Entity-Relationship Models and Sketches

Michael Johnson*
School of Mathematics and Computing
Macquarie University

Robert Rosebrugh*
Department of Mathematics and Computer Science
Mount Allison University

R. J. Wood*
Department of Mathematics, Statistics
and Computing Science
Dalhousie University

Abstract

Entity-Relationship models are a common tool for specification and design of information systems. They use a graphical technique for displaying the objects of the system and relationships among them. The design process can be enhanced by specifying constraints of the system and the natural environment for these is the categorical notion of sketch. Here we argue that the finite-limit, finite-coproduct sketches with a terminal node are the appropriate class and call them ER sketches. A model for an ER sketch is a 'snapshot' of a database. The models of an ER sketch in a lextensive category are a category with determined values on the entities and additional properties in case the entities have key attributes. Among the models is the query language associated to the ER sketch. We also show that database updates are simply spans of models. Moreover, the equipment of spans of models of an ER sketch is the equipment of models of the sketch in the equipment of spans.

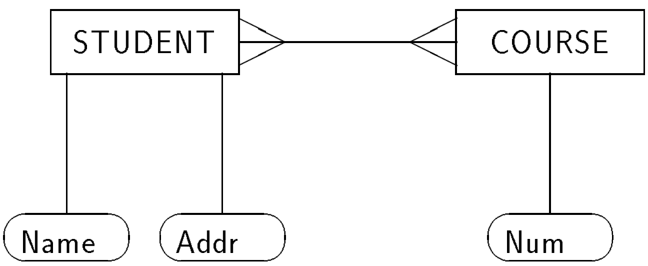
*Research partially supported by grants from the Australian ARC and NSERC Canada.
Diagrams typeset using Michael Barr's diagram macros

1 Introduction

In [9] information models based on entity-relationship (ER) diagrams are shown to be enhanced by the inclusion of commutative diagrams and specification of finite limits and coproducts to model constraints and queries. In this article we extend this work by defining a class of sketches called ER sketches that are suitable for description of ER models and their constraints. Models of an ER sketch in a lextensive category are states of databases for the sketch in the lextensive category. The theory of the sketch expresses its query language. Our main result is that the bicategory of updates of models of an ER sketch is the bicategory of spans of models and that the underlying equipment (see [4]) of this bicategory is itself an equipment of models of the same sketch.

Before proceeding we note that some other authors have also proposed sketches as data models. Diskin and Cadish [7] describe enhancements to entity-relationship models also using sketches. Their perspective differs from ours in defining 'diagram operations' for queries and in suggesting a generalization of sketches. Piessens [10, 11] also considers models of sketches, but allows the attribute domains to be infinite and gives less consideration to the constraints and query language. Motivated by the problem of view integration he obtains some very interesting results on algorithmic determination of equivalence of model categories for certain classes of sketches. Recently, Benson [3] has proposed viewing the dual of a Diers category (which is sketchable and viewed as a category of data) as the category of queries.

We begin with a brief discussion of our categorical view of ER models. For a fuller description see [8, 9]. An *entity* is a class of objects about which a database owner has information. For example a school might include STUDENT, COURSE, PROFESSOR, ... as entities in its information model. Entities have certain *attributes* or properties — for example students have a name, address, degree program and so on. Among the attributes of an entity there may be a specified *key attribute*. A *relationship* among entities is a (many-many) relation — for example students are enrolled in courses. Graphically, entities are represented by boxes, attributes by ovals, and relationships by lines joining the boxes (with "crows-feet" indicating a 'many-many' relationship) as in:



The information in such a diagram can be represented by a directed graph in the following way. The first step is to replace a line representing a relationship by a new entity and (in the case of a binary relationship) two directed lines (arrows) to the entities related. We thus intend to represent the relationships by *tabulating spans*. Note that this places entities and relationships on the same level (represented by boxes) with arrows among them. Hence we drop the distinction between entities and relationships. This point of view is also espoused by other writers on database theory, for example C. J. Date, [6].

Next we direct the lines joining entities and attributes towards the attributes. The attributes are meant to stand for fixed finite domains of values. We add a terminal object 1 and an arrow from it to an attribute for each element of the domain of values of the attribute. The graph which takes entities (including relationships!), attributes and 1 as nodes and has arrows as described above underlies the ER sketch. The set of arrows from 1 to an attribute will be a *finite coproduct specification* in the ER sketch.

Commutative diagrams arise as constraints in the model. An example which illustrates another use of finite coproducts might arise as follows. Enrollments in 1997 might be an entity ENROL97 included in an entity ENROL — a so-called *is_a* relation in ER theory — as part of a coproduct specification. Both ENROL97 and ENROL have (projection) arrows to STUDENT and the composite of the latter arrow and the inclusion must commute with the former arrow.

Finite limit diagrams may also arise as constraints. For example the arrow from an entity to a key attribute is necessarily monic. An example of a more complex constraint, taken from [9] follows. A database involving students, courses, classes and class-times may require no student to have a timetable conflict. We assume that courses may have several classes with corresponding time-slots. The pullback of the arrows from ENROL and CLASS into COURSE provides a time-table entity T-TABLE with an arrow to the product of STUDENT and TIME-SLOT. The non-conflict constraint requires this arrow to be monic (see Figure 1).

In Section 2 we begin by giving the general definition of a sketch and its models in a category. Motivated by considerations raised in our example we will require that our sketches allow finite limit cones and finite coproduct cocones in order to capture the properties of an ER scheme. This leads to the definition of ER sketch. As a special case we mention the 'keyed' ER sketches in which every entity has a specified key attribute. The models of an ER sketch in an appropriate (i. e. lextensive) category **S** are called database states in **S**. We give some properties of the category of database states. For the special case of keyed ER sketches we find that the database states have the property that transition morphisms are monic. Consequently the category of models then simplifies to a preorder, which is finite if **S** has finitely many subobjects of 1. Next, in Section 3 we describe the query language of an ER sketch and find that it is the theory of the sketch. We also consider *spans* of models of ER sketches. We interpret a span of models as an update of database states. In the special case

