Conjunctive Concept Algebras

Named Perspective

Jens Kötters, Stefan E. Schmidt

Technische Universität Dresden, Germany

Abstract

Concept lattices of relational structures establish a database-theoretic variant of Formal Concept Analysis (FCA). As shown in recent work, these concept lattices naturally extend to concept algebras, by means of a semigroup action. Extensionally, these concept algebras form subalgebras of (a variant of) SPJR table algebras (the conjunctive query fragment of Codd's relational algebra). By that means, an axiomatic characterization of the concept $\wedge$-subalgebras (up to isomorphism, u.t.i.) has been obtained. However, the axioms are difficult to memorize, and in some respects, the semigroup action proved cumbersome to work with. In this paper, we reformulate the axioms, using the signature of Tarski's cylindric algebras (an algebraization of first-order predicate logic). The axioms compare surprisingly well to the cylindric algebra axioms, and the concept $\wedge$-subalgebras correspond to cylindric set algebras. We also obtain an axiomatic characterization of the concept $\bigwedge$-subalgebras (u.t.i.).

Keywords

Concept Algebras, Cylindric Algebra, Conjunctive Queries, Database Theory, Algebraic Logic

1. Motivation

Formal Concept Analysis (FCA) [1] is a mathematical theory of concepts. The central notion in FCA is the *concept lattice*, a complete lattice which describes a hierarchy of concepts. As the *Basic Theorem of FCA* states [1, p. 20], every complete lattice can be represented as a concept lattice. So in this sense, FCA is the theory of complete lattices, from a different perspective.

In the first publication on FCA [2], Rudolf Wille explains what this different perspective was meant to achieve. He was inspired by von Hentig [3], who warned that, as an effect of growing specialization, sciences were becoming disconnected from their surroundings and original motivations, and needed to be *restructured* to re-enable such connections. Wille writes:

> "*Restructuring lattice theory* is an attempt to reinvigorate connections with our general culture by interpreting the theory as concretely as possible, and in this way to promote better communication between lattice theorists and potential users of lattice theory." [2, emphasis added]

> "For this purpose we go back to the origin of the lattice concept in nineteenth-century attempts to formalize logic, where the study of hierarchies of concepts played a central rôle [...]." [2]

More than a decade later, when FCA was already established and had been successfully applied, Wille announced a second project [4], called *restructuring mathematical logic.*

> "The connections of logic to reality have been narrowed since Frege's turn to predicate logic, the leading paradigm of mathematical logic today. Thus, restructuring has to establish a broader understanding of mathematical logic, in particular, by elaborating the pragmatic dimension.

FCA4AI 2024: The 12th International Workshop "What can FCA do for Artificial Intelligence?", October 19 2024, Santiago de Compostela, Spain
jkoetters5@gmail.com (J. Kötters); midt1@msn.com (S. E. Schmidt)
0009-0005-3557-3313 (J. Kötters); 0000-0001-6621-005X (S. E. Schmidt)

For activating real communication and argumentation, it seems to be most important to build enough bridges from the logic-mathematical theory to reality. One way to do this is to revitalize the traditional paradigm of logic given by 'the three essential main functions of thinking - *concepts, judgments and conclusions*' ([5, p. 6])." [4, in-text citation adapted]

A formalization of concepts had already been achieved by FCA. In a follow-up paper [6], Wille observed that another well-known theory of concepts, *Conceptual Graphs* [7] by John Sowa, already offered a formalization of judgments and conclusions. More concretely, Wille points to a mathematization of conceptual graphs by Chein and Mugnier [8]. We gather that *S-graphs* (with "*S* for Sowa, for simple, [...]" [8]), which formalize the most basic type of conceptual graph, represent judgments, and that conclusions can be characterized by graph homomorphism [8, Thm. 1]. The question was then how the two theories can be unified, and Wille proceeds with a proposal. First, he introduces *abstract concept graphs* as slightly modified S-graphs. Then he introduces *power context families*, which represent relational data, and also support the usual notion of concepts for FCA. Finally, he introduces *concept graphs*, which combine abstract concept graphs with concepts from a power context family, thereby obtaining a formalization of judgments that builds on FCA concepts. A summary of concept graphs is presented in [9, Sect. 6.5].

Wille's inspirational paper [6] marked the beginning of a new era of FCA, where relations entered the stage. *Relational Concept Analysis* (RCA) [10][11] provides a deeper integration of concepts with relations, adapts to relational databases through *conceptual scaling* [1], supports different kinds of logical quantifiers, and is being applied in practice. Baader et al. [12] combine FCA with *Description Logics*, a more human-centered branch of logic (cf. Wille's criticism w.r.t. predicate logic above). Kötters [13] introduces a database-theoretic FCA variant; a detailed and refined presentation is given in [14, Sects. 3–5], and the paper at hand continues the theoretical developments.

Conjunctive queries [15, Ch. 4] are a natural and fundamental class of database queries. They were introduced by Chandra and Merlin [16], and have their origin in mathematical logic. Unifying the logical and database-theoretic viewpoints, we identify conjunctive queries with *primitive-positive formulas* (i.e. first-order formulas built from atoms using $\{\exists, \wedge\}$), evaluated in *relational structures*, where

$$\mathrm{res}_{\mathfrak{G}}(\varphi) := \{t \in G^X \mid \mathfrak{G} \models \varphi[t]\} \tag{1}$$

defines the *result table* of a formula φ (with set $X := \mathrm{free}(\varphi)$ of free variables) in a relational structure $\mathfrak{G}$ (with universe G).[1] A *relational database* is a finite relational structure [16, p. 77]. For any relational database $\mathfrak{G}$, the result operation $\mathrm{res}_{\mathfrak{G}}$ is part of a Galois connection, which means that we obtain a concept lattice $\mathfrak{B}(\mathfrak{G})$. This establishes a fundamental connection between FCA and database theory.

Tableau queries [15, p. 43] are structural representations of conjunctive queries. Accordingly, we represent a formula φ with $X := \mathrm{free}(\varphi)$ by a pair $(\mathfrak{N}, \nu)$, consisting of a relational structure $\mathfrak{N}$ and a *window* $\nu : X \to N$, elsewhere called the *summary* [15, p. 43], and obtain the result table as a set

$$\mathrm{res}_{\mathfrak{G}}(\mathfrak{N}, \nu) = \{f \circ \nu \mid f : \mathfrak{N} \to \mathfrak{G}\} \tag{2}$$

of homomorphisms as "seen through the window". From a graph-theoretical perspective, a relational structure is a graph [19]. Likewise, a tableau query can be considered a graph (cf. [14, Sect. 3.1] for our drawing conventions). Under this perspective, $(\mathfrak{N}, \nu)$ is a query graph, and $\mathrm{res}_{\mathfrak{G}}(\mathfrak{N}, \nu)$ contains the pattern matches in the data graph $\mathfrak{G}$. Tableau queries offer a natural way to express infinite conjunctive queries, and indeed, we have not required that $(\mathfrak{N}, \nu)$ must be finite. In order to maintain the logical perspective, a *graph logic* [14, Sect. 3.4] can be formulated. Homomorphisms $f : (\mathfrak{N}_1, \nu_1) \to (\mathfrak{N}_2, \nu_2)$ of tableau queries are defined in the obvious way, and correspond to logical implication in the graph logic.

[1] Details of the unification: Note that queries can be represented in *prenex normal form* [17, Sect. 8.4]; constants are not allowed, but unary relations can play the role of constants [17, Sect. 8.1]; equality is allowed, e.g. the query $x{=}x$ requests a list of all objects in the database, even though such a query is not natively supported in Codd's data model [18]; equality does not enhance expressivity greatly, because in many instances, equality can be eliminated by substitution [15, p. 47f.] or would be expressed by variable repetition in a *conjunctive calculus query* [15, p. 45].

