ENTITY-RELATIONSHIP-ATTRIBUTE MODELS AND SKETCHES

MICHAEL JOHNSON, ROBERT ROSEBRUGH AND R.J. WOOD

ABSTRACT. Entity-Relationship-Attribute ideas are commonly used to specify and design 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-sum sketches with a terminal node are the appropriate class and call them EA sketches. A model for an EA sketch in a lextensive category is a 'snapshot' of a database with values in that category. The category of models of an EA sketch is an object of models of the sketch in a 2-category of lextensive categories. Moreover, modelling the *same* sketch in certain objects in other 2-categories defines both the query language for the database and the updates (the dynamics) for the database.

1. Introduction

It should be said at the outset that there is a conflict between the terminology used in the study of databases and that used in category theory. Since this paper is an application of the latter to the former we find ourselves with several words that must be sorted out. Most troubling of these is 'model'. Because the primary audience for this paper is the category theory community we will use categorical terminology. In particular we will use 'model' as it is already understood in category theory and in Section 2 we will present a categorical generalization of the idea.

Thus, other than in this paragraph, we will avoid speaking of 'information models' and 'entity-relationship-attribute models'. In [10] information models based on entity-relationship-attribute (ERA) diagrams were shown to be enhanced by the inclusion of commutative diagrams and specification of finite limits and finite sums to model constraints and queries. In this article we extend that work by defining a class of sketches called EA sketches that are suitable for description of ERA models and their constraints. As the categorical reader may already suspect, 'ERA models' are more nearly described as 'ERA theories' from a categorical perspective. Still, we will try to avoid adding such new terminology.

Early database writers used to exhort their readers to avoid confusing the formal 'database design' with particular instances or states of the database. For us the former is the sketch and the latter are its models. We will use 'database state' to mean a model of an EA sketch in a lextensive category. The interesting result in this article is that the categorical model idea, applied to a fixed EA sketch, can be used to capture not only the category of states or 'statics' but also *both* the category of queries on the database design

The authors gratefully acknowledge financial support from the Australian Research Council and the Canadian NSERC. Diagrams typeset using M. Barr's diagram macros.

1991 Mathematics Subject Classification: 18C30, 68P15.

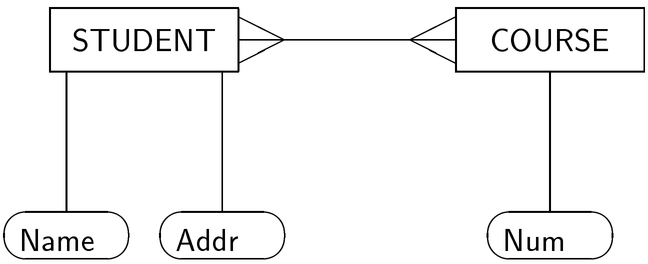
Key words and phrases: sketch, model, database, update.

and the updates (or 'dynamics') of database states.

Certain sketches, in particular our EA sketches, have what has been called in [2] an *associated theory* which is, in a sense, an enlargement of the sketch to include all possible derived operations and specifications. There are a number of special cases of such theories that may come to mind but an early observation of ours was that the theory expresses the query language of the given database design. It transpires though that the theory of an EA sketch can itself be seen as an *object of models* of the sketch in a special object in a special 2-category.

In order to understand updates of a database we were led to spans of monic arrows between database states, incorporating the idea that an update involves a deletion followed by an insertion. For an EA sketch $\mathbb{E}$ and a lextensive category $\mathcal{S}$, there are various ways that spans in the models of $\mathbb{E}$ in $\mathcal{S}$, $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$, can be regarded as a mathematical structure. Most common is the structure of a bicategory. However, if spans in $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ is regarded as an *equipment* in the sense of [4] it turns out to be an *object of models* of $\mathbb{E}$ in the starred, pointed equipment of spans in $\mathcal{S}$, in the 2-category of all such equipments.

We include here a brief discussion of our categorical view of the entity-relationship-attribute idea. For a fuller description see [6, 10]. 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 specification. Entities have certain *attributes* or properties — for example students have a name, address, degree programme 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 (e. g. in the case of a binary relationship, two) directed edges (arrows) to the entities related. Thus we represent the relationships by *tabulating spans.* This places entities and relationships on the same level (represented by boxes) with arrows among them. Then 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, [7]. It explains why we have chosen to speak of 'EA' sketches rather than 'ERA' sketches.

Next we replace the lines joining entities and attributes with edges directed towards the attributes. The attributes are meant to stand for fixed finite domains of values. To specify this we add a terminal object 1 and then an edge from the terminal object to an attribute for each element of the domain of values of the attribute. The resulting directed graph has entities (now including the former relationships!), the attributes and the terminal 1 as nodes. It has edges as described above and underlies the EA sketch.

The cocone of all edges (arrows) from 1 to an attribute will be a *finite sum specification* in the EA sketch.

Some of the constraints in data specifications may be modelled using *commutative diagrams*. An example which illustrates this follows. Suppose there are entities TEACHER, DEPARTMENT, and EQUIPMENT. A teacher is a member of a department so there is an edge (arrow) TEACHER $\longrightarrow$ DEPARTMENT. An equipment item belongs to a department and is assigned to a teacher. These two facts generate two more arrows, one from EQUIPMENT to TEACHER and a second from EQUIPMENT to DEPARTMENT, giving a triangle. If it is a rule that equipment must be assigned to a teacher who is a member of the department to which the equipment belongs, then the triangle of three arrows just described is a *commutativity specification* in the EA sketch.

Finite limit specifications may also be used to model constraints. For example the arrow from an entity to a key attribute is necessarily monic. An example of a more complex constraint, taken from [10] follows. A database involving students, courses, classes and class-times may require that no student 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).

We continue with an outline of the article. In Section 2 we make precise what we mean by modelling a sketch in an object in a 2-category. We examine what becomes an obvious construction of the required object. It seems to be of independent interest. The modellings that we mentioned earlier which describe queries and updates actually take place in 2-categories that are quite different from the 2-category of categories. We should also point out that in this paper we need to have objects of models in the 2-category whose objects are categories with pullbacks and whose arrows are functors that preserve pullbacks. While this is merely a non-full sub-2-category of the 2-category of categories, the modelling formalism here will underscore the need to model database 'statics' in a lextensive category.

