphs in Figure 3. G1 represents the most general pattern between patients and drugs. It shows that all patients take drugs expected to give fatigue and diarrhea, and that all drugs are taken by an adult with fatigue. G2-4 are specializations of G1. For example, G4 specializes G1 to patients with bleeding. G2 specializes G1 to patients with hairloss and oedema, which all take drugs giving vomiting and rash, which are taken by a patient with hives, and by a female patient. G5 and G6 represent patterns that are specific to individual patients and drugs,

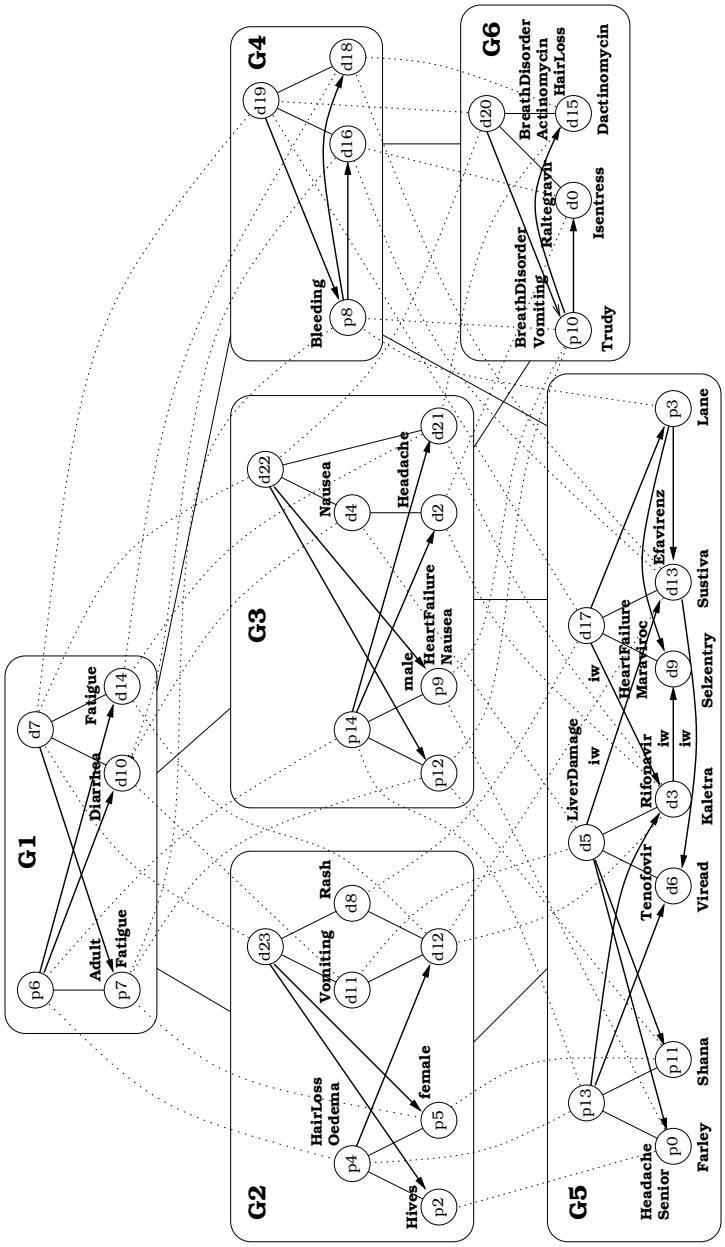


**Fig. 3.** Hierarchy of concept graphs: arrows represent relational attributes (*takes* from patients to drugs, *itb* from drugs to patients, *iw* between drugs), dotted lines represent lattice edges across concept graphs

and are therefore less interesting from a data-mining perspective. G5 shows that some patients (p13) take drugs (d6 and d3) that cause liver dammage (d5) and are in interaction with drug Sustiva (d13) which is taken by patient Lane (p3).

*Discussion.* The hierarchy of concept graphs has a number of good theoretical properties. First, it is a *complete* representation because it keeps all concepts and edges from the concept lattices. It is also parcimonious in that it does not duplicate any concept, edge, or label. Second, it is more *readable* because it displays relation attributes as relation edges, and because its layout offers a better balance between lattice edges and relational edges. Moreover, when the many concepts are clustered in a small number of concept graphs, the RCLF can be read at a higher level of granularity. Third, it is *efficient* to compute because the SCCs can be extracted in time linear with the size of $G_{\boldsymbol{L}}$, and hence in the cumulated size of lattices in $\boldsymbol{L}$.