The following reasons suggest that the database-theoretic FCA variant matches Wille's intention with the restructuring project:

- Wille indicates [6, pp. 291f.,300] that suitable notions of judgments and conclusions are offered by S-graphs and their homomorphisms. Since S-graphs represent primitive-positive formulas [8, Sect. 9.1], we might as well consider tableau queries and their homomorphisms; the difference being that S-graphs represent closed formulas (i.e. sentences), whereas tableau queries may, and generally do, represent open formulas (having one or more free variables). By allowing free variables, we obtain concept extensions (cf. eq. (1)).[2]

- Conceptual graphs were initially motivated as a human-centered query language for relational databases [20].[3]

- The widespread use of relational databases suggests practical relevance and good availability of data.

- The result operation corresponds to the activity of querying a database, which suggests a pragmatic dimension.

- The implementation of the classical flight example [21] is not based on concept graphs, but on abstract concept graphs, interpreted as conjunctive queries.

We provide some logical background in Sect. 2, and give a short account of cylindric algebra in Sect. 3. In Sect. 4 we summarize recent results on table algebras [22][23], and also extend a result in Sect.4.4. In Sect. 5, we introduce conjunctive concept algebras and present our main results (Props. 10 and 11).

2. Preliminaries

We assume it is generally known what is meant by a *first-order formula*, and what it means that a first-order formula φ *holds* in a *structure* $\mathfrak{A}$ under a *variable assignment* α, written $\mathfrak{A} \models \varphi[\alpha]$ or $(\mathfrak{A}, \alpha) \models \varphi$, cf. [17]. A *signature* is generally a set M of function symbols, constants, and relation symbols. The set of first-order formulas over the signature M is denoted by $\mathrm{FO}(M)$. If M contains only relation symbols, it is called a *relational signature*, and a structure $\mathfrak{A}$ over M is called a *relational structure*. Because of our take on database theory, we always assume that M is a relational signature; this does not limit expressivity in general [17, Sect. 8.1]. Each symbol $m \in M$ has an *arity* $|m| \geq 1$. For technical convenience, we identify the countably infinite set of variables with the ordinal $\omega = \{0, 1, 2, \dots\}$. An *atomic formula* in $\mathrm{FO}(M)$ is either a *relational atom* $Rx_1 \dots x_n$, an *equality atom* $x{=}y$, or one of the special atoms true (the *tautology*) or false (the *contradiction*), for arbitrary $x_1, \dots, x_n, x, y \in \omega$.

Logical implication between formulas $\varphi, \psi \in \mathrm{FO}(M)$ in the *standard semantics* is introduced as in [17]. We say that φ *logically implies* ψ, and denote this by $\varphi \models \psi$, if $(\mathfrak{G}, \alpha) \models \varphi$ implies $(\mathfrak{G}, \alpha) \models \psi$ for all structures $\mathfrak{G}$ of signature M and all variable assignments $\alpha \in G^\omega$. From (5) we obtain that $\varphi \models \psi$ holds if and only if $\varphi^\mathfrak{G} \subseteq \psi^\mathfrak{G}$ for all structures $\mathfrak{G}$ (of signature M), i.e. logical implication is conveniently expressed via the solution sets. Accordingly, φ and ψ are *logically equivalent*, denoted by $\varphi \mathrel{=\joinrel|\joinrel\models} \psi$, if $\varphi^G = \psi^G$ for all $\mathfrak{G}$.

Then clearly, in the *table semantics*, φ and ψ should be *logically equivalent* if $\mathrm{res}_\mathfrak{G}(\varphi) = \mathrm{res}_\mathfrak{G}(\psi)$ for all $\mathfrak{G}$. The formulas $x{=}x$ and $y{=}y$ are then not equivalent, because the result tables have schemas $\{x\}$ and $\{y\}$, respectively, at least for nonempty $\mathfrak{G}$ and different x, y. So while the special atom true is a tautology, the equality atoms $x{=}x$ and $y{=}y$ are not. A logic with undefined variables provides a formal underpinning: the modified result operation

$$\mathrm{res}^*_\mathfrak{G}(\varphi) := \{t \in \mathrm{Tup}(G) \mid (\mathfrak{G}, t) \models \varphi\} \tag{3}$$

[2] Beyond the formal analogy, the distinction between concepts and judgments in the conjunctive query approach needs to be clarified.

[3] Interestingly, Sowa's article predates Chandra and Merlin's [16] by a year.

uses the finite tuples in $\mathrm{Tup}(G) := \bigcup\{G^X \mid X \in \mathcal{P}_{\mathrm{fin}}(\omega)\}$ as variable assignments, and if an assignment t is not defined on all variables in $\mathrm{free}(\varphi)$, then $(\mathfrak{G}, t) \not\models \varphi$. We refer to this as the *tuple set semantics*. The function

$$h : \begin{cases} \mathrm{Tab}(G) \to \mathcal{P}_{\mathrm{fin}}(\mathrm{Tup}(G)) \\ T \mapsto \{t \in \mathrm{Tup}(G) \mid t|_{\mathrm{schema}(T)} \in T\} \end{cases} \tag{4}$$

satisfies $\mathrm{res}^*_{\mathfrak{G}} = h \circ \mathrm{res}_{\mathfrak{G}}$, so it relates table semantics and tuple set semantics. It forms an embedding $h : (\mathrm{Tab}(G), \bowtie) \to (\mathcal{P}_{\mathrm{fin}}(\mathrm{Tup}(G)), \cap)$ of meet-semilattices, i.e. an injective homomorphism; and as such, it also forms an order embedding $h : (\mathrm{Tab}(G), \leq) \to (\mathcal{P}_{\mathrm{fin}}(\mathrm{Tup}(G)), \subseteq)$; thereby providing a set interpretation of the tables and their order, see also [14, Sect. 3.5][24]. In particular, $\varphi \lesssim \psi :\Leftrightarrow \forall \mathfrak{G} : \mathrm{res}_{\mathfrak{G}}(\varphi) \leq \mathrm{res}_{\mathfrak{G}}(\psi) \Leftrightarrow \forall \mathfrak{G} : \mathrm{res}^*_{\mathfrak{G}}(\varphi) \subseteq \mathrm{res}^*_{\mathfrak{G}}(\psi)$ denotes *logical implication* in both the table semantics and the tuple set semantics. So both semantics are equivalent; we can use either of them, depending on the purpose. Finally, we write $\varphi \simeq \psi$ if and only if $\varphi \lesssim \psi$ and $\psi \lesssim \varphi$, which coincides with our initially postulated notion of equivalence.

The following proposition relates table semantics with standard semantics.

Proposition 1. *Let $\varphi, \psi \in \mathrm{PP}(M)$. Then $\varphi \lesssim \psi$ if and only if $\varphi \models \psi$ and $\mathrm{free}(\psi) \subseteq \mathrm{free}(\varphi)$.*

Proof. The case $\varphi = \mathsf{false}$ is trivial. Now let $\varphi \neq \mathsf{false}$ and $\psi = \mathsf{false}$. Because φ is primitive-positive (and not the contradiction), it is satisfiable (cf. [17, Ex. 3.4]), i.e. there exists $\mathfrak{G}$ such that $\varphi^{\mathfrak{G}} \not\subseteq \emptyset = \psi^{\mathfrak{G}}$, so $\varphi \not\models \psi$; and likewise, we obtain $\varphi \not\lesssim \psi$. The case $\varphi, \psi \neq \mathsf{false}$ is covered in [14, Prop. 3]. □ □

3. Cylindric Algebra

The two most fundamental disciplines of logic, as of today, are *propositional logic* and *predicate logic*; and by predicate logic, we usually mean *first-order logic*. *Boolean algebras* are well-known algebraizations of propositional logic. Likewise, *cylindric algebras* by Alfred Tarski are algebraizations of first-order logic. The classical monographs on cylindric algebras are the works of Henkin, Monk and Tarski [25][26], and for an introduction, we refer to the papers of Németi [27] and Monk [28]. We first present *cylindric set algebras* (Sect. 3.1), then turn to *cylindric algebras* in general (Sect. 3.2).

