Abstract. Main contribution of this paper is an investigation of expressive power of the database category $DB$. An object in this category is a database-instance (set of n-ary relations). Morphisms are not functions but have complex tree structures based on a set of complex query computations. They express the semantics of view-based mappings between databases. The higher (logical) level scheme mappings between databases, usually written in some high expressive logical language, may be functorially translated into this base "computation" $DB$ category. The behavioral point of view for databases is assumed, with behavioural equivalence of databases corresponding to isomorphism of objects in $DB$ category. The introduced observations, which are view-based computations without side-effects, are based (from Universal algebra) on monad endofunctor $T$, which is the closure operator for objects and for morphisms also. It was shown that $DB$ is symmetric (with a bijection between arrows and objects) 2-category, equal to its dual, complete and cocomplete.

In this paper we demonstrate that $DB$ is concrete, locally small and finitely presentable. Moreover, it is enriched over itself monoidal symmetric category with a tensor products for matching, and has a parameterized merging database operation. We show that it is an algebraic lattice and we define a database metric space and a subobject classifier: thus, $DB$ category is a monoidal elementary topos.

1 Introduction

The relational databases are complex structures, defined by sets of n-ary relations, and the mappings between them are based on sets of view-mappings between the source database $A$ to the target database $B$. We consider the views as an universal property for databases (possible observations of the information contained in some database).

We assume a view of a database $A$ the relation (set of tuples) obtained by a "Select-Project-Join + Union" (SPJRU) query $q(x)$ where $x$ is a list of attributes of this view. We denote by $\mathcal{L}_A$ the set of all such queries over a database $A$, and by $\mathcal{L}_A/\approx$ the quotient term algebra obtained by introducing the equivalence relation $\approx$, such that $q(x) \approx q'(x)$ if both queries result with the same relation (view). Thus, a view can be equivalently considered as a term of this quotient-term algebra $\mathcal{L}_A/\approx$ with carrier set of relations in $A$ and a finite arity of their operators, whose computation returns with a set of tuples of this view. If this query is a finite term of this algebra it is called a "finitary view."
Notice that a finitary view can have an infinite number of tuples also. Such an instance level database category $DB$ has been introduced first time in Technical report [1], and used also in [2]. General information about categories the reader can find in classic books [3], while more information about this particular database category $DB$, with set of its objects $Ob_{DB}$ and set of its morphisms $Mor_{DB}$, are recently presented in [4]. In this paper we will only emphasize some of basic properties of this $DB$ category, in order to render more self-contained this presentation.

Every object (denoted by $A, B, C, \ldots$) of this category is a database instance, composed by a set of $n$-ary relations $a_i \in A$, $i = 1, 2, \ldots$ called also "elements of $A$". Universal database instance $\Upsilon$, is defined as the union of all database instances, i.e., $\Upsilon = \{ a_i | a_i \in A, A \in Ob_{DB} \}$. It is a top object of this category.

It was defined [4] the power view-operator $T$, with domain and codomain equal to the set of all database instances, such that for any object (database) $A$, the object $TA$ denotes a database composed by the set of all views of $A$. The object $TA$, for a given database instance $A$, corresponds to the quotient-term algebra $L_A/\approx$, where carrier is a set of equivalence classes of closed terms of a well defined formulae of a relational algebra, "constructed" by $\Sigma_R$-constructors (relational operators in SPJRU algebra: select, project, join and union) and symbols (attributes of relations) of a database instance $A$, and constants of attribute-domains. More precisely, $TA$ is "generated" by this quotient-term algebra $L_A/\approx$, i.e., for a given evaluation of queries in $L_A$, $Eval_A : L_A \rightarrow TA$, which is surjective function, from a factorization theorem, holds that there is a unique bijection $is_A : L_A/\approx \rightarrow TA$, such that the following diagram in $Set$ category (where objects are sets, and arrows are functions) commutes

$$
\begin{array}{ccc}
L_A & \xrightarrow{Eval_A} & TA \\
\downarrow nat_{\approx} & & \downarrow is_A \\
L_A/\approx & & \\
\end{array}
$$

where the surjective function $nat_{\approx} : L_A \rightarrow L_A/\approx$ is a natural representation for the equivalence $\approx$.

For every object $A$ holds that $A \subseteq TA$, and $TA = TT A$, i.e., each (element) view of database instance $TA$ is also an element (view) of a database instance $A$.

Closed object in $DB$ is a database $A$ such that $A = TA$. Notice that also when $A$ is finitary (has a finite number of relations) but with at least one relation with infinite number of tuples, then $TA$ has an infinite number of relations (views of $A$), thus can be an infinitary object. It is obvious that when a domain of constants of a database is finite then both $A$ and $TA$ are finitary objects. As default we assume that a domain of every database is arbitrary large set but is finite. It is reasonable assumption for real applications. We have that $T = TT$, because every view $v \in TT$ is a database instance also, thus $v \in T$; and vice versa, every element $r \in T$ is also a view of $T$, thus $r \in TT$.

Every object (database) $A$ has also an empty relation $\bot$. The object (database) composed by only this empty relation is denoted by $\bot^0$ and we have that $T\bot^0 = \bot^0 = \{ \bot \}$.

Any empty database (a database with only empty relations) is isomorphic to this bottom
Consequently, an atomic morphism \( f \) map\( q \). We can introduce two functions, \( \partial_0, \partial_1 \) is not set of such view-mappings, thus \( \text{dom}, \text{cod} \) standard category functions \( \text{Mor} \). Consequently, an atomic morphism \( f : A \to B \), from a database \( A \) to database \( B \), is a set of such view-mappings, thus it is not generally a function.

We can introduce two functions, \( \partial_0, \partial_1 : \text{Mor}_{DB} \to \mathcal{P}(T) \) (which are different from standard category functions \( \text{dom}, \text{cod} : \text{Mor}_{DB} \to \text{Ob}_{DB} \)), such that for any view-map \( q_{A_1} : A \to TA \), we have that \( \partial_0(q_{A_1}) = \{r_1, \ldots, r_k\} \subseteq A \) is a subset of relations of \( A \) used as arguments by this query \( q_{A_1} \), and \( \partial_1(q_{A_1}) = \{v\}, v \in TA \) (\( v \) is a resulting view of a query \( q_{A_1} \)). In fact, we have that they are functions \( \partial_0, \partial_1 : \text{Mor}_{DB} \to \mathcal{P}(T) \) (where \( \mathcal{P} \) is a powerset operation), such that for any morphism \( f : A \to B \) between databases \( A \) and \( B \), which is a set of view-mappings \( q_{A_1} \), such that \( \|q_{A_1}\| \in B \), we have that \( \partial_0(f) \subseteq A \) and \( \partial_1(f) \subseteq TA \cap B \subseteq B \). Thus, we have

\[
\partial_0(f) = \bigcup_{q_{A_1} \in f} \partial_0(q_{A_1}) \subseteq \text{dom}(f) = A, \quad \partial_1(f) = \bigcup_{q_{A_1} \in f} \partial_1(q_{A_1}) \subseteq \text{cod}(f) = B
\]

We may define equivalent (categorically isomorphic) objects (database instances) from the behavioral point of view based on observations: each arrow (morphism) is composed by a number of "queries" (view-maps), and each query may be seen as an observation over some database instance (object of \( DB \)). Thus, we can characterize each object in \( DB \) (a database instance) by its behavior according to a given set of observations. Thus databases \( A \) and \( B \) are equivalent (bisimilar) if they have the same set of its observable internal states, i.e. when \( TA \) is equal to \( TB \) \( A \approx B \iff TA = TB \).

Basic properties of this database category \( DB \) as its symmetry (bijective correspondence between arrows and objects, duality (\( DB \) is equal to its dual \( DB^{0P} \)) so that each limit is also colimit (ex. product is also coproduct, pullback is also pushout, \( \bot^0 \) is zero objet, that is, both initial and terminal object, etc..), and that it is a 2-category has been demonstrated in [1, 4].

Generally, database mappings are not simply programs from values (relations) into computations (views) but an equivalence of computations: because of that each mapping, from any two databases \( A \) and \( B \), is symmetric and gives a duality property to the category \( DB \). The denotational semantics of database mappings is given by morphisms of the Kleisli category \( DB_T \) which may be "internalized" in \( DB \) category as "computations" [5].

The product \( A \times B \) of a databases \( A \) and \( B \) is equal to their coproduct \( A + B \), and the semantics for them is that we are not able to define a view by using relations of both databases, that is, these two databases have independent DBMS for query evaluation. For example, the creation of exact copy of a database \( A \) in another DB server corresponds to the database \( A + A \). In this paper we will introduce the denotational se-
mantics for other two fundamental database operations as matching and merging (and
data federation), and we will show other more advanced properties.

Plan of this paper is the following: After brief technical preliminaries taken from [1, 4,
6], in Section 2 we will consider some Universal algebra considerations and relations-
ships of DB and standard Set category. In Section 3 we will introduce the categorial
(functors) semantics for two basic database operations: matching and merging, while in
Section 4 we will define the algebraic database lattice and will show that DB is con-
crete, small and locally finitely presentable (lfp) category. In Section 5 we will show
that DB is also V-category enriched over itself. Finally in Section 6 we will develop a
metric space and a subobject classifier for this category, and we will show that it is a
weak monoidal topos.

1.1 Technical preliminaries

Based on atomic morphisms (sets of view-mappings) [4,6] which are complete arrows
c-arrows), we obtain that their composition generates tree-structures, which can be
incomplete (p-arrows), in the way that for a composed arrow
h = g ◦ f : A → C
of two atomic arrows f : A → B and g : B → C, we can have the situations where
∂0(f) ⊂ ∂0(h).

Definition 1. The following BNF defines the set MorDB of all morphisms in DB:
p-arrow : c-arrow | c-arrow ◦ c-arrow (for any two c-arrows f : A → B and
g : B → C)
morphism := p-arrow | c-arrow ◦ p-arrow (for any p-arrow f : A → B
and c-arrow g : B → C)

whereby the composition of two arrows, f (partial) and g (complete), we obtain the
following p-arrow (partial arrow)

h = g ◦ f = ∪

\{qBj\} ⊂ ∩\{qAi \mid qBj \in g \& \partial_0(qBj) \cap \partial_1(f) \neq \emptyset

∪

\{qA_i(tree)\}

= \{qBj \circ \{qA_i(tree) \mid \partial_1(qA_i) \subseteq \partial_0(qBj)\} \mid qBj \in g \& \partial_0(qBj) \cap \partial_1(f) \neq \emptyset

= \{qBj(tree) \mid qBj \in g \& \partial_0(qBj) \cap \partial_1(f) \neq \emptyset

where qA_i(tree) is the tree of the morphisms f below qA_i.

We define the semantics of mappings by function B_T : MorDB → ObDB, which,
given any mapping morphism f : A → B returns with the set of views ("information
flux") which are really "transmitted" from the source to the target object.

1. for atomic morphism, \( \tilde{f} = B_T(f) \) \( \triangleq T\{\|f_i\| \mid f_i \in f\} \).
2. Let g : A → B be a morphism with a flux \( \tilde{g} \), and f : B → C an atomic morphism
with flux \( \tilde{f} \) defined in point 1, then \( \tilde{f} \circ g = B_T(f \circ g) \) \( \triangleq \tilde{f} \cap \tilde{g} \).

We introduce an equivalence relation over morphisms by, \( f \approx g \) if \( f \) \( \tilde{f} = \tilde{g} \).
Notice that between any two databases $A$ and $B$ there is at least an “empty” arrow $f : A \rightarrow B$ such that $\partial_0(f) = \partial_1(f) = \tilde{f} = \perp^0$. Thus we have the following fundamental properties:

**Proposition 1** Any mapping morphism $f : A \rightarrow B$ is a closed object in DB, i.e., $\tilde{f} = T \tilde{f}$, such that $\tilde{f} \subseteq TA \cap TB$ and

1. each arrow such that $\tilde{f} = TB$ is an epimorphism $f : A \twoheadrightarrow B$,
2. each arrow such that $f = TA$ is a monomorphism $f : A \hookrightarrow B$,
3. each monic and epic arrow is an isomorphism.