## 5 Use Cases

In this section we present two use cases in order to compare graph concepts with results of Graph-FCA and cpo-patterns from [9]. The first one describes the royal family. The second use case is about flu patients and medical examinations.

### 5.1 Royal Family: Genealogical Data

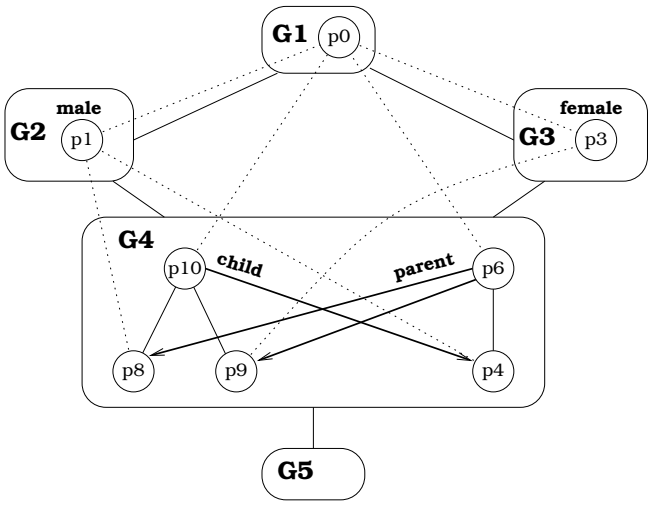


**Fig. 4.** Hierarchy of concept graphs about the royal family (G5 not detailed)

The first use case is the one used for Graph-FCA [6]. It describes a subset of the British royal family: Charles, Diana, William, Harry, Kate,

George, and Charlotte. The power context family uses two attributes (*male* and *female*), and two relations (*parent* and its inverse *child*). Note that the relations have the same object type as domain and range: people. RCA produces one lattice containing 19 concepts. Figure 4 shows the hierarchy of concept graphs obtained from that lattice with our approach. Concepts are clustered in 5 concept graphs. G5 is not detailed because it contains very specific concepts, and hence does not bring new knowledge. The hierarchy of concept graphs enables to reach the following interpretations for each concept:

| | |
|---|---|
| p0 people | p8 male parents (fathers) |
| p1 male people (men) | p9 female parents (mothers) |
| p3 female people (women) | p6 people with a father and mother (children) |
| p10 people with a child (parents) | p4 male children (sons) |

The concept of "daughters" has a single instance (Charlotte), and is found in G5, a specialization of G4. The relation from p10 (parents) to p4 (sons) shows that, in the context, every parent has a son but not necessarily a daughter. The relations froms p6 to p8 and p9 shows that, in the context, every child who has a known father also has a known mother, and reciprocally. Concept graph G4 exhibits the relational pattern of a nuclear family, relating children to their father and mother as parents, with the specificity in this context that all parents have a son.

Comparing those results to Graph-FCA, it is interesting to note that the graph patterns of Graph-FCA are equivalent to the RCA concept graphs, up to a few representation changes. Graph-FCA patterns only represent relational edges, not lattice edges. The generalization ordering between concepts and patterns is therefore not explicitly represented. In Graph-FCA the use of inverse relations is implicit so that the *child* relation is redundant with the *parent* relation. Note that Graph-FCA also defines n-ary concepts such as "couple" or "sibling". The equivalence with Graph-FCA on this example must not be generalized. In fact, it does not hold on the running example about patients and drugs.

### 5.2 Medical Histories and Comparison to cpo-Patterns

The second use case is the one used for cpo-Patterns [9]. It describes flu patients through their symptoms, their viral tests and their medical examinations. The specificity of that dataset is the sequentiality of the data. For instance for patient $p1$ we know that a viral test on 28/09 is preceded by a medical examination on 26/09 which is also preceded by another medical examination on 25/09. The power context family uses 6 symptom attributes (COUGHmoderate, FEVERmoderate, ?moderate, COUGHhigh, FEVERhigh, ?high) and two relations: RME-ipb-ME (sequential relation between medical examinations), RVT-ipb-ME (sequential relation from viral tests to medical examinations). It describes five viral tests and ten medical examinations. RCA produces two lattices. The viral test lattice contains 12 concepts and the medical examination lattice contains 18 concepts.