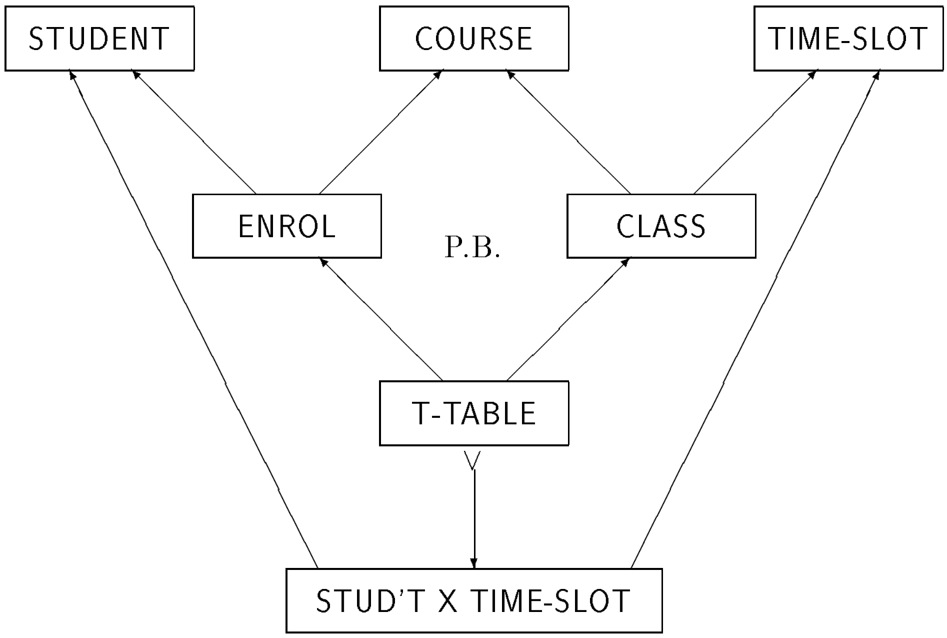


Figure 1: A finite limit constraint

of keyed ER sketches, where the arrows in a span are necessarily monic, this interpretation coincides exactly with our intuition about updates. Since the category of database states has pullbacks we get a bicategory with database states as objects and with spans as arrows which expresses database dynamics. However, by considering the *equipment*, in the sense of [4], which underlies this bicategory we are able to say more. In fact, *the equipment of spans of models of an ER sketch is the equipment of models of the sketch in the equipment of all spans.*

Section 4 outlines the basics of the theory of equipments from [4] and we apply that in Section 5 to obtain the italicised result above, after developing just what we need of the idea of modelling a sketch in an object in a 2-category. In essence, the result says that we can model the dynamics of a database using the same sketch as for the statics.

Some similar results for a categorical model of relational databases can be found in [12].

2 ER Sketches

To establish our notation we briefly describe the language of sketches, their theories and their models. For fuller treatment we refer the reader to [1] or [2].

Let $G = (V, E)$ be a directed graph and J a graph morphism from a directed graph D to G. A *cone* $C = (v, \lambda)$ to J consists of a node v of G (the *vertex* of C) and, for each node d in D, an edge λ_d in G from v to $J(d)$. *Cocones* are dual—the edges λ_d are directed to the vertex.

Definition 1 *A sketch $I\!\!E$ is a 4-tuple $I\!\!E = (G, \mathbf{D}, \mathcal{L}, \mathcal{C})$ where G is a directed graph, $\mathbf{D}$ is a set of pairs of paths in G with common source and target (the commutative diagrams) and $\mathcal{L}$ (respectively $\mathcal{C}$) is a set of cones (resp. cocones) in G.*

Recall that a category has an underlying sketch.

Definition 2 *A* model *M of $I\!\!E$ in a category $\mathbf{S}$ is a morphism of sketches from $I\!\!E$ to the underlying sketch of $\mathbf{S}$, or in elementary terms, an assignment of vertices and edges of G to objects and arrows of $\mathbf{S}$ so that the images of pairs of paths in $\mathbf{D}$ have equal composites in $\mathbf{S}$ and cones (resp. cocones) in $\mathcal{L}$ (resp. in $\mathcal{C}$) have images which are (co)limit (co)cones.*

For a sketch $I\!\!E$ we denote the category generated by G subject to the relations $\mathbf{D}$ by $\widetilde{I\!\!E}$, so we will also refer to vertices of G as objects and edges of G as arrows.

Clearly a model M of $I\!\!E$ in $\mathbf{S}$ extends to a functor $\widetilde{M} : \widetilde{I\!\!E} \longrightarrow \mathbf{S}$ which takes every (co)cone in $\mathcal{L}$ ($\mathcal{C}$) to a (co)limit (co)cone. If M and M' are models a *homomorphism* $\phi : M \longrightarrow M'$

is simply a natural transformation from $\widetilde{M}$ to $\widetilde{M'}$. Models and homomorphisms determine a category of models of $\mathbb{E}$ in $\mathbf{S}$ denoted by $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$.

We speak of (limit-class, colimit-class) sketches when $\mathcal{L}$ and $\mathcal{C}$ are required to contain (co)cones only from a specified (co)limit-class. For example, (finite-product, $\emptyset$) sketches correspond to (multi-sorted) algebraic theories.

Definition 3 *An* ER sketch $\mathbb{E}$ *is a (finite limit, finite coproduct) sketch such that the empty cone is a limit cone whose vertex is called* 1*. Arrows with domain* 1 *are called* elements*. Objects which are vertices of coproduct (discrete) cocones of elements are called* attributes*. An ER sketch is* keyed *if each object* E *has a specified monic arrow* $k_E : E \rightarrowtail A_E$ *to some attribute* A_E*.*

The fragment of a data model for a school outlined in the Introduction provides an example of an ER sketch. Indeed, our thesis is that sketches with the special properties of an ER sketch have precisely the expressive power needed to describe entity-relationship-attribute data models with constraints.

The appropriate receiving categories for models of ER sketches should have finite limits and finite coproducts which will be assumed to be disjoint and universal. In fact such categories have already been studied [5]. As examples of distributive categories, they have many applications in theoretical Computer Science [13]. For the record,

Definition 4 *A category* $\mathbf{S}$ *is* lextensive *iff it has finite limits and finite coproducts which are disjoint and universal.*

For the balance of this article, *let* $\mathbf{S}$ *be a lextensive category.* We note that the definition of lextensive in [5] differs from that above, though the definition there is, as they point out, equivalent to Definition 4.

Remark 5 Notice that the definition of ER-sketch does not rule out the possibility of an *inconsistent sketch*, i. e. a sketch with no non-trivial models. For example, the sketch might include attributes A_2 and A_3 with 2 and 3 elements respectively and then specify that their sum is 1 which implies that the sketch has no models in $\mathbf{S}$ unless $\mathbf{S}$ is degenerate, i. e. $0 \cong 1$ in $\mathbf{S}$.

Definition 6 *A* database state D *in* $\mathbf{S}$ *for an ER sketch* $\mathbb{E}$ *is a model of* $\mathbb{E}$ *in* $\mathbf{S}$*. The* category of database states of $\mathbb{E}$ in $\mathbf{S}$ *is the category of models* $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ *of* $\mathbb{E}$ *in* $\mathbf{S}$*.*

An example of a database state in the lextensive category of finite sets $\mathbf{set}_0$ for the ER sketch of our school data model is provided by any database which models the specifications described *and* the constraints implied by the commutative diagrams, the limit cones and the coproduct cocones. The next proposition gives a property of the category of models of an ER sketch which is crucial to our discussion of database updates and we provide a proof. In fact the model categories have all connected finite limits.

Proposition 7 *Let $\mathbb{E}$ be an ER sketch. The category* $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ *has pullbacks and these are computed 'pointwise'. If* $\mathbf{S}$ *is a Grothendieck topos* $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ *has filtered colimits.*

Proof. For brevity denote $\mathrm{Mod}(\mathbb{E}, \mathbf{S}) by \mathcal{M}$. Suppose that $D_1 \xrightarrow{\alpha} D_0 \xleftarrow{\alpha'} D_2$ is a cospan in $\mathcal{M}$. Since $\mathbf{S}$ has pullbacks, for any object E of $\mathbb{E}$ we can define $D(E)$ to be the pullback of $D(\alpha_E)$ and $D(\alpha'_E)$, i. e. the following square is a pullback.

$$\begin{tikzcd}
D(E) \arrow[r] \arrow[d] & D_2(E) \arrow[d, "\alpha'_E"] \\
D_1(E) \arrow[r, "\alpha_E"'] & D_0(E)
\end{tikzcd}$$

By the universality of pullbacks we can extend D to arrows of $\mathbb{E}$ so that the commutative diagrams of $\mathbb{E}$ are respected. Moreover, since it is defined by pullbacks, it is immediate that D preserves the finite limit cones of $\mathbb{E}$ since these commute with the pullbacks defining D.

To complete the proof we need the observation that D preserves the finite coproduct cones of $\mathbb{E}$. However since the finite coproducts in $\mathbf{S}$ are disjoint and universal this follows since pullbacks commute with such coproducts.

If $\mathbf{S}$ is a Grothendieck topos, filtered colimits in $\mathcal{M}$ are constructed similarly using the commutativity of finite limits with filtered colimits in $\mathbf{S}$. ∎

The values of a database state at an attribute are determined up to a canonical isomorphism:

Proposition 8 *Let D and D' be database states in* $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$. *For any attribute A with $\langle 1 \xrightarrow{a_i} A\rangle_{i\in I}$ a coproduct cocone there is a canonical isomorphism $i_A : D(A) \xrightarrow{\cong} D'(A)$. Moreover, for any morphism of database states $\alpha : D \longrightarrow D'$ we have $i_A = \alpha_A : D(A) \xrightarrow{\cong} D'(A)$.*

Proof. This is simply the observation that $D(A)$ is isomorphic to $D'(A)$ since D and D' both preserve the cocone defining A. Indeed, $D(1) \cong D'(1)$ are terminal objects of $\mathbf{S}$, and hence i_A is determined by the family $\langle D(1) \cong D'(1) \xrightarrow{D(a_i)} D'(A)\rangle$, and similarly for i_A^{-1}. We have that $i_A = \alpha_A$ since naturality of α requires that all of the squares:

$$\begin{tikzcd}
D(1) \arrow[r, tail, "D(a_i)"] \arrow[d, "\cong"'] & D(A) \arrow[d, "\alpha_A"] \\
D'(1) \arrow[r, tail, "D'(a_i)"'] & D'(A)
\end{tikzcd}$$

must commute. ∎

Proposition 9 *If $\alpha : D \longrightarrow D'$ is a morphism of database states for a keyed ER sketch $\mathbb{E}$ and E is an entity of $\mathbb{E}$ then $\alpha_E : D(E) \rightarrowtail D'(E)$ is monic and determined by a key for E.*

Proof. Suppose that the monic specification $k_E : E \rightarrowtail A_E$ in $\mathbb{E}$ is a key for E where A is an attribute. Now consider the commutative square:

$$\begin{tikzcd}
D(E) \arrow[r, tail, "D(k_E)"] \arrow[d, "\alpha_E"'] & D(A_E) \arrow[d, "\alpha_{A_E}"] \\
D'(E) \arrow[r, tail, "D'(k_E)"'] & D'(A_E)
\end{tikzcd}$$

By the previous proposition, we have that $\alpha_{A_E} : D(A_E) \xrightarrow{\cong} D'(A_E)$ is an isomorphism determined by the cocone structure of A. The horizontal arrows are monic since D and D' are models, so α_E is monic as claimed. ∎

Proposition 10 *Let $\mathbb{E}$ be a keyed ER sketch and $\mathbf{S}$ a lextensive category. The model category $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ is a preorder which has meets of bounded pairs of models. If $\mathrm{sub}(1)$ in $\mathbf{S}$ is finite then $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ is finite.*

Proof. By Propositions 8 and 9, all components α_E of any $\alpha : D \longrightarrow D'$ are monic so all α are monic. To see that $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ is a preorder suppose that $\alpha, \alpha' : D \longrightarrow D'$ are model homomorphisms and E is an entity. Since $\mathbb{E}$ is keyed there is a monic specification $k_E : E \longrightarrow A$ so $D'(k_E)$ is monic. By Proposition 8 we have $\alpha_A = i_A = \alpha'_A$. Using this and naturality we get

$$D'(k_E)\alpha_E = \alpha_A D(k_E) = \alpha'_A D(k_E) = D'(k_E)\alpha'_E$$

so $\alpha_E = \alpha'_E$ since $D'(k_E)$ is monic. Thus $\alpha = \alpha'$ on all components so they are equal and hence $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ is a preorder.

Now $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ has meets of bounded pairs since it has pullbacks by Proposition 7.