Not all of our readers will have read [4] or know otherwise about *equipments*. We have made no attempt to make this paper self-contained by repeating the material from [4]. To do so would have made this paper unbalanced and obscured our main points. In Section 3 we review **CAT**-powers of equipments since the material on this topic in [4] is somewhat scattered. In Section 3 we also include a discussion about inverters in a certain 2-category of equipments, for inverters did not appear in [4] and we need them for a full account here. We have put an asterisk beside this section and subsection 5.8 and Theorem 5.9

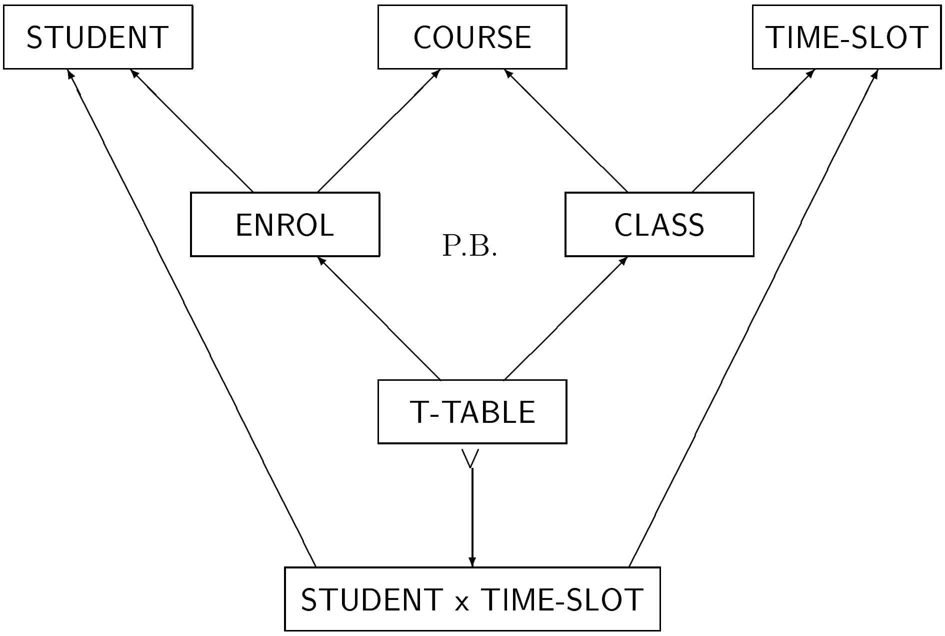


Figure 1: A finite limit constraint

to identify them as containing material about equipments. They can be skipped without interrupting continuity but at the expense of losing what we have to say about database updates.

In Section 4 we formally define EA sketches. The category of models $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ of an EA sketch $\mathbb{E}$ in an appropriate — lextensive — category $\mathcal{S}$ are called $\mathbb{E}$-database states in $\mathcal{S}$. $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ is an object of models of $\mathbb{E}$ in $\mathcal{S}$. We show some properties of the category of models. As a special case we mention the 'keyed' EA sketches in which every entity has a specified monomorphism to a key attribute. Here we find that database states have the property that transition arrows are monic. Furthermore, the category of models simplifies to a preorder which is finite if $\mathbb{E}$ is so and $\mathcal{S}$ has finitely many subobjects of 1.

Finally, in Section 5 we describe the associated theory of an EA sketch and show how to view it as the query language. We are able to exploit the considerations of Section 2 and see the query language as the object of models of $\mathbb{E}$ in the free lextensive category on **1** *in the opposite of the 2-category of lextensive categories.* In this Section we also consider *spans* of models of EA sketches. We interpret a span of models as an update of database states. In the special case of keyed EA 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. This 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 EA sketch $\mathbb{E}$ in $\mathcal{S}$ is the object of models of $\mathbb{E}$ in the equipment of of all spans in $\mathcal{S}$ *in the 2-category of starred,* *pointed equipments.*

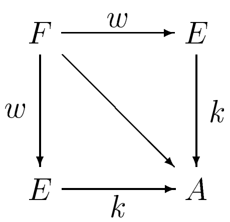
We should point out that using sketches for data specification has led to recent progress in understanding the view updateability problem [11]. Before proceeding we note that some other authors have also proposed sketches for data specification. Piessens and Steegmans [12, 13] also consider models of sketches, but give less consideration to the query language and updates. Motivated by the problem of view integration they obtain some very interesting results on algorithmic determination of equivalence of model categories for certain classes of sketches. Diskin and Cadish [8] describe extensions to entity-relationship-attribute specifications using sketches. 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. Some results about query languages and updates in a categorical model of relational databases can be found in [14].

2. Modelling a Sketch in an Object in a 2-Category

2.1. A *sketch* $\mathbb{S} = (\mathbb{G}, D, L, R)$ is usually understood to consist of a directed graph $\mathbb{G}$ together with a set D of diagrams in $\mathbb{G}$, a set L of cones in $\mathbb{G}$ and a set R of cocones in $\mathbb{G}$. A *model* M of $\mathbb{S}$ in a category $\mathcal{C}$ is a graph homomorphism $M : \mathbb{G} \longrightarrow \mathcal{C}$ which sends diagrams belonging to D to commutative diagrams, cones belonging to L to limit cones and cocones belonging to R to colimit cocones. A *homomorphism of models* $h : M \longrightarrow N$ is a natural transformation. For a fuller treatment we refer the reader to [1] or [2]. Models and their homomorphisms determine a category that we denote by $\mathrm{Mod}(\mathbb{S}, \mathcal{C})$.

If we write $\mathbb{C}$ for the category generated by $\mathbb{G}$ subject to the relations D then a graph homomorphism $M : \mathbb{G} \longrightarrow \mathcal{C}$ that sends diagrams in D to commutative diagrams is the same thing as a functor $M : \mathbb{C} \longrightarrow \mathcal{C}$. A natural transformation $h : M \longrightarrow N : \mathbb{G} \longrightarrow \mathcal{C}$ between such graph homomorphisms is the same thing as a natural transformation $h : M \longrightarrow N : \mathbb{C} \longrightarrow \mathcal{C}$. The presentation of $\mathbb{C}$ in terms of generators and relations is significant for the applications but somewhat cumbersome for our observations in this section. Thus we will also write $\mathbb{S} = (\mathbb{C}, L, R)$ for a sketch, where $\mathbb{C}$ is a category, and then models M of $\mathbb{S}$ in $\mathcal{C}$ are functors $M : \mathbb{C} \longrightarrow \mathcal{C}$ which send cones belonging to L to limit cones and cocones belonging to R to colimit cocones. It is even more convenient to write $\mathbb{S}$ for both the sketch $\mathbb{S}$ and the category $\mathbb{C}$. We continue to write $\mathrm{Mod}(\mathbb{S}, \mathcal{C})$ for the category of models of sketch $\mathbb{S}$ in category $\mathcal{C}$.

If $\mathbb{S} = (\mathbb{S}, L, R)$ is a sketch and $k : E \longrightarrow A$ is an arrow in $\mathbb{S}$ then, for the diagram $k : E \longrightarrow A \longleftarrow E : k$ in $\mathbb{S}$, consider the following cone to it:



If this cone is in L then it follows that, for any model D of $\mathbb{S}$, Dk is a monomorphism.

When this is the case it is convenient to say k *is a specified monomorphism of* $\mathbb{S}$.