If $f$ is epic then $TA \supseteq TB$; if it is monic then $TA \subseteq TB$. Thus we have an isomorphism of two objects (databases), $A \cong B$, iff $TA = TB$, i.e., when they are observationally equivalent $A \approx B$. Thus, for any database $A$ we have that $A \cong TA$.

Let us extend the notion of the type operator $T$ into a notion of the power-view endofunctor in $DB$ category:

**Theorem 1** There exists an endofunctor $T = (T^0, T^1) : DB \rightarrow DB$, such that

1. for any object $A$, the object component $T^0$ is equal to the type operator $T$, i.e., $T^0(A) \cong TA$
2. for any morphism $f : A \rightarrow B$, the arrow component $T^1$ is defined by
   $$T(f) \equiv T^1(f) = \bigcup_{\partial_{i}(q_{TA})(v) = \partial_{j}(q_{TA})(v) \& v \in \tilde{f}} \{q_{TA} : TA \rightarrow TB\}$$
3. Endofunctor $T$ preserves the properties of arrows, i.e., if a morphism $f$ has a property $P$ (monic, epic, isomorphic), then also $T(f)$ has the same property: let $P_{\text{mono}}, P_{\text{epi}}$ and $P_{\text{iso}}$ are monomorphic, epimorphic and isomorphic properties respectively, then the following formula is true
   $$\forall f \in \text{Mor}_{DB}(P_{\text{mono}}(f) \equiv P_{\text{mono}}(Tf) \text{ and } P_{\text{epi}}(f) \equiv P_{\text{epi}}(Tf) \text{ and } P_{\text{iso}}(f) \equiv P_{\text{iso}}(Tf)).$$

**Proof:** It is easy to verify that $T$ is a 2-endofunctor and to see that $T$ preserves properties of arrows: for example, if $P_{\text{mono}}(f)$ is true for an arrow $f : A \rightarrow B$, then $\tilde{f} = TA$ and $T \tilde{f} = T \tilde{f} = T(TA) = TA$, thus $P_{\text{mono}}(Tf)$ is true. Viceversa, if $P_{\text{mono}}(Tf)$ is true then $T \tilde{f} = T \tilde{f} = T(TA)$, i.e., $\tilde{f} = TA$ and, consequently, $P_{\text{mono}}(f)$ is true.

The equivalence relations on objects and morphisms are based on the “inclusion” Partial Order (PO) relations, which define the DB as a 2-category:

**Proposition 2** The subcategory $DB_I \subseteq DB$, with $\text{Ob}_{DB_I} = \text{Ob}_{DB}$ and with only monomorphic arrows, is a Partial Order category with PO relation of “inclusion” $A \preceq B$ defined by a monomorphism $f : A \hookrightarrow B$. The “inclusion” PO relations for objects and arrows are defined as follows:

$$A \preceq B \iff TA \subseteq TB$$
they determine observation equivalences, i.e.,

\[ f \preceq g \iff \tilde{f} \preceq \tilde{g} \quad (\text{i.e., } \tilde{f} \subseteq \tilde{g}) \]

The power-view endofunctor \( T : DB \rightarrow DB \) is a 2-endofunctor and the closure operator for this PO relation: any object \( A \) such that \( A = TA \) will be called "closed object".

\( DB \) is a 2-category, 1-cells are its ordinary morphisms, while 2-cells (denoted by \( \sqrt{\tau} \)) are the arrows between ordinary morphisms: for any two morphisms \( f, g : A \rightarrow B \), such that \( f \preceq g \), a 2-cell arrow is the "inclusion" \( \sqrt{\alpha} : f \rightarrow g \). Such a 2-cell arrow is represented by an ordinary monic arrow in \( DB \), \( \alpha : \tilde{f} \hookrightarrow \tilde{g} \).

The following duality theorem tells that, for any commutative diagram in \( DB \) there is also the same commutative diagram composed by the equal objects and inverted equivalent arrows: This "bidirectional" mappings property of \( DB \) is a consequence of the fact that the composition of arrows is semantically based on the set-intersection commutativity property for "information fluxes" of its arrows. Thus any limit diagram in \( DB \) has also its "reversed" equivalent colimit diagram with equal objects, any universal property has also its equivalent couniversal property in \( DB \).

**Theorem 2** there exists the controvariant functor \( \mathcal{S} = (S^0, S^1) : DB \rightarrow DB \) such that

1. \( S^0 \) is the identity function on objects.
2. for any arrow in \( DB \), \( f : A \rightarrow B \) we have \( S^1(f) : B \rightarrow A \), such that \( S^1(f) \triangleq f^{inv} \), where \( f^{inv} \) is (equivalent) reversed morphism of \( f \) (i.e., \( f^{inv} = \tilde{f} \)),

\[
(T f)^{inv} = \bigcup_{\partial_0(q_{TB}) = \partial_1(q_{TB}) = \{v\} \& v \in \tilde{f}} \{q_{TB} : TB \rightarrow TA\}
\]

3. The category \( DB \) is equal to its dual category \( DB^{op} \).

**Proof:** it can be found in [6]

\( \square \)

2 Universal algebra considerations

In order to explore universal algebra properties for the category \( DB \) [7], where, generally, morphisms are not functions (this fact complicates a definition of mappings from its morphisms into homomorphisms of the category of \( \Sigma_R \)-algebras), we will use an equivalent to \( DB \) "functional" category, denoted by \( DB_{sk} \), such that its arrows can be seen as total functions.
Proposition 3 Let us denote by $DB_{sk}$ the full skeletal subcategory of $DB$, composed by closed objects only.

Such a category is equivalent to the category $DB$, i.e., there exists an adjunction of a surjective functor $T_{sk} : DB \to DB_{sk}$ and an inclusion functor $In_{sk} : DB_{sk} \to DB$ such that $T_{sk}In_{sk} = Id_{DB_{sk}}$ and $In_{sk}T_{sk} \simeq Id_{DB}$.

There exists the faithful forgetful functor $F_{sk} : DB_{sk} \to Set$, and $F_{DB} = F_{sk} \circ T : DB \to Set$, thus $DB_{sk}$ and $DB$ are concrete categories.

Proof: Let us define $T_{sk}^0 = T^0$ and $T_{sk}^1 = T^1$, while $In_{sk}^0$ and $In_{sk}^1$ are two identity functions. It is easy to verify that these two categories are equivalent. In fact, there exists an adjunction $< T_{sk}, In_{sk}, \eta_{sk}, \varepsilon_{sk} > : DB \to DB_{sk}$, because of the bijection $DB_{sk}(T_{sk}A, B) \simeq DB(A, In_{sk}B)$ which is natural in $A \in DB$ and $B \in DB_{sk}$ ($B$ is closed, i.e., $B = TB$). In facts, $DB_{sk}(T_{sk}A, B) = \{ f/| f : TA \to B \} = \{ \tilde{g} | g : A \to In_{sk}B \} = DB(A, In_{sk}B)$. The skeletal category $DB_{sk}$ has closed objects only, so, for any two closed objects $TA$ and $TB$, each arrow between them $f : TA \to TB$ can be expressed in a following "total" form $f_T = f$ (such that $\partial_0(f_T) = TA$)

$$f_T \triangleq \bigcup_{\partial_0(q_{TA}) = \partial_0(q_{TA})} \{ q_{TA} \} \cup \{ q_{TA} \}$$

Thus, a morphism $f_T$ can be seen as a (total) function from $TA$ to $TB$, such that for any $v \in TA$ we have that $f_T(v) = v$ if $v \in f_T$, otherwise. Such an analog property is valid for its reversed equivalent morphism $f_T^{\text{rev}} : TB \to TA$ also.

Let us define the functor $F_{sk} : DB_{sk} \to Set$ by: $F_{sk}^0$ an identity function on objects and for any arrow $f : TA \to TB$ in $DB_{sk}$ we obtain a function $g = F_{sk}^0(f)$ from a set $TA$ to a set $TB$ such that for any relation $v \in TA$, $g(v) \triangleq \{ v | f \in f_T \}$, otherwise.

It is easy to verify that $F_{sk}^1(f) = F_{sk}(h)$ implies $f = h$, i.e., $F_{sk}$ is a faithful functor and also $F_{DB} = F_{sk} \circ T$ is a faithful. Thus $DB_{sk}$ and $DB$ are concrete categories. □

In a given inductive definition one defines a value of a function (in our example the endofunctor $T$) on all (algebraic) constructors (relational operators). What follows is based on the fundamental results of the Universal algebra [8].

Let $\Sigma_R$ be a finitary signature (in the usual algebraic sense : a collection $F_{\Sigma}$ of function symbols together with a function $ar : F_{\Sigma} \to N$ giving the finite arity of each function symbol) for a single-sorted (sort of relations) relational algebra.

We can speak of $\Sigma_R$-equations and their satisfaction in a $\Sigma_R$-algebra, obtaining the notion of a $(\Sigma_R, E)$-algebra theory. In a special case, when $E$ is empty, we obtain a purely syntax version of Universal algebra, where $K$ is a category of all $\Sigma_R$-algebras, and the quotient-term algebras are simply term algebras.

An algebra for the algebraic theory (type) $(\Sigma_R, E)$ is given by a set $X$, called the carrier of the algebra, together with interpretations for each of the function symbols in $\Sigma_R$. A function symbol $f \in \Sigma_R$ of arity $k$ must be interpreted by a function $f_X : X^k \to X$. Given this, a term containing $n$ distinct variables gives rise to a function $X^n \to X$ defined by induction on the structure of the term. An algebra must also satisfy the equations given in $E$ in the sense that equal terms give rise to identical
functions (with obvious adjustments where the equated terms do not contain exactly the same variables). A homomorphism of algebras from an algebra $X$ to an algebra $Y$ is given by a function $g : X \rightarrow Y$ which commutes with operations of the algebra $g(f(x_1, \ldots, x_k)) = f(g(x_1), \ldots, g(x_k))$.

This generates a variety category $K$ of all relational algebras. Consequently, there is a bifunctor $E : DB_{sk} \times K \rightarrow Set$ (where $Set$ is the category of sets), such that for any database instance $A$ in $DB_{sk}$ there exists the functor $E(A, \_ : K \rightarrow Set$ with an universal element $(U(A), g)$, where $g \in E(A, U(A))$, $g : A \rightarrow U(A)$ is an inclusion function and $U(A)$ is a free algebra over $A$ (quotient-term algebra generated by a carrier database instance $A$), such that for any function $f \in E(A, X)$ there is a unique homomorphism $h$ from the free algebra $U(A)$ into an algebra $X$, with $f = E(A, h) \circ g$.

From the so-called "parameter theorem" we obtain that there exists:

- a unique universal functor $U : DB_{sk} \rightarrow K$ such that for any given database instance $A$ in $DB_{sk}$ it returns with the free $\Sigma_R$-algebra $U(A)$ (which is a quotient-term algebra, where a carrier is a set of equivalence classes of closed terms of a well defined formulae of a relational algebra, "constructed" by $\Sigma_R$-constructors (relational operators: select, project, join and union S|P|R|U) and symbols (attributes and relations) of a database instance $A$, and constants of attribute-domains. An alternative for $U(A)$ is given by considering $A$ as a set of variables rather than a set of constants, then we can consider $U(A)$ as being a set of derived operations of arity $A$ for this theory. In either case the operations are interpreted syntactically $\hat{f}(\{t_1, \ldots, t_k\}) = [f(t_1, \ldots, t_k)]$, where, as usual, brackets denote equivalence classes), while, for any "functional" morphism (correspondent to the total function $F_{sk}(f_T)$ in Set, $F_{sk} : DB_{sk} \rightarrow Set$) $f_T : A \rightarrow B$ in $DB_{sk}$ we obtain the homomorphism $f_U = U^1(f_T)$ from the $\Sigma_R$-algebra $U(A)$ into the $\Sigma_R$-algebra $U(B)$, such that for any term $\rho(a_1, \ldots, a_n) \in U(A)$, $\rho \in \Sigma_R$, we obtain $f_U(\rho(a_1, \ldots, a_n)) = \rho(f_U(a_1), \ldots, f_U(a_n))$, so, $f_U$ is an identity function for algebraic operators and it is equal to the function $F_{sk}(f_T)$ for constants.

