Electronic Notes in Theoretical Computer Science 78 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume78.html 18 pages

Three approaches to partiality in the sketch data model [⋆]

Michael Johnson

Departments of Mathematics and Computing
Macquarie University
Sydney, Australia

Robert Rosebrugh

Department of Mathematics and Computer Science
Mount Allison University
Sackville, NB, Canada

Abstract

Partial information is common in real-world databases. Yet the theoretical foundations of data models are not designed to support references to missing data (often termed `nulls`). Instead, we usually analyse a clean data model based on assumptions about complete information, and later retrofit support for `nulls`.

The sketch data model is a recently developed approach to database specification based on category theory. The sketch data model is general enough to support references to missing information within itself (rather than by retrofitting).

In this paper we explore three approaches to incorporating partial information in the sketch data model. The approaches, while fundamentally different, are closely related, and we show that under certain fairly strong hypotheses they are Morita equivalent (that is they have the same categories of models, up to equivalence). Despite this equivalence, the query languages arising from the three approaches are subtly different, and we explore some of these differences.

Key words: Database, semantic data model, null, category theory.

1 Introduction

Partial information is common in real-world databases. To offer a banal example, most address books contain many entries (records) with an empty address field. In database terms, we sometimes say that the address field is `null`.

[⋆] Research partially supported by the Australian Research Council and NSERC Canada.

The theoretical foundations of databases are most frequently based upon the mathematical study of relations. Relations, formally subsets of cartesian products, cannot have fields which are `null`, and attempts to extend the notion of relation to permit `null`s have led to many anomalies (for a textbook treatment of these problems see [18], chapter 18). The problem is essentially that the current foundations depend upon a simple data model based on assumptions about the completeness of information, and later attempts have been made to retrofit support for `null`s.

The sketch data model is a recently developed approach to database specification based on category theory. The sketch data model is general enough to support references to missing information within itself rather than by retrofitting.

In this paper we explore three approaches to incorporating partial information in database specifications structured as sketch data models. For the purposes of this analysis in all three cases we begin with a specification which would correspond to the database with incomplete information prohibited, along with a specification of which attribute values might be allowed to be `null`. The approaches differ in how they encode the information about permitted `null`s. Our main results show that under certain fairly strong hypotheses the three approaches are Morita equivalent (that is they have the same categories of models, up to equivalence). The three approaches are well understood by category theorists and have been applied in a range of areas of theoretical computer science. Their application to data modelling is, to the authors' knowledge, entirely new, and the delicacies of their interaction with the database specification are somewhat surprising.

In the sketch data model the function which assigns to a given entity instance a certain attribute value is specified by an arrow in a directed graph, which we will call here an *attribute arrow*. The approach that we will take is to include with a specification a subset R of its attribute arrows. The arrows in R will be the ones for which null values are permitted (all other attribute functions are required to be fully defined). To simplify these initial explorations of the three approaches we will put some limitations on allowable elements of R. These are the "fairly strong hypotheses" referred to above. The limitations ensure that permitting partiality does not interact too much with other aspects of the specification. Relaxing these limitations will be the subject of future work. It is worth noting that there are several inequivalent approaches to relaxing the limitations, so including them here would necessitate comparing approximately eight approaches to partiality — far too many to treat with any rigour in one paper. The eight approaches divide into three closely related groups, so we have chosen to explore here the three groups under hypotheses that eliminate the other differences.

We conclude this introduction with a brief look at related research.

There has been considerable work on the incorporation of nulls into implementations of standard data models. For example outer joins and their optimization for query processing are considered in [22] and [23]. More theoretical treatments involving extensions to the relational model include [19] which introduces the "probabilistic relational model" and [33] which extends the relational model to incorporate "maybe information". Probably the most theoretical treatment of partial information and its incorporation into classical data models is given by Date. He is also the most prolific author in this field [17], [5], [13], [14], [15] and [16], and he is generally scathing about the theoretical foundations so far provided for partial information.

Apart from the authors' own work there has been considerable use of sketches to support data modelling. Piessens and Steegmans developed a notion of data specification including sketches. They have obtained results on the algorithmic determination of equivalences of model categories [34] and [35] which were intended to support plans for view integration. Diskin and Cadish have used sketches for a variety of modelling purposes including for example [20] and [21]. They have concentrated on developing the diagrammatic language of "diagram operations". Several others, including Lippe and ter Hofstede [32], Islam and Phoa [25], Tuijn and Gyssens [38], Rosebrugh and Wood [36] and Baclawski et al. [2], have been applying category theory to data modelling. None of this work has so far considered modelling partial information and its interaction with sketches.