3.1. Cylindric Set Algebras

Every relational structure $\mathfrak{G}$ with signature M induces a *solution operation* $(\cdot)^{\mathfrak{G}} : \mathrm{FO}(M) \to \mathcal{P}(G^\omega)$ that maps each first-order formula φ to its *solution set*

$$\varphi^{\mathfrak{G}} := \{\alpha \in G^\omega \mid (\mathfrak{G}, \alpha) \models \varphi\} \ . \tag{5}$$

The algebra $\mathbf{FO}(M) = (\mathrm{FO}(M), \vee, \wedge, \neg, \mathsf{false}, \mathsf{true}, \exists_x, x{=}y)_{x,y \in \omega}$ interprets $\vee$, $\wedge$, $\neg$ and $\exists_x$ (for all $x \in \omega$) as syntactic operations, e.g. $\vee(\varphi, \psi) := (\varphi \vee \psi)$ and $\exists_x(\varphi) := (\exists x \varphi)$. Moreover, it contains false, true, and all equality atoms $x{=}y$ as distinguished elements. The solution operation forms a homomorphism $(\cdot)^{\mathfrak{G}} : (\mathrm{FO}(M), \vee, \wedge, \neg, \mathsf{false}, \mathsf{true}) \to (\mathcal{P}(G^\omega), \cup, \cap, (\cdot)^\complement, \emptyset, G^\omega)$. In this sense, the logical operations are represented by set operations. Likewise, existential quantification over x is represented by the *cylindrification* $C_x : \mathcal{P}(G) \to \mathcal{P}(G)$, defined by

$$C_x(A) := \{\alpha \in G^\omega \mid \exists g \in G : \alpha\tfrac{g}{x} \in A\} \ , \tag{6}$$

where $\alpha\frac{g}{x} \in G^\omega$ is the *modification* of α that satisfies $\alpha\frac{g}{x}(x) = g$ and $\alpha\frac{g}{x}(y) = \alpha(y)$ for all $y \in \omega \setminus \{x\}$. Finally, the equality atoms $x{=}y$ are represented by the *diagonals*

$$D_{xy} := \{\alpha \in G^\omega \mid \alpha(x) = \alpha(y)\} \ . \tag{7}$$

This motivates $\mathbf{Cs}(G) := (\mathcal{P}(G^\omega), \cup, \cap, (\cdot)^\complement, \emptyset, G^\omega, C_x, D_{xy})_{x,y \in \omega}$ as a set-theoretic counterpart of $\mathbf{FO}(M)$; but note that in principle, G and M are independent. In summary, the relational structure $\mathfrak{G}$ induces the *solution homomorphism* $(\cdot)^{\mathfrak{G}} : \mathbf{FO}(M) \to \mathbf{Cs}(G)$.

The homomorphic image $\mathbf{Cs}(\mathfrak{G}) := [\mathbf{FO}(M)]^{\mathfrak{G}}$ is the subalgebra of $\mathbf{Cs}(G)$ that consists of the solution sets. More generally, a subalgebra of $\mathbf{Cs}(G)$ is called a *cylindric set algebra* with base G and dimension ω. We now pose two questions, and state the answers below, as found in Monk [28]:

a) How are the solution set algebras $\mathbf{Cs}(\mathfrak{G})$ characterized from among all cylindric set algebras of dimension ω?

b) Is there an axiomatic characterization for the cylindric set algebras of dimension ω?

a) The algebras $\mathbf{Cs}(\mathfrak{G})$ are precisely the *locally finite-dimensional* and *regular* cylindric set algebras of dimension ω (cf. [28, Thm. 12.2]), b) The cylindric set algebras of dimension ω are not first-order axiomatizable (cf. [28, p. 279]).

3.2. Cylindric Algebras

Because of negative results with regard to first-order axiomatization of cylindric set algebras and other concrete notions, *cylindric algebra* were introduced. They are defined by a finite schema of equations, to provide for a good theory, and are meant to circumscribe the interesting classes of concrete algebras sufficiently well. In that sense, the notion of cylindric algebra is arbitrary, cf. [27, Sect. 4].

Definition 2. *A* cylindric algebra *is an algebra* $(V, \vee, \wedge, \neg, 0, 1, c_x, d_{xy})_{x,y\in\omega}$ *consisting of a binary* supremum $\vee$*, a binary* infimum $\wedge$*, a unary* complement $\neg$*, a* zero *element* 0*, a* one *element* 1*, a unary* cylindrification c_x *for each* $x \in \omega$*, and a* diagonal *element* d_{xy} *for each* $(x, y) \in \omega \times \omega$*, which satisfies*

(CA0) $(V, \vee, \wedge, \neg, 0, 1)$ *is a Boolean algebra*

(CA1) $c_x(0) = 0$

(CA2) $u \leq c_x(u)$

(CA3) $c_x(u \wedge c_x(v)) = c_x(u) \wedge c_x(v)$

(CA4) $c_x(c_y(u)) = c_y(c_x(u))$

(CA5) $d_{xx} = 1$

(CA6) $x \neq y, z \Rightarrow d_{yz} = c_x(d_{yx} \wedge d_{xz})$

(CA7) $x \neq y \Rightarrow c_x(d_{xy} \wedge u) \wedge c_x(d_{xy} \wedge \neg u) = 0$

for all $u, v \in V$ *and all* $x, y, z \in \omega$*.*

4. Table Algebras

From an extensional point of view, concept lattices of relational structures are table algebras. In quest for a Basic Theorem, this motivates the study of table algebras.

4.1. DPJR Algebras

The *SPJR algebra* [15, Sect. 4.4] allows to specify conjunctive queries using algebraic operations; these are the table operations of *selection*, *projection*, *(natural) join* and *renaming*, indicated by the letters. It is also called the *named conjunctive algebra*, because it operates on tables with named columns (as opposed to tables with ordered columns). While Abiteboul et al. [15] refer to SPJR algebra as a query language, it better suits our extensional viewpoint to think of it as an algebra of tables, with concrete operations.

We define a *table* as a set $T \subseteq G^X$, where $X \subseteq \omega$ is a finite set of *column names* (not column numbers), an element $t \in T$ is a *row*, $t(x)$ is the *entry* in row t and column x, and G is an arbitrary set. Hence,

$$\mathrm{Tab}(G) = \bigcup\{\mathcal{P}(G^X) \mid X \subseteq \omega \text{ finite}\} \tag{8}$$

contains all tables with entries in G. Note that while X must be finite, a table can have an infinite number of rows if G is infinite. Naturally, the empty set $\emptyset$ represents the *empty table*. The *schema* of a table $T \in \mathrm{Tab}(G)$ is uniquely defined by

$$\mathrm{schema}(T) := \begin{cases} X & \text{if } T \in G^X \text{ and } T \neq \emptyset \\ \omega & \text{if } T = \emptyset \end{cases} . \tag{9}$$

Note that $G^\emptyset$ has a single element $\emptyset$, called the *empty tuple*, and $\{\emptyset\} \in \mathcal{P}(G^\emptyset)$ is the unique table with schema $\emptyset$.