- its adjoint forgetful functor $F : K \rightarrow DB_{sk}$ such that for any free algebra $U(A)$ in $K$ the object $F \circ U(A)$ in $DB_{sk}$ is equal to its carrier-set $A$ (each term $\rho(a_1, \ldots, a_n) \in U(A)$ is evaluated into a view of this closed object $A$ in $DB_{sk}$) and for each arrow $U^1(f_T)$ holds that $F^1U^1(f_T) = f_T$, i.e., we have that $FU = Id_{DB_{sk}}$ and $UF = Id_K$.

Consequently, $U(A)$ is a quotient-term algebra, where carrier is a set of equivalence classes of closed terms of a well defined formulae of a relational algebra, "constructed" by $\Sigma_R$-constructors (relational operators in S|P|R|U algebra: select, project, join and union) and symbols (attributes of relations) of a database instance $A$, and constants of attribute-domains.

It is immediate from the universal property that the map $A \mapsto U(A)$ extends to the endofunctor $F \circ U : DB_{sk} \rightarrow DB_{sk}$. This functor carries monad structure $(F \circ U, \eta, \mu)$ with $F \circ U$ an equivalent version of $T$ but for this skeletal database category $DB_{sk}$.

The natural transformation $\eta$ is given by the obvious "inclusion" of $A$ into $F \circ U(A)$. 


\[ a \rightarrow [a] \] (each view \( a \) in an closed object \( A \) is an equivalence class of all algebra terms which produce this view). Notice that the natural transformation \( \eta \) is the unit of this adjunction of \( U \) and \( F \), and that it corresponds to an inclusion function in \( \text{Set} \), \( \rho : A \rightarrow U(A) \), given above. The interpretation of \( \mu \) is almost equally simple. An element of \((F \circ U)^2(A)\) is an equivalence class of all algebra terms built up from elements of \( F \circ U(A) \), so that instead of \( t(x_1, \ldots, x_k) \), a typical element of \((F \circ U)^2(A)\) is given by the equivalence class of a term \( t([t_1], \ldots, [t_k]) \). The transformation \( \mu \) is defined by map \([t([t_1], \ldots, [t_k])] \mapsto [t(t_1, \ldots, t_k)]\). This make sense because a substitution of provably equal expressions into the same term results in provably equal terms.

3 Matching and Merging database operations

In this section we will investigate the properties of \( DB \) category and, especially, its functorial constructs for the algebraic high-level operators over databases: for example \([9]\), matching, merging, etc..

3.1 Matching tensor product

Since the data residing in different databases may have inter-dependencies (they are based on the partial overlapping between databases, which is information about a common part of the world) we can define such an (partial) overlapping by morphisms of the category \( DB \): “information flux” of each mapping between two objects \( A \) and \( B \) in \( DB \) is just a subset of this overlapping between these two databases, denoted by \( A \otimes B \). It is “bidirectional”, i.e.,(by duality) for any mapping \( f \) from \( A \) into \( B \) there exists an equivalent mapping \( f^\text{inv} \) from \( B \) into \( A \). This overlapping represents the common matching between these two databases, and is equal to the maximal “information flux” which can be defined between these two databases. Consequently, we can introduce formally a denotational semantics for database matching operation \( \otimes \), as follows:

**Proposition 4** \( DB \) is a strictly symmetric idempotent monoidal category \((DB, \otimes, \mathcal{T}, \alpha, \beta, \gamma)\), where \( \mathcal{T} \) is the total object for a given universe for databases, with the ”matching” tensor product \( \otimes : DB \times DB \rightarrow DB \) defined as follows:

1. for any two database instances (objects) \( A \) and \( B \), \( A \otimes B \) is the overlapping (matching) between \( A \) and \( B \), defined by the bisimulation equivalence relation (i.e., by their common observations): \( A \otimes B \equiv (\bigcap \mathcal{T})(A, B) = TA \cap TB \)
2. for any two arrows \( f : A \rightarrow C \) and \( g : B \rightarrow D \), \( f \otimes g \equiv (f, g) \triangleq \bigcup \{ q(A \otimes B)_i \}_{\partial h(q(A \otimes B)_i) = \partial_i(q(A \otimes B)_i) = \{ v \} & v \in \bigcap g \}
3. for any two objects \( A, B \), every morphism \( f : A \rightarrow B \) satisfy \( \bot^0 \subseteq \bigcap \rightarrow A \otimes B \).

**Proof:** It is easy to verify that \( \otimes \) is monoidal bifunctor, with natural isomorphic transformations (which generate an identity arrow for each object in \( DB \)): \( \alpha : (\_ \otimes \_) \otimes _\_ \rightarrow \_ \otimes (\_ \otimes \_), \) associativity
for any two objects A, B, the arrow
Each object A together with two arrows, an isomorphism
for any monomorphism $\otimes$ The following properties for arrows in $DB$ are valid:

Notice that for any 2-cell $h$ that
Moreover, for any database $A$ we have that $A \otimes 1^0 = 1^0$.

$\bigotimes$ of the monoidal category $DB$ is not unique in contrast with the Cartesian product (we can have $A \otimes B = C \otimes B$ such that $C = A \bigcup A \simeq B$).

Notice that each $A \otimes B$ is a closed object (intersection of two closed objects $TA$ and $TB$), and that the "information flux" of any morphism from $A$ to $B$ is a closed object included in this "maximal information flux" (i.e., overlapping) between $A$ and $B$. Two completely disjoint databases have as overlapping (the maximal possible "information interchange flux") the empty bottom object $1^0$.

**Proposition 5** Each object $A$ together with two arrows, an isomorphism $\mu_A : A \otimes A \rightarrow A$ and an epimorphism $\eta_A : T \rightarrow A$, is a monoid in the monoidal category $(DB, \otimes, T, \alpha, \beta, \gamma)$.

**Proof:** It is easy to verify that is valid $\mu_A \circ (\mu_A \otimes id_A) \circ \alpha_{A,A,A} = \mu_A \circ (id_A \otimes \mu_A)$ and $\beta_A = \mu_A \circ (\eta_A \otimes id_A), \gamma_A = \mu_A \circ (id_A \otimes \eta_A)$.

**Proposition 6** The following properties for arrows in $DB$ are valid:

- for any two objects $A, B$, the arrow $h : A \rightarrow B$ such that $\hat{h} = A \otimes B$ is a principal morphism.
- for any monomorphism $f : A \rightarrow B$ and its reversed epimorphism $f^{inv} : B \rightarrow A$, $(f, f^{inv})$ is a retraction pair.
- for any object $A$ there is a category of idempotents on $A$ (denoted by $Ret_A$) defined as follows:
  1. objects of $Ret_A$ is the set of all arrows from $A$ into $A$, i.e., $Ob_{Ret_A} = DB(A, A)$.
  2. for any two objects $f, g \in Ob_{Ret_A}$ arrows between them are defined by the bijection $\varphi : DB(f, g) \simeq Ret_A(f, g)$ such that for any $h \in DB(f, g)$ holds $h \approx \varphi(h)$.

**Proof:** 1. for any $h : A \rightarrow B$ with $\hat{h} = A \otimes B$ holds that $\forall f : A \rightarrow B, \exists \tilde{g} : A \rightarrow A$, such that $f = h \circ g$ (in fact for $g \approx f$ it is satisfied).
2. for any monomorphism $f : A \rightarrow B$ and epimorphism $f^{inv} : B \rightarrow A$ ($f^{inv} \approx f$) holds that $f^{inv} \circ f = id_A$ (in fact, $f^{inv} \circ f = f^{inv} \bigcap \tilde{f} = \tilde{f} = TA = id_A$).
3. for each $f : A \rightarrow A$ holds $f \circ \tilde{f} = f$. Thus, it is idempotent and, consequently, an object in $Ret_A$. For any $h \in DB(f, g)$ in $DB$, the arrow $k = \varphi(h) \in Ret_A(f, g)$, such that $h \approx k$ satisfies $k = g \circ k \circ f$ in $DB$. Demonstration: from $h : \tilde{f} \rightarrow \tilde{g}$ it holds that $\hat{h} \subseteq T\tilde{f} \bigcap T\tilde{g} = \tilde{f} \bigcap \tilde{g}$, consequently $\tilde{k} = \tilde{k} \bigcap \tilde{f} \bigcap \tilde{g} = g \circ k \circ f$.

Notice that for any 2-cell $h : \tilde{f} \leq \tilde{g}$ we have that $\varphi(h) = f \in Ret_A(f, g)$ (in fact, $h$ is monomorphism, thus, $\hat{h} = \tilde{f}$ and also $\tilde{h} = \varphi(h)$, thus $\varphi(h) = \tilde{f}$, i.e., $\varphi(h) = f$).

$\square$
3.2 Merging operator

Merging of two databases $A$ and $B$ is similar to the concept of union of two databases in one single database. As we will show, this similarity corresponds to an isomorphism in $DB$. That is, the union of two databases is isomorphic to the database obtained by their merging, from the behavioral point of view. Any view which can be obtained from union of two databases, can also be obtained from merging these two databases, and vice versa.

In what follows, similarly to matching tensor products which, for any two given databases, returns with only closed objects, also the merging operator will return with closed objects. As we will see these two operators will result as meet and joint operators of complete algebraic database lattice where $\bot^0$ and $T$ are bottom and top elements respectively.

**Proposition 7** For any fixed database (object) $A$ in $DB$ we define the parameterized “merging with $A$” operator as an endofunctor $A \oplus -$ : $DB \longrightarrow DB$ as follows:

1. for any database instance (object) $B$, $A \oplus B$ is a merging of $A$ and $B$, defined by the bisimulation equivalence relation: $A \oplus B \equiv \oplus(A, B) \triangleq (T \cdot \bigcup)(A, B) = T(A \cup B)$
2. for any arrow $f : B \rightarrow C$, $A \oplus (f) \triangleq (id_A \cup f) : A \oplus B \rightarrow A \oplus C$, such that $A \oplus (f) = A \oplus \tilde{f}$.

**Proof:** It is easy to verify that $T(A \cup B) = T(TA \cup TB)$, that is $A \oplus B = TA \oplus TB$, and $A \cup B \simeq TA \cup TB \simeq A \oplus B$.

Now we can verify that $A \oplus -$ is an endofunctor. In fact, for any identity arrow $id_B : B \rightarrow B$, we have that $A \oplus (id_B) = id_A \cup id_B$, so that $A \oplus (id_B) = T(id_A \cup id_B) = TTA \cup TTB = TA \cup TB = id_{A \oplus B}$. Consequently, for identity arrows holds functorial property, $A \oplus (id_B) = id_{A \oplus B}$.

From the fact that for any object (database) $B$, we have that $A \subseteq A \oplus B$, each arrow resulting by application of this endofunctor contains a sub arrow $id_A$. Thus, given two arrows $f : B \rightarrow C$ and $g : C \rightarrow D$, we have the compositional endofunctors property, $A \oplus (g) \circ A \oplus (f) = (id_A \cup g) \circ (id_A \cup f) = id_A \cup (g \circ f) = A \oplus (g \circ f)$.