2.2. Definition. *For a sketch* $\mathbb{S} = (\mathbb{S}, L, R)$ *and an object* $\mathcal{K}$ *in a 2-category* $\mathbf{K}$*, an object of* models of $\mathbb{S}$ in $\mathcal{K}$ *is an object* $\mathrm{Mod}(\mathbb{S},\mathcal{K})$ *in* $\mathbf{K}$*, together with a model* M *of* $\mathbb{S}$ *in the category* $\mathbf{K}(\mathrm{Mod}(\mathbb{S},\mathcal{K}),\mathcal{K})$*, for which, for all* $\mathcal{A}$ *in* $\mathbf{K}$*, the assignment*

$$\mathcal{A} \xrightarrow{F} \mathrm{Mod}(\mathbb{S},\mathcal{K}) \quad \longmapsto \quad \mathbb{S} \xrightarrow{M} \mathbf{K}(\mathrm{Mod}(\mathbb{S},\mathcal{K}),\mathcal{K}) \xrightarrow{\mathbf{K}(F,\mathcal{K})} \mathbf{K}(\mathcal{A},\mathcal{K})$$

provides an equivalence of categories

$$\mathbf{K}(\mathcal{A},\mathrm{Mod}(\mathbb{S},\mathcal{K})) \xrightarrow{\sim} \mathrm{Mod}(\mathbb{S},\mathbf{K}(\mathcal{A},\mathcal{K}))$$

2.3. We are taking the point of view that $\mathrm{Mod}(\mathbb{S},\mathcal{K})$ is a weighted bilimit in $\mathbf{K}$ and, as such, may or may not exist. It may be useful to point out an aspect of the definition that is somewhat masked in familiar 2-categories of categories. Consider a typical element π of the set L of cones of $\mathbb{S}$ and a typical element σ of the set R of cocones of $\mathbb{S}$.

It follows from our definition that MP is a limit of the diagram $M\mathcal{P}$ in the category $\mathbf{K}(\mathrm{Mod}(\mathbb{S},\mathcal{K}),\mathcal{K})$, which is preserved by composition with all $F : \mathcal{A} \longrightarrow \mathrm{Mod}(\mathbb{S},\mathcal{K})$ in $\mathbf{K}$. Similarly, MS is a colimit of the diagram $M\mathcal{S}$, which is preserved by composition with all such F. The preservation properties are often automatic for complete objects $\mathcal{K}$ in familiar $\mathbf{K}$, such as $\mathbf{CAT}$. With reference to the diagrams above, note that we will write $\Pi = \Pi(\pi)$, $\mathcal{P} = \mathcal{P}(\pi)$, $S = S(\sigma)$, and so on, to name the data implicit in the statement that π is a cone or that σ is a cocone.

2.4. The question immediately arises as to how $\mathrm{Mod}(\mathbb{S},\mathcal{K})$ might be calculated in terms of more familiar bilimits. The only answer that we need for our present work is a generalization of the description of $\mathrm{Mod}(\mathbb{S},\mathcal{C})$ for $\mathcal{C}$ a category. We will assume that the 2-category $\mathbf{K}$ admits $\mathbf{CAT}$-*powers* (being what most authors call $\mathbf{CAT}$-cotensor-products.) Thus, for all $\mathcal{K}$ in $\mathbf{K}$ and $\mathbb{C}$ in $\mathbf{CAT}$ there is an object $\mathcal{K}^{\mathbb{C}}$ in $\mathbf{K}$ and a functor $E : \mathbb{C} \longrightarrow \mathbf{K}(\mathcal{K}^{\mathbb{C}},\mathcal{K})$ for which, for all $\mathcal{A}$ in $\mathbf{K}$, the assignment

$$\mathcal{A} \xrightarrow{F} \mathcal{K}^{\mathbb{C}} \quad \longmapsto \quad \mathbb{C} \xrightarrow{E} \mathbf{K}(\mathcal{K}^{\mathbb{C}},\mathcal{K}) \xrightarrow{\mathbf{K}(F,\mathcal{K})} \mathbf{K}(\mathcal{A},\mathcal{K})$$

provides an equivalence of categories

$$\mathbf{K}(\mathcal{A},\mathcal{K}^{\mathbb{C}}) \xrightarrow{\sim} \mathbf{CAT}(\mathbb{C},\mathbf{K}(\mathcal{A},\mathcal{K}))$$

For all $\mathcal{K}$ in $\mathbf{K}$, we have a 2-functor $\mathcal{K}^{(-)} : \mathbf{CAT}^{op} \longrightarrow \mathbf{CAT}$. In particular, for all $\mathcal{K}$ in $\mathbf{K}$ and all $\mathbb{C}$ in $\mathbf{CAT}$ there is an arrow $\mathcal{K}^{!} : \mathcal{K} \longrightarrow \mathcal{K}^{\mathbb{C}}$ in $\mathbf{K}$ arising from $! : \mathbb{C} \longrightarrow \mathbf{1}$, where we have identified $\mathcal{K}^{\mathbf{1}}$ with $\mathcal{K}$. The object $\mathcal{K}$ in $\mathbf{K}$ is said to have $\mathbb{C}$-limits if $\mathcal{K}^{!} : \mathcal{K} \longrightarrow \mathcal{K}^{\mathbb{C}}$ has a right adjoint, denoted $\lim_{\leftarrow} : \mathcal{K}^{\mathbb{C}} \longrightarrow \mathcal{K}$, in $\mathbf{K}$. Similarly, $\mathcal{K}$ is said to have $\mathbb{C}$-colimits if $\mathcal{K}^{!} : \mathcal{K} \longrightarrow \mathcal{K}^{\mathbb{C}}$ has a left adjoint, denoted $\lim_{\rightarrow} : \mathcal{K}^{\mathbb{C}} \longrightarrow \mathcal{K}$.

2.5. Consider the elements π of L and σ of R displayed in 2.3. We will now suppose that $\mathcal{K}$ has $\Pi(\pi)$-limits, for all π in L, and $\Sigma(\sigma)$-colimits, for all σ in R. Consider also

$$\begin{array}{ccc} \mathcal{K}^{\mathbb{S}} & \xrightarrow{\ \mathcal{K}^{P}\ } & \mathcal{K}^{\Pi} \\ & {\scriptstyle \mathcal{K}^{p}}\searrow \quad \Uparrow \mathcal{K}^{\pi} \quad \nearrow{\scriptstyle \mathcal{K}^{!}} & \\ & \mathcal{K} & \end{array} \qquad \begin{array}{ccc} \mathcal{K}^{\mathbb{S}} & \xrightarrow{\ \mathcal{K}^{S}\ } & \mathcal{K}^{\Sigma} \\ & {\scriptstyle \mathcal{K}^{s}}\searrow \quad \Downarrow \mathcal{K}^{\sigma} \quad \nearrow{\scriptstyle \mathcal{K}^{!}} & \\ & \mathcal{K} & \end{array}$$

$$\begin{array}{ccc} \mathcal{K}^{\mathbb{S}} & \xrightarrow{\ \mathcal{K}^{P}\ } & \mathcal{K}^{\Pi} \\ & {\scriptstyle \mathcal{K}^{p}}\searrow \quad \xrightarrow[\widetilde{\pi}]{} \quad \swarrow{\scriptstyle \lim_{\leftarrow}} & \\ & \mathcal{K} & \end{array} \qquad \begin{array}{ccc} \mathcal{K}^{\mathbb{S}} & \xrightarrow{\ \mathcal{K}^{S}\ } & \mathcal{K}^{\Sigma} \\ & {\scriptstyle \mathcal{K}^{s}}\searrow \quad \xleftarrow[\widetilde{\sigma}]{} \quad \swarrow{\scriptstyle \lim_{\rightarrow}} & \\ & \mathcal{K} & \end{array}$$

where the first pair of triangles result from applying the 2-functor $\mathcal{K}^{(-)}$ to π and σ and the second pair are obtained from the first pair by adjointness. Define a transformation

$$\mathcal{K}^{\mathbb{S}} \underset{\longrightarrow}{\overset{\longrightarrow}{\Downarrow \widetilde{\mathbb{S}}}} \mathcal{K}^{L+R}$$

whose π-component, for all π in L, is $\widetilde{\pi}$ and whose σ-component, for all σ in R, is $\widetilde{\sigma}$.