For the last statement of the proposition, note that any $D(A)$ has finitely many subobjects since coproducts in $\mathbf{S}$ are disjoint and universal and so there are only finitely many possible values for any $D(E)$ whence the result. ∎

3 Queries and updates

We now turn to the query language for an ER sketch. We first note that ER sketches are *finite-sum* or *FS sketches* in the sense of [2]. That is, they have only finite limit cones and finite coproduct cocones. In [2] Barr and Wells construct a lextensive category $T(\mathbb{E})$ called the *associated FS theory* from any FS sketch $\mathbb{E}$. The construction begins with a finite limit closure of $\mathbb{E}$ and uses a suitable subcategory of sheaves for a Grothendieck topology built using the finite coproduct cocones in $\mathbb{E}$. Note that $T(\mathbb{E})$ determines a sketch constructed by taking the graph, commutativities, finite limit cones and finite coproduct cocones of $T(\mathbb{E})$ itself. The result of this construction on an ER sketch will usually not be an ER sketch — $T(\mathbb{E})$ is usually not finite. However $T(\mathbb{E})$ should be thought of as the category of 'derived operations' of the sketch, and there is a model of $\mathbb{E}$ in $T(\mathbb{E})$ which is initial.

The results in Section 8.2 of [2] are as follows:

Theorem 11 *Let $\mathbb{E}$ be an FS sketch. Any model of $\mathbb{E}$ extends to a model of the associated FS theory $T(\mathbb{E})$. Moreover, $\mathbb{E}$ has a model in $T(\mathbb{E})$ which induces a bijection on models.*

The property of $T(\mathbb{E})$ is that the model J of $\mathbb{E}$ in $T(\mathbb{E})$ in the theorem mediates the bijection between models of $\mathbb{E}$ and models of $T(\mathbb{E})$: any model M of the sketch $T(\mathbb{E})$ can be composed with J to provide a model MJ of $\mathbb{E}$. Conversely, any model D of $\mathbb{E}$ *extends uniquely* to a model $\overline{D}$ of $T(\mathbb{E})$. Moreover, $\overline{MJ} = M$ and $\overline{D}J = D$.

Our interest in the lextensive category $T(\mathbb{E})$ for an ER sketch $\mathbb{E}$ is that it is precisely the *query language* for the ER sketch. Indeed, the theory $T(\mathbb{E})$ is exactly the *classifying category* for the ER information model as described in [9]. For example, the result of a simple query such as `select STUDENT where NAME = ˋJonesˊ` is the equalizer of the attribute defining arrow $\mathsf{STUDENT} \longrightarrow \mathsf{NAME}$ and the composite $\mathsf{STUDENT} \longrightarrow 1 \longrightarrow \mathsf{NAME}$. (The latter arrow just determines the constant `` `Jones´ ``.) As another example, equi-joins are simply pullbacks—as we saw in the Introduction in the context of expressing a constraint. Now $T(\mathbb{E})$ has objects for all such finite limits, and so it has an object for each of the 'select, project, join' queries which can be expressed using the entities and attributes of the ER sketch. Since $T(\mathbb{E})$ has disjoint universal sums it also includes expressions for queries involving sums. In short, $T(\mathbb{E})$ is a syntactic category whose objects are precisely queries on the original sketch.

Now since any database state D for $\mathbb{E}$ in $\mathbf{S}$ extends to a model $\overline{D}$ of $T(\mathbb{E})$ in $\mathbf{S}$, evaluating the extension $\overline{D}$ at any query Q, i. e. object of $T(\mathbb{E})$, gives precisely the result of evaluating the query Q on the database state D.

The remainder of this article is concerned with updating database states and we begin with a definition.

Definition 12 *Let D, D' be database states for an ER sketch $\mathbb{E}$. An* update *from D to D' is a span from D to D' in* $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$.

The motivation for this definition is as follows. An update of D should change it to a new state D'. In general, the value of the database state D at an entity E (which is not an attribute) can have data *added, deleted* or *changed.* The last operation, however, can be expressed by a deletion of data followed by the addition of new data. Because of this it can be expressed by a span of monos $D(E) \longleftarrow S(E) \longrightarrow D'(E)$ where $S(E)$ is $D(E)$ after deletions and the mono from $S(E)$ to $D'(E)$ expresses all necessary additions. Note that for an update from D to D' we require that S be a model i. e. a legal database state and also that a simple morphism $D(E) \longrightarrow D'(E)$ will simply not do the job. (There may be many models S which represent the same update, possibly including a maximal such S.) Thus in the case of a keyed ER sketch where all arrows in $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ are monic, our definition of update matches this motivation. In general the values of a database state are determined on the attributes, and we allow the more general notion of update envisaged above. Provided that $\mathbf{S}$ has pushouts and epi-mono factorizations we can always factor a span through a pair of monics anyway.

Proposition 13 *Using pullbacks to compose updates, we obtain a bicategory* $\mathrm{Upd}(\mathbb{E}, \mathbf{S})$ *with database states for $\mathbb{E}$ as objects and updates as arrows.* ∎

Remark 14 An alternative definition of an update as a cospan obtained by first adding new data and then deleting data gives nothing new since any cospan update can be replaced by an equivalent span update obtained by pullback.

4 Equipments

For $\mathbf{F}$ a category with pullbacks, we begin by writing $\mathbf{spnF}$ for the familiar bicategory whose objects are those of $\mathbf{F}$ and whose arrows (1-cells) are spans, with composition given by pullback. In particular then, our bicategory $\mathrm{Upd}(\mathbb{E}, \mathbf{S})$ is precisely $\mathbf{spn}(\mathrm{Mod}(\mathbb{E}, \mathbf{S}))$. For each such bicategory, we have the homomorphism of bicategories $\mathbf{F} \longrightarrow \mathbf{spnF}$ which is the identity on objects and sends each arrow to its graph. By analogy with the terminology in ring theory, let us say that this homomorphism allows us to see $\mathbf{spnF}$ as an $\mathbf{F}$-*algebra*. Such a point of view immediately suggests that there is perhaps a simpler structure, an $\mathbf{F}$-*module* of some kind, that underlies the $\mathbf{F}$-*algebra* structure.

This point of view was studied at length in [4]. In this section we recall just enough of [4] to make an observation about $\mathbf{spn}(\mathrm{Mod}(\mathbb{E}, \mathbf{S}))$ that enhances our view of database dynamics. The basic idea is to forget about arbitrary composites of arrows in bicategories such as $\mathbf{spnF}$, remembering just those given by composing an arbitrary arrow, on either side, with an arrow that comes from $\mathbf{F}$. It is suggestive to refer to a general arrow as a *vector* arrow and to refer to an arrow of $\mathbf{F}$ as a *scalar* arrow. In this terminology, the analogy of the paragraph above is reinforced by saying that the scalar arrows *act* on the vector arrows but the vector arrows do not necessarily admit a composition operation.

The structure to which we allude above and which is defined below is called an *equipment*. When the relevant arrows between equipments are defined and further examined it becomes clear that there is merely a 2-category of equipments — whereas if we do the same thing for bicategories then we get a tricategory of all bicategories. The simplification that results is necessary for the statement of our results in the next section. We point out, but do not develop the idea here, that for many equipments that underlie bicategories, the composition of vectors is in fact an operation that is derivable within the context of equipments alone.

An *equipment* is a 4-sorted structure having *objects* $\cdots, X, A, \cdots$, *scalar arrows* $f : X \longrightarrow A$, *vector arrows* $\mu : X \longrightarrow\!\!\!\!\!\!\!+\;\, A$ and *vector transformations* $\Phi : \mu \longrightarrow \nu : X \longrightarrow\!\!\!\!\!\!\!+\;\, A$. The objects and scalar arrows carry the structure of a category; for each pair of objects X, A, the vector arrows from X to A and the vector transformations between these carry the structure of a category; finally, there are actions of the scalars on the vectors, as suggested by

$$\begin{tikzcd} Y \arrow[r, "x"] & X \arrow[r, bend left, "\mu" slashed] \arrow[r, bend right, "\nu"' slashed] \arrow[r, phantom, "\Downarrow\Phi"] & A \end{tikzcd} \quad = \quad \begin{tikzcd} Y \arrow[r, bend left, "\mu x" slashed] \arrow[r, bend right, "\nu x"' slashed] \arrow[r, phantom, "\Downarrow\Phi x"] & A \end{tikzcd}$$

and

$$\begin{tikzcd} X \arrow[r, bend left, "\mu" slashed] \arrow[r, bend right, "\nu"' slashed] \arrow[r, phantom, "\Downarrow\Phi"] & A \arrow[r, "a"] & B \end{tikzcd} \quad = \quad \begin{tikzcd} X \arrow[r, bend left, "a\mu" slashed] \arrow[r, bend right, "a\nu"' slashed] \arrow[r, phantom, "\Downarrow a\Phi"] & B \end{tikzcd}$$

which are strictly unitary and coherently associative in the three possible senses. To see **spnF** as such a structure, begin by taking the scalar category to be **F** itself and, for each pair of objects X, A in **F**, taking the category of vectors from X to A to be the usual category $\mathbf{spnF}(X, A)$ of spans from X to A; a typical object of which we denote by $y : X \longleftarrow S \longrightarrow A : b$. For $x : Y \longrightarrow X$ in **F**, $(y, S, b)x$ is the span $p : Y \longleftarrow P \longrightarrow A : bq$ where

$$\begin{tikzcd} P \arrow[r, "q"] \arrow[d, "p"'] & S \arrow[d, "y"] \\ Y \arrow[r, "x"'] & X \end{tikzcd}$$

is a pullback; while for $a : A \longrightarrow B$, $a(y, S, b) = (y, S, ab)$. An equipment may be *starred* which is to say that for each scalar z the action $(-)z$ has a left adjoint, denoted by $(-)z^*$ and for each scalar c the action $c(-)$ has a right adjoint, denoted by $c^*(-)$; satisfying conditions of the Beck-Chevalley type to ensure that the $(-)z^*$ associate with the $b(-)$ and the $c^*(-)$ (which then imply that the $c^*(-)$ associate with the $(-)x$). The equipments of the form **spnF**, for **F** with pullbacks are starred. For $Z \longleftarrow X : z$ in **F**, $(y, S, b)z^*$ is given by composition and for $A \longleftarrow C : c$ in **F**, $c^*(y, S, b)$ is given by pullback. An equipment may carry the further structure of a *pointing*, which provides, for each object X, a distinguished vector arrow $\iota_X : X \longrightarrow\!\!\!\!\!+ \; X$ and isomorphisms $f\iota_X \xrightarrow{\cong} \iota_A f$, for each scalar $f : X \longrightarrow A$; satisfying $(1\iota_X \xrightarrow{\cong} \iota_X 1) = 1_{\iota_X}$ and the familiar hexagonal coherence condition. In the case of **spnF**, such "identity vectors" are provided by the spans $1 : X \longleftarrow X \longrightarrow X : 1$. Finally, a *starred pointed equipment* is a starred equipment together with a pointing for which the mates $\iota_X f^* \longrightarrow f^*\iota_A$ of the $f\iota_X \xrightarrow{\cong} \iota_A f$ are again isomorphisms. In fact, **spnF** is a starred pointed equipment.

For an equipment $\mathcal{A}$, it is convenient to write $\mathcal{A}_\#$ for the category of scalars of $\mathcal{A}$ and, for each pair of objects A, C in $\mathcal{A}$, $\mathcal{A}(A, C)$ for the category of vector arrows from A to C. For a further equipment $\mathcal{B}$, an *equipment arrow* $T : \mathcal{A} \longrightarrow \mathcal{B}$ consists of a functor $T_\# : \mathcal{A}_\# \longrightarrow \mathcal{B}_\#$, abbreviated to T whenever possible, functors $T_{A,C} : \mathcal{A}(A, C) \longrightarrow \mathcal{B}(TA, TC)$ and *action comparisons* $Tc.T\mu \longrightarrow T(c\mu)$, $T(\mu a) \longrightarrow T\mu.Ta$, which are natural in μ and satisfy coherence criteria (but are not required to be invertible). For equipment arrows $T, S : \mathcal{A} \longrightarrow \mathcal{B}$, an *equipment transformation* $u : T \longrightarrow S : \mathcal{A} \longrightarrow \mathcal{B}$ consists of a natural transformation $u_\# : T_\# \longrightarrow S_\# : \mathcal{A}_\# \longrightarrow \mathcal{B}_\#$ and, for each vector arrow $\mu : A \longrightarrow C$ in $\mathcal{A}$, a vector transformation u_μ as displayed below

$$\begin{tikzcd} TA \arrow[r, "u_A"] \arrow[d, "T\mu"', "\mid" marking] \arrow[dr, phantom, "\xrightarrow{u_\mu}"] & SA \arrow[d, "S\mu", "\mid" marking] \\ TC \arrow[r, "u_C"'] & SC \end{tikzcd} \qquad (*)$$

natural with respect to vector transformations $\Phi : \mu \longrightarrow \nu$ and compatible with the action comparisons for T and S. In [4] there is a very detailed account of how these definitions arise, of composites of equipment arrows and transformations and of the fact that equipments, their arrows and transformations constitute a 2-category called **EQT**. The precise account shows that no further dimension has been simply forgotten. The full sub-2-category of **EQT** determined by the starred equipments is called ***EQT**. For pointed equipments $\mathcal{A}$ and $\mathcal{B}$, an equipment arrow $T : \mathcal{A} \longrightarrow \mathcal{B}$ is *pointed* by the further data of isomorphisms $p : T\iota_A \xrightarrow{\cong} \iota_{TA}$, subject to coherence criteria. For pointed equipment arrows $(T, p), (S, q) : \mathcal{A} \longrightarrow \mathcal{B}$, a *pointed equipment transformation* $u : (T, p) \longrightarrow (S, q) : \mathcal{A} \longrightarrow \mathcal{B}$ is an equipment transformation $u : T \longrightarrow S : \mathcal{A} \longrightarrow \mathcal{B}$ that is compatible with p and q. These definitions lead to the 2-category of pointed equipments $\mathbf{EQT}_*$ and to the full sub-2-category thereof determined by the starred pointed equipments — which is denoted by $^*\mathbf{EQT}_*$. Writing **PBK** for the *full* sub-2-category of **CAT** determined by the categories which have pullbacks one can make the **spn** construction 2-functorial, arriving at $\mathbf{spn} : \mathbf{PBK} \longrightarrow {}^*\mathbf{EQT}_*$. Now in fact, for F an arrow in **PBK**, $\mathbf{spn}F$ provides an example of an equipment arrow T with the action comparisons $Tc.T\mu \longrightarrow T(c\mu)$ invertible. Moreover, if F preserves pullbacks then also the $T(\mu a) \longrightarrow T\mu.Ta$ are invertible. We write $\mathbf{PBK_{pbk}}$ for the locally-full sub-2-category of **PBK** determined by the pullback preserving functors. An equipment arrow T with all action comparisons invertible is called a *homomorphism* and we write $^*\mathbf{EQT_{*hom}}$ for the locally-full sub-2-category of $^*\mathbf{EQT}_*$ determined by the homomorphisms (and similarly for the other 2-categories of equipments). The **spn** construction restricts to provide a 2-functor $\mathbf{spn} : \mathbf{PBK_{pbk}} \longrightarrow {}^*\mathbf{EQT_{*hom}}$. All the 2-categories of equipments mentioned above admit **CAT**-*powers* (being what many authors call cotensor products). The 2-functor **spn** preserves powers.

5 Updates are models

Let us reconsider the construction of the category $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$. We will find it convenient here to let $\mathbb{E}$ stand also for $\widetilde{\mathbb{E}}$, the quotient modulo the commutative diagrams of $\mathbb{E}$ of the category freely generated by the graph of $\mathbb{E}$. The category $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ is in the first instance a full sub-category of the power $\mathbf{S}^{\mathbb{E}}$. (The 2-category **CAT** is of course cartesian closed but the functor category $\mathbf{S}^{\mathbb{E}}$ can also be seen as the $\mathbb{E}$th power of $\mathbf{S}$.) How are the objects of $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ to be carved out of $\mathbf{S}^{\mathbb{E}}$? The familiar condition on such an object D is that certain arrows constructed from it in $\mathbf{S}$ be invertible. If the required invertibility conditions can be parameterized by a set C then the requirement for D to be in $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ can be reduced to a single invertibility in $\mathbf{S}^C$. These heuristics, which we will make precise below, suggest — and this seems to be a new observation — that $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ is given in **CAT** by an inverter diagram of the form

$$\begin{tikzcd} \mathrm{Mod}(\mathbb{E}, \mathbf{S}) \arrow[r] & \mathbf{S}^{\mathbb{E}} \arrow[r, shift left=2, ""{name=U, below}] \arrow[r, shift right=2, ""{name=D, above}] & \mathbf{S}^C \arrow[from=U, to=D, Rightarrow] \end{tikzcd}$$

¿From the details which follow it will become apparent that we can consistently *define* $\mathrm{Mod}(\mathbb{E}, \mathcal{A})$, for $\mathbb{E}$ a sketch in the usual sense and $\mathcal{A}$ a "suitable" object in any 2-category $\mathbf{K}$ that admits enough powers and inverters. Moreover, if both $\mathbf{K}$ and $\mathbf{L}$ admit these limits and $F : \mathbf{K} \longrightarrow \mathbf{L}$ preserves them, then F also preserves $\mathrm{Mod}(\mathbb{E}, -)$, in the sense that, for any "suitable" $\mathcal{A}$ in $\mathbf{K}$, the canonical arrow $F\mathrm{Mod}(\mathbb{E}, \mathcal{A}) \longrightarrow \mathrm{Mod}(\mathbb{E}, F\mathcal{A})$ in $\mathbf{L}$ is an isomorphism.

Let $\mathbb{E}$ be a sketch and consider an element of $\mathcal{L}$, the set of designated cones. It does no harm to consider such a cone to be a diagram of the following kind in **CAT**

$$\begin{tikzcd} \Pi \arrow[rr, "P"] \arrow[dr, "!"'] & \arrow[d, phantom, "\Uparrow\pi"] & \mathbb{E} \\ & \mathbf{1} \arrow[ur, "p"'] & \end{tikzcd}$$

where Π is the category generated (in the sense understood for $\mathbb{E}$) by the base of the cone and P is the (not necessarily full) inclusion. If for an object $\mathcal{A}$ in a 2-category $\mathbf{K}$ the necessary powers exist then we obtain the following diagram in $\mathbf{K}$

$$\begin{tikzcd} \mathcal{A}^{\mathbb{E}} \arrow[rr, "\mathcal{A}^P"] \arrow[dr, "\mathcal{A}^p"'] & \arrow[d, phantom, "\Uparrow\mathcal{A}^\pi"] & \mathcal{A}^{\Pi} \\ & \mathcal{A} \arrow[ur, "\mathcal{A}^!"'] & \end{tikzcd}$$

Now assume that the object $\mathcal{A}$ has Π-limits, meaning of course that $\mathcal{A}^! : \mathcal{A} \longrightarrow \mathcal{A}^{\Pi}$ has a right adjoint $\lim_{\leftarrow} : \mathcal{A}^{\Pi} \longrightarrow \mathcal{A}$ (in $\mathbf{K}$). In this case the transformation displayed above yields by adjointness a transformation $\tilde{\pi} : \mathcal{A}^p \longrightarrow \lim_{\leftarrow} \mathcal{A}^P : \mathcal{A}^{\mathbb{E}} \longrightarrow \mathcal{A}$. If $\mathbf{K}$ is $\mathbf{CAT}$ the inverter of this transformation is precisely the full subcategory of $\mathcal{A}^{\mathbb{E}}$ determined by those functors D that take the cone in question to a limit cone.