Figure 5 shows the hierarchy of concept graphs obtained from those lattices with our approach. We note that it is a special case, indeed each concept is a

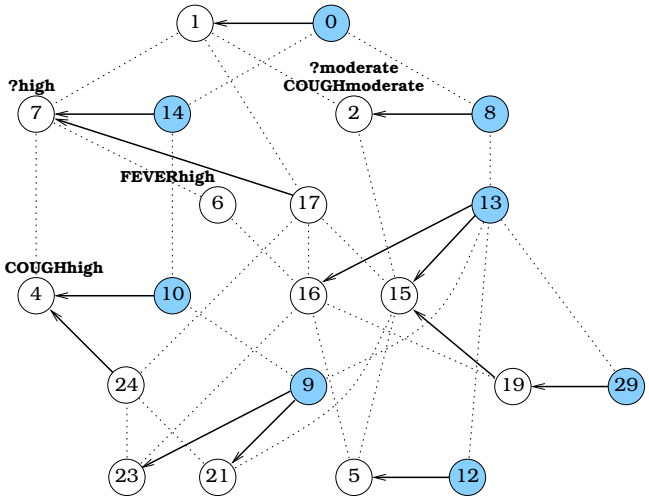


**Fig. 5.** Hierarchy of concept graphs about medical histories. Each concept is a concept graph on its own. Greyed concepts are about viral tests, others are about medical examinations. Only concepts with support greater than one are shown.

concept graph by its own. It is due to the fact that the relation "is-preceded-by" has no inverse relation, and forms no cycle because of its sequential nature. It is thus impossible to find more than one concept in a strongly connected component. The greyed concepts are the concepts from the viral test lattice, other concepts come from the medical examination lattice. In the graph only concepts with support greater than one are shown. We can read in the hierarchy that all viral tests (top concept 0) are preceded by a medical examination (top concept 1). That relational pattern has two specialisations. The first one where the symptom during the examination is moderate cough (concepts 2 and 8). The second one where there is a high symptom (concepts 7 and 14). We can also note that parts of sequential patterns are shared by several concepts. For instance, concept 24 and 10 are preceded by a medical examination with high cough (concept 4). In fact, concept 9 specializes concept 10 by inserting between the viral test and the high cough (concept 4) two additional medical examinations (concepts 21 and 23), which are themselves specialisations of concept 24. It highlights the overlaps between sequential patterns.

We have also conducted experiments when considering the inverse relation of "is-preceded-by" (ipb), i.e. "is-followed-by" (ifb). The power context family is thus extended with two relations: RME-ifb-ME and RME-ifb-VT. RCA still produces two lattices but the hierarchy of concept graphs is different. Indeed, eleven concept graphs are extracted. Each of them contains several concepts and only one viral test concept. Figure 6 shows an excerpt of the hierarchy of concept graphs with those inverse relations. For the sake of readability and compactness, we modified the representation of concept graphs in two ways: (a) only the most

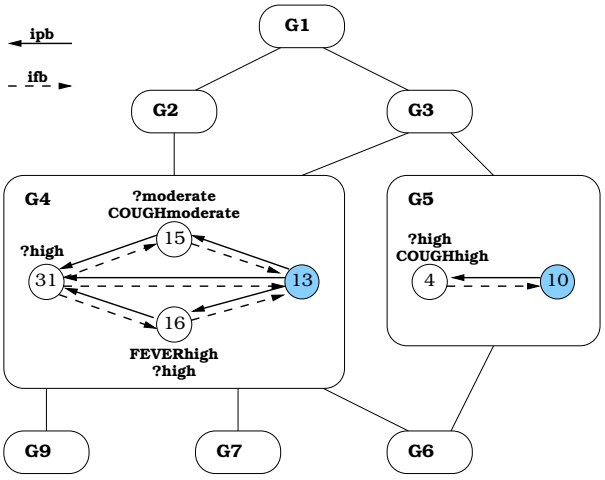


**Fig. 6.** Hierarchy of concept graphs about medical histories with inverse relation *ifb*. Only concept graphs whose concepts have their support greater than one are shown.

specific concepts of a concept graph are kept, and (b) the full intent of those concepts is shown, instead of the reduced intent, so that each concept graph can be read in isolation. Eight concept graphs among the eleven are shown and only two of them (G4 and G5) are detailled in the figure. Concept graph G4 can be read as "a viral test (concept 13) preceded by a medical examination with moderate cough (concept 15) and a medical examination with high fever (concept 16) and both of them are preceded by a medical examination with a high symptom (concept 31)".