Moreover, we have that $A \oplus B = B \oplus A$, $A \oplus \emptyset = A$, $A \oplus \bot^0 = A$, $A \oplus TA = TA$ and $A \oplus A \simeq A$.

□

Matching and merging operators are dual operators in the category $DB$: in fact they are also dual lattice operators (meet and join respectively) w.r.t. the database ordering $\preceq$, as we will show in what follows.

Notice that for the objects in database category, the commutative operation of merging $\oplus = T \cdot \bigcup$ is a generalization of the set union operation $\bigcup$ in the category of sets $Set$.

**Remark:** Data federation of two databases $A$ and $B$ is their union, that is a database $A \cup B$. It is easy to see that $A \cup B \simeq A \oplus B$, that is, from the behavioral point of view, data federation is equivalent to data merging, that is for any query over data federation $A \cup B$, which returns with a view $r$, there exists a query over data merging $A \oplus B$ which returns with the same view $r$; and vice versa.
4 Algebraic database lattice

We have seen that the set of all closed objects (i.e., objects of the skeletal category $DB_{sk}$, equivalent to $DB$), denoted by $C \triangleq Ob_{DB_{sk}}$, defines a closed set system $< Y, C >$, where $Y$ is a closed "total" object (a merging, or up to isomorphism a union (we have that $A \cup B \simeq A \oplus B$), of all objects (database instances) of $DB$), correspondent to the closure operator $T$. Thus, the lattice $< C, \subseteq >$ with respect to the set-inclusion $\subseteq$ is a complete lattice [8]. We recall the fact that a complete lattice is a poset $P$ such that for any subset $S$ both $\inf S$ (greatest lower bound) and $\sup S$ (least upper bound) exist in $P$: for any $A, B, C \in C$, and binary operations "join" $\lor$ and "meet" $\land$ (in the case of $Set$ category these operators are set-union and set-intersection respectively, while for $DB$ category we will show that they are merging and matching operators respectively), the following identities are satisfied

1) $A \lor B = B \lor A, \quad A \land B = B \land A$ commutative laws
2) $A \lor (B \land C) = (A \lor B) \lor A, \quad A \land (B \lor C) = (A \land B) \lor C$ associative laws
3) $A \lor A = A, \quad A \land A = A$ idempotent laws
4) $A = A \lor (A \land B), \quad A = A \land (A \lor B)$ absorption laws

By definition, a closed-set system is algebraic if $C$ is closed under unions of upward directed subsets, i.e., for every $S \subseteq C, \lor S \in C$. Equivalently, the closure operator $J$ of a closure-set system $< Y, C >$ is algebraic if it satisfy the following "finitary" property: for any upward directed subset $X \subseteq Y$

$$J(X) = \bigcup\{J(X') \mid X' \subseteq_\omega X\}$$

where $X' \subseteq_\omega X$ means that $X'$ is a finite subset of $X$.

A lattice is algebraic if it is complete and compactly generated: a lattice $< C, \subseteq >$ is compactly generated if every element of $C$ is a sup of compact elements less then or equal to it, i.e., for every $A \in C, A = \sup\{B \in \text{Comp} C \mid B \subseteq A\}$ (an element is compact $B \in \text{Comp} C$ if, for every $X \subseteq C$ such that $\sup X$ exists, $B \subseteq \sup X$ implies there exists a $X' \subseteq_\omega X$ such that $B \subseteq \sup X'$). Set of compact elements in an algebraic lattice is the set of all closed elements obtained from finite subsets.

We define the finite objects in $DB$ the databases with a finite number of n-ary ($n$ is a finite number $n \in \omega$, the nullary relation is $\bot$ and is an element of each object in $DB$ category) relations (elements); the extension of relations does not necessarily be finite - in such a case for a finite object $A$ in $DB$, the object $TA$ is composed by infinite number of relations, that is $TA$ is an infinite object.

We will demonstrate that this database lattice is an algebraic lattice.

Proposition 8 Let $C = Ob_{DB_{sk}}$ denotes the set of all closed objects of $DB$ category. The following properties for a database closure are valid:

- A closed-set system $< Y, C >$ consists of the "total" closed object (top database instance) $Y \in C$, which is a merging (or, up to isomorphism, a union) of all objects in $DB$, and the set $C \subseteq \mathcal{P}(Y)$, such that $C$ is closed under intersections of arbitrary subsets. That is, for any $K \subseteq C, \bigcap K \in C$.
- The closure operator $T$ is algebraic.
- $< C, \subseteq >$ is an algebraic lattice with meet $\otimes$ and join $\oplus$ operators. The compact elements of $< C, \subseteq >$ are closed objects of $DB$ category $T(A)$ generated by finite objects $A \subseteq_\omega Y$. 

12
Proof: It is easy to verify that $C$ is closed under intersection, it is a poset of closed objects of $DB$ category (i.e., a set of objects of the equivalent skeletal category $DB_{sk}$), which is a subset of the total object $T$, with set inclusion as a partial ordering.

The closure operator is $T : Ob_{DB} \rightarrow Ob_{DB}$. We have that each object $A \in Ob_{DB}$ is a subset of $T$, and vice versa, each subset of $T$ is a database instance, thus an object in $DB$ category. From Universal algebra theory it holds that each closure operator, and its equivalent closure-set system $< T, C >$, generates a complete lattice $< C, \subseteq >$, such that for any subset $K \subseteq C$ of closed objects $K = \{ TA_i \mid i \in I, A_i$ closed set of $DB \}$ we have that:

Greatest lower bound $\bigwedge K = \bigwedge_{i \in I} TA_i = \bigcap_{i \in I} TA_i = \bigcap K = \oplus_{i \in I} TA_i$, that is, meet lattice operator $\bigwedge$ corresponds to the matching operation $\oplus$.

Least upper bound $\bigvee K = \bigvee_{i \in I} TA_i = T(\bigcup_{i \in I} A_i) = \oplus_{i \in I} TA_i$, that is, join lattice operator $\bigvee$ corresponds to the merging operation $\oplus$.

so that for $K = C$ we obtain $\bigvee C = T(\bigcup_{A_i \in Ob_{DB}} A_i) = T$.

Let us prove that $T$ is algebraic: let $T_{\Sigma} = U(T)$, where $U : DB_{sk} \rightarrow K$ is the unique universal functor described previously in Section 2, be a $\Sigma_{R}$-algebra generated by $T$, and $A \subseteq \Sigma$ a database instance (each object $A$ in $DB$ satisfy $A \subseteq \Sigma$). The $A$ is subuniverse of $T_{\Sigma}$ if for all $\Sigma_{R}$-algebra operators $\sigma \in \Sigma_{R}$, $ar(\sigma) = n$, and $a_{1}, \ldots, a_{n} \in A$, $\sigma(a_{1}, \ldots, a_{n}) \in A$, i.e., $A$ is closed under $\sigma$ for each $\sigma \in \Sigma_{R}$. Thus, each subuniverse $A$ of $T_{\Sigma}$ is a closed object in $DB$. The set of all subuniverses of $T_{\Sigma}$ (i.e., the set of all closed objects of $DB$) is denoted by $Sub(T_{\Sigma})$.

$T_{\Sigma}$ defines, for every $A \subseteq \Sigma$ the subuniverse generated by $A$, $Sg(A) = \bigcap\{ B \mid A \subseteq B$ and $B \in Sub(T_{\Sigma}) \}$, where $Sg : P(\Sigma) \rightarrow Sub(T_{\Sigma})$ is an algebraic operator (Theorem of Universal algebra). Let us verify that $T \equiv Sg$:

In fact, for any $A \in P(\Sigma), A \in Ob_{DB}$, we obtain $Sg(A) = \bigcap\{ B \mid A \subseteq B$ and $B \in Sub(T_{\Sigma}) \} = \bigcap\{ B \mid A \subseteq B$ and $B$ is closed object in $DB \} = \bigwedge\{ B \mid A \subseteq B$ and $B$ is closed object in $DB \} = TA$ because $TA$ is the least closed object $B = TA$ in $DB$ such that $A \subseteq B$ (from the property of the closure operator $T$). Thus, $T$ is an algebraic closure operator and, consequently, the lattice $< C, \subseteq >$ and the closed-set system $< T, C >$ are algebraic.

Now we can extend the lattice $< C, \subseteq >$ of only closed objects of $DB$ into a lattice of all objects of $DB$ category:

**Proposition 9** The set $Ob_{DB}$ of all database instances (objects)of $DB$, together with merging and matching tensor products $\oplus$ and $\otimes$ (read "join" and "meet" respectively) is a lattice.

Proof: We have to prove that:

1) $A \oplus B \simeq B \oplus A, \quad A \otimes B \simeq B \otimes A$, commutative laws
2) $A \oplus (B \oplus C) \simeq (A \oplus B) \oplus C, \quad A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$, associative laws
3) $A \oplus A \simeq A, \quad A \otimes A \simeq A$, idempotent laws
4) $A \simeq A \oplus (A \oplus B), \quad A \simeq A \otimes (A \otimes B)$, absorption laws.

The commutative, associative and idempotent laws holds directly from functorial definition of $\oplus$ and $\otimes$. Let us prove (4): We have that $A \oplus (A \oplus B) = T(A \cup (TA \cap TA)) \subseteq T(A \cup TA) = TTA = TA$, thus we obtain that $A \oplus (A \oplus B) = TA \simeq A$. 

13
Analogously, \( A \otimes (A \oplus B) = TA \cap T(A \cup B) = TA \simeq A \).

\[\square\]

Let us denote by \( T_I : DB_I \rightarrow DB_I \) the restriction of closure endofunctor \( T : DB \rightarrow DB \). We have seen in Proposition 2 that \( DB_I \) is a PO category where each arrow \( f : A \leftarrow B \) is a monomorphism, i.e., \( A \preceq B \). Thus, we obtained a partial order \( \prec Ob_{DB}, \preceq > \). Let us show that it is a lattice ordered set; i.e., that every pair of objects \( A, B \in Ob_{DB} \) has a least-upper-bound (sup) and the greatest-lower-bound (inf).

**Proposition 10** Poset \( \prec Ob_{DB}, \preceq > \) is a lattice ordered set where \( A \preceq B \) if \( A \preceq A \oplus B \) (or equivalently \( B \simeq A \oplus B \)), so that for all \( A, B \in Ob_{DB}, \inf (f(A, B)) = A \oplus B \) and \( \sup (A, B) = A \oplus B \). It is a complete lattice.

**Proof:** In fact, if \( A \preceq A \oplus B \) then \( TA \cap TB \preceq A \preceq TA \), thus \( TA \cap TB = TA \) and \( TA \preceq TB \), i.e., \( A \preceq B \). Or, equivalently, if \( B \preceq A \oplus B \) then \( TB = T(A \cup B) \supseteq TA \), thus \( A \preceq B \). We have also that \( ob_{DB} \supseteq C \), where \( \prec C, \preceq > \) is algebraic lattice of closed objects. Thus, for any subset \( K \subseteq ob_{DB} \) we have that \( inf(K) = \bigcap_{A_i \in K} TA_i \subseteq C \), thus, from \( C \subseteq ob_{DB} \) we obtain that \( inf(K) \in ob_{DB} \), i.e., the lattice \( \prec ob_{DB} \preceq > \) for every subset \( K \) has a least-upper-bound and, consequently, it is a complete lattice.

\[\square\]

**Corollary 1** PO subcategory \( DB_I \subseteq DB \), \( DB_I = \prec ob_{DB}, \preceq > \) is an algebraic lattice isomorphic to the lattice \( \prec C, \preceq > \).