Observe that if $\mathbf{K}$ is $\mathbf{CAT}$, then $\mathcal{K}^{\mathbb{S}}$ is the usual functor category. In this case, to say that the M-component of $\widetilde{\pi}$ is invertible is precisely to say that M sends the cone π to a limit cone and to say that the M-component of $\widetilde{\sigma}$ is invertible is precisely to say that M sends the cocone σ to a colimit cocone. It follows that the M-component of $\widetilde{\mathbb{S}}$ is invertible if and only if M sends every cone belonging to L to a limit cone and every cocone belonging to R to a colimit cocone.

2.6. Proposition. *For a sketch $\mathbb{S} = (\mathbb{S}, L, R)$ and an object $\mathcal{K}$ in a bicategory $\mathbf{K}$, if $\mathbf{K}$ has powers and inverters and $\mathcal{K}$ has $\Pi(\pi)$-limits, for all π in L, and $\Sigma(\sigma)$-colimits, for all σ in R, then* $\mathrm{Mod}(\mathbb{S}, \mathcal{K})$ *exists and is given by the following inverter diagram:*

$$\mathrm{Mod}(\mathbb{S}, \mathcal{K}) \xrightarrow{\ I\ } \mathcal{K}^{\mathbb{S}} \underset{\longrightarrow}{\overset{\longrightarrow}{\Downarrow \widetilde{\mathbb{S}}}} \mathcal{K}^{L+R}$$

with the required model $M : \mathbb{S} \longrightarrow \mathbf{K}(\mathrm{Mod}(\mathbb{S}, \mathcal{K}), \mathcal{K})$ given by

$$\mathbb{S} \xrightarrow{\ E\ } \mathbf{K}(\mathcal{K}^{\mathbb{S}}, \mathcal{K}) \xrightarrow{\ \mathbf{K}(I, \mathcal{K})\ } \mathbf{K}(\mathrm{Mod}(\mathbb{S}, \mathcal{K}), \mathcal{K})$$

Proof. The discussion preceding the Proposition provides the result in $\mathbf{CAT}$, from which the general result follows since the definitions of powers, inverters and objects of models in $\mathbf{K}$ are all given in terms of $\mathbf{CAT}$-valued representability. ∎

2.7. Corollary. *If* $\mathcal{F}:\mathbf{K}\longrightarrow\mathbf{L}$ *is a 2-functor that preserves powers and inverters then it preserves modelling of* $\mathbb{S}$ *in the sense that the evident comparison arrow*

$$\mathcal{F}(\mathrm{Mod}(\mathbb{S},\mathcal{K}))\longrightarrow\mathrm{Mod}(\mathbb{S},\mathcal{F}\mathcal{K})$$

in $\mathbf{L}$ *is an equivalence.*

Proof. The only point that requires comment is the construction of the transformation $\tilde{\mathbb{S}}$ in $\mathbf{L}$. However, 2-functors send adjunctions to adjunctions and since $\mathcal{F}$ preserves powers, it follows that $\mathcal{F}\mathcal{K}$ has the necessary limits and colimits when $\mathcal{K}$ does. ∎

2.8. A comment on the presentation of $\mathbb{S}=(\mathbb{S},L,R)$ in the form $\mathbb{S}=(\mathbb{G},D,L,R)$ as touched on in 2.1 is in order. In a 2-category $\mathbf{K}$, it might well be the case that powers $\mathcal{K}^{\mathbb{S}}$ are constructed in a fairly concrete way and one can hope to exploit a presentation of a category $\mathbb{S}$ in terms of a graph $\mathbb{G}$ and a set of diagrams D. Here it is understood that a *diagram* δ in $\mathbb{G}$ is to consist of a pair of vertices $S(\delta)$ and $T(\delta)$ of $\mathbb{G}$ together with a pair of paths $p(\delta),q(\delta):S(\delta)\rightrightarrows T(\delta)$ which we wish to see as

$$\mathbf{1}\ \underset{T(\delta)}{\overset{S(\delta)}{\rightrightarrows}}\ \mathbb{G}\qquad p(\delta)\Downarrow\ \Downarrow q(\delta)$$

and that $\mathbb{S}$ is obtained from the free category on $\mathbb{G}$ by identifying, for each $\delta\in D$, the arrows resulting from $p(\delta)$ and $q(\delta)$. If $\mathbf{K}$ admits powers of an object $\mathcal{K}$ for all the data in the display above then we obtain

$$\mathcal{K}^{\mathbb{G}}\ \underset{\mathcal{K}^{T(\delta)}}{\overset{\mathcal{K}^{S(\delta)}}{\rightrightarrows}}\ \mathcal{K}\qquad \mathcal{K}^{p(\delta)}\Downarrow\ \Downarrow\mathcal{K}^{q(\delta)}$$

and if such is taken as the δ-component, for $\delta\in D$, of a diagram

$$\mathcal{K}^{\mathbb{G}}\ \rightrightarrows\ \mathcal{K}^{D}\qquad \Downarrow\ \Downarrow$$

then it is a straightforward matter to show that an equifier for it will provide the power $\mathcal{K}^{\mathbb{S}}$ as in:

$$\mathcal{K}^{\mathbb{S}}\longrightarrow\mathcal{K}^{\mathbb{G}}\ \rightrightarrows\ \mathcal{K}^{D}\qquad \Downarrow\ \Downarrow$$

In the 2-category to be considered in subsection 5.8 it is clear that the considerations of this subsection do apply.

3.* Powers and Inverters in a 2-Category of Equipments

3.1. As pointed out in the Introduction, one of the main points in this paper, Theorem 5.9 and the subsection 5.8 preceding it, does require some knowledge of *equipments* as studied in [4]. In particular, our work requires an understanding of the formation of spans for a category $\mathcal{K}$ with pullbacks, herein and in [4] denoted $\mathbf{spn}\mathcal{K}$, as a 2-functor. In this paper we can restrict our attention to the case where the domain of **spn** is $\mathbf{PBK_{pbk}}$, the 2-category of categories with pullbacks, pullback-preserving functors and all natural transformations between these. In this case **spn** takes values in $^*\mathbf{EQT}_{*\mathbf{hom}}$. Here the objects are *starred pointed equipments*, arrows are *homomorphisms* of equipments, and transformations are all equipment transformations between these.

3.2. In $\mathbf{PBK_{pbk}}$, **CAT**-powers exist and are constructed as in **CAT**. More precisely, if $\mathcal{K}$ is in $\mathbf{PBK_{pbk}}$ and $\mathbb{C}$ is in **CAT** then the functor category $\mathcal{K}^{\mathbb{C}}$, together with evaluation of functors, provides a power object in $\mathbf{PBK_{pbk}}$. The 2-category $^*\mathbf{EQT}_{*\mathbf{hom}}$ also has all **CAT**-powers. For an equipment $(\mathcal{K},\mathcal{M})$ and a category $\mathbb{C}$, we will follow [4] to describe $(\mathcal{K},\mathcal{M})^{\mathbb{C}}$. It is helpful to recall first the category $\mathbf{gr}\mathcal{M}$ associated to the equipment $(\mathcal{K},\mathcal{M})$. The objects of $\mathbf{gr}\mathcal{M}$ are triples (K,μ,L), where K and L are objects of the *category of scalars* $\mathcal{K}$ and μ is an object of the vector category $\mathcal{M}(K,L)$. It is convenient to call μ a *vector arrow* and write $\mu : K \dashrightarrow L$. Arrows between vector arrows are given by squares

$$\begin{array}{ccc} K & \xrightarrow{k} & K' \\ \mu \big\downarrow & \xrightarrow{\Phi} & \big\downarrow \mu' \\ L & \xrightarrow[l]{} & L' \end{array}$$

Here $\Phi : l\mu \longrightarrow \mu' k$ is an arrow in the vector category $\mathcal{M}(K,L')$, whose domain $l\mu$ employs the left action of $\mathcal{K}$ on vectors and whose codomain $\mu' k$ employs the right action of $\mathcal{K}$ on vectors. Arrows such as Φ from the vector categories are also known as *vector transformations* of the equipment $(\mathcal{K},\mathcal{M})$. Composition in $\mathbf{gr}\mathcal{M}$ is given by horizontal pasting of squares. There are evident domain and codomain functors $\partial_0, \partial_1 : \mathbf{gr}\mathcal{M} \longrightarrow \mathcal{K}$.

The category of scalars of $(\mathcal{K},\mathcal{M})^{\mathbb{C}}$ is just the functor category $\mathcal{K}^{\mathbb{C}}$. For P and Q in $\mathcal{K}^{\mathbb{C}}$, the objects of the vector category $(\mathcal{K},\mathcal{M})^{\mathbb{C}}(P,Q)$ are given by functors $\rho : \mathbb{C} \longrightarrow \mathbf{gr}\mathcal{M}$ with $\partial_0\rho = P$ and $\partial_1\rho = Q$. Arrows $T : \rho \longrightarrow \sigma$ in $(\mathcal{K},\mathcal{M})^{\mathbb{C}}(P,Q)$ are given by natural transformations $T : \rho \longrightarrow \sigma : \mathbb{C} \longrightarrow \mathbf{gr}\mathcal{M}$ with the property that $\partial_0 T = 1_P$ and $\partial_1 T = 1_Q$. For $p : P' \longrightarrow P$ in $\mathcal{K}^{\mathbb{C}}$ and $\rho : P \dashrightarrow Q$ in $(\mathcal{K},\mathcal{M})^{\mathbb{C}}(P,Q)$, $(\rho.p)(C) = \rho(C)p(C)$ and, similarly, for $q : Q \longrightarrow Q'$ in $\mathcal{K}^{\mathbb{C}}$, $(q.\rho)(C) = q(C)\rho(C)$. If $(\mathcal{K},\mathcal{M})$ has the structure of a pointing given by vectors $\iota_K : K \dashrightarrow K$ then $(\mathcal{K},\mathcal{M})^{\mathbb{C}}$ is pointed by $\iota_P(C) = \iota_{P(C)}$. If $(\mathcal{K},\mathcal{M})$ has the starred property then so does $(\mathcal{K},\mathcal{M})^{\mathbb{C}}$.

3.3. For $\mathcal{K}$ a category with pullbacks, $\mathbf{spn}\mathcal{K} = (\mathcal{K}, \mathbf{spn}\mathcal{K})$. In other words, the scalar category of the equipment $\mathbf{spn}\mathcal{K}$ is $\mathcal{K}$ and, for K and L objects of $\mathcal{K}$, the vector category $\mathbf{spn}\mathcal{K}(K,L)$ is the usual category of spans from K to L. Great detail about this example of an equipment can be found in [4]. In the case of **spn**$\mathcal{K}$ the squares of 3.2, arrows in **gr**(**spn**$\mathcal{K}$), take the form of comutative diagrams:

$$\begin{array}{ccc} K & \xrightarrow{\quad k \quad} & K' \\ {\scriptstyle m_0}\uparrow & & \uparrow{\scriptstyle m_0'} \\ M & \xrightarrow{\quad f \quad} & M' \\ {\scriptstyle m_1}\downarrow & & \downarrow{\scriptstyle m_1'} \\ L & \xrightarrow[\quad l \quad]{} & L' \end{array}$$

With this observation at hand it is easy to verify:

3.4. Proposition. *The 2-functor* $\mathbf{spn} : \mathbf{PBK_{pbk}} \longrightarrow {}^*\mathbf{EQT}_{*\mathbf{hom}}$ *preserves powers.* ■

3.5. Inverters exist in $\mathbf{PBK_{pbk}}$ and are constructed as in **CAT**. Inverters also exist in the 2-category ${}^*\mathbf{EQT}_{*\mathbf{hom}}$ but these were not described in [4]. We find it convenient here to use further notation from [4] and write $\mathcal{A} = (\mathcal{A}_\#, \mathcal{A})$ for an equipment. In fact, extraction of scalar components, $(-)_\# : {}^*\mathbf{EQT}_{*\mathbf{hom}} \longrightarrow \mathbf{CAT}$, is a 2-functor. Consider a transformation in ${}^*\mathbf{EQT}_{*\mathbf{hom}}$

$$\begin{array}{ccc} & \xrightarrow{\qquad T \qquad} & \\ \mathcal{A} & \Downarrow u & \mathcal{B} \\ & \xrightarrow[\qquad S \qquad]{} & \end{array}$$

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

$$\begin{array}{ccccc} & & \mathcal{B}(TA, TC) & & \\ & {\scriptstyle T_{A,C}} \nearrow & & \searrow {\scriptstyle \mathcal{B}(TA, u_C)} & \\ \mathcal{A}(A, C) & & \Downarrow u_- & & \mathcal{B}(TA, SC) \\ & {\scriptstyle S_{A,C}} \searrow & & \nearrow {\scriptstyle \mathcal{B}(u_A, SC)} & \\ & & \mathcal{B}(SA, SC) & & \end{array}$$

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, for $\mu : A \longrightarrow\!\!\!\!\!\!\!\!+\;\; C$, the μ-component of u_- is

$$\begin{array}{ccc} TA & \xrightarrow{\quad u_A \quad} & SA \\ {\scriptstyle T\mu}\downarrow & \xrightarrow{\; u_\mu \;} & \downarrow{\scriptstyle S\mu} \\ TC & \xrightarrow[\quad u_C \quad]{} & SC \end{array}$$

as provided by the definition of a transformation in ${}^*\mathbf{EQT}_{*\mathbf{hom}}$. 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. The scalar actions for $\mathcal{I}$ are inherited from those in $\mathcal{A}$ — this making sense because T and S are homomorphisms of equipments. Because u is a transformation of pointed equipments it follows that u_{ι_A} is an isomorphism whereupon ι_A is in $\mathcal{I}(A, A)$ and provides the components for a pointing for $\mathcal{I}$. The starred property for $\mathcal{I}$ is inherited from that of $\mathcal{A}$.