Similarly, for each element $\sigma : S \longrightarrow s! : \Sigma \longrightarrow \mathbb{E}$ in $\mathcal{C}$, the set of designated cocones, we construct, assuming that $\mathcal{A}$ has Σ-colimits (meaning that $\mathcal{A}^! : \mathcal{A} \longrightarrow \mathcal{A}^\Sigma$ has a left adjoint) a transformation $\tilde{\sigma} : \lim_{\rightarrow} \mathcal{A}^S \longrightarrow \mathcal{A}^s : \mathcal{A}^{\mathbb{E}} \longrightarrow \mathcal{A}$. Say that $\mathcal{A}$ in $\mathbf{K}$ is *suitable for modelling* $\mathbb{E}$ if $\mathcal{A}$ has Π-limits for all Π arising from $\mathcal{L}$ and Σ-colimits for all Σ arising from $\mathcal{C}$. In this case we define a transformation

$$\begin{tikzcd} \mathcal{A}^{\mathbb{E}} \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Downarrow \mathcal{A}_{\mathbb{E}}"] & \mathcal{A}^{\mathcal{L}+\mathcal{C}} \end{tikzcd}$$

whose π-component, for all π in $\mathcal{L}$ is $\tilde{\pi}$ and whose σ-component, for all σ in $\mathcal{C}$ is $\tilde{\sigma}$. If $\mathbf{K}$ has inverters, we *define* $\mathrm{Mod}(\mathbb{E}, \mathcal{A})$ to be the inverter of $\mathcal{A}_{\mathbb{E}}$. Since powers and inverters are defined by (2-)representability criteria we have immediately

Proposition 15 *There is in the category* $\mathbf{K}(\mathrm{Mod}(\mathbb{E}, \mathcal{A}), \mathcal{A})$ *a model of* $\mathbb{E}$ *with the property that for all* $\mathcal{B}$ *in* $\mathbf{K}$*, composition yields an isomorphism of categories*

$$\mathbf{K}(\mathcal{B}, \mathrm{Mod}(\mathbb{E}, \mathcal{A})) \xrightarrow{\simeq} \mathrm{Mod}(\mathbb{E}, \mathbf{K}(\mathcal{B}, \mathcal{A})).$$

∎

Remark 16 Interpreting the definition in $\mathbf{K}^{op}$, where powers are *multiples* in $\mathbf{K}$ and inverters are coinverters in $\mathbf{K}$, we obtain the dual notion which might be denoted by $\mathbb{E} \star \mathcal{A}$ and satisfies

$$\mathbf{K}(\mathbb{E} \star \mathcal{A}, \mathcal{B}) \xrightarrow{\simeq} \mathrm{Mod}(\mathbb{E}, \mathbf{K}(\mathcal{A}, \mathcal{B})).$$

It must be noted though that "suitability" is in this case defined in terms of multiples and tends to make $\mathcal{A}$ trivial in $\mathbf{CAT}$. Still, there are naturally occurring 2-categories in which both $\mathrm{Mod}(\mathbb{E}, \mathcal{A})$ and $\mathbb{E} \star \mathcal{A}$ are interesting.

Any 2-functor $F : \mathbf{K} \longrightarrow \mathbf{L}$ takes adjunctions to adjunctions, so if F preserves the necessary powers and $\mathcal{A}$ is suitable for modelling $\mathbb{E}$ in $\mathbf{K}$, it follows that $F\mathcal{A}$ is suitable for modelling $\mathbb{E}$ in $\mathbf{L}$.

Proposition 17 *If* $F : \mathbf{K} \longrightarrow \mathbf{L}$ *is a 2-functor that preserves powers and inverters then it preserves modelling of* $\mathbb{E}$ *in the sense that the evident arrow*

$$F\mathrm{Mod}(\mathbb{E}, \mathcal{A}) \longrightarrow \mathrm{Mod}(\mathbb{E}, F\mathcal{A})$$

in $\mathbf{L}$ *is an isomorphism.* ∎

The 2-category $\mathbf{PBK_{pbk}}$ admits inverters; they are constructed as in $\mathbf{CAT}$.

Proposition 18 *The 2-category* ${}^*\mathbf{EQT}_{*\mathbf{hom}}$ *admits inverters (as do also the 2-categories* $\mathbf{EQT_{hom}}$*,* ${}^*\mathbf{EQT_{hom}}$ *and* $\mathbf{EQT}_{*\mathbf{hom}}$*).*

Proof. This follows from general considerations in [4]. ∎

The reader may nevertheless find an explicit description of inverters in ${}^*\mathbf{EQT}_{*\mathbf{hom}}$ interesting, so consider a transformation therein

$$\begin{tikzcd} \mathcal{A} \arrow[r, bend left, "T"] \arrow[r, bend right, "S"'] \arrow[r, phantom, "\Downarrow u"] & \mathcal{B} \end{tikzcd}$$

and denote its inverter by $I : \mathcal{I} \longrightarrow \mathcal{A}$. The scalar category $\mathcal{I}_\#$ is simply the inverter of $u_\#$ which is the full subcategory of $\mathcal{A}_\#$ determined by those objects A in $\mathcal{A}$ for which $u_A : TA \longrightarrow SA$ is an isomorphism. The functor $I_\#$ is the inclusion. The category of vector arrows from A to C in $\mathcal{I}$ is $\mathcal{I}(A, C)$, given as the inverter of

$$\begin{tikzcd} & \mathcal{B}(TA, TC) \arrow[dr, "{\mathcal{B}(TA, u_C)}"] & \\ \mathcal{A}(A, C) \arrow[ur, "T_{A,C}"] \arrow[dr, "S_{A,C}"'] \arrow[rr, phantom, "\Downarrow u_-"] & & \mathcal{B}(TA, SC) \\ & \mathcal{B}(SA, SC) \arrow[ur, "{\mathcal{B}(u_A, SC)}"'] & \end{tikzcd}$$

where $\mathcal{B}(TA, u_C)$ is given by post-action of u_C, $\mathcal{B}(u_A, SC)$ is given by pre-action of u_A and the μ-component of u_- is precisely the vector transformation u_μ displayed in the square (*) of Section 4. Thus $\mathcal{I}(A, C)$ is the full subcategory of $\mathcal{A}(A, C)$ determined by those μ for which u_μ is an isomorphism. (Readers familiar with [4] may prefer to start with the observation that the Grothendieck category of $\mathcal{I}$ is the inverter of the Grothendieck transformation of u.) The scalar actions for $\mathcal{I}$ are inherited from those in $\mathcal{A}$ (because the action comparisons for T

and S are invertible) and these admit adjoints when those in $\mathcal{A}$ do. For pointed u and A an object of $\mathcal{I}$, it follows that u_{ι_A} is an isomorphism whereupon ι_A is in $\mathcal{I}(A,A)$ and provides the components around which one finds a pointing for $\mathcal{I}$, which is then starred pointed when $\mathcal{A}$ is so.

Proposition 19 *The 2-functor* $\mathbf{spn} : \mathbf{PBK_{pbk}} \longrightarrow {}^*\mathbf{EQT_{*hom}}$ *described in [4] preserves inverters.* ∎

A proof of Proposition 19 would require a complete set of definitions but we have enough at hand to indicate what is at issue. Given an inverter diagram in $\mathbf{PBK_{pbk}}$

$$\begin{tikzcd} \mathbf{D} \arrow[r, "H"] & \mathbf{S} \arrow[r, bend left, "G"{name=G}] \arrow[r, bend right, "F"'{name=F}] & \mathbf{F} \arrow[from=G, to=F, Rightarrow, "t"] \end{tikzcd}$$

we must show that the induced arrow $K : \mathbf{spnD} \longrightarrow \mathcal{I}$ in ${}^*\mathbf{EQT_{*hom}}$, where $\mathcal{I}$ is the inverter of $\mathbf{spn}t$, is invertible. From the discussion after Proposition 18 it is clear that $K_\#$ is invertible and that the vector arrows in $\mathcal{I}$ are certain spans, with mediating object in $\mathbf{S}$. However, given a vector arrow $a : A \longleftarrow S \longrightarrow C : c$ in $\mathbf{spnS}$, the square for $(\mathbf{spn}t)_{(a,S,c)}$, as in the diagram (*) of Section 4, takes here the form of a commutative diagram

$$\begin{tikzcd} GA \arrow[r, "t_A"] & FA \\ GS \arrow[u, "Ga"] \arrow[r, "t_S"] \arrow[d, "Gc"'] & FS \arrow[u, "Fa"'] \arrow[d, "Fc"] \\ GC \arrow[r, "t_C"'] & FC \end{tikzcd}$$

Said otherwise, $(\mathbf{spn}t)_{(a,S,c)} = t_S$ and hence is invertible precisely if S is in $\mathbf{D}$; so a vector arrow in $\mathcal{I}$ is a span in $\mathbf{D}$.

Corollary 20 *For any finite limit, finite-sum sketch* $\mathbb{E}$ *and lextensive category* $\mathbf{S}$*, the 2-functor* $\mathbf{spn} : \mathbf{PBK_{pbk}} \longrightarrow {}^*\mathbf{EQT_{*hom}}$ *preserves modelling of* $\mathbb{E}$ *in the sense that the evident arrow*

$$\mathbf{spn}(\mathrm{Mod}(\mathbb{E}, \mathbf{S})) \longrightarrow \mathrm{Mod}(\mathbb{E}, \mathbf{spnS})$$

is an isomorphism of starred pointed equipments.

Proof. The preservation property is immediate from the discussion above as soon as $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ is recognized to be an inverter of powers *in* $\mathbf{PBK_{pbk}}$. Inspection of the construction before Proposition 15 shows that only pullback preservation of *left* adjoints of the $\mathbf{S}^! : \mathbf{S} \longrightarrow \mathbf{S}^\Sigma$ is an issue — dealt with by the assumption that such left adjoints are to provide finite sums in the lextensive category $\mathbf{S}$. ∎

The purely syntactic concept of ER sketch seems ideally suited to describe any aspects that might be contemplated in the design of a particular database. As we saw in Section 2, modelling such a sketch classically in a suitable semantic *category*, a lextensive category such as $\mathbf{set}_0$, produces a category of models whose objects are the possible states of the resulting database. Our investigation of database dynamics in Section 3 and Corollary 20 show that modelling the *same* sketch in a semantic *equipment* of spans produces an equipment whose vector arrows are the possible updates of the database.

References

[1] M. Barr and C. Wells. *Category theory for computing science, second edition.* Prentice-Hall, 1995.

[2] M. Barr and C. Wells. *Toposes, Triples and Theories. Grundlehren Math. Wiss. 278*, Springer Verlag, 1985.

[3] David B. Benson. Stone duality between queries and data. 1996. preprint.

[4] Aurelio Carboni, G. M. Kelly, D. Verity, and R. J. Wood. A 2-categorical approach to change of base and geometric morphisms 2. *Theory and Applications of Categories*, 4:82-136, 1998.

[5] Aurelio Carboni, Steven Lack, and R. F. C. Walters. Introduction to extensive and distributive categories. *Journal of Pure and Applied Algebra*, 84:145–158, 1993.

[6] C. J. Date. *Introduction to Database Systems, Sixth Edition.* Addison-Wesley, 1995.

[7] Zinovy Diskin and Boris Cadish. Algebraic graph-based approach to management of multidatabase systems. In *Proceedings of The Second International Workshop on Next Generation Information Technologies and Systems (NGITS '95)*, 1995.

[8] C. N. G. Dampney, Michael Johnson, and G. P. Monro. An illustrated mathematical foundation for era. In *The unified computation laboratory*, pages 77–84, Oxford University Press, 1992.

[9] Michael Johnson and C. N. G. Dampney. On the value of commutative diagrams in information modelling. In *Springer Workshops in Computing*, Springer-Verlag, 1994.

[10] Frank Piessens. *Semantic data specifications: an analysis based on a categorical formulation.* PhD thesis, Katholieke Universiteit Leuven, 1996.

[11] Frank Piessens and Eric Steegmans. Categorical data specifications. *Theory and Applications of Categories*, 1:156–173, 1995.

[12] Robert Rosebrugh and R. J. Wood. Relational databases and indexed categories. In *Proceedings of the Int'l Category Theory Meeting 1991, CMS Conference Proceedings, 13*, pages 391–407, American Mathematical Society, 1992.

[13] R. F. C. Walters. *Categories and Computer Science.* Cambridge University Press, 1991.