For finite $X \subseteq \omega$, the set $\mathrm{Tab}(G)[X] := \mathcal{P}(G^X)$ is the X*-slice* of $\mathrm{Tab}(G)$. The *natural join* of tables $S \in \mathrm{Tab}(G)[X]$ and $T \in \mathrm{Tab}(G)[Y]$ is a table $T \in \mathrm{Tab}(G)[X \cup Y]$, defined by

$$S \bowtie T := \{t \in G^{X\cup Y} \mid t|_X \in S \text{ and } t|_Y \in T\} \ . \tag{10}$$

Moreover, for all $x, y \in \omega$, we define the *diagonal*

$$E_{xy} := \{t \in G^{\{x,y\}} \mid t(x) = t(y)\} \ . \tag{11}$$

The natural join is associative and commutative [15, p. 58], and trivially idempotent, i.e. $(\mathrm{Tab}(G), \bowtie)$ is a meet-semilattice, with the implied table order $T_1 \leq T_2 :\Leftrightarrow T_1 = T_1 \bowtie T_2$. The tables $\emptyset$ and $\{\emptyset\}$ are the absorbing element and neutral element, respectively, w.r.t. to the join. This means that they are also the smallest and greatest elements in the lattice order.

A *finite partial transformation* of ω is a partial function $\lambda : \omega \rightharpoonup \omega$, defined on a finite set $\mathrm{def}(\lambda) = X \subseteq \omega$, and we set $\mathrm{rng}(\lambda) = \{\lambda(x) \mid x \in \mathrm{def}(\lambda)\}$. We use $\mathcal{T}_{\mathrm{fp}}(\omega)$ to denote the set of finite partial transformations on ω. The pair $(\mathcal{T}_{\mathrm{fp}}, \circ)$ is a semigroup, with $\circ$ as composition of partial functions, which naturally acts on the tables through the *right multiplication*

$$\cdot \begin{cases} \mathrm{Tab}(G) \times \mathcal{T}_{\mathrm{fp}}(\omega) \to \mathrm{Tab}(G) \\ (T, \lambda) \mapsto T \cdot \lambda := \{t \circ \lambda \mid t \in T\} \end{cases} . \tag{12}$$

The right multiplicaton encodes three different table operations: projection, renaming and column duplication. For the *partial identity* $\pi_X : \omega \rightharpoonup \omega$, which can be written $\{(x, x) \mid x \in X\}$ as a relation, $T \cdot \pi_X$ is the *projection* of T on the column set X. Note that right multiplication is totally defined, so generally $\mathrm{schema}(T \cdot \pi_X) = \mathrm{schema}(T) \cap X$. A *partial bijection* is an injective function $\xi : \omega \rightharpoonup \omega$, and it acts as a *renaming* on $\mathrm{Tab}(G)$. Moreover, a *folding* is a partial function $\delta : \omega \rightharpoonup \omega$ with $\delta \circ \delta = \delta$, and for each $x \in \mathrm{def}(\delta)$, the table $T \cdot \delta$ has a column x which is a copy of $\delta(x)$; the column $\delta(x)$ is fixed because of $\delta \circ \delta = \delta$. This completely describes right multiplication, since every $\lambda \in \mathcal{T}_{\mathrm{fp}}(\omega)$ acts as a sequence of these operations [22, Lemma 1]; more concretely, there is a decomposition $\lambda = \pi_X \circ \xi \circ \delta$, and furthermore $T \cdot (\pi_X \circ \xi \circ \delta) = ((T \cdot \pi_X) \cdot \xi) \cdot \delta$. For the above reason, we call

$$\mathrm{DPJR}(G) = (\mathrm{Tab}(G), \bowtie, \emptyset, \{\emptyset\}, \cdot, E_{xy}, \mathrm{schema})_{x,y\in\omega} \tag{13}$$

the *full DPJR algebra* with *base* G. A *DPJR algebra* with *base* G is a subalgebra of $\mathrm{DPJR}(G)$. Before we proceed, the relation with SPJR algebras shall be explained.

Abiteboul et al. [15, p. 57] refer to two kinds of selection, denoted by $\sigma_{A=a}$ and $\sigma_{A=B}$, where A and B are column names, and a denotes an object in the universe. The reference to a reflects a database-theoretic convention, whereby objects in the universe are exposed as constants. Note however, that in our formalization of conjunctive queries, which unifies the database-theoretic and logical viewpoints (cf. the footnote in Sect. 1), we strictly allow relation symbols only. So the corresponding variant of SPJR algebra would only use the second kind of selection (i.e. $\sigma_{A=B}$, which deletes all rows having different entries in the A and B columns). It is a moderately easy exercise to show that DPJR algebra (without diagonals) is equivalent to this variant of SPJR algebra. The diagonals are not part of SPJR algebra; their inclusion in the DPJR algebra also caters to the unified viewpoint.

4.2. Conjunctive Table Algebras

We motivate conjunctive table algebras in the same way we have motivated cylindric set algebras in Sect. 3.1. A first-order formula is *primitive-positive* if it is built from atoms using $\{\wedge, \exists\}$. The set of primitive-positive formulas over the relational signature M is denoted by $\mathrm{PP}(M)$. The algebra $\mathbf{PP}(M) := (\mathrm{PP}(M), \wedge, \mathsf{false}, \mathsf{true}, \exists_x, x{=}y, \mathrm{free})_{x,y\in\omega}$ extends $\mathrm{PP}(M)$ with the respective syntactic operations and constants (cf. the algebra $\mathbf{FO}(M)$ in Sect. 3.1), and it also includes the function free : $\mathrm{PP}(M) \to \mathcal{P}(\omega)$, which maps each formula to its set of free variables; for the special atoms, we define $\mathrm{free}(\mathsf{true}) = \emptyset$ and $\mathrm{free}(\mathsf{false}) = \omega$.

Every relational structure $\mathfrak{G}$, with universe G and signature M, induces a *result operation* $\mathrm{res}_{\mathfrak{G}} : \mathrm{PP}(M) \to \mathrm{Tab}(G)$ that maps each formula φ to its *result table*, given by

$$\mathrm{res}_{\mathfrak{G}}(\varphi) := \{t \in G^{\mathrm{free}(\varphi)} \mid (\mathfrak{G}, t) \models \varphi\} \ . \tag{14}$$

In particular, we have $\mathrm{res}_{\mathfrak{G}}(\mathsf{false}) = \emptyset$ and $\mathrm{res}_{\mathfrak{G}}(\mathsf{true}) = \{\emptyset\}$. Note that each variable in $\mathrm{free}(\varphi)$ corresponds to a column in the result table $\mathrm{res}_{\mathfrak{G}}(\varphi)$.

Next, we identify the table operations which correspond to the logical operations. Existential quantification is matched by column deletion; we define the *deletion operation* $\mathrm{del}_x : \mathrm{Tab}(G) \to \mathrm{Tab}(G)$ by

$$\mathrm{del}_x(S) := \{t|_{X\setminus\{x\}} \mid t \in S\} \ . \tag{15}$$

Note that $\mathrm{del}_x(S) = S$ if $x \notin X$. The other required operations have already been defined in Sect. 4.1. As expected, we have

$$\begin{aligned}
\mathrm{res}_{\mathfrak{G}}(\varphi \wedge \psi) &= \mathrm{res}_{\mathfrak{G}}(\varphi) \bowtie \mathrm{res}_{\mathfrak{G}}(\psi) \\
\mathrm{res}_{\mathfrak{G}}(\mathsf{false}) &= \emptyset \\
\mathrm{res}_{\mathfrak{G}}(\mathsf{true}) &= \{\emptyset\} \\
\mathrm{res}_{\mathfrak{G}}(\exists_x \varphi) &= \mathrm{del}_x(\mathrm{res}_{\mathfrak{G}}(\varphi)) \\
\mathrm{res}_{\mathfrak{G}}(x = y) &= E_{xy} \ ,
\end{aligned}$$

and if $\mathrm{res}_{\mathfrak{G}}(\varphi) \neq \emptyset$, then also $\mathrm{schema}(\mathrm{res}_{\mathfrak{G}}(\varphi)) = \mathrm{free}(\varphi)$. This motivates to define $\mathbf{Tab}(G) := (\mathrm{Tab}(G), \bowtie, \emptyset, \{\emptyset\}, \mathrm{del}_x, E_{xy}, \mathrm{schema})_{x,y\in\omega}$ as the *full conjunctive table algebra* with *base* G.