The remainder of the paper is organised as follows. In Section 2 we briefly review the definitions required by the sketch data model and refer readers if required to other papers where a more gentle introduction of the model along with appropriate motivation is provided. In Section 3 we indicate the general nature of the three approaches and make precise the hypotheses we put upon $\mathcal{R}$. In Sections 4, 5 and 6 we develop the three approaches in detail, and prove the main results, the two Morita equivalence theorems. Finally Section 7 considers the resulting query languages and Section 8 concludes.

2 Background

A review of the basic definitions follows. General introductions to the sketch data model can be found in for example [27], [28] and [30]. The basic idea is to use the sketches of categorical universal algebra as a database specification tool. The sketches are considerably more powerful than standard entity relationship (ER) models [4] [24] supporting as they do constraint information [26] that would normally be recorded outside of the standard ER framework. Nevertheless, they are sufficiently like ER models that practitioners can work with them, and they have already been valuable in large scale consultancies [6], [7], [8], [10], [37] and [12]. The extra constraint information has proved valuable theoretically too, leading to a new treatment of the view update problem [11] and to new techniques for database interoperability [29] and [9].

For background material on the theory of sketches the reader can consult [3] or [1].

Definition 2.1 *A* cone $C = (v_C, I_C : B_C \longrightarrow G, \langle p_b \rangle_{b \in (B_C)_0})$ *in a graph G consists of a node v_C of G (the* vertex *of C), a graph morphism $I_C : B_C \longrightarrow G$ (the* base diagram *of C), and, for each node b in B_C, an edge $p_b : v_C \longrightarrow I_C b$.* Cocones *are dual (that is we reverse all the edges of G which occur in the definition, so the new definition* *is the same except that the last phrase requires edges $j_b : I_C b \longrightarrow v_C$). The edges p_b in a cone (respectively cocone) are called* projections *(respectively* coprojections *or* injections*).*

Definition 2.2 *A* sketch $\mathbb{E} = (G, \mathbf{D}, L, C)$ *consists of a directed graph G, a set* $\mathbf{D}$ *of pairs of directed paths in G with common source and target (called the commutative diagrams) and sets of cones* L *and cocones* C *in G. The category generated by the graph G modulo the commutative diagrams in* $\mathbf{D}$ *is denoted* $C(\mathbb{E})$*.*

Definition 2.3 *Let* $\mathbb{E} = (G, \mathbf{D}, L, C)$ *and* $\mathbb{E}' = (G', \mathbf{D}', L', C')$ *be sketches. A* sketch morphism $H : \mathbb{E} \longrightarrow \mathbb{E}'$ *is a graph morphism* $H : G \longrightarrow G'$ *which carries, by composition, commutative diagrams in* $\mathbf{D}$*, cones in* L *and cocones in* C *to respectively commutative diagrams in* $\mathbf{D}'$*, cones in* L' *and cocones in* C'*.*

Definition 2.4 *A* model *M of a sketch* $\mathbb{E}$ *in a category* $\mathbf{S}$ *is a graph morphism from G to the underlying graph of the category* $\mathbf{S}$ *such that the images of pairs of paths in* $\mathbf{D}$ *have equal composites in* $\mathbf{S}$ *and cones (respectively cocones) in* L *(respectively in* C*) have images which are limit cones (respectively colimit cocones) in* $\mathbf{S}$*.*

That is, a model is precisely a sketch morphism from $\mathbb{E}$ to the underlying sketch of the category $\mathbf{S}$. Equivalently, we can also express models in terms of functors as follows. A model of $\mathbb{E}$ in $\mathbf{S}$ is exactly the same thing as a functor $M : C(\mathbb{E}) \longrightarrow \mathbf{S}$ such that the cones and cocones in L and C are sent to limit cones and colimit cocones in $\mathbf{S}$. Thus, the models are some of the objects of the category $[C(\mathbb{E}), \mathbf{S}]$ of functors from $C(\mathbb{E})$ to $\mathbf{S}$.

It is important to note that the category $\mathbf{S}$ need not satisfy any particular exactness conditions, though lack of exactness will clearly reduce the number of potential models.

Definition 2.5 *If M and M′ are models of* $\mathbb{E}$ *(viewed as functors) a* homomorphism $\phi : M \longrightarrow M'$ *is a natural transformation from M to* M'*.*

Models and their homomorphisms form the category of models of $\mathbb{E}$ in $\mathbf{S}$ denoted by $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$, namely the full subcategory of the functor category $[\mathbb{C}, \mathbf{S}]$ determined by those functors which are indeed models. Frequently we will write simply $\mathrm{Mod}(\mathbb{E})$ when their is unlikely to be confusion about the identity of $\mathbf{S}$.