**Proof:** If we define an equivalence classes for \( \prec ob_{DB}, \preceq > \) w.r.t. the equivalence relation ”\( \preceq \)”, such that \( [A] = \{ B \mid B \in ob_{DB} \text{ and } B \preceq A \} \), so that \( \prec C, \preceq > \) is its quotient lattice (we consider lattices as algebras) such that elements of this quotient lattice (algebra) are closed objects \( [A] = TA \) only. The function \( \alpha : \prec ob_{DB}, \preceq > \rightarrow \prec C, \preceq > \), such that for any \( A \in ob_{DB}, \alpha(A) = \alpha(TA) = TA \), i.e., \( \alpha \equiv T \), is an order-preserving bijection (\( A \) and \( TA \) are indistinguishable elements in the lattice \( \prec ob_{DB}, \preceq > \), thus \( | \prec ob_{DB}, \preceq > | = | \prec C, \preceq > | \)), while the function \( \alpha^{-1} : \prec C, \preceq > \rightarrow \prec ob_{DB}, \preceq > \) is an order-preserving identity function. Thus, \( \alpha \) is an isomorphism of lattices, and, consequently, also \( \prec ob_{DB}, \preceq > \) is algebraic.

\[\square\]

Database lattice \( DB_I = \prec ob_{DB}, \preceq > \) is bounded: it has the largest element \( \top \) (element that is upper bound of every element of the lattice), and also the smallest element \( \bot^0 \). The algebraic property is very useful in order to demonstrate the properties of \( DB \) category: in order to prove theorems in general we need to be able to extend inductive process of proof beyond \( \omega \) steps to the transfinite. Zorn’s lemma (equivalent to the Axiom of Choice of set theory) allows us to do this. The database lattice \( \prec ob_{DB}, \preceq > \) is a (nonempty) poset with the property that every chain \( K \subseteq ob_{DB} \) (i.e., linearly ordered subset) has an upper bound \( \forall K = \bigcup K \) (because this poset is algebraic) in \( ob_{DB} \). Then we can apply the Zorn’s lemma which asserts that \( \prec ob_{DB}, \preceq > \) has a maximal element.

**Remark:** From the fact that \( DB_I = \prec ob_{DB}, \preceq > \) is an algebraic lattice we obtain that for the total object \( \top \) the following is valid: \( \top = \top \top = \bigcup \{ TA \mid A \subseteq_i \top \} = \bigvee \{ TA \mid A \text{ is finite such that } A \preceq \top \} \), it is the union of all closed objects generated by only finite objects of \( DB \), i.e., the union of all compact elements of \( \prec C, \preceq > \). Similarly,
For each object $A, B$ in a poset, such that for any two objects $J = \langle J, \leq \rangle$, the total order relation $j \leq k$, i.e., $\omega = \{0 \rightarrow 1 \rightarrow 2 \rightarrow \ldots \}$. An endofunctor $H : C \rightarrow D$ is $\omega$-cocontinuous if preserves the colimits of functors $J : \omega \rightarrow C$, that is when $H \text{Colim} J \simeq \text{Colim} H J$ (the categories $C$ and $D$ are thus supposed to have these colimits). Notice that a functor $J : \omega \rightarrow C$ is a diagram in $C$ of the form $\{C_0 \rightarrow C_1 \rightarrow C_2 \rightarrow \ldots \}$. For $\omega$-cocontinuous endofunctors the construction of the initial algebra is inductive [10].

**Proposition 11** For each object $A$ in the category $DB$ the ”merging with $A$” endofunctor $\sum_A = A \oplus \cdot : DB \rightarrow DB$ is $\omega$-cocontinuous.

**Proof:** Let us consider any chain in $DB$ (all arrows are monomorphisms, i.e., ”$\leq$” in a correspondent chain of the $\langle Ob_{DB}, \leq \rangle$ algebraic lattice), is a following diagram $D \supseteq_0 (\sum_A \supseteq_0) \supseteq_1 (\sum_A \supseteq_0) \supseteq_2 \ldots \supseteq_A$, where $\supseteq_0$ is the initial object in $DB$, with unique monic arrow $\supseteq_0 : \supseteq_0 i \supseteq (\sum_A) \supseteq_0$ with $\supseteq_0 = \supseteq_0$, and consecutive arrows $\supseteq_n = \sum_A \supseteq_0 \cdot (\sum_A \supseteq_0) i \supseteq_0$ with $\sum_A \supseteq_0 = T A$, for all $n \geq 1$, as representation of a functor $J : \omega \rightarrow DB$. The endofunctor $\sum_A$ preserves colimits because it is monotone and $\sum_A = T A$ is its fixed point, i.e., $\sum_A = T A = T(A \cup T A) = T(A \cup \sum_A) = \sum_A(T \sum_A)$. Thus, the colimit $\text{Colim} J = \sum_A$ of the base diagram $D$ given by the functor $J : \omega \rightarrow DB$, is equal to $\text{Colim} J = (A \oplus \cdot) \supseteq_0 i \supseteq_0 T A$. Thus $\sum_A \text{Colim} J = T(A \cup T \text{Colim} J) = T(A \cup \sum_A) = T(T A) = T A = \text{Colim} \sum_A J$ (where $\text{Colim} \sum_A J$ is a colimit of the diagram $\sum_A D$).

The $\omega$-cocompleteness amounts to chain-completeness, i.e., to the existence of least upper bound of $\omega - \text{chains}$. Thus $\sum_A$ is $\omega$-cocontinuous endofunctor: a monotone function which preserves lubs of $\omega - \text{chains}$.

In what follows we will pass from lattice based concepts, as lubs of directed subsets, compact subsets, and algebraic lattices, to categorically generalized concepts as directed colimits, finitely presentable (fp) objects, and locally finitely presentable (lfp) categories respectively:

A directed colimit in $DB$ is a colimit of the functor $F : \langle J, \leq \rangle \rightarrow DB$, where $\langle J, \leq \rangle$ is a directed partially ordered set, such that for any two objects $j, k \in J$ there is an object $l \in J$ such that $j \preceq l, k \preceq l$, considered as a category. For example, when $J = \text{Ob}_{DB}$ we obtain the algebraic (complete and compact) lattice which is an directed PO-set, such that for any two objects $A, B \in J$ there is an object $C \in J$ with $A \preceq C$ and $B \preceq C$ (when $C = \text{sup}(A, B) \in J$).

An object $A$ is said to be finitely presentable (fp), or finitary, if the functor $DB(A, \_ : DB \rightarrow \text{Set}$ preserves directed colimits (or, equivalently, if it preserves filtered colimits). We write $DB_{fp}$ for the full subcategory of $DB$ on the finitely presentable objects: it is essentially small. Intuitively, fp objects are ”finites objects”, and a category is lfp if it can be generated from its finite objects: a strong generator $M$ of a category is its small
full subcategory such that \( f : A \rightarrow B \) is an isomorphism iff for all objects \( C \) of this subcategory, given a hom-functor \( M(C, \_ ) : M \rightarrow \text{Set} \), the following isomorphism of hom-sets \( M(\tilde{C}, f) : M(C, A) \rightarrow M(\tilde{C}, B) \) in \( \text{Set} \) is valid.

From Th.1.11 [11] a category is locally finitely presentable (lfp) iff it is cocomplete and has a strong generator.

**Corollary 2** \( DB \) and \( DB_{sk} \) are concrete, locally small, and locally finitely presentable categories (lfp).

**Proof:** Given any two objects \( A, B \) in \( DB \), the hom-set \( DB(A, B) \) of all arrows \( f : A \rightarrow B \) corresponds to the directed subset \( K = \{ f \mid \bot \leq f \leq A \otimes B \} \subseteq C \), which is bounded algebraic (complete and compact) sublattice of \( C \). Thus, the set of all arrows \( f : \mathcal{T} \rightarrow \mathcal{T} \) corresponds to the directed set \( K = \{ f \mid \bot \leq f \leq \mathcal{T} \otimes \mathcal{T} = \mathcal{T} \} \), which is equal to the lattice \( C, \subseteq \). Thus, \( DB \) is locally small (has small hom-sets), and, by \( DB \supseteq DB_{sk} \), also \( DB_{sk} \) is locally small.

Let us show that the full subcategory \( DB_{fin} \), composed by closed objects obtained from finite database objects, is a strong generator of \( DB \): in fact, if \( TA, TB \in DB \) and \( A \simeq B \) are two finite databases (so that \( TA = TB \)) then for all \( C \in DB_{fin} \), \( |DB(C, TA)| \) is a rank of the complete sublattice \( \bot \leq D \leq C \otimes TA \), while \( |DB(C, TB)| \) is a rank of the complete sublattice \( \bot \leq D_{1} \leq C \otimes TB \). From \( A \simeq B \) we deduce \( C \otimes TA = TC \otimes TA = TC \otimes TB = C \otimes TB \), thus \( |DB(C, TA)| = |DB(C, TB)| \), i.e, there is a bijection \( \nu : |DB(C, TA)| \cong |DB(C, TB)| \) which is an isomorphism in \( \text{Set} \). Thus, \( DB \), which is cocomplete and has this strong generator \( DB_{fin} \), is a lfp.

We define a representable functor \( DB(A, \_ ) : DB \rightarrow \text{Set} \), such that \( DB(A, B) \) is the set of functions \( \{ F_{DB}(f) \mid \text{for each } f : A \rightarrow B \text{ in } DB \} \), and for any arrow \( g : B \rightarrow C, DB(A, g) \) is the function such that for any function \( f \in DB(A, B) \) we obtain the function \( h = DB(A, g) \trianglelefteq F_{DB}(g) \circ f \in DB(A, C) \).

We say that a functor \( H : DB \rightarrow \text{Set} \) preserves colimits if the image \( H_{\nu} : HF \rightarrow H\text{Colim}F \) for the colimit \( (\nu, \text{Colim}F) \) of a functor \( F \in DB^{J} \) is a colimiting cone (or cocone) for \( HF \) (in this case we are interested for \( H = DB(T, \_ ) \)).

Let us show, for example, that the object \( T \) is a finitely presentable (fp) (it was demonstrated previously by remark that \( \mathcal{T} = TT = \bigcup \{ TA \mid A \subseteq \mathcal{T} \} = \square \{ TA \mid A \text{ is finite such that } A \subseteq \mathcal{T} \} \)), i.e., the fact that its hom-functor \( DB(T, \_ ) : DB \rightarrow \text{Set} \) preserves directed colimits:

**Proposition 12** Total object (matching monoidal unit) \( T \) is a finitely presentable (fp).

**Proof:** Let us have a \( \text{Colim}F \) in \( DB \) (a colimit of the functor \( F \in DB^{J} \), where \( F \) can be seen as a base diagram for this colimit, composed by a finite number of objects \( B_{1}, ..., B_{n} \) with PO-arrows \( "\simeq" \) between them), such that arrows \( h_{i} : B_{i} \rightarrow \text{Colim}F \), are components of the cone \( (\nu, \text{Colim}F) \) where \( \nu : F \rightarrow \Delta\text{Colim}F \) is a natural transformation and \( \Delta \) is a diagonal constant functor.

Let us show that for any other cocone \( E \) in \( \text{Set} \), for the same cocone-base \( HF \) (where \( H = DB(T, \_ ) : DB \rightarrow \text{Set} \)) there is an unique arrow (function) from \( DB(T, \text{Colim}F) \) to the set \( E \) (vertex of a cocone \( E \)). We can see that, for a set of all objects in the diagram (functor) \( F, S = \{ B_{i} \mid B_{i} \in F \} \subseteq Ob_{DB}, holds that \( \text{Colim}F = \sup(S) = \)
there is an unique arrow in
Set
∈
\mathcal{E}
\text{ension functions}
< \mathbf{T}v
for any
E
\subseteq S,
lattice
On the other hand
H\text{ColimF} = DB(T, \text{Colim}F)
is isomorphic to the complete sub-
lattice \langle S, \subseteq \rangle, where \( S = \{ \tilde{\mathbf{f}} \subseteq \sum_{B_i \in F} TB_i \} \). Thus, all arrows of the cocone
H\mathbf{\nu}, DB(T, h_i) : DB(T, B_i) \hookrightarrow DB(T, \text{Colim}F) are inclusion functions
\langle \{ \tilde{\mathbf{f}} \subseteq \mathbf{T}B_i \}, \subseteq \rangle \subseteq \langle S, \subseteq \rangle (also each arrow in the base diagram \( H\mathbf{F} \) in Set are inclusion functions
\langle \{ \tilde{\mathbf{f}} \subseteq \mathbf{T}B_j \}, \subseteq \rangle \subseteq \langle \{ \tilde{\mathbf{f}} \subseteq \mathbf{T}B_k \}, \subseteq \rangle).
All arrows of the cocone \( E, \ k_i : \langle \{ \tilde{\mathbf{f}} \subseteq \mathbf{T}B_i \}, \subseteq \rangle \to E \), must be equal functions (only with different domains) in order to preserve the commutativity of this colimiting cocone \( E \): thus the function \( k : \langle S, \subseteq \rangle \to E \) is an unique function such that, for any \( v \in \langle S, \subseteq \rangle, k(v) = k_i(v) \) for some \( k_i : \langle \{ \tilde{\mathbf{f}} \subseteq \mathbf{T}B_i \}, \subseteq \rangle \to E \) and \( v \in \langle \{ \tilde{\mathbf{f}} \subseteq \mathbf{T}B_i \}, \subseteq \rangle \). From \( H\text{Colim} = DB(T, \text{Colim}F) \cong S \) we conclude that there is an unique arrow in Set from \( H\text{Colim}F \) to \( E \). Thus, \( H\text{Colim} \) is a colimit in Set, i.e., \( H = DB(T, \cdot) \) preserves directed colimits and, consequently, \( T \) is a finitely presentable.

Remark: We emphasize the fact that \( T \) is fp object for a more general considerations of the theory of enriched categories, which will be elaborated in Section , as demonstration that the monad based on the power-view endofunctor \( T : DB \to DB \) is an enriched monad. The Kelly-Power theory applies in the case of a symmetric monoidal closed category, which is lf and closed category (which is equivalent to demanding that the underlying ordinary category is lf, and that the monoidal structure on this ordinary category restricts to one on its fp objects. For details see \([12, 13]\), but note in particular that the unit \( T \) must be finitely presentable.

A locally finitely presentable category \([14]\) is the category of models for an essentially algebraic theory, which allows operations whose domain is an equationally defined subset of some product of the previously defined domains (the canonical example is a composition in a category, which is defined only on composable, not arbitrary pairs of arrows). In fact, we deduce from the algebraic (complete and compact) lattice \langle \text{Ob}_{DB}, \subseteq \rangle that for any object \( A \) holds that \( A \cong TA = \oplus \{ TB \mid B \subseteq A \} = \oplus S \) (remember that \( \oplus \) is a generalization in DB of the union operation \( \bigcup \) for sets and \( X \oplus Y = TX \oplus TY \)), where the set \( S = \{ B \mid B \subseteq A \} \) is upward directed, i.e., for any two finite \( B_1, B_2 \subseteq A \) there is \( C = B_1 \bigcup B_2 \in S \) such that \( B_1 \preceq C, B_2 \preceq C \), with \( TC = B_1 \bigoplus B_2 \), i.e., any object in DB is generated from finite objects and this generated object is just a directed colimit of these fp objects.

An important consequence of this freedom is that we can express conditional equations in the logic for databases.

Other important result from the fact that DB is a complete and cocomplete lf category that it can be used as the category of models for essentially algebraic theory \([15, 16]\) as is a relational database theory. Thus it is a category of models for a finite limit sketch, where sketches are called graph-based logic \([17, 18]\), and it is well known that a relational database scheme can readily be viewed, with some inessential abstraction involved, as a sketch. By Liar’s theorem, a category DB is accessible \([19, 20]\), because

\[ \sup \{ B_i \mid B_i \in F \} = \sum_{B_i \in F} B_i. \]
it is sketchable.

**Remark:** differently from standard application of sketches used to define a theory of a single database scheme, so that objects of this graph-based logic theory are single relations of such a database and arrows between them are used to define the common database functional dependencies, inclusion dependencies and other database constraints, in the case of inter-database mappings we need to use the whole databases as objects in this lfp $DB$ category: the price for this higher level of abstraction is that arrows in $DB$ are much more complex than in standard setting and that generally are not functions.

5 Enrichment

It is not misleading, at least initially, to think of an enriched category as being a category in which the hom-sets carry some extra structure (partial order $\leq$ of algebraic sublattice $<\text{Ob}_{DB}, \preceq>$ in our case) and in which that structure is preserved by composition. The notion of enriched category [12] is more general and allows for the hom-objects ("hom-sets") of the enriched category to be objects of some monoidal category, traditional called $V$.

Let us now prove that $DB$ is a monoidal closed category: for any two objects $B$ and $C$ the set of all arrows $\{f_1, f_2, \ldots\} : B \rightarrow C$, from $B$ into $C$, can be represented by an unique arrow $\bigcup_{f_i \in DB(B,C)} f_i : B \rightarrow C$, so that the object $C^B$ is equal to the information flux of this arrow $\bigcup_{f_i \in DB(B,C)} f_i$. Thus, we define the hom-object $C^B \triangleq \bigoplus_{f \in DB(B,C)} f$ (merging of all closed objects obtained from a hom-set of arrows from $B$ to $C$), i.e., merging of compact elements $A \preceq B \otimes C$ (where $B \otimes C$ is the "distance" between $B$ and $C$, following Lawvere’s idea, as follows from the definition of metric space for $DB$ category in a Section 6.1) which "internalize" the hom-sets.

Thus we obtain that $C^B = T(\bigcup_{f \in DB(B,C)} f) = T(\bigcup_{f \in DB(B,C)} f \preceq B \otimes C)) = T(B \otimes C) = B \otimes C$.

Generally a monoid $M$ acting on set $\text{Ob}_{DB}$ may be seen as general metric space where for any $B \in \text{Ob}_{DB}$ the distance $C^B$ is a set of $v \in S$ (views on our case) whose action send $B$ to $C$ (gives a possibility to pass from the "state" $B$ to "state C of the database "system" of objects in $DB$).

A monoidal category is closed if the functor $\_ \otimes B : DB \rightarrow DB$, for every object $B$, has a right adjoint $(\_)^B : DB \rightarrow DB$ and the counit $\varepsilon_C : C^B \otimes B \rightarrow C$ called the evaluation at $C$ (denoted by $\text{eval}_{B,C}$).

**Proposition 13** Strictly symmetric idempotent monoidal category $(DB, \otimes, T)$ is a monoidal bi-closed: for every object $B$, there exists an isomorphism $A : DB(A \otimes B, C) \simeq DB(A, C^B)$ such that for any $f \in DB(A \otimes B, C)$, $A(f) \approx f$, the hom-object $C^B$ together with
a monomorphism $\text{eval}_{B,C} : C^B \otimes B \hookrightarrow C$ the following "exponent" diagram

\[
\begin{array}{ccc}
C^B \otimes B & \xrightarrow{\text{eval}_{B,C}} & C \\
A(f) & \xrightarrow{id_B} & \\
A \otimes B & \xrightarrow{f} & \\
\end{array}
\]

commutes, with $f = \text{eval}_{B,C} \circ (A(f) \otimes id_B)$.

**Proof:** From a definition of hom-object we have $C^B \triangleq T(\bigcup \{ \tilde{g} \mid \perp^0 \subseteq \tilde{g} \subseteq B \otimes C \}) = T(\bigcup \{ \tilde{g} \mid \perp^0 \subseteq \tilde{g} \subseteq B \otimes C \}) = B \otimes C$. Thus, from $\tilde{f} \subseteq A \otimes B \otimes C = TA \cap TB \cap TC$, and the fact that for a monomorphism $\text{eval}_{B,C} = T(C^B \otimes B) = T(B \otimes C) = B \otimes C$, we obtain for the commutativity of this exponential diagram that, $\tilde{f} = \text{eval}_{B,C} \circ (A(f) \otimes id_B)$

$= \text{eval}_{B,C} \cap (A(f) \otimes id_B) = \text{eval}_{B,C} \cap (A(f) \otimes id_B) = B \otimes C \cap A(f) \cap TB = TB \cap TC \cap A(f) = A(f)$, from the fact that $A(f) \subseteq C^B = TB \cap TC$.

Thus, $f = \text{eval}_{B,C} \circ (A(f) \otimes id_B)$ if $A(f) \approx f$.

A is a bijection, because $DB(A \otimes B, C) = \{ g \mid \perp^0 \subseteq \tilde{g} \subseteq A \otimes B \otimes C \} \cong \{ \tilde{g} \mid \tilde{g} \in K \}$, where $K$ is a bounded algebraic sublattice (of closed objects) of the lattice $C$, and $\cong$ denotes a bijection, i.e., $K = \{ a \mid a \in C \text{ and } a \subseteq TA \cap TB \cap TC \}$. Also $DB(A, C^B) = DB(A, B \otimes C) \cong K$, thus $|DB(A \otimes B, C)| = |DB(A, C^B)| = |K|$. Thus, $A$ is a bijection, such that for any $f \in DB(A \otimes B, C)$, $\tilde{f} = f \in K$, i.e., $A(f) \approx f$.

Consequently, $DB$ is closed and symmetric, that is, biclosed category.

\hfill $\square$

**Remark:** from duality we have that, for any two objects $A$ and $B$ that $|DB(A, B)| = |DB(B, A)|$, i.e., $A^B \equiv B^A \equiv A \otimes B$. That is, the cotensor (hom object) of any two objects $A^B$ which is a particular limit in $DB$ is equal to the correspondent colimit of these two objects, that is tensor product $A \otimes B$: this fact is based on the duality property of $DB$ category.

We have seen that all objects in $DB$ are finitely representable. Let us denote by $V = DB(T, -) : DB \rightarrow \text{Set}$ the representable functor $DB(T, -)$. By putting $A = T$ in $\alpha$, and by using the isomorphism $\beta : T \otimes A \cong B$, we get a natural isomorphism $DB(B, C) \cong V(C^B) = DB(T, C^B)$. Then $C^B$ is exhibited as a lifting through $V$ of the hom-set $DB(B, C)$, i.e., hom-object $C^B$ is a set of all views which gives a possibility to pass from a "state" $T$ to a "state" $C$. It is called the internal hom of $B$ and $C$.

By putting $B = T$ in $\alpha$ and by using the isomorphism $\gamma : A \otimes T \cong A$ we deduce a natural isomorphism $i : C \cong C^T$ (it is obvious by $C^T = C \otimes T \cong C$).

The fact that a monoidal structure is closed means that we have an internal Hom functor, $(\_)^{(\_)} : DB^{op} \times DB \rightarrow DB$, which 'internalizes' the external Hom functor, $\text{Hom} : DB^{op} \times DB \rightarrow \text{Set}$, such that for any two objects $A, B$, the hom-object $B^A = (\_)^{(\_)}(A, B)$, represents the hom-set $\text{Hom}(A, B)$ (the set of all morphisms from $A$ to $B$).

We have that $(A \oplus B) \otimes C = (TA \cup TB) \cap TC = (TA \cap TC) \cup (TB \cap TC) =$
The endofunctor \( T \) of the category \( DB \) is defined as "internalized" hom-objects (cotensors) of objects in \( DB \) and morphisms \( f \) between \( DB \) objects. The composition law \( AB \) is equal also for the endofunctor identity for any \( A \), \( B \), \( C \) objects in \( DB \).

Monoidal closed categories generalize Cartesian closed ones in that they also possess exponent objects \( B^A \) which "internalize" the hom-sets. One may then ask if there is a way to "internally" describe the behavior of functors on morphisms. That is, given a monoidal closed category \( C \) and a functor \( F : C \to C \), consider, say, \( f \in C(A, B) \) then \( F(f) \in C(F(A), F(B)) \). Since hom-object \( B^A \) and \( F(B)^{F(A)} \) represent hom-sets \( C(A, B) \) and \( C(F(A), F(B)) \) in \( C \), one may study the conditions under which \( F \) is "represented" by morphism in \( C(B^A, F(B)^{F(A)}) \), for each \( A \) and \( B \).

**Proposition 14** The endofunctor \( T : DB \to DB \) is closed.

\( DB \) is a \( V \)-category enriched over itself, with the composition law monomorphism \( m_{A,B,C} : C^B \otimes B^A \to C^A \) and identity element (epimorphism) \( j_A : Y \to A^A \) which "picks up" the identity in \( A^A \).

The monad \( (T, \eta, \mu) \) is an enriched monad, thus, \( DB \) is an object of \( V \)-cat, and endofunctor \( T : DB \to DB \) is an arrow of \( V \)-cat.

**Proof:** It is easy to verify that for each two objects (databases) \( A \) and \( B \) in \( DB \) there exists \( f_{AB} \in DB(B^A, (TB)^{TA}) \), called "an action of \( T \) on \( B^A \) such that for all \( g \in DB(A, B) \) is valid

\[
T_f \circ A(g) \circ \beta(A) = A(T_f \circ \beta(T_A)) : Y \to (TB)^{TA},
\]
where \( \beta : \otimes \to I \) is a left identity natural transformation of a monoid \( (DB, \otimes, T, \alpha, \beta, \gamma) \), thus \( \beta(A) = id_A \), \( \beta(T_A) = id_{T_A} \). In fact, we take \( f_{AB} = id_{B^A} \), and we obtain

\[
f_{AB} \circ A(g) \circ \beta(A) = id_{B^A} \circ A(g \circ id_A) = A(g) = \tilde{g} = T(\tilde{g}) = A(\tilde{T(\tilde{g})}) = A(T_f \circ id_{T_A}) = A(T_f \circ \beta(T_A)).
\]

Consequently, \( T \) is a closed endofunctor.

The composition law \( m_{A,B,C} \) may be equivalently represented by a natural transformation \( j : (B \otimes) \otimes (\otimes B) \to \otimes \), and an identity element \( j_A \) by natural transformation \( j : Y \to \otimes T \), where \( \triangle : DB \to DB \times DB \) is a diagonal functor, while \( Y : DB \to DB \) is a constant endofunctor, \( Y(A) \equiv Y \) for any \( A \) and \( Y(f) \equiv id_T \) for any arrow \( f \) in \( DB \). It is easy to verify that two coherent diagrams (associativity and unit axioms) commute, thus \( DB \) is enriched over itself \( V \)-category (as, for example, \( Set \) category).

\( T \) is a \( V \)-functor: for each pair of objects \( A, B \) there exists an identity map (see above) \( f_{AB} : B^A \to (TB)^{TA} \), subject to the compatibility with composition \( m \) and with the identities expressed by the commutativity law \( f_{AB} \circ m_{A,B,C} = m_{T_A B, T_B C} \circ (f_{AB} \otimes f_{AB}) \) and \( j_T \circ \tilde{f}_{AB} = \tilde{f}_{AB} \circ j_A \). It is easy to verify that also natural transformations \( \eta : I_{DB} \to T \), \( \mu : T^2 \to T \) satisfy the \( V \)-naturality condition. This \( V \)-natural transformation \( \eta \) and \( \mu \) are an \( Ob_{DB} \)-indexed family of components \( \delta_A : Y \to TA \) in \( DB \) (for \( \eta, \delta_B : Y \to (TA)^A, (TA)^A = TA \); while for \( \mu, \delta_A : Y \to TT^2A, (TA)^2 = TA \)). This map \( f_{AB} \) is equal also for the endofunctor identity \( I_{DB} \), and for the endofunctor \( T^2 \), because \( B^A = B \otimes A = (TB)^{TA} = (T^2B)^{T^2A} \).

\( \square \)

In fact, each monoidal closed category is itself a \( V \)-category: hom-sets from \( A \) to \( B \) are defined as "internalized" hom-objects (cotensors) \( B^A \). The composition is given by the...
image of the bijection $A : DB(D \otimes A, C) \simeq DB(D, C^A)$, where $D = C^B \otimes B^A$, of the arrow $\varepsilon_B \circ (id_{C^B} \otimes \varepsilon_A) \circ \alpha_{C^B,B^A,A}$, i.e., $m_{A,B,C} = A(\varepsilon_B \circ (id_{C^B} \otimes \varepsilon_A) \circ \alpha_{C^B,B^A,A}) = A(\alpha_{C^B,B^A,A} \circ (id_{C^B} \otimes \varepsilon_A) \circ \alpha_{C^B,B^A,A})$ (it is a monomorphism, in fact, $\bar{m}_{A,B,C} = A(\varepsilon_B \circ (id_{C^B} \otimes \varepsilon_A) \circ \alpha_{C^B,B^A,A}) = TA \cap TB \cap TC = T(CB \otimes B^A)$.

The identities are given by the image of the isomorphism $\beta_A : \Upsilon \otimes A \rightarrow A$, under the bijection $A : DB(T \otimes A, A) \simeq DB(\Upsilon, A^A)$, i.e., $j_A = A(\beta_A) : \Upsilon \rightarrow A^A$ ($j_A$ is an epimorphism because $\bar{j}_A = A(\beta_A) = \beta_A = TA = T(A^A)$).

Moreover, for a $V$-category $DB$ holds the following isomorphism (which extends the tensor-cotensor isomorphism $A$ of exponential diagram in Proposition 13) valid in all enriched Lawvere theories [16], $DB(D \otimes A, C) \simeq DB(D, C^A) \simeq DB(A, DB(D, C))$.

Finally, from the fact that $DB$ is a lfp category enriched over the lfp symmetric monoidal closed category with a tensor product $\otimes$ (matching operator for databases), and the fact that $T$ is a finitary enriched monad on $DB$, by Kelly-Power theorem we have that $DB$ admits a presentation by operations and equations, what was implicitly assumed in the definition of this power-view operator in [4, 6].

6 Topological properties

In this Section we will investigate some topological properties of database category $DB$. That is we will consider its metric, subobject classifier and topos properties.

We will show that $DB$ is a metric space, weak monoidal topos and some negative results as: it is not well-pointed, has no power objects and pullbacks does not preserve epics.

6.1 Database metric space

In a metric space $X$, we denote by $X(A, B)$ the non negative real quantity of X-distance from the point $A$ to the point $B$. In a database context, for any two given databases $A$ and $B$, their matching is inverse proportional to their distance: The maximal distance, $\infty$, between any two objects is equal to the minimal possible matching, i.e., $\infty$ is represented by the closed object $\bot^0$, while the minimal distance, $0$, we obtain for their maximal matching, i.e., when these two objects are isomorphic ($A \simeq B$).

Following this reasoning, we can define formally the concept of the database distance:

**Definition 2.** If $A$ and $B$ are any two objects in $DB$, then their distance, denoted by $d(A,B)$, is defined as follows:

$$d(A,B) = \begin{cases} \Upsilon, & \text{if } A \simeq B \\ A^B, & \text{otherwise} \end{cases}$$

The (binary) partial distance relation $\sqsubseteq$, on closed database objects, is defined as inverse of the set inclusion relation $\subseteq$.

Notice that each distance is a closed database object (such that $A = T(A)$): the minimal distance $\Upsilon$ (total object), the maximal distance $\bot^0$ (zero object), and hom-objects $B^A$ ($B^A = T(A) \cap T(B)$), intersection of two closed objects is a closed object also).
Thus, a database metric space \( DB_{met} \), where points are databases and their distances are \textit{closed} databases, is a subcategory of \( DB \), composed by only epimorphic arrows: each epimorphism \( f : A \rightarrow B \) (i.e., \( A \cong B \)) in \( DB_{sb} \), correspond to the distance relation \( A \preceq B \). Thus we can say that a database metric space is embedded in \( DB \) category, where distances are closed databases and distance relations are epimorphisms between closed databases.

Let us show that this definition of distance for databases satisfies the general metric space properties.

A categorical version of metric space under the name enriched category or \( V \)-category, is introduced by \([21, 22]\), where distances became hom-objects. In this paper the definition of database distance in the \( V \)-category is introduced by \([21, 22]\), where distances became hom-objects. In this paper the definition of database distance in the \( V \)-category \( DB \) (which is a strictly symmetric monoidal category \( (DB, \otimes, T) \)) is different, as we can see, for example, for every \( A \neq T \), \( d(A, A) = T \supset A^A \).

**Proposition 15** The transitivity law for distance relation \( \preceq \), and the triangle inequality \( d(A, B) \otimes d(B, C) \sqsubseteq d(A, C) \) for a database metric space are valid. Moreover,

- There exists strong connection between the database PO-relation \( \preceq' \) and the distance PO-relation \( \preceq' \)
  
  \[ A \preceq B \iff \forall (C \neq A)(d(A, C) \sqsubseteq d(B, C)), \text{ thus} \]
  
  \[ A \cong B \iff \forall C(d(A, C) = d(B, C)) \]

- The distances in \( DB \) are locally closed. That is, for each object \( A \) there exists the bijection \( \phi : \{ d(A, B) \mid B \neq A \} \simeq DB(A, A) \)

where \( DB(A, A) \) is the hom set of all endomorphisms of \( A \).

**Proof:** The transitivity of \( \sqsubseteq \) holds because it is inverse set inclusion relation. Let us show the triangle inequality:

1. case when \( A \cong C \), then \( d(B, C) = d(B, A) = d(A, B) \), thus \( d(A, B) \otimes d(B, C) = d(A, B) \sqsubseteq T = d(A, C) \).
2. case when \( A \ncong B \), then
  
  2.1 case \( A \ncong B \), then \( T \otimes d(B, C) = d(B, C) = d(A, C) \) (by \( A \ncong B \)), i.e., \( d(A, B) \otimes d(B, C) = d(A, C) \).
  
  2.2 case \( B \cong C \), (see 2.1).
  
  2.3 case \( A \ncong B \) and \( B \ncong C \), then \( d(A, B) \otimes d(B, C) = T(A) \cap T(B) \cap T(C) \sqsubseteq T(A) \cap T(C) = d(A, C) \), i.e., \( d(A, B) \otimes d(B, C) \sqsubseteq d(A, C) \).

\[ \Box \]

Notice that locally closed property means that for any distance \( d(A, B) \) from a database \( A \), we have a morphism \( \phi(d(A, B)) = f : A \rightarrow A \), such that \( d(A, B) = \bar{f} \).

From the definition of distance we have that for the infinite distance \( \perp^0 \) (which is the terminal and initial object in \( DB \) category; denominated infinite object also) we obtain: \( d(\perp^0, \perp^0) = T \) (zero distance is the total object in \( DB \) category), and for any other database \( A \ncong \perp^0 \), \( d(A, \perp^0) = \perp^0 \), the distance from \( A \) to the infinite object (database) is \textit{infinite}. Thus, the bottom element \( \perp^0 \) and the top element \( T \) in the database lattice are, for this database metric system, \textit{infinite and zero distances} (closed objects) respectively.
Let us make a comparison between this database metric space and the general metric space (Frechet axioms):

<table>
<thead>
<tr>
<th>Frechet axioms</th>
<th>DB metric space</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d(A, B) + d(B, C) \geq d(A, C))</td>
<td>(d(A, B) \otimes d(B, C) \cong d(A, C))</td>
</tr>
<tr>
<td>(0 \geq d(A, A)) if (d(A, B) = 0) then (A = B)</td>
<td>(\top = d(A, A)) if (d(A, B) = \top) then (A \simeq B)</td>
</tr>
<tr>
<td>(d(A, B) &lt; \infty)</td>
<td>(d(A, B) \subseteq \bot^0)</td>
</tr>
<tr>
<td>(d(A, B) = d(B, A))</td>
<td>(d(A, B) = d(B, A))</td>
</tr>
</tbody>
</table>

### 6.2 Subobject classifier

Every subset \(A \subseteq B\) in the category Set can be described by its characteristic function \(C_f : B \rightarrow \Omega\), such that \(C_f(x) = \text{True}\) if \(x \in A\), \(\text{False}\) otherwise, where \(\Omega = \{\text{True, False}\}\) is the set of truth values. In order to generalize this idea for any two database instances \(A \preceq B\) (i.e., a monomorphism \(f : A \hookrightarrow B\) in DB category, the subobject classifier \(\Omega\) or truth-value object in DB will now be defined.

**Proposition 16** Subobject classifier for DB is the object \(\Omega = \top\) with the arrow \(\text{true} : \bot^0 \rightarrow \Omega\) that satisfies the \(\Omega\)-axiom:

For each monomorphism \(\text{in}_A : A \hookrightarrow B\) there is one and only one characteristic arrow \(C_{\text{in}_A} : B \rightarrow \Omega\), where \(C_{\text{in}_A} = C_{\text{Tin}_A} \circ \text{is}_B\), with \(\text{is}_B : B \rightarrow TB\) an isomorphism and \(C_{\text{Tin}_A} : TB \rightarrow \Omega\) the characteristic arrow for \(\text{Tin}_A : TA \hookrightarrow TB\), \(C_{\text{Tin}_A} \triangleq \{\text{id}_\bot\} \cup \partial_{\text{h}(\text{Tin}_A)} \in TB - TA\ \{\text{q}_B\}\), such that the diagram

\[
\begin{array}{ccc}
A & \xrightarrow{\text{in}_A} & B \\
\downarrow t_A & & \downarrow C_{\text{in}_A} \\
\bot^0 & \xrightarrow{\text{true}} & \Omega = \top
\end{array}
\]

is a pullback square.

Thus, DB is a monoidal elementary topos.

**Proof:** Let us verify that this pullback square commutes. The arrow \(t_A : A \rightarrow \bot^0\) is a unique arrow from \(A\) to the terminal object \(\bot^0\), while the arrow \(\text{true} : \bot^0 \rightarrow \Omega\) (such that \(\partial_0(\text{true}) = \partial_1(\text{true}) = \bot\)) is a unique arrow from the initial object \(\bot^0\) to the subobject classifier \(\Omega = \top\), thus, \(\text{true} \circ t_A = \text{true} \cap t_A = \bot \cap \bot = \bot\). While, \(C_{\text{in}_A} \circ \text{in}_A = C_{\text{Tin}_A} \circ \text{is}_B \circ \text{in}_A\). Thus, \(C_{\text{in}_A} \circ \text{in}_A = \)
\[ C_{\text{in}_A} \circ \text{id}_B \circ \text{in}_A = \text{in}_A \cap \text{id}_B \cap \text{in}_A = (TB - TA) \cap TB \cap TA = (TB - TA) \cap TA = \bot, \] and, consequently, diagram commutes. Let us show that it is a pull-back. For any \( h : C \rightarrow B \) and \( t_C : C \rightarrow \bot \), such that \( C \circ h = \text{true} \circ t : C \) it must hold that \( h \cap (TB - TA) = \bot \) and \( h \subseteq TB \cap TC \), thus \( h \subseteq TB \) and, consequently, \( h \subseteq TA \). But, in that case, there exists \( k : C \rightarrow A \) such that \( k \subseteq h \) (in fact, \( k \subseteq TA \cap TC \subseteq TA \)). Let us show that \( k \) is unique. In fact, for any other \( k_1 : C \rightarrow A \) we have \( h = \text{in}_A \circ k_1 = \text{in}_A \cap k_1 = TA \cap k_1 = k_1 \) (because it holds that \( k_1 \subseteq TA \cap TC \)), so, \( k_1 = k \), i.e., \( k_1 = k \).

An elementary topos is a Cartesian Closed category with subobject classifier. The monoidal elementary topos is a Monoidal Closed category, finitely complete and cocomplete, with hom-object (“exponentiation”) and a subobject classifier: all properties which are satisfied by \( DB \) category.

\[ \square \]

### 6.3 Weak monoidal topos

The standard topos is a Cartesian Closed Category with subobject classifier, that is a finitely complete and cocomplete category with exponents and subobject classifier.

In the previous chapter we defined the database category \( DB \) as the weak monoidal topos, which differs from a standard topos by the fact that, instead of exponents (with cartesian product in the exponent diagrams), we have the hom-objects which satisfy the "exponent" diagrams where the cartesian product \( ' \times ' \) is replaced by the monoidal tensor product \( ' \otimes ' \). Let us now compare these two kinds of toposes.

In the weak monoidal topos \( DB \) the following standard topos properties that all monomorphisms and epimorphisms are regular:

\[
\text{ISOMORPHIC} \equiv \text{MONIC} + \text{EPIC} \\
\text{EQUALIZER} \equiv \text{MONIC}
\]

Recall that in every category is valid ‘ISOMORPHIC imply MONIC + EPIC’ and ‘EQUALIZER imply MONIC’ only.

**Proposition 17** If \( f : A \rightarrow B \) is a monic arrow then \( f \) is an equalizer of \( C_{\text{in}_A} : B \rightarrow \Omega \) and \( \text{true}_B = \text{true} \circ t_B : B \rightarrow \Omega \), where \( t_B : B \rightarrow \bot \) is a terminal arrow for \( B \).

**Proof:** Easy to verify.

\[ \square \]

**Proposition 18** \( \tilde{f} \) is the smallest subobject of \( B \) through which \( f : A \rightarrow B \) factors. That is, if \( f = l \circ h \) for any \( h : A \rightarrow C \) and monic \( l : C \rightarrow B \), then there is a unique \( k : \tilde{f} \rightarrow C \) making

\[
\begin{array}{ccc}
A & \xrightarrow{\tau f} & \tilde{f} \\
\downarrow h & & \downarrow k \\
C & \xrightarrow{l} & B \\
\end{array}
\]

\[
\tilde{f} \cap \tau f^{-1} = \bot
\]

\[ 24 \]
commute, and hence $\tilde{\tau}_f^{-1} \subseteq \tilde{\eta}$.

**Proof:** $\tilde{f} = \tilde{h} \cap \tilde{\eta}$, thus $\tilde{h} \supseteq \tilde{f}$ and $TC \supseteq \tilde{\eta} \supseteq \tilde{f}$. From $\tilde{f} \supseteq \tilde{k}$ and $\tilde{\tau}_f^{-1} = \tilde{f} = \tilde{k} \cap \tilde{\eta}$, i.e., $\tilde{k} \supseteq \tilde{f}$, we obtain that $\tilde{k} = \tilde{f}$, thus $k$ is the unique monomorphisms.

$\square$

**Proposition 19** Coproduct preserve pullbacks. If

\[
\begin{array}{ccc}
A & \xrightarrow{f} & D \\
g & & k \\
B & \xrightarrow{h} & E
\end{array}
\quad
\begin{array}{ccc}
A_1 & \xrightarrow{f_1} & D \\
g_1 & & k \\
B_1 & \xrightarrow{h_1} & E
\end{array}
\]

are pullbacks in the $DB$ category, than so is

\[
\begin{array}{ccc}
A + A_1 & \xrightarrow{[f, f_1]} & D \\
& \downarrow{g + g_1} & \downarrow{k} \\
B + B_1 & \xrightarrow{[h, h_1]} & E
\end{array}
\]

where $[f, f_1] = ep_D \circ (f + f_1)$, with the epimorphism $ep_D : D + D \twoheadrightarrow D$, such that $ep_D = \{i_r : D \rightarrow D | \partial_0(i_r) = \partial_1(i_r) = r \in D\}$.

**Proof:** Easy to verify.

$\square$

Let us now consider the topos properties which are not satisfied in $DB$ category.

**Proposition 20** The following topos properties in $DB$ category does not hold:

- Pullbacks does not preserve epics.
- $DB$ category has no power objects.
- $DB$ category is not well-pointed.

**Proof:** 1. Let

\[
\begin{array}{ccc}
D = \tilde{f} \cap \tilde{\eta} & \xrightarrow{h_A} & A \\
& \downarrow{h_B} & \downarrow{f} \\
B & \xrightarrow{g} & C
\end{array}
\]

be a pullback square with epimorphism $f : A \rightarrow C$, (i.e., $\tilde{f} = TC$), then $D = \tilde{\eta} = \tilde{h_B}$. Thus, for any $B$ such that $TB \supseteq \tilde{g} = D$, $h_B \subset TB$, so $h_B$ is not an epimorphism.
2. By definition, the power object of \( A \) (if it exists) is an object \( P(A) \) which represents the contravariant functor \( \text{Sub}(\_ \times A) : DB \to \text{Set} \), where for any object \( B \), \( \text{Sub}(B) = \{ \tilde{f} | f \text{ is a subobject of } A \} = \{ \tilde{f} | \tilde{f} \subseteq TA \} \) is the set of all subobjects (monomorphic arrows with the target object \( B \)) of \( B \). Let us show that for any object \( A \not\cong \bot^0 \) there is no the power object \( P(A) \) such that in \( \text{Set} \) holds the bijection \( DB(B, P(A)) \cong \text{Sub}(\_ \times A) \).

In fact, \( \text{Sub}(B \times A) = \text{Sub}(B + A) = \{ \tilde{f} | \tilde{f} \subseteq T(A + B) = TA + TB \} = \text{Sub}(A) + \text{Sub}(B) \). So, \(|DB(B, P(A))| = \|\{ \tilde{f} | \tilde{f} \subseteq TB \cap T(P(A)) \}\| \subseteq \|\{ \tilde{f} | \tilde{f} \subseteq TB \}\| \subseteq \|\text{Sub}(B \times A)\| \).

3. The extensionality principle for arrows ”if \( f, g : A \to B \) is a pair of distinct parallel arrows, then there is an element \( x : \bot^0 \to A \) of \( A \) such that \( f \circ x \neq g \circ x \)” does not hold, because \( \widehat{f \circ x} = \widehat{g \circ x} = \bot^0 \) for the (unique) element (arrow) \( x : \bot^0 \to A \) such that \( \widehat{x} = \bot^0 \). \( \square \)

7 Conclusions

In previous work we defined a category \( DB \) where objects are databases and morphisms between them are extensional GLAV mappings between databases. We defined equivalent (categorically isomorphic) objects (database instances) from the behavioral point of view based on observations: each arrow (morphism) is composed by a number of ”queries” (view-maps), and each query may be seen as an observation over some database instance (object of \( DB \)). Thus, we characterized each object in \( DB \) (a database instance) by its behavior according to a given set of observations. In this way two databases \( A \) and \( B \) are equivalent (bisimilar) if they have the same set of its observable internal states, i.e. when \( TA \) is equal to \( TB \). It has been shown that such a \( DB \) category is equal to its dual, it is symmetric in the way that the semantics of each morphism is an closed object (database) and viceversa each database can be represented by its identity morphism, so that \( DB \) is a 2-category.

In this paper we considered some Universal algebra considerations and relationships of \( DB \) category and standard \( \text{Set} \) category. We introduced the categorial (functors) semantics for two basic database operations: matching and merging (and data federation), and we defined the algebraic database lattice. After that we have shown that \( DB \) is concrete, small and locally finitely presentable (lfp) category, and that \( DB \) is also monoidal symmetric \( V \)-category enriched over itself.

Based on these results we developed a metric space and a subobject classifier for this category, and we have shown that it is a weak monoidal topos.

Finally we have shown some negative results for \( DB \) category: it is not well-pointed, it has no power objects, and its pullbacks does not preserve epics.

These, and some other, results suggest the need for further investigation of categorial coalgebraic semantics for GLAV database mappings based on monads, and of general (co)algebraic and (co)induction properties for databases.
References