As indicated, in the case $\mathrm{res}_{\mathfrak{G}}(\varphi) = \emptyset$, the free variables of φ can not be recovered from the result table, and in this sense they are not preserved. Consequently, we do not consider $\mathrm{res}_{\mathfrak{G}} : \mathbf{PP}(M) \to \mathbf{Tab}(G)$ to be a proper homomorphism, and refer to it as a *zero-tolerant homomorphism*, a slightly weaker kind of homomorphism. But it does preserve all logical operations and constants, so the homomorphic image $\mathbf{Tab}(\mathfrak{G}) := \mathrm{res}_{\mathfrak{G}}[\mathbf{PP}(M)]$ is a subalgebra of $\mathbf{Tab}(G)$. This motivates our main definition.

Definition 3 (Conjunctive Table Algebra)**.** *A* conjunctive table algebra *with* base G *is a subalgebra* $\mathfrak{A}$ *of* $\mathbf{Tab}(G)$.

The X-*slice* of $\mathfrak{A}$, for each $X \in \mathcal{P}_{\mathrm{fin}}(\omega)$, is the set $\mathfrak{A}[X] := \{T \in A \mid T \in G^X\}$. For convenience, we define $\mathfrak{A}^*[X] := \{T \in A \mid \mathrm{schema}(T) = X\} = \mathfrak{A}[X] \setminus \{\emptyset\}$. Note that $n = \{0, \dots, n-1\}$, so $\mathfrak{A}[n] = \mathfrak{A}[\{0, \dots, n-1\}]$ and $\mathfrak{A}^*[n] = \mathfrak{A}[\{0, \dots, n-1\}]$.

In Sect. 3.1, we have presented two questions (and their answers) on cylindric set algebras. We formulate their counterparts in our database-theoretic setting:

a) How are the algebras $\mathbf{Tab}(\mathfrak{G})$ characterized from among all conjunctive table algebras?

b) Is there an axiomatic characterization for the conjunctive table algebras?

Proposition 4. *Conjunctive table algebras and DPJR algebras are equivalent:*

i) Every conjunctive table algebra is closed under right multiplication.

ii) Every DPJR algebra is closed under deletions.

Proof. i) Let $\mathfrak{A}$ be a conjunctive table algebra. We show $T \cdot \lambda \in A$ for all $T \in \mathfrak{A}[Y]$, $Y \in \mathcal{P}_{\mathrm{fin}}(\omega)$, and $\lambda \in \mathcal{T}_{\mathrm{fp}}(\omega)$. Since $T \cdot \lambda = T \cdot \lambda|_{\lambda^{-1}(Y)}$, we may assume w.l.o.g. that $\mathrm{rng}(\lambda) \subseteq Y$, i.e. $\lambda : X \to Y$ for some $X \in \mathcal{P}_{\mathrm{fin}}(\omega)$. If $X \cap Y = \emptyset$, then $T \cdot \lambda = \mathrm{del}_Y(T \bowtie E_\lambda) \in \mathfrak{A}[X]$. Otherwise, let $\xi : Y \to Z$ be a bijection onto some $Z \in \mathcal{P}_{\mathrm{fin}}(\omega)$ with $Z \cap X = \emptyset$ and $Z \cap Y = \emptyset$. By reduction to the previous case, we first obtain $T \cdot \xi^{-1} \in \mathfrak{A}[Z]$, and then $T \cdot \lambda = (T \cdot \xi^{-1}) \cdot (\xi \circ \lambda) \in \mathfrak{A}[X]$.

ii) Let $\mathfrak{A}$ be a DPJR algebra. For all $T \in \mathfrak{A}[X]$, $X \in \mathcal{P}_{\mathrm{fin}}(\omega)$ and $x \in \omega$, we have $\mathrm{del}_x(T) = T \cdot \pi_{X\setminus\{x\}} \in \mathfrak{A}[X \setminus \{x\}]$. $\square$

Proposition 5. *The conjunctive table algebras are precisely the result table algebras* $\mathbf{Tab}(\mathfrak{G})$ *of relational structures* $\mathfrak{G}$*.*

Proof. By definition, every algebra $\mathbf{Tab}(\mathfrak{G})$ is a conjunctive table algebra. Now let $\mathfrak{A}$ be a conjunctive table algebra with base G. Let $M_{\mathfrak{A}} = \bigcup_{n\geq 1} \mathfrak{A}^*[n]$ be the relational signature which uses $\mathfrak{A}^*[n]$ as its set of n-ary relation symbols. Each $T \in \mathfrak{A}^*[n]$ is also a set of n-tuples, i.e. an n-ary relation. Let $\mathfrak{G}_{\mathfrak{A}}$ be the relational structure with universe G and signature $M_{\mathfrak{A}}$, given by the map $I : M_{\mathfrak{A}} \to M_{\mathfrak{A}}$ that maps each $T \in \mathfrak{A}^*[n]$ (as a symbol) to $T \in \mathfrak{A}^*[n]$ (as a relation), i.e. $I = \mathrm{id}_{M_{\mathfrak{A}}}$. It remains to show $\mathrm{res}_{\mathfrak{G}_{\mathbf{A}}}[\mathbf{PP}(M_{\mathfrak{A}})] = \mathfrak{A}$.

"$\subseteq$:" By definition of $\mathfrak{G}_{\mathfrak{A}}$, we have $\mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}(T(0,\dots,n-1)) = T \in M_{\mathfrak{A}} \subseteq A$ for all relational atoms $T(0,\dots,n-1)$. Let $\sigma : n \to X$ be a substitution of variables, such that $n \cap X = \emptyset$. Then $T(\sigma(0),\dots,\sigma(n-1))$ is equivalent to $\varphi_{T,\sigma} := \exists 0 \dots \exists n-1 : (T(0,\dots,n-1) \wedge 0{=}\sigma(0) \wedge \dots \wedge n-1{=}\sigma(n-1))$. So $\mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}(T(\sigma(0),\dots,\sigma(n-1))) = \mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}(\varphi_{T,\sigma}) = \mathrm{del}_0 \dots \mathrm{del}_{n-1}(T \bowtie E_{0\sigma(0)} \bowtie \dots \bowtie E_{n-1,\sigma(n-1)}) \in A$. Every relational atom $T(x_1,\dots,x_n)$ is obtained from $T(0,\dots,n-1)$ by two such substitutions, i.e. $\mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}(T(x_1,\dots,x_n)) \in A$ for all relational atoms. By induction, $\mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}(\varphi) \in A$ for all $\varphi \in \mathrm{PP}(M)$.

"$\supseteq$:" Let $T \in \mathfrak{A}^*[X]$ for some $X \in \mathcal{P}_{\mathrm{fin}}(\omega)$ with cardinality $n := \#X$. We choose an arbitrary bijection $\xi : n \to X$, and obtain $T \cdot \xi \in \mathfrak{A}^*[n] \subseteq \mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}[\mathbf{PP}(M_{\mathfrak{A}})]$. By Prop. 4, the homomorphic image is closed under right multiplication, so we also have $T = (T \cdot \xi) \cdot \xi^{-1} \in \mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}[\mathbf{PP}(M_{\mathfrak{A}})]$. $\square$

Proposition 5 provides a simple answer to our question a) above: The algebras $\mathbf{Tab}(\mathfrak{G})$ are precisely the conjunctive table algebras. The primary question is how the algebras $\mathbf{Tab}(\mathfrak{G})$ can be axiomatized. As we have seen now, the formal framework of cylindric set algebras fits the question perfectly (which was not the case for cylindric set algebra, cf. question a) in Sect. 3.1). An answer to our question b) is given in Sect. 4.3.

4.3. Projectional Semilattices

The main result of [23] is the axiomatic characterization of conjunctive table algebras by projective semilattices. The given axiomatization is not a first-order axiomatization, but a comparison with cylindric algebra axioms, given below, should convince the reader of their value.

Definition 6 ([23, Def. 2]). *A* projectional semilattice *is an algebra* $(V, \wedge, 0, 1, c_x, d_{xy}, \mathrm{dom})_{x,y\in\omega}$ *consisting of a binary infimum* $\wedge$*, a* zero *element* 0*, a* one *element* 1*, a unary* cylindrification c_x *for each* $x \in \omega$*, a* diagonal *element* d_{xy} *for each* $(x,y) \in \omega \times \omega$*, and a* domain function $\mathrm{dom} : V \to \mathcal{P}(\omega)$*, which satisfies*

- **(PS0)** $(V, \wedge, 0, 1)$ *is a bounded semilattice*
- **(PS1)** $c_x(0) = 0$
- **(PS2)** $u \leq c_x(u)$
- **(PS3)** $c_x(u \wedge c_x(v)) = c_x(u) \wedge c_x(v)$
- **(PS4)** $c_x(c_y(u)) = c_y(c_x(u))$
- **(PS5)** $u \neq 0 \Rightarrow (u \neq c_x(u) \Leftrightarrow u \leq d_{xx})$
- **(PS6)** $x \neq y, z \Rightarrow d_{yz} = c_x(d_{yx} \wedge d_{xz})$
- **(PS7)** $x \neq y \Rightarrow d_{xy} \wedge c_x(d_{xy} \wedge u) \leq u$
- **(PS8)** $u \neq 0 \Rightarrow \mathrm{dom}(u)$ *finite*
- **(PS9)** $\mathrm{dom}(u) = \{x \in \omega \mid u \leq d_{xx}\}$
- **(PS10)** $\mathrm{dom}(u) = \emptyset \Rightarrow u = 1$
- **(PS11)** $d_{xx} \neq 0$
- **(PS12)** $d_{xy} = d_{yx}$

for all $u, v \in V$ *and* $x, y, z \in \omega$*.*

Proposition 7 ([23, Thms. 1,3]). *The conjunctive table algebras over non-empty universes are precisely (up to isomorphism) the projectional semilattices.*

The axioms (**PS0**), …, (**PS7**) correspond to the axioms (**CA0**),…,(**CA7**) for cylindric algebras. Axiom (**CA0**) asserts a Boolean algebra; since we do not consider disjunction and negation, axiom (**PS0**) only asserts a bounded semilattice. The Axioms (**CA1**), (**CA2**), (**CA3**), (**CA4**) and (**CA6**) are identical to (**PS1**), (**PS2**), (**PS3**), (**PS4**) and (**PS6**), respectively. Cylindric algebra axiom (**CA5**) states $d_{xx} = 1$, reflecting that $x{=}x$ is a tautology; however, the *table semantics* in eq. (1) corresponds to a logic with undefined variables, where $x{=}x$ is not a tautology! We consider (**PS5**) to be a suitable replacement: Under the definition axiom (**PS9**), axiom (**CA5**) asserts $\mathrm{dom}(u) = \omega$ for all $u \neq 0$; whereas axiom (**PS5**) asserts $\mathrm{dom}(u) = \{x \in \omega \mid c_x(u) \neq u\}$ for all $u \neq 0$; the latter set is known as the *dimension set* $\Delta(u)$ in the terminology of cylindric algebras. Axiom (**PS7**) is the historical axiom (**CA7**); the contemporary axiom (**CA7**) is equivalent but involves negation! Historically, there was also an axiom (**CA8**), stating that $\Delta(u)$ is finite for all $u \in V$. Since $\mathrm{dom}(u) = \Delta(u)$ for $u \neq 0$, we can identify (**CA8**) with (**PS8**), disregarding the case $u = 0$.

4.4. Complete Projectional Semilattices

The table algebras $\mathrm{Tab}(G)$ are complete lattices [14, Sect. 3.5]. The join $\bowtie_{i\in I} T_i$ of a family $(T_i)_{i\in I}$ is the empty table if $\bigcup_{i\in I} \mathrm{schema}(T_i)$ is infinite (because no other tables with infinite schema are contained in $\mathrm{Tab}(G)$), and is otherwise defined in the natural way.

A conjunctive table algebra $\mathfrak{A}$ is *complete* if $\bowtie_{i\in I} T_i \in A$ for all families $(T_i)_{i\in I}$ in A. In this section, we provide an axiomatic characterization of complete conjunctive table algebras. Likewise, we say that a projectional semilattice $(V, \wedge, 0, 1, c_x, d_{xy}, \mathrm{dom})_{x,y\in\omega}$ is *complete* if $(V, \leq)$ is a complete lattice.

Proposition 8. *The complete conjunctive table algebras over non-empty universes are precisely (up to isomorphism) the complete projectional semilattices.*

Proof. Trivially, every complete conjunctive table algebra is a complete projectional semilattice. Now let $\mathfrak{A}$ be a complete projectional semilattice. In the proof of [23], an embedding $\mathrm{ext}_\alpha : \mathfrak{A} \to \mathrm{Tab}(G)$ into a full table algebra with non-empty base G is obtained, where $\alpha : \bigcup_{X\in\mathcal{P}_{\mathrm{fin}}(\omega)} G^X \to A$ is a tuple labeling of $\mathfrak{A}$ (cf. [22, Def. 4]), in particular it satisfies $\mathrm{schema}(\alpha(t)) = \mathrm{def}(t)$ and

$$\alpha(t) \cdot \lambda = \alpha(t \circ \lambda) \tag{16}$$

for all $\lambda \in \mathcal{T}_{\mathrm{fp}}(\omega)$. The embedding ext_α is defined by

$$\mathrm{ext}_\alpha(u) := \{t \in G^X \mid \alpha(t) \leq u\} \tag{17}$$

for all $u \in \mathfrak{A}[X]$ and $X \in \mathcal{P}_{\mathrm{fin}}(\omega)$. Our proof amounts to an adaptation of the infimum case in the proof of [22, Thm. 2]. From that paper, we also obtain [22, Prop. 3x)]

$$\alpha(t) \leq u_i \Leftrightarrow \alpha(t) \cdot \pi_{X_i} \leq u_i \ . \tag{18}$$

Now let $(u_i)_{i\in I}$ be a family of elements in $\mathfrak{A}$. We have to show $\mathrm{ext}_\alpha(\bigwedge_{i\in I} u_i) = \bowtie_{i\in I} \mathrm{ext}_\alpha(u_i)$. If $\bigcup_{i\in I} \mathrm{dom}(u_i)$ is infinite, we obtain $\mathrm{ext}_\alpha(\bigwedge_{i\in I} u_i) = \mathrm{ext}_\alpha(0) = \emptyset = \bowtie_{i\in I} \mathrm{ext}_\alpha(u_i)$. Otherwise,

$$\begin{aligned} t \in \mathrm{ext}_\alpha(\bigwedge_{i\in I} u_i) &\underset{(17)}{\Leftrightarrow} \forall i \in I : \alpha(t) \leq u_i \underset{(18)}{\Leftrightarrow} \forall i \in I : \alpha(t) \cdot \pi_{X_i} \leq u_i \\ &\underset{(16)}{\Leftrightarrow} \forall i \in I : \alpha(t|_{X_i}) \leq u_i \underset{(17)}{\Leftrightarrow} \forall i \in I : t|_{X_i} \in \mathrm{ext}_\alpha(u_i) \Leftrightarrow t \in \bowtie_{i\in I} \mathrm{ext}_\alpha(u_i) \ . \end{aligned}$$

□

5. Conjunctive Concept Algebras

For every relational structure $\mathfrak{G}$, the result operation $\mathrm{res}_\mathfrak{G}$ of eq. (2) is part of a Galois connection, from which a concept lattice is obtained in the usual way, cf. [14, Sect. 5][13]. The pair of maps can be stated as

$$\mathrm{res}_\mathfrak{G}(\mathfrak{N}, \nu) := \{t \in G^{\mathrm{def}(\nu)} \mid (\mathfrak{N}, \nu) \precsim (\mathfrak{G}, t)\} \tag{19}$$

$$\mathrm{info}_{\mathfrak{G}}(T) := \prod_{t \in T} (\mathfrak{G}, t) \tag{20}$$

where $(\mathfrak{N}, \nu) \lesssim (\mathfrak{G}, t) :\Leftrightarrow \exists f\, f : (\mathfrak{N}, \nu) \to (\mathfrak{G}, t)$ denotes the existence of a tableau query homomorphism, and $\prod_{t \in T}(\mathfrak{G}, t)$ is the direct product of tableau queries. A *concept* of $\mathfrak{G}$ is a pair $(T, (\mathfrak{N}, \nu))$ such that $T = \mathrm{res}_{\mathfrak{G}}(\mathfrak{N}, \nu)$ and $(\mathfrak{N}, \nu) = \mathrm{info}_{\mathfrak{G}}(T)$. The table $\mathrm{ext}(T, (\mathfrak{N}, \nu)) := T$ is the concept's *extent*, and the tableau query $\mathrm{int}(T, (\mathfrak{N}, \nu)) := (\mathfrak{N}, \nu)$ is the concept's *intent*. For practical purposes, the intents can be simplified by reduction to connected components and query minimization, cf. [13, Figs. 5,2]. Complexity of intents can be further reduced by pattern projections [14, Sect. 6.2][29], but this amounts to considering an $\bigwedge$-sublattice of $\mathfrak{B}(\mathfrak{G})$. For theoretical purposes, we use eqs. (19) and (20) as they are. The concept lattice of $\mathfrak{G}$ is denoted by $\mathfrak{B}(\mathfrak{G})$. It is a complete lattice; we denote the *infimum* by $\wedge$, the *supremum* by $\vee$, the *top concept* by $\top$ and the *bottom concept* by $\perp$. Every concept of $\mathfrak{B}(\mathfrak{G})$ has a *domain* $\mathrm{dom}(T, (\mathfrak{N}, \nu)) := \mathrm{schema}(T) \subseteq \omega$, and the X*-slice* of $\mathfrak{B}(\mathfrak{G})$ is the set $\mathfrak{B}(\mathfrak{G})[X] := \{C \in \mathfrak{B}(\mathfrak{G}) \mid \mathrm{dom}(C) = X\} \cup \{\perp\}$.

The operations of the DPJR algebra can be lifted to concepts, which results in *orbital concept lattices* [30]. The right multiplication on concepts is defined by $(T, [(\mathfrak{N}, \nu)]) \cdot \lambda := (T \cdot \lambda, [(\mathfrak{N}, \nu \circ \lambda)]) \in \mathfrak{B}(\mathfrak{G})$, where intents are classes of equivalent tableau queries, or their representives (for technical details see [14, Sect. 4.3]). Note that if $C \in \mathfrak{B}(\mathfrak{G})[Y]$ and $\lambda : X \to Y$, then $C \cdot \lambda \in \mathfrak{B}(\mathfrak{G})[X]$. Also in [30], we have introduced *equality concepts* $\mathfrak{E}_{xy}$ for each $(x, y) \in \omega \times \omega$. We now introduce a *deletion operation* del_x on $\mathfrak{B}(\mathfrak{G})$ for every $x \in \omega$, given by $\mathrm{del}_x(C) := C \cdot \pi_{X \setminus \{x\}}$ for $C \in \mathfrak{B}(\mathfrak{G})[X]$. The following definition is inspired by the definition of cylindric set algebras.

Definition 9. *The algebra* $\mathfrak{C}(\mathfrak{G}) := (\mathfrak{B}(\mathfrak{G}), \wedge, \perp, \top, \mathrm{del}_x, \mathfrak{E}_{xy}, \mathrm{dom})_{x,y \in \omega}$ *is the* full conjunctive concept algebra *with base* $\mathfrak{G}$. *A* conjunctive concept algebra *with base* $\mathfrak{G}$ *is a subalgebra of* $\mathfrak{C}(\mathfrak{G})$.

We infer from Prop. 4 that right multiplication is a derived operation on $\mathfrak{C}(\mathfrak{G})$; i.e. the conjunctive concept algebras coincide with the subalgebras of orbital concept lattices. Note that the primitive-positive formulas correspond to the finite tableau queries [14, Sect. 3.2][15]. The subalgebra $\mathfrak{C}_{\mathrm{fin}}(\mathfrak{G}) := \{C \in \mathfrak{B}(\mathfrak{G}) \mid \mathrm{ext}(C) \in \mathrm{res}_{\mathfrak{G}}(\mathrm{PP}(M))\}$ consists of the *primitive-positive definable* concepts of $\mathfrak{G}$. The concept algebra $\mathfrak{C}_{\mathrm{fin}}(\mathfrak{G})$ is essentially a concept algebra in the sense of Andreká and Németi [31], applied there specifically to cylindric set algebras, and used with other kinds of logic in [32].

By Prop. 5, for each set G, there exists a relational structure $\mathfrak{G}$ such that $\mathbf{Tab}(G) = \mathrm{res}_{\mathfrak{G}}[\mathrm{PP}(M)]$. In other words, $\mathfrak{C}_{\mathrm{fin}}(\mathfrak{G})$ is isomorphic to $\mathbf{Tab}(G)$. Then necessarily, we have $\mathfrak{C}_{\mathrm{fin}}(\mathfrak{G}) = \mathfrak{C}(\mathfrak{G})$. So in conclusion, for every set G, there exists a conjunctive concept algebra $\mathfrak{C}(\mathfrak{G})$ that is isomorphic to the table algebra $\mathbf{Tab}(G)$. This means that Props. 7 and 8 translate to concepts:

Proposition 10. *The conjunctive concept algebras (up to isomorphism) with non-empty base* $\mathfrak{G}$ *are precisely the projectional semilattices.*

Proposition 11. *The complete conjunctive concept algebras (up to isomorphism) with non-empty base* $\mathfrak{G}$ *are precisely the complete projectional semilattices.*

While we have not arrived at a Basic Theorem, a substantial connection to algebraic logic has been made. The remaining question is whether every complete subalgebra of a concept lattice $\mathfrak{B}(\mathfrak{G})$ is itself isomorphic to a concept lattice. We conjecture that this is the case.

Conjecture 12. *The concept algebras* $\mathfrak{C}(\mathfrak{G})$ *(up to isomorphism) over non-empty* $\mathfrak{G}$ *are precisely the complete projectional semilattices.*

6. Related Work

Imieliński and Lipski [24] have described a mapping from a relational algebra into a cylindric set algebra, which acts as an embedding under certain assumptions. As Düntsch and Mikulás[33] have pointed out, the table schema is not preserved by this mapping, so this mapping can not be truly considered an embedding. In order to preserve the table schema, they include a new element in the cylindric set algebra, which does not occur in tables. This new element amounts to a value of "undefined", so that the sets in the cylindric set algebra become sets of partial functions.

In this paper, we suggest to take a different route, and adapt the axioms of cylindric algebra to the database-theoretic setting. In his survey paper, Németi [27] presents variants of cylindric algebras, and also discusses the merits of such an approach [27, Sect. 7(4)], citing Howard [34] and Craig [35] as protagonists. However, they work with a different signature, which includes negation/complements, and which supports the unnamed perspective, while we present an axiomatization in the named perspective (cf. [15] for perspectives), which is closer to the original axioms. Variants of cylindric algebras, which are based on other first-order fragments (cylindrification only, cylindrification with union, cylindrification with union and intersection) are presented by Hansen [36].

7. Conclusion

We have characterized the $\wedge$-subalgebras of concept algebras $\mathfrak{C}(\mathfrak{G})$ by axioms in the style of cylindric algebras, and have more specifically likened them to cylindric set algebras. This establishes a connection between FCA and algebraic logic in the database-theoretic setting. In addition, we have obtained an axiomatic characterization of the $\bigwedge$-subalgebras of concept algebras $\mathfrak{C}(\mathfrak{G})$. Since $\bigwedge$-sublattices correspond to pattern projections [29], we have thus axiomatized conjunctive pattern concept algebras (to be defined in a suitable way). Moreover, we have conjectured that the concept algebras $\mathfrak{C}(\mathfrak{G})$ are precisely (u.t.i.) the complete projectional semilattices. The results raise the question how concept $\vee$-subalgebras and concept $\bigvee$-subalgebras can be axiomatically characterized. Moreover, while conjunctive concept algebras correspond to cylindric set algebras, is there also a well-motivated counterpart of cylindric algebras in this setting? Finally, we suggest to use *relational concept algebra* as a generic notion, and consider conjunctive concept algebras, as well as their orbital counterpart [30], as special kinds of relational concept algebras.

Acknowledgments

We wish to thank István Németi, who described a related notion of concept algebra in his work, for the kind feedback, and for pointing us to the article of Ben Hansen [36].

Declaration on Generative AI The author(s) have not employed any Generative AI tools.

References

[1] B. Ganter, R. Wille, Formal concept analysis: mathematical foundations, Springer, Berlin, 1999.
[2] R. Wille, Restructuring lattice theory: An approach based on hierarchies of concepts, in: I. Rival (Ed.), Ordered Sets, Reidel, Dordrecht–Boston, 1982, pp. 445–470.
[3] H. von Hentig, Magier oder Magister? Über die Einheit der Wissenschaft im Verständigungsprozeß, Suhrkamp, Frankfurt, 1974.
[4] R. Wille, Restructuring mathematical logic: an approach based on Peirce's pragmatism, in: A. Ursini, P. Aglianò (Eds.), Logic and Algebra, Marcel Dekker, New York, 1996, pp. 267–281.
[5] I. Kant, Logic, Dover, New York, 1988.
[6] R. Wille, Conceptual graphs and formal concept analysis, in: D. Lukose, H. S. Delugach, M. Keeler, L. Searle, J. F. Sowa (Eds.), Proceedings of ICCS 1997, 5th International Conference on Conceptual Structures, volume 1257 of *LNCS*, Springer, Heidelberg, 1997, pp. 290–303.
[7] J. F. Sowa, Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, Reading, MA, 1984.
[8] M. Chein, M.-L. Mugnier, Conceptual graphs: Fundamental notions, Revue d'Intelligence Artificielle 6 (1992) 365–406.

[9] R. Wille, Formal concept analysis and contextual logic, in: P. Hitzler, H. Schärfe (Eds.), Conceptual Structures in Practice, Chapman & Hall/CRC, 2009, pp. 137–173.
[10] M. Huchard, C. Roume, P. Valtchev, When concepts point at other concepts: the case of UML diagram reconstruction, in: Proceedings of the 2nd Workshop on Advances in Formal Concept Analysis for Knowledge Discovery in Databases (FCAKDD 2002), 2002, pp. 32–43.
[11] M. Huchard, M. R. Hacene, C. Roume, P. Valtchev, Relational concept discovery in structured datasets, Annals of Mathematics and Artificial Intelligence 49 (2007) 39–76.
[12] F. Baader, R. Molitor, Building and structuring description logic knowledge bases using least common subsumers and concept analysis, in: B. Ganter, G. W. Mineau (Eds.), Proceedings of ICCS 2000, volume 1867 of *LNCS*, Springer, Berlin, Heidelberg, 2000, pp. 292–305.
[13] J. Kötters, Concept lattices of a relational structure, in: H. D. Pfeiffer, D. I. Ignatov, J. Poelmans, N. Gadiraju (Eds.), Proceedings of ICCS 2013, volume 7735 of *LNCS*, Springer, 2013, pp. 301–310.
[14] J. Kötters, P. W. Eklund, Conjunctive query pattern structures: A relational database model for formal concept analysis, Discrete Applied Mathematics 273 (2020) 144–171.
[15] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, 1995.
[16] A. K. Chandra, P. M. Merlin, Optimal implementation of conjunctive queries in relational databases, in: Proceedings of the ninth annual ACM symposium on Theory of computing, STOC '77, ACM, New York, NY, USA, 1977, pp. 77–90.
[17] H.-D. Ebbinghaus, J. Flum, W. Thomas, Mathematical Logic, second ed., Springer, New York, 1994.
[18] E. F. Codd, A relational model of data for large shared data banks, Commun. ACM 13 (1970) 377–387.
[19] P. Hell, J. Nešetřil, Graphs and Homomorphisms, Oxford University Press, Oxford,New York, 2004.
[20] J. F. Sowa, Conceptual graphs for a data base interface, IBM Journal of Research and Development 20 (1976) 336–357.
[21] B. Groh, A Contextual-Logic Framework based on Relational Power Context Families, Ph.D. thesis, Griffith University, 2002.
[22] J. Kötters, S. E. Schmidt, Orbital semilattices, 2022. URL: https://arxiv.org/abs/2206.07790. doi:10.48550/ARXIV.2206.07790.
[23] J. Kötters, S. E. Schmidt, Conjunctive table algebras, 2024. URL: https://arxiv.org/abs/2404.01354.
[24] T. Imieliński, W. Lipski, The relational model of data and cylindric algebras, J. Comput. Syst. Sci. 28 (1984) 80–102.
[25] L. Henkin, J. D. Monk, A. Tarski, Cylindric Algebras, Part 1, North-Holland, 1971.
[26] L. Henkin, J. D. Monk, A. Tarski, Cylindric Algebras, Part 2, North-Holland, 1985.
[27] I. Németi, Algebraizations of quantifier logics, an introductory overview, https://old.renyi.hu/pub/algebraic-logic/survey.html, 1997. 12.1th Version.
[28] J. D. Monk, Lectures on cylindric set algebras, in: C. Rauszer (Ed.), Algebraic Methods in Logic and Computer Science, volume 28 of *Banach Center Publications*, Institute of Mathematics, Polish Academy of Sciences, Warsaw, 1993, pp. 253–290.
[29] B. Ganter, S. O. Kuznetsov, Pattern structures and their projections, in: H. S. Delugach, G. Stumme (Eds.), Proceedings of ICCS 2001, volume 2120 of *LNCS*, Springer, 2001, pp. 129–142.
[30] J. Kötters, S. E. Schmidt, Orbital concept lattices, in: T. Braun, D. Cristea, R. Jäschke (Eds.), Proceedings of ICCS 2022, volume 13403 of *LNAI*, Springer, 2022, pp. 66–81.
[31] H. Andréka, I. Németi, How many varieties of cylindric algebras are there, Transactions of the American Mathematical Society 369 (2017) 8903–8937. Http://dx.doi.org/10.1090/tran/7083.
[32] H. Andréka, Z. Gyenis, I. Németi, I. Sain, Universal Algebraic Logic. Dedicated to the Unity of Science., Birkhäuser, 2022.
[33] I. Düntsch, S. Mikulás, Cylindric structures and dependencies in relational databases, Theor. Comput. Sci. 269 (2001) 451–468.
[34] C. M. Howard, An Approach to Algebraic Logic, Ph.D. thesis, UC Berkeley, 1965.
[35] W. Craig, Logic in Algebraic Form, North-Holland, Amsterdam, 1974.
[36] B. Hansen, Finitizability questions for some reducts of cylindric algebras, in: Logic Colloquium '92, CSLI Publications and FOLLI, 1995, pp. 115–134.