The following defines the class of sketches which we use for the sketch data model. We generally work in model categories $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$ where $\mathbf{S}$ is a lextensive category, that is $\mathbf{S}$ has finite limits and disjoint universal finite sums.

Definition 2.6 *An* EA sketch $\mathbb{E} = (G, \mathbf{D}, L, C)$ *is a (finite limit, finite coproduct) sketch such that*

- *There is a specified cone with empty base in* L*. Its vertex will be called* 1*. Arrows with domain* 1 *are called* elements*.*
- *Nodes which are vertices of cocones all of whose injections are elements are called* attributes*.*
- *The graph of G is finite.*

Nodes which are neither attributes, nor 1*, are called* entities.

We say that the EA sketch is *keyed* if for each entity E there is a specified attribute A_E called its *key attribute* and a chosen monic specification $k_E : E \rightarrowtail A_E$ from the entity to the specified attribute. This is essentially the requirement of *entity integrity*, for it means that there is a chosen primary key. Notice that such primary keys cannot be composite since they are monic specifications to a single attribute. Nevertheless, there is nothing in the model which forbids other possibly composite candidate keys — they would be specified by a monic specification to a product of attributes.

3 Approaches to incomplete information

We propose three ways to consider incomplete information, or null values in attributes, in the sketch data model. Note that the semantics of the sketch data model requires that the value of an entity in any model can never be `null`. Moreover, the value of any arrow between entities in any model can also never be only partially defined.

Suppose that we are given an EA sketch $\mathbb{E}$ and we want to add a treatment for nulls in $\mathrm{Mod}(\mathbb{E}, \mathbf{S})$. The question arising is how to change the sketch $\mathbb{E}$, the modelling category $\mathbf{S}$, or possibly the notion of model in order to allow for unknown values at specified attributes of specified entities.

In the first two approaches there is no change to the entities in the sketch. In the first technique, only attribute cocones are modified, essentially by adding a `null` value to the attribute. Thus we change the EA sketch to explicitly include `null` values wherever they are to be permitted.

In the second technique, we change the modelling category, normally taken to be $\mathbf{S} = \mathbf{set}_0$, the category of finite sets. The idea here is to keep the EA sketch the same, but to allow it to take values in a category of "lifted sets" — sets which already incorporate a special value $\perp$ which will stand for `null`. The change of the model category in this case necessitates a change of the notion of model because lifted sets do not form a lextensive category. So we introduce a new notion called here R p-model. This approach may be extended to allow more general ordered sets (as used in domain theory) to model complex partial information, but the treatment of R p-models is delicate.

The third approach involves considerable modification of the EA sketch, including the introduction of new entities, though the modelling category $\mathbf{S}$ and the definition of model do not change. In the third approach each attribute arrow $E \longrightarrow A$ which is allowed to be `null` is replaced by a new entity E' and a span of arrows $E \longleftarrow E' \longrightarrow A$. In addition we require E' to be complemented, so we add another entity E'' and a coproduct specification that ensures that in a model M, $ME = ME' + ME''$ with the injection $ME' \longrightarrow ME$ given by the image of the added arrow $E' \longrightarrow E$. Incidentally, this implies that $ME' \longrightarrow ME$ is mono, and the idea here is that ME' is the subobject of ME for which the attribute is fully defined (non-`null`).

In all three cases we will suppose that the attribute arrows for which null values are permitted are given in a set R. Throughout this paper we will assume that arrows in R are arrows from a sketch $\mathbb{E} = (G, \mathbf{D}, L, C)$ with domain an entity and codomain an attribute such that

- no arrow in R occurs in a diagram in $\mathbf{D}$
- no arrow in R occurs in a cone in L
- no arrow in R occurs in a cocone in C
- the codomain (attribute) of each arrow in R is not the domain of any arrow of G.

Such a set R is called $\mathbb{E}$-independent.

Of course, if $\mathbb{E}$ is keyed, we will expect key attributes to never take null values, so key attribute arrows cannot occur in R.

4 Attributes with `null`

Suppose given a sketch $\mathbb{E}$ and an $\mathbb{E}$-independent set R.

The apparently simplest approach is to suppose that each attribute A which occurs in the codomain of an arrow of R has added to it a specified element called `null`. However, since the same attribute A might occur as the codomain of an arrow in R and another arrow not in R we will modify the EA sketch $\mathbb{E}$ by adding for each $f : E \longrightarrow A \in R$ a new attribute A_f whose elements are those of A and an extra specified element called $\mