Concept Lattices of a Relational Structure

Jens Kötters

Abstract. Conceptual patterns can be described by graphs, entailment by graph homomorphism. The mapping of a pattern to its set of instantiations, represented as a table, constitutes one half of a Galois connection. The join operation is the infimum in a complete lattice of tables, and a most descriptive pattern can be assigned to each table by means of a categorial product construction. This construction constitutes the other half of the Galois connection. In this approach, relational structures assume the role of formal contexts in standard Formal Concept Analysis (FCA). Concepts arise as connected components of powers of these relational structures. The ordered set of these concepts may be conceived as a navigation space.

Keywords: Formal Concept Analysis, Relational Structures, Category Theory, Databases.

1 Introduction

The idea of using concept lattices to browse data can be traced back to [7]. In [7], a set of attributes is considered a query, and the set of objects having all the attributes (which is a concept extent) is the corresponding result set. The downward (upward) edges in a lattice's line diagram indicate the ways in which a query can be refined (weakened) to effect a minimal change in the result set. The capability to successively modify queries in this fashion is thought to make data more accessible to the information seeker.

More advanced applications of lattice-based browsing make use of conceptual scales to incorporate and distinguish between different types of values in the data. Relational scales can be used to account for inter-object relations in the data. The reader is referred to [1,3] for recent applications that treat relational data. In this paper, we describe mathematically a navigation space akin to those underlying the mentioned systems, but it is obtained directly from a relational structure and not by means of relational scaling.

For an example, consider the family tree in Fig. 1. The nodes represent the family members A(nne), B(ob), C(hris), D(ora) and E(mily). The arcs say that Anne is the m(other) of Bob and Chris, and Bob is the f(ather) of Dora and Emily. This graph defines (and visually represents) a relational structure $\mathcal{F}$ with underlying set $\{A, B, C, D, E\}$, unary relations ♂ and ♀ and binary relations m and f. We assume that $\mathcal{F}$ also has a p(arent) relation p, which is not drawn here. In Fig. 2, the nine nodes arranged as a cube form a particular concept lattice.

The concept intents, drawn to the right of each concept, are relational structures representing conjunctive queries [2]. The black nodes designate the subject(s)

H.D. Pfeiffer et al. (Eds.): ICCS 2013, LNAI 7735, pp. 301–310, 2013.

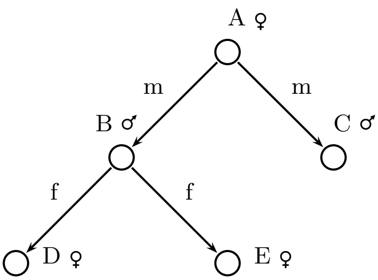

Fig. 1. Example: family tree $\mathcal{F}$

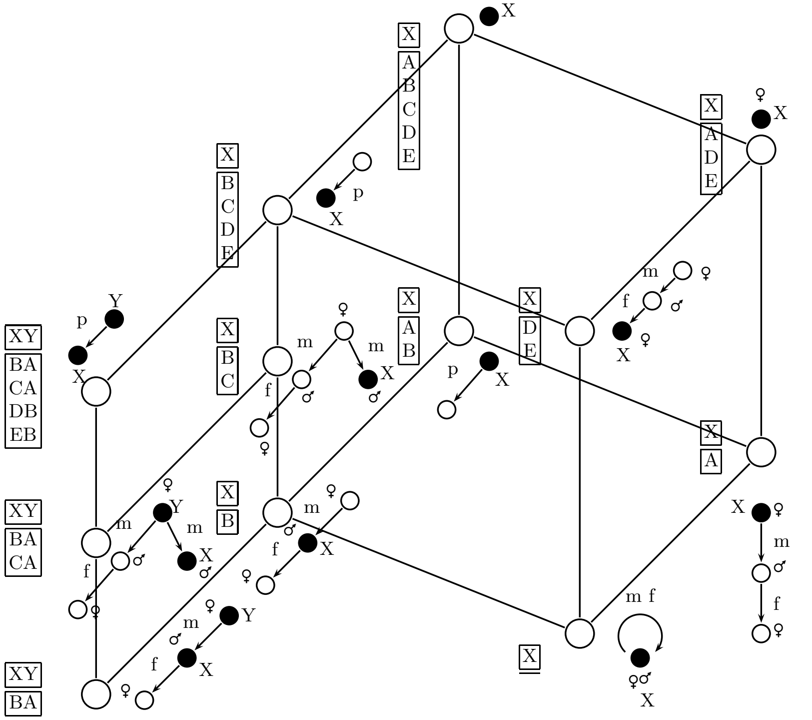

Fig. 2. Example: concept lattice $\mathfrak{C}_{\mathcal{F}}[\{x\}]$ (plus three concepts from $\mathfrak{C}_{\mathcal{F}}[\{x,y\}]$)

of a query, variable names are assigned to them (technically by a function ν). The white nodes correspond to existentially quantified variables. The top concept and its lower neighbors (from left to right) are 'person', 'child', 'parent' and 'female'. The concept extents, drawn to the left of each concept, are result sets of the intent queries. Each result arises from a homomorphism from the intent to $\mathcal{F}$. Consider now the three lower neighbors of 'child'. The right one can be identified with 'granddaughter', but note that the most precise description (the intent) tells us more. The middle one can *not* be identified with 'uncle' because we do not (and can not) express that the males are different persons. We could make up a name "parentship" for the left one, the intent has two free variables x, y and the extent consists of all instances of 'parentship'. This concept does not belong to the "cube" $\mathfrak{C}_{\mathcal{F}}[\{x\}]$ (the lattice of all concepts definable with one free variable x), it belongs to the concept lattice $\mathfrak{C}_{\mathcal{F}}[\{x, y\}]$. Only three concepts of $\mathfrak{C}_{\mathcal{F}}[\{x, y\}]$ are shown in Fig. 2.

Graphs are a natural candidate for the formal representation of queries over RDF. Chandra and Merlin's result on query optimization by graph folding [2] may exemplify on a more general level the benefits of such representation. Pattern Structures [5] formalize the idea of representing concept intents by some kind of pattern rather than by an attribute set. In [5], the authors mention Conceptual Graphs and formalize chemical graphs (see also [4]) as examples for patterns. The notions of homomorphism which accompany these graphs make their approach seem very similar to the one presented here. A difference is that in [5], extents are still sets of objects, while here we use tables for their representation (although one-column tables are naturally identified with object sets). One could argue that such a construction is no longer a concept lattice, but in fact we have identified concepts in the foregoing example.

We will stipulate that concepts are described by connected graphs. However, the Galois connection in Sect. 5 extends to all windowed structures (i.e. conjunctive queries, Sect. 3), it does not harm to allow even infinite ones. The concept lattices lie embedded in the complete lattice that arises from this Galois connection (Sect. 6). Before the Galois connection is defined, the preordered class of windowed structures (Sect. 3) and the complete lattice of tables (Sect.4) are introduced independent of each other.

2 Preliminaries

A relational signature is a set $\mathcal{S}$ of relation symbols. The arity of a symbol $R \in \mathcal{S}$ is a natural number $|R| \geq 1$. A relational structure over the signature $\mathcal{S}$, also called an $\mathcal{S}$-structure, is a pair $\mathcal{A} = (A, (\mathcal{A}(R))_{R\in\mathcal{S}})$, where $\mathcal{A}(R) \subseteq A^{|R|}$ for all $R \in \mathcal{S}$. The set A is called the underlying set of $\mathcal{A}$ and can also be denoted by $|\mathcal{A}|$. A homomorphism from an $\mathcal{S}$-structure $\mathcal{G}_1$ to an $\mathcal{S}$-structure $\mathcal{G}_2$ is a map $\varphi : |\mathcal{G}_1| \to |\mathcal{G}_2|$ such that for all $R \in \mathcal{S}$ and $(x_1, \ldots, x_{|R|}) \in \mathcal{G}_1(R)$ we have $(\varphi(x_1), \ldots, \varphi(x_{|R|})) \in \mathcal{G}_2(R)$.

The product $\prod_{i\in I}\mathcal{G}_i$ of $\mathcal{S}$-structures is defined by

$$|\prod_{i\in I}\mathcal{G}_i| = \bigtimes_{i\in I}|\mathcal{G}_i|$$

and, for all symbols R of $\mathcal{S}$,

$$(x_1,\dots,x_{|R|}) \in (\prod_{i\in I}\mathcal{G}_i)(R) \;:\Leftrightarrow\; \forall i \in I : (x_1(i),\dots,x_{|R|}(i)) \in \mathcal{G}_i(R)\ .$$

The I-th power of a structure $\mathcal{G}$ is the product $\bigtimes_{i\in I}\mathcal{G}$ and is denoted by $\mathcal{G}^I$. The product of relational structures is a product in the sense of category theory:

Proposition 1. *Let $(\mathcal{G}_i)_{i\in I}$ be a family of $\mathcal{S}$-structures. Then each projection $\pi_i : \bigtimes_{i\in I}|\mathcal{G}_i| \to |\mathcal{G}_i|$ is a homomorphism from $\prod_{i\in I}\mathcal{G}_i$ to $\mathcal{G}_i$. Furthermore, for each $\mathcal{S}$-structure $\mathcal{H}$ and family $(\varphi_i : \mathcal{H} \to \mathcal{G}_i)_{i\in I}$, there exists a unique $\varphi : \mathcal{H} \to \prod_{i\in I}\mathcal{G}_i$ such that $\varphi_i = \pi_i \circ \varphi$ for all $i \in I$.*

When we talk about the nodes of $\mathcal{A}$, what we mean are the elements of $|\mathcal{A}|$. A sequence $(a_1,\dots,a_n)$ of nodes of $\mathcal{A}$ is called a path from a_1 to a_n in $\mathcal{A}$, if for all $1 \le i < n$ there exists an $R \in \mathcal{S}$ such that $\{a_i, a_{i+1}\} \subseteq \{x_1,\dots,x_{|R|}\}$ for some $(x_1,\dots,x_{|R|}) \in \mathcal{A}(R)$. We call a structure $\mathcal{A}$ connected if there exists a path from a to b for all $a, b \in |\mathcal{A}|$. We define an equivalence relation

$$a \sim b \;\Leftrightarrow\; \text{there exists a path from } a \text{ to } b$$

over $|\mathcal{A}|$. A connected component of $\mathcal{A}$, or simply a component of $\mathcal{A}$, is an $\mathcal{S}$-structure $\mathcal{C}$ for which $|\mathcal{C}|$ is a class of $\sim$ and $\mathcal{C}(R) = \mathcal{A}(R) \cap |\mathcal{C}|^{|R|}$ for all $R \in \mathcal{S}$.

Throughout the paper, we will use Var to denote a countably infinite set of variables. By ι (or any variety such as $\tilde{\iota}, \iota_1, \dots$) we shall always denote an inclusion map from some set X_1 to some set X_2, i.e. a map with $\iota(x) = x$ for all $x \in X_1$, where $X_1 \subseteq X_2$ is implied. The sets X_1 and X_2 will be clear from the context.

3 Windowed Structures

Definition 1. *Let $\mathcal{S}$ be a relational signature. A **windowed $\mathcal{S}$-structure** is a triple $(X, \nu, \mathcal{G})$ consisting of a set $X \subseteq \mathrm{Var}$, an $\mathcal{S}$-structure $\mathcal{G}$ and a map $\nu : X \to |\mathcal{G}|$.*

Definition 2. *Let $W_1 = (X_1, \nu_1, \mathcal{G}_1)$ and $W_2 = (X_2, \nu_2, \mathcal{G}_2)$ be windowed $\mathcal{S}$-structures. A **homomorphism** $\varphi : W_1 \to W_2$ **of windowed structures** is a structure homomorphism $\varphi : \mathcal{G}_1 \to \mathcal{G}_2$ with $\varphi \circ \nu_1 = \nu_2 \circ \iota$, where $X_1 \subseteq X_2$ is assumed.*

Homomorphisms of windowed $\mathcal{S}$-structures are closed under composition. Also, the identity id : $|\mathcal{G}| \to |\mathcal{G}|$ is a homomorphism of any windowed $\mathcal{S}$-structure $(X, \nu, \mathcal{G})$ onto itself. These two facts imply that windowed $\mathcal{S}$-structures with homomorphisms form a category. Furthermore, they imply that the following relation on the class of windowed $\mathcal{S}$-structures is a preorder:

Definition 3 (Homomorphism Preorder). *For windowed $\mathcal{S}$-structures W_1 and W_2, we set*

$$W_1 \lesssim W_2 \;:\Leftrightarrow\; \exists\, \varphi : W_1 \to W_2 \;\; .$$

The homomorphism preorder induces an equivalence relation on the class of windowed $\mathcal{S}$-structures:

Definition 4 (Homomorphic Equivalence). *For windowed $\mathcal{S}$-structures W_1 and W_2, we set*

$$W_1 \simeq W_2 \;:\Leftrightarrow\; W_1 \lesssim W_2 \wedge W_2 \lesssim W_1 \;\; .$$

Definition 5. *The* ***product*** *of a family $((X_i, \nu_i, \mathcal{G}_i))_{i \in I}$ of windowed $\mathcal{S}$-structures is the windowed $\mathcal{S}$-structure*

$$\prod_{i \in I} (X_i, \nu_i, \mathcal{G}_i) := (\bigcap_{i \in I} X_i, \nu_I, \prod_{i \in I} \mathcal{G}_i) \;\; ,$$

where $\nu_I(x) := (\nu_i(x))_{i \in I}$.

The product of the empty family is $(\text{Var}, \nu_\emptyset, (\emptyset, (\{\emptyset\}^{|R|})_{R \in \mathcal{S}}))$, where $\nu_\emptyset(x) = \emptyset$ for all $x \in \text{Var}$.

The product is indeed a product in the category theoretical sense, as the following proposition shows:

Proposition 2. *Let $((X_i, \nu_i, \mathcal{G}_i))_{i \in I}$ be a family of windowed $\mathcal{S}$-structures. Each projection $\pi_i : \bigtimes_{i \in I} |\mathcal{G}_i| \to |\mathcal{G}_i|$ is a homomorphism from $\prod_{i \in I}(X_i, \nu_i, \mathcal{G}_i)$ to $(X_i, \nu_i, \mathcal{G}_i)$. Furthermore, for each windowed $\mathcal{S}$-structure $(Y, \mu, \mathcal{H})$ and family $(\varphi_i : (Y, \mu, \mathcal{H}) \to (X_i, \nu_i, \mathcal{G}_i))_{i \in I}$, a unique $\varphi : (Y, \mu, \mathcal{H}) \to \prod_{i \in I}(X_i, \nu_i, \mathcal{G}_i)$ exists such that $\varphi_i = \pi_i \circ \varphi$ for all $i \in I$.*

Proof. From Prop. 1 we obtain $\pi_i : \prod_{i \in I} \mathcal{G}_i \to \mathcal{G}_i$ for $i \in I$. The definition of ν_I provides $\pi_i \circ \nu_I = \nu_i \circ \iota_i$ for all $i \in I$ (see the right circuit in Fig. 4). This proves the first claim. Now let $(\varphi_i : (Y, \mu, \mathcal{H}) \to (X_i, \nu_i, \mathcal{G}_i))_{i \in I}$ be a family of homomorphisms on some $(Y, \mu, \mathcal{H})$. In particular, we have $\varphi_i \circ \mu = \nu_i \circ \tilde{\iota}_i$ for all $i \in I$ (outer circuit). Also, Y must be a subset of each X_i, and so we have an inclusion map $\iota : Y \to \bigcap_{i \in I} X_i$. The equations $\tilde{\iota}_i = \iota_i \circ \iota$ (upper circuit) hold trivially. Again from Prop. 1 we obtain $\varphi : \mathcal{H} \to \prod_{i \in I} \mathcal{G}_i$ with $\varphi_i = \pi_i \circ \varphi$ (lower circuit). Altogether, we obtain

$$\pi_i \circ \varphi \circ \mu = \varphi_i \circ \mu = \nu_i \circ \tilde{\iota}_i = \nu_i \circ \iota_i \circ \iota = \pi_i \circ \nu_I \circ \iota$$

for all $i \in I$, and thus $\varphi \circ \mu = \nu_I \circ \iota$ (left circuit). Note that this last equation can not be inferred from the commutativity of the diagram!

We have shown that φ is a homomorphism from $(Y, \mu, \mathcal{H})$ to $\prod_{i \in I}(X_i, \nu_i, \mathcal{G}_i)$. From $\varphi_i = \pi_i \circ \varphi$ it follows that $\varphi(x) := (\varphi_i(x))_{i \in I}$, so φ is unique. □

The coproduct $\coprod_{i \in I}(X_i, \nu_i, \mathcal{G}_i)$ of windowed graphs is identical to a pushout of $\mathcal{S}$-structures, if all X_i are the same. In the general case, it is constructed from the disjoint union of the $\mathcal{G}_i$, $i \in I$, by identifying all nodes $\nu_i(x)$ and $\nu_j(x)$ where $x \in X_i \cap X_j$.

The product and coproduct can be understood as infimum and supremum in the homomorphism preorder. This is made precise in the corollary:

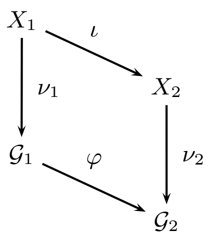

Fig. 3. Windowed graph morphism

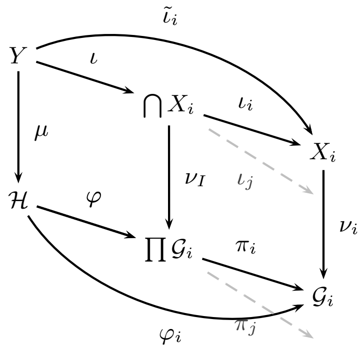

Fig. 4. Product of windowed graphs

Corollary 1. *Let $(W_i)_{i\in I}$ be a family of windowed $\mathcal{S}$-structures. The following equivalences hold for all windowed $\mathcal{S}$-structures W:*

$$W \lesssim \prod_{i\in I} W_i \;\Leftrightarrow\; \forall i \in I : W \lesssim W_i \ , \tag{1}$$

$$\coprod_{i\in I} W_i \lesssim W \;\Leftrightarrow\; \forall i \in I : W_i \lesssim W \ . \tag{2}$$

4 Tables

Definition 6. *Let G be a set. A **table over** G is a pair (X, Λ), where $X \subseteq \mathrm{Var}$ and $\Lambda \subseteq G^X$. The set of all tables over G is denoted by $\mathrm{Tab}(G)$.*

Definition 7. *For tables (X_1, Λ_1) and (X_2, Λ_2) over G, we define*

$$(X_1, \Lambda_1) \leq (X_2, \Lambda_2) \;:\Leftrightarrow\; X_2 \subseteq X_1 \,\wedge\, \Lambda_1 \circ \iota \subseteq \Lambda_2 \ .$$

Proposition 3. *The pair $(\mathrm{Tab}(G), \leq)$ is a complete lattice. The infimum of a family $((X_i, \Lambda_i))_{i\in I}$ of tables is given by the join operation*

$$\bowtie_{i\in I} (X_i, \Lambda_i) := (\bigcup_{i\in I} X_i, \Lambda_I) \ , \tag{3}$$

where $\Lambda_I := \{\lambda : \bigcup_{i\in I} X_i \to G \mid \forall i \in I : \lambda \circ \iota_i \in \Lambda_i\}$. The supremum is

$$\bigotimes_{i\in I} (X_i, \Lambda_i) := (\bigcap_{i\in I} X_i, \bigcup_{i\in I} (\Lambda_i \circ \iota_i)) \ . \tag{4}$$

5 Galois Connection

Definition 8. *Let $\mathcal{D}$ be an $\mathcal{S}$-structure. The **solution (in $\mathcal{D}$)** of a windowed $\mathcal{S}$-structure $(X, \nu, \mathcal{G})$ is a table over $|\mathcal{D}|$, given by*

$$(X, \nu, \mathcal{G})' := (X, \mathrm{Hom}(\mathcal{G}, \mathcal{D}) \circ \nu) \ . \tag{5}$$

*The **description (over $\mathcal{D}$)** for a table (X, Λ) over $|\mathcal{D}|$ is a windowed $\mathcal{S}$-structure, given by*

$$(X, \Lambda)' := (X, \nu_\Lambda, \mathcal{D}^\Lambda) \ , \tag{6}$$

*where $\nu_\Lambda(x) := (\lambda(x))_{\lambda \in \Lambda}$ for $x \in X$. The two operations thus defined are both denoted by the same sign and are called the **derivation operations with respect to $\mathcal{D}$**.*

Proposition 4. *Let $\mathcal{D}$ be an $\mathcal{S}$-structure. The derivation operators w.r.t. $\mathcal{D}$ form a Galois connection. That is, the following equivalence holds for all windowed $\mathcal{S}$-structures $(X, \nu, \mathcal{G})$ and for all tables (Y, Λ) over $|\mathcal{D}|$:*

$$(X, \nu, \mathcal{G}) \lesssim (Y, \Lambda)' \ \Leftrightarrow \ (Y, \Lambda) \leq (X, \nu, \mathcal{G})' \ . \tag{7}$$

Proof. The left side of the statement can be transformed into the right side by a series of equivalences (explained below):

$$\begin{aligned}(X, \nu, \mathcal{G}) \lesssim (Y, \nu_\Lambda, \mathcal{D}^\Lambda) \ &\Leftrightarrow \ \forall \lambda \in \Lambda : (X, \nu, \mathcal{G}) \lesssim (Y, \lambda, \mathcal{D}) \\ &\Leftrightarrow \ \Lambda \circ \iota \subseteq \mathrm{Hom}(\mathcal{G}, \mathcal{D}) \circ \nu \\ &\Leftrightarrow \ (Y, \Lambda) \leq (X, \mathrm{Hom}(\mathcal{G}, \mathcal{D}) \circ \nu) \ .\end{aligned}$$

To see the first equivalence, use that $(Y, \Lambda)'$ is the product of all $(Y, \lambda, \mathcal{D})$, $\lambda \in \Lambda$, and then apply Cor. 1. For the second equivalence, note that the statements on either side assert that for each $\lambda \in \Lambda$ there exists $\varphi : \mathcal{G} \to \mathcal{D}$ with $\lambda \circ \iota = \varphi \circ \nu$. The last equivalence follows from Def. 7. □

In Props. 5 and 6 we state some consequences of (7) which are well-known in their general form. Proofs can e.g. be found in the introductory chapter of [6]. These carry over to our case of a Galois connection involving a preordered class (note Prop.5(iii), however).

Proposition 5. *Let $\mathcal{D}$ be an $\mathcal{S}$-structure. The following holds for all tables T, T_1 and T_2 over $|D|$, and for all windowed $\mathcal{S}$-structures W, W_1 and W_2:*

(i) $T \leq T''$	*(i')* $W \lesssim W''$
(ii) $T_1 \leq T_2 \Rightarrow T_2' \lesssim T_1'$	*(ii')* $W_1 \lesssim W_2 \Rightarrow W_2' \leq W_1'$
(iii) $T' \simeq T''' = T'''''$	*(iii')* $W' = W'''$

Proposition 6. *Let $\mathcal{D}$ be an $\mathcal{S}$-structure. The following holds for all families $(W_i)_{i \in I}$ of windowed $\mathcal{S}$-structures and for all families $(T_i)_{i \in I}$ of tables over $|\mathcal{D}|$, respectively:*

$$(\coprod_{i \in I} W_i)' = \bowtie_{i \in I} W_i' \ ,$$

$$(\bigtimes_{i \in I} T_i)' \simeq \prod_{i \in I} T_i' \ .$$

6 Concepts and Lattices

As in Formal Concept Analysis, we proceed to define a set of pairs which are stable under the Galois connection,

$$\mathfrak{L}_{\mathcal{D}} := \{(T, W) \mid T \in \mathrm{Tab}(\mathcal{D}) \wedge T' = W \wedge W' = T\} \ , \tag{8}$$

and define an order on that set,

$$(T_1, W_1) \leq (T_2, W_2) \ :\Leftrightarrow\ T_1 \leq T_2 \ \Leftrightarrow\ W_2 \lesssim W_1 \ . \tag{9}$$

The second equivalence in (9) follows from Prop. 5(ii)(ii'). Note that the elements of $\mathfrak{L}_{\mathcal{D}}$ are precisely the pairs (W', W''), or equivalently the pairs (T'', T'''), generated by the windowed $\mathcal{S}$-structures W and tables $T \in \mathrm{Tab}(\mathcal{D})$, respectively (see Prop. 5(iii)(iii')). For $X \subseteq \mathrm{Var}$ we define

$$\begin{aligned} \mathfrak{L}_{\mathcal{D}}[X] &:= \{((X, \Lambda)'', (X, \Lambda)''') \mid (X, \Lambda) \in \mathrm{Tab}(\mathcal{D})\} \\ &= \{((X, \nu, \mathcal{G})', (X, \nu, \mathcal{G})'') \mid (X, \nu, \mathcal{G}) \text{ windowed } \mathcal{S}\text{-structure}\} \ . \end{aligned} \tag{10}$$

The following definition of concept is suggested:

Definition 9. *A **concept** is a pair $(T, (X, \nu, \mathcal{G})) \in \mathfrak{L}_{\mathcal{D}}$ for which all nodes $\nu(x)$, $x \in X$, belong to the same component of $\mathcal{G}$. The set of all concepts of the relational structure $\mathcal{D}$ is denoted by $\mathfrak{C}_{\mathcal{D}}$.*

In analogy to (10), we define

$$\mathfrak{C}_{\mathcal{D}}[X] := \mathfrak{L}_{\mathcal{D}}[X] \cap \mathfrak{C}_{\mathcal{D}} \ . \tag{11}$$

We may identify a concept intent with the component containing $\nu(X)$.

Theorem 1. *The ordered set $(\mathfrak{L}_{\mathcal{D}}, \leq)$ is a complete lattice. Infimum and supremum are given by*

$$\bigwedge_{i \in I} (T_i, W_i) = \left(\bowtie_{i \in I} T_i, (\coprod_{i \in I} W_i)'' \right) \ , \tag{12}$$

$$\bigvee_{i \in I} (T_i, W_i) = \left((\mathop{\Large\boxtimes}_{i \in I} T_i)'', (\prod_{i \in I} W_i)'' \right) \ . \tag{13}$$

For all $X \subseteq \mathrm{Var}$, the suborders $(\mathfrak{L}_{\mathcal{D}}[X] \cup \{\top\}, \leq)$ and $(\mathfrak{C}_{\mathcal{D}}[X] \cup \{\top\}, \leq)$, where $\top$ denotes the maximum of $(\mathfrak{L}_{\mathcal{D}}, \leq)$, are $\bigwedge$-sublattices of $(\mathfrak{L}_{\mathcal{D}}, \leq)$.

Proof. The formulas for the infimum and supremum follow from Prop. 6. Now let $(C_i)_{i \in I}$ be a family in $\mathfrak{L}_{\mathcal{D}}[X] \cup \{\top\}$ and $C := \bigwedge_{i \in I} C_i$, and let us further define $C_i =: ((X, \Lambda_i), (X, \nu_i, \mathcal{G}_i))$. If $C_i = \top$ for all $i \in I$, the infimum is $\top$. Else (3) simplifies to

$$\bowtie_{i \in I} (X, \Lambda_i) = (X, \bigcap_{i \in I} \Lambda_i) \ , \tag{14}$$

which means in particular that $C \in \mathfrak{L}_{\mathcal{D}}[X]$. We write $C =: ((X, \Lambda), (X, \nu, \mathcal{G}))$.

If $(C_i)_{i \in I}$ is a family in $\mathfrak{C}_{\mathcal{D}}[X] \cup \{\top\}$, we have to show in addition that $C \in \mathfrak{C}_{\mathcal{D}}$ if $C_i \in \mathfrak{C}_{\mathcal{D}}$ for some $i \in I$. In this case, there exists $\varphi : \mathcal{G}_i \to \mathcal{G}$. Homomorphisms preserve paths, so C must also be a concept. $\square$

7 Construction

In this section, a brute force construction algorithm for $\mathfrak{C}_{\mathcal{D}}[X]$ is given (X and $\mathcal{D}$ finite), where intents are (and need be) computed up to homomorphical equivalence only. A key observation is that all concept intents are components of powers of $\mathcal{D}$, complemented by some assignment from X to the nodes (cf. (6)). It can be shown that conversely, each windowed structure $(X, \nu, \mathcal{C})$, where $\mathcal{C}$ is a component of a power of $\mathcal{D}$ and ν is chosen arbitrarily, is homomorphically equivalent to some concept intent. If we pick one of these windowed structures from each $\simeq$-class, we have determined all concept intents up to homomorphical equivalence. Note that the components $\mathcal{C}$ can be taken from the power structures $\mathcal{D}^1, \ldots, \mathcal{D}^n$, $n := ||D|^X|$, because a power $\mathcal{D}^{\Lambda}$, $\Lambda \subseteq |\mathcal{D}|^X$, is isomorphic to $\mathcal{D}^{|\Lambda|}$. The following terminating condition can be proven: If we compute the powers $\mathcal{D}^1, \ldots, \mathcal{D}^n$ in sequence and reach some $\mathcal{D}^i$, $1 < i \leq n$, such that every windowed structure obtained from a component of $\mathcal{D}^i$ is homomorphically equivalent to one computed earlier, then the set of concept intents (up to isomorphism) is complete. To build the line diagram (or check for homomorphical equivalence), it may be more convenient to compare extents. If data is stored in a database, extents could be computed by the query engine (this would involve translating windowed structures into some other form of query).

The nine concepts of $\mathfrak{C}_{\mathcal{D}}[\{x\}]$ from our initial example are obtained from the ten components of $\mathcal{F}$ and $\mathcal{F}^2$ (see Figs. 1 and 5). Thirty windowed structures are obtained from these components (as many as there are nodes), each can be folded onto some equivalent graph in Fig. 2. Higher powers of $\mathcal{F}$ do not yield any further concepts.

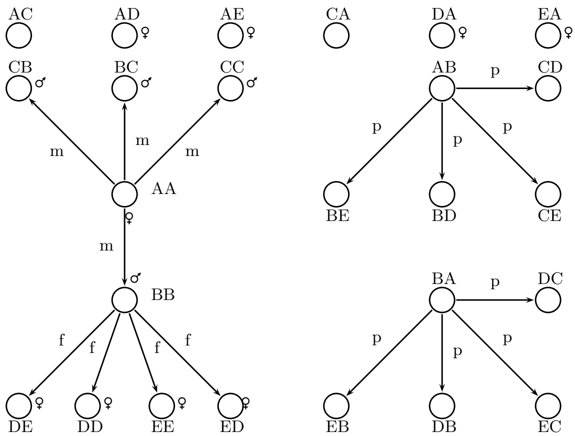

Fig. 5. Example: family tree (squared)

8 Conclusion

We have generated concept lattices directly from a relational structure. The representation of concept intents and extents by graphs and tables establishes connections to graph theory and database theory with their proven formalisms. This gives hope that notions and results from these areas may produce new insights into questions related to lattice-based navigation, and thus guide the development of applications. The similarity of the model to well-known Pattern Structures requires further, detailed comparison. The model will also have to be compared with other formal approaches dealing with relational data, including Concept Graphs [9,8] and Relational Semantic Systems [10].

References

1. Azmeh, Z., Huchard, M., Napoli, A., Hacene, M.R., Valtchev, P.: Querying relational concept lattices. In: Proc. of the 8th Intl. Conf. on Concept Lattices and their Applications (CLA 2011), pp. 377–392 (2011)
2. Chandra, A.K., Merlin, P.M.: Optimal implementation of conjunctive queries in relational databases. In: Proceedings of the Ninth Annual ACM Symposium on Theory of Computing, STOC 1977, pp. 77–90. ACM, New York (1977)
3. Ferré, S.: Conceptual Navigation in RDF Graphs with SPARQL-Like Queries. In: Kwuida, L., Sertkaya, B. (eds.) ICFCA 2010. LNCS, vol. 5986, pp. 193–208. Springer, Heidelberg (2010)
4. Ganter, B., Grigoriev, P.A., Kuznetsov, S.O., Samokhin, M.V.: Concept-Based Data Mining with Scaled Labeled Graphs. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) ICCS 2004. LNCS (LNAI), vol. 3127, pp. 94–108. Springer, Heidelberg (2004)
5. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In: Delugach, H.S., Stumme, G. (eds.) ICCS 2001. LNCS (LNAI), vol. 2120, pp. 129–142. Springer, Heidelberg (2001)
6. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer, Berlin (1999)
7. Godin, R., Saunders, E., Gecsei, J.: Lattice model of browsable data spaces. Inf. Sci. 40(2), 89–116 (1986)
8. Wille, R.: Conceptual Graphs and Formal Concept Analysis. In: Lukose, D., Delugach, H., Keeler, M., Searle, L., Sowa, J. (eds.) ICCS 1997. LNCS, vol. 1257, pp. 290–303. Springer, Heidelberg (1997)
9. Wille, R.: Formal concept analysis and contextual logic. In: Hitzler, P., Schärfe, H. (eds.) Conceptual Structures in Practice, pp. 137–173. Chapman & Hall/CRC (2009)
10. Wolff, K.E.: Relational Scaling in Relational Semantic Systems. In: Rudolph, S., Dau, F., Kuznetsov, S.O. (eds.) ICCS 2009. LNCS, vol. 5662, pp. 307–320. Springer, Heidelberg (2009)