The interesting result is that the eleven concept graphs match exactly the eleven cpo-patterns extracted by [9]. However, there are some differences in the display of the patterns. Indeed, in order to compute the strongly connected components, the inverse relations have to be added and they appear in the result. For instance between concepts 13 and 15 there are two arrows, one in each direction, because the relation is-followed-by (ifb) is the inverse relation of is-preceded-by (ipb). In the same vein, ifb and ipb are transitive relations, and thus some arrows are redundant. For example, the arrows between concepts 13 and 31 can be deduced from the paths through concepts 15, and 16 by transitivity. In [9], the representation was specialized for sequential data, and so that kind of redanduncies were avoided. On the contrary, our approach is general and allows to take into account any kind of relations without any assumption on them. In order to avoid those redundancies the description language of power context families should be modified in order to add a way to specify relation properties (e.g., is transitive, is symmetric, has an inverse), and then it should be taken into account when computing and displaying the concept graphs. The same can be said for the redundancy on attributes. Indeed, when looking at the intent of concept 15, we can note the redundancy between "?moderate" and

"COUGHmoderate". It is due to the conceptual scaling used on the symptom attributes. By taking into account the hierarchy between attributes, the display of the concept graphs can be simplified without loosing information.

## 6 Conclusion and Perspectives

We have proposed a novel and general representation of RCA concept lattices, called *hierarchy of concept graphs*, in order to facilitate their interpretation. The key idea is to exhibit relational patterns by having a better balance in the display between lattice edges and relational edges. Each concept graph clusters a set of concepts (from different lattices) whose intents are mutually dependent, and exhibits a relational pattern. Concept graphs are organized into a hierarchy so that generalization ordering between concepts is lifted to concept graphs. As future work, we plan to study the impact of relation properties (e.g., inverse, transitivity) on the trade-off between the number of concept graphs and the size of each concept graph. It will also be necessary to develop tools for the dynamic visualization of large hierarchies of concept graphs, *à la* Conexp[1].

## References


1. Chein, M., Mugnier, M.L.: Graph-based knowledge representation: computational foundations of conceptual graphs. Advanced Information and Knowledge Processing, Springer (2008)
2. Džeroski, S.: Relational Data Mining, pp. 887–911. Springer US, Boston, MA (2010), https://doi.org/10.1007/978-0-387-09823-4_46
3. Even, S.: Graph algorithms. Cambridge University Press (2011)
4. Falleri, J.R., Arévalo, G., Huchard, M., Nebut, C.: Use of model driven engineering in building generic fca/rca tools. In: CLA. vol. 7, pp. 225–236 (2007)
5. Ferré, S.: A proposal for extending formal concept analysis to knowledge graphs. In: Baixeries, J., Sacarea, C., Ojeda-Aciego, M. (eds.) Int. Conf. Formal Concept Analysis (ICFCA). pp. 271–286. LNCS 9113, Springer (2015)
6. Ferré, S., Cellier, P.: Graph-FCA in practice. In: Haemmerlé, O., et al. (eds.) Int. Conf. Conceptual Structures (ICCS). pp. 107–121. LNCS 9717, Springer (2016)
7. Ferré, S., Ridoux, O., Sigonneau, B.: Arbitrary relations in formal concept analysis and logical information systems. In: ICCS (2005)
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York. (1999)
9. Nica, C., Braud, A., Dolques, X., Huchard, M., Le Ber, F.: Extracting hierarchies of closed partially-ordered patterns using relational concept analysis. In: Int. Conf. Conceptual Structures. pp. 17–30. Springer (2016)
10. Prediger, S.: Simple concept graphs: A logic approach. In: Int. Conf. Conceptual Structures. pp. 225–239. LNCS 1453 (Aug 1998)
11. Rouane-Hacene, M., Huchard, M., Napoli, A., Valtchev, P.: Relational concept analysis: mining concept lattices from multi-relational data. Annals of Mathematics and Artificial Intelligence 67(1), 81–108 (2013)
12. Sowa, J.: Conceptual structures. Information processing in man and machine. Addison-Wesley, Reading, US (1984)


[1] http://conexp.sourceforge.net/