3.6. Given an inverter diagram in $\mathbf{PBK_{pbk}}$

$$\mathcal{D} \xrightarrow{\;H\;} \mathcal{E} \underset{F}{\overset{G}{\rightrightarrows}} \mathcal{F} \qquad (t : G \Rightarrow F)$$

we consider the induced arrow $K : \mathbf{spn}\mathcal{D} \longrightarrow \mathcal{I}$ in ${}^*\mathbf{EQT}_{*\mathbf{hom}}$, where $\mathcal{I}$ is the inverter of $\mathbf{spn}t$. Because $(-)_\#\mathbf{spn} : \mathbf{PBK_{pbk}} \longrightarrow \mathbf{CAT}$ is the inclusion, it follows from 3.5 that $K_\#$ is invertible — in fact it can even be taken to be the identity on $\mathcal{D}$, provided that the inverter of t is constructed by the canonical full sub-category construction both in $\mathbf{PBK_{pbk}}$ and when seen as $(-)_\#\mathbf{spn}t$. Moreover the vector arrows in $\mathcal{I}$ are certain spans, with mediating object in $\mathcal{E}$. However, given a vector arrow $a : A \longleftarrow S \longrightarrow C : c$ in $\mathbf{spn}\mathcal{E}$, the square for $(\mathbf{spn}t)_{(a,S,c)}$, as in the last diagram of 3.5 takes here the form of a commutative diagram

$$\begin{array}{ccc} GA & \xrightarrow{\;t_A\;} & FA \\ {\scriptstyle Ga}\uparrow & & \uparrow{\scriptstyle Fa} \\ GS & \xrightarrow{\;t_S\;} & FS \\ {\scriptstyle Gc}\downarrow & & \downarrow{\scriptstyle Fc} \\ GC & \xrightarrow[\;t_C\;]{} & FC \end{array}$$

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

3.7. Proposition. *The 2-functor* $\mathbf{spn} : \mathbf{PBK_{pbk}} \longrightarrow {}^*\mathbf{EQT}_{*\mathbf{hom}}$ *preserves inverters.* ∎

4. EA Sketches and Database States

4.1. Definition. *An* EA sketch $\mathbb{E} = (\mathbb{E}, L, R)$ *is a sketch for which*

i) every π *in* L *has* $\Pi(\pi)$ *finite;*

ii) every σ *in* R *has* $\Sigma(\sigma)$ *finite and discrete;*

iii) there is an object 1 *in* $\mathbb{E}$ *and the cone*

$$\begin{array}{ccc} \mathbf{0} & \xrightarrow{\quad !\quad} & \mathbb{E} \\ & {\scriptstyle !}\searrow \;\; \uparrow 1_! \;\; \nearrow{\scriptstyle 1} & \\ & \mathbf{1} & \end{array}$$

is in L.

If

$$\begin{array}{ccccc} \Sigma & \xrightarrow{\;!\;} & \mathbf{1} & \xrightarrow{\;1\;} & \mathbb{E} \\ & {\scriptstyle !}\searrow & \downarrow{\scriptstyle \sigma} & \nearrow{\scriptstyle A} & \\ & & \mathbf{1} & & \end{array}$$

is a cocone in R *then* A *is called an* attribute. *An EA sketch is* keyed *if, for each object* E *in* $\mathbb{E}$*, there is a specified monomorphism* $k_E : E \rightarrowtail A_E$*, where* A_E *is an attribute.*

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

The objects in which EA sketches are modelled must necessarily have finite limits and finite sums. In the applications sums need to be well-behaved so that, at least when modelling in a category, the *lextensive* axiom should hold. Recall that a category is said to be *lextensive* if it has finite limits and finite sums which are disjoint and universal. A basic reference for lextensive categories is [5]. It is possible to define lextensive objects in 2-categories other than **CAT** but we do not need to do so here. Henceforth, $\mathcal{S}$ will denote a lextensive category. We will write **LXT** for the 2-category of lextensive categories, functors that preserve finite limits and finite sums, and natural transformations between these. Note that for $\mathcal{S}$ in **LXT**, $- + - : \mathcal{S} \times \mathcal{S} \longrightarrow \mathcal{S}$ preserves pullbacks — in fact it preserves all finite connected limits.

4.2. Remark. Notice that the definition of EA-sketch does not rule out the possibility of an *inconsistent sketch*, that is one for which models are trivial. From Definition 4.1 it follows that for any $M : \mathbb{E} \longrightarrow \mathcal{S}$ to be a model we have, in particular, $M1 \xrightarrow{\simeq} 1$. Thus if, for example, the cocone $1 \longrightarrow 1 \longleftarrow 1$ is in R then we must have $1 \longrightarrow 1 \longleftarrow 1$ a sum diagram in $\mathcal{S}$. In a lextensive category this implies $0 \xrightarrow{\simeq} 1$ so that either $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ is empty or $\mathcal{S} \xrightarrow{\simeq} \mathbf{1}$. Of course, an inconsistency in a sketch can be quite subtle.

4.3. Definition. *For an EA sketch* $\mathbb{E}$*, an* $\mathbb{E}$-database state *in* $\mathcal{S}$ *is a model of* $\mathbb{E}$ *in* $\mathcal{S}$*. The* category of $\mathbb{E}$-database states in $\mathcal{S}$ *is the category* $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$.

An example of a database state in the lextensive category of finite sets, for the EA sketch of our school data specification is provided by any database which models the specifications described *and* the constraints implied by the commutative diagrams, the limit cones, and the sum cocones. The next proposition gives a property of the category of models of an EA sketch which is crucial to our discussion of database updates.

4.4. Proposition. *For* $\mathbb{E}$ *an EA sketch and* $\mathcal{S}$ *a lextensive category,* $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ *has pullbacks (in fact finite connected limits) and these are computed pointwise. If* $\mathcal{S}$ *is a Grothendieck topos then* $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ *has filtered colimits.*

Proof. For $D_1 \longrightarrow D_0 \longleftarrow D_2$ in $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ and E in $\mathbb{E}$, define $D(E)$ to be the pullback

$$\begin{array}{ccc} D(E) & \longrightarrow & D_2(E) \\ \downarrow & & \downarrow \\ D_1(E) & \longrightarrow & D_0(E) \end{array}$$

in $\mathcal{S}$. By the universality of pullbacks we can extend D to arrows of $\mathbb{E}$ so that the commutative diagrams of $\mathbb{E}$ are respected. Because D is defined by pullbacks, it is immediate that D sends cones in L to limit cones in $\mathcal{S}$, since finite limits commute with pullbacks. To complete the proof we need to show that D sends cocones in R to sums in $\mathcal{S}$. However, since $\mathcal{S}$ is lextensive the finite sum functors $\mathcal{S}^n \longrightarrow \mathcal{S}$ preserve pullbacks (more generally all finite connected limits) and from this it follows that D is a model of $\mathbb{E}$ and the pullback of $D_1 \longrightarrow D_0 \longleftarrow D_2$ in $\mathcal{S}^{\mathbb{E}}$ is the pullback in $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$.

If $\mathcal{S}$ is a Grothendieck topos then filtered colimits commute with finite limits in $\mathcal{S}$. In this case, filtered colimits can be constructed pointwise in $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$. ∎

We will need later the 2-category $\mathbf{PBK_{pbk}}$ of categories with pullbacks, functors which preserve pullbacks, and all natural transformations between these. In particular, we will need:

4.5. Proposition. *For* $\mathbb{E}$ *an EA sketch and* $\mathcal{S}$ *a lextensive category,* $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ *provides an object of models of* $\mathbb{E}$ *in* $\mathcal{S}$ *in the 2-category* $\mathbf{PBK_{pbk}}$.

Proof. Since $\mathbf{PBK_{pbk}}$ has inverters and powers which are constructed as in $\mathbf{CAT}$, this follows from Proposition 2.6 as soon as we show that the transformation $\widetilde{\mathbb{E}}$ lies in $\mathbf{PBK_{pbk}}$ for $\mathbb{E}$ a finite-limit, finite-sum sketch and $\mathcal{S}$ lextensive. This last follows by inspection of the construction of $\widetilde{\mathbb{E}}$ in 2.5. One must check that all (1-)arrows in the case at hand are pullback-preserving functors. The only ones which require comment are those of the form $\lim_{\rightarrow} : \mathcal{S}^{\Sigma(\sigma)} \longrightarrow \mathcal{S}$, for $\sigma \in R$. Again, all these are finite summations since $\mathbb{E}$ is an EA sketch and finite summations are pullback-preserving since $\mathcal{S}$ is lextensive. ∎

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

4.6. Proposition. *For* D *and* D' *database states and* A *an attribute of* $\mathbb{E}$*, there is a canonical isomorphism* $i_A : D(A) \xrightarrow{\cong} D'(A)$*. Moreover, for any* $\alpha : D \longrightarrow D'$ *in* $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$*,* $\alpha_A = i_A : D(A) \xrightarrow{\cong} D'(A)$.

Proof. We have already observed that for any D in $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$, $D1 \xrightarrow{\cong} 1$. If A is an attribute in virtue of an element σ in R, with $\Sigma(\sigma)$ the discrete category with n objects, then $DA \cong n \cdot 1$ in $\mathcal{S}$. Explicitly, if σ is $\langle a_j : 1 \longrightarrow A \rangle_{j \in n}$ then we can define i_A to be the unique arrow making all n of the following squares commute

$$\begin{array}{ccc} D(1) & \xrightarrow{D(a_j)} & D(A) \\ {\scriptstyle i_1}\big\downarrow & & \big\downarrow{\scriptstyle i_A} \\ D'(1) & \xrightarrow[D'(a_j)]{} & D'(A) \end{array}$$

where i_1 is the unique isomorphism. For any $\alpha : D \longrightarrow D'$ in $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$, $\alpha_1 = i_1$, from which it follows by naturality and $D(A)$ being a sum that $\alpha_A = i_A$. ∎

4.7. PROPOSITION. *If $\alpha : D \longrightarrow D'$ is a morphism of database states for a keyed EA 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_E is an attribute. Now consider the commutative square:

$$\begin{array}{ccc} D(E) & \xrightarrow{D(k_E)} & D(A_E) \\ {\scriptstyle \alpha_E}\big\downarrow & & \big\downarrow{\scriptstyle \alpha_{A_E}} \\ D'(E) & \xrightarrow[D'(k_E)]{} & D'(A_E) \end{array}$$

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

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

Proof. By Propositions 4.6 and 4.7, all components α_E of any $\alpha : D \longrightarrow D'$ are monic so all α are monic. To see that $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ 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 4.6 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 $\alpha = \alpha'$ and hence $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ is a preorder.

Since $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ has pullbacks by Proposition 4.4 it has meets of bounded pairs.

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

5. Database Queries and Updates

5.1. We first note that EA sketches are examples of the *finite-sum sketches*, abbreviated *FS sketches*, in [2]. For $\mathbb{E}$ an FS sketch, Barr and Wells constructed in [2] a lextensive category $T(\mathbb{E})$ called the *associated FS theory*. The construction begins with the finite limit completion of $\mathbb{E}$ and uses a suitable subcategory of sheaves for a Grothendieck topology built using the finite sum cocones in $\mathbb{E}$. Note that, for an EA sketch $\mathbb{E}$, $T(\mathbb{E})$ determines an EA sketch by taking the graph, commutativities, finite limit cones and finite sum cocones of $T(\mathbb{E})$ itself. With this point of view, arrows $T(\mathbb{E}) \longrightarrow \mathcal{S}$ in **LXT**, being functors that preserve finite limits and finite sums, can be seen as models of $T(\mathbb{E})$ in $\mathcal{S}$. If $\mathbb{E}$ is finite, $T(\mathbb{E})$ is not necessarily so but $T(\mathbb{E})$ should be thought of as the category of 'derived operations' of the sketch. There is a model J of $\mathbb{E}$ in $T(\mathbb{E})$ and its rôle as described in Section 8.2 of [2] can be summarised as follows:

5.2. THEOREM. *For $\mathbb{E}$ an FS sketch, composition with $J : \mathbb{E} \longrightarrow T(\mathbb{E})$ provides, for any lextensive category $\mathcal{S}$, an equivalence of categories*

$$\mathbf{LXT}(T(\mathbb{E}), \mathcal{S}) \xrightarrow{\sim} \mathrm{Mod}(\mathbb{E}, \mathcal{S})$$

5.3. Our interest in the lextensive category $T(\mathbb{E})$, for an EA sketch $\mathbb{E}$, is that it is precisely the *query language* for the data specification that is described by the EA sketch. Indeed, $T(\mathbb{E})$ is exactly the *classifying category* for the EA information specification as described in [10]. For example, the result of a simple query such as `select` STUDENT `where` NAME `= 'Jones'` is the equalizer of the attribute-defining arrow STUDENT $\longrightarrow$ NAME and the composite STUDENT $\longrightarrow 1 \xrightarrow{\texttt{'Jones'}}$ NAME. 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 EA 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 Q are precisely queries on the original sketch.

To answer a query Q on database state D in $\mathcal{S}$ is to extend $D : \mathbb{E} \longrightarrow \mathcal{S}$ to $\overline{D} : T(\mathbb{E}) \longrightarrow \mathcal{S}$, as in Theorem 5.2, and evaluate $\overline{D}(Q)$.

5.4. The free lextensive category on a category $\mathbb{C}$ can be constructed as the free finite sum completion of the free finite limit completion of $\mathbb{C}$. In fact **LXT** is the 2-category of algebras for the distributive law

$$\mathrm{LexFam} \longrightarrow \mathrm{FamLex}$$

where Fam is the KZ-doctrine whose algebras are categories with finite sums and Lex is the co-KZ-doctrine whose algebras are categories with finite limits. (Details of this will appear elsewhere.) In particular, the free lextensive category on $\mathbf{1}$, which we will denote by $\mathcal{F}$, can be seen as $\mathrm{Fam}({\mathbf{set}_0}^{op})$, where $\mathbf{set}_0$ is the category of finite sets. The free finite sum completion Fam is given by finite families and described, for example, in [9]. Now, referring to Section 2, we consider what it is to model an EA sketch $\mathbb{E}$ in the particular object $\mathcal{F}$ in the 2-category $\mathbf{LXT}^{op}$. An object of models of $\mathbb{E}$ in $\mathcal{F}$ in $\mathbf{LXT}^{op}$ is a lextensive category $\mathcal{L}$ together with a model M of $\mathbb{E}$ in $\mathbf{LXT}^{op}(\mathcal{L},\mathcal{F})$ so that composition with M mediates an equivalence of categories

$$\mathbf{LXT}^{op}(\mathcal{S},\mathcal{L}) \xrightarrow{\sim} \mathrm{Mod}(\mathbb{E}, \mathbf{LXT}^{op}(\mathcal{S},\mathcal{F}))$$

For any lextensive category $\mathcal{S}$ we have an equivalence $\mathcal{S} \xrightarrow{\sim} \mathbf{LXT}(\mathcal{F},\mathcal{S}) = \mathbf{LXT}^{op}(\mathcal{S},\mathcal{F})$ and for any $F : \mathcal{L} \longrightarrow \mathcal{S}$ in $\mathbf{LXT}$ we have a natural isomorphism:

$$\begin{array}{ccc}
\mathbf{LXT}^{op}(\mathcal{L},\mathcal{F}) & \xrightarrow{\mathbf{LXT}^{op}(F,\mathcal{F})} & \mathbf{LXT}^{op}(\mathcal{S},\mathcal{F}) \\
\sim\uparrow & \xrightarrow{\simeq} & \uparrow\sim \\
\mathcal{L} & \xrightarrow[F]{} & \mathcal{S}
\end{array}$$

It follows that if we take $\mathcal{L} = T(\mathbb{E})$ and M to be the composite

$$\mathbb{E} \xrightarrow{J} T(\mathbb{E}) \xrightarrow{\sim} \mathbf{LXT}^{op}(T(\mathbb{E}),\mathcal{F})$$

where $J : \mathbb{E} \longrightarrow T(\mathbb{E})$ is as in Theorem 5.2, then that result can be paraphrased as saying

5.5. Corollary. *For $\mathbb{E}$ an FS sketch (in particular an EA sketch), $T(\mathbb{E})$ provides* $\mathrm{Mod}(\mathbb{E},\mathcal{F})$*, an object of models of $\mathbb{E}$ in $\mathcal{F}$ in $\mathbf{LXT}^{op}$, where $\mathcal{F}$ is the free lextensive category on the category* $\mathbf{1}$. ■

So *if an EA sketch has been constructed to model the* statics *of a particular data specification then the* same *sketch also serves to model the* queries *of the data specification.*

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

5.6. Definition. *For $\mathbb{E}$-database states D and D' in $\mathcal{S}$, an* update *from D to D' is a span from D to D' in* $\mathrm{Mod}(\mathbb{E},\mathcal{S})$.

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 *inserted, deleted* or *changed.* The last operation, however, can be expressed by a deletion of data followed by an insertion of new data. Because of this it can be expressed by a span of monic arrows $D(E) \longleftarrow S(E) \longrightarrow D'(E)$, where $S(E)$ is $D(E)$ after deletions and the mono from $S(E)$ to $D'(E)$ expresses all necessary insertions. Note that for an update from D to D' we require that S be a model — that is a database state — and also that a simple arrow $D(E) \longrightarrow D'(E)$ will not do the job. There may be many models S which represent the same update, possibly including a maximal such S. In the case of a keyed EA sketch, where all arrows in $\mathrm{Mod}(\mathbb{E},\mathcal{S})$ are monic, our definition of update matches the motivation. In any event, by Proposition 4.4, $\mathrm{Mod}(\mathbb{E},\mathcal{S})$ is a category with pullbacks.

5.7. Proposition. *Using pullbacks to compose updates, we obtain a bicategory that we call* $\mathrm{Upd}(\mathbb{E},\mathcal{S})$*, which has $\mathbb{E}$-database states in $\mathcal{S}$ as objects and updates as arrows.* ■

5.8.* Of course $\mathrm{Upd}(\mathbb{E}, \mathcal{S})$ is simply the bicategory of spans in $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$. The latter provides the *maps* — arrows with right adjoints — for the bicategory $\mathrm{Upd}(\mathbb{E}, \mathcal{S})$. We have

$$\mathrm{Mod}(\mathbb{E}, \mathcal{S}) \longrightarrow \mathrm{Upd}(\mathbb{E}, \mathcal{S})$$

whereby an arrow is sent to its graph, regarded as a span. The point of view of [4] is that this displayed homomorphism of bicategories can be regarded as making $\mathrm{Upd}(\mathbb{E}, \mathcal{S})$ into a $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$-*algebra*, meaning at first no more than an analogy with ring theory. Underlying this $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$-algebra structure is a $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$-*bimodule* structure that forgets the general horizontal composites of $\mathrm{Upd}(\mathbb{E}, \mathcal{S})$. It remembers those special horizontal composites in which precisely one factor is a map and regards these as actions of the category $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$. In the terminology of [4], $\mathbf{spn}(\mathrm{Mod}(\mathbb{E}, \mathcal{S}))$ is the *starred, pointed equipment* of spans in $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ underlying the bicategory $\mathrm{Upd}(\mathbb{E}, \mathcal{S})$. Following [4] we have written ${}^*\mathbf{EQT}_{*\mathbf{hom}}$ for the 2-category of starred, pointed equipments; pointed equipment homomorphisms; and transformations between these. (It is crucial for our closing point that equipments form 2-categories in natural ways, rather than 3-dimensional structures as is the case with bicategories.) We recall from Proposition 4.5 that, for any finite-limit, finite-sum sketch $\mathbb{E}$ and lextensive category $\mathcal{S}$, $\mathrm{Mod}(\mathbb{E}, \mathcal{S})$ can be seen as an object of models of $\mathbb{E}$ in $\mathcal{S}$ in the 2-category $\mathbf{PBK_{pbk}}$. Our final theorem shows that the equipment of updates $\mathbf{spn}(\mathrm{Mod}(\mathbb{E}, \mathcal{S}))$ can be seen as an object of models of $\mathbb{E}$ in $\mathbf{spn}\mathcal{S}$ in the 2-category of starred pointed equipments and their homomorphisms.

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

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

is an equivalence of starred pointed equipments.

Proof. This follows immediately from Corollary 2.7, Proposition 3.4 and Proposition 3.7. ∎

So *if an EA sketch has been constructed to model the* statics *of a particular data specification then the* same *sketch also serves to model the* dynamics *and, as remarked after Corollary 5.5, the* queries *of the data specification.*

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. 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.

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

[8] 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.

[9] Robbie Gates. On extensive and distributive categories. Thesis, University of Sydney, 1997

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

[11] Michael Johnson and Robert Rosebrugh. View updateability based on the models of a formal specification. *Lecture Notes in Computer Science* 2021:534–549, Springer-Verlag, 2001.

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

[13] Frank Piessens and Eric Steegmans. Proving semantical equivalence of data specifications. *Journal of Pure and Applied Algebra* 116:291–322, 1997.

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

School of Mathematics and Computing
Macquarie University
Sydney 2109, Australia
and

Department of Mathematics and Computer Science
Mount Allison University

Sackville, NB, E0A 3C0 Canada

and

Department of Mathematics and Statistics
Dalhousie University
Halifax, NS, B3H 3J5 Canada
Email: mike@ics.mq.edu.au and rrosebrugh@mta.ca and rjwood@mathstat.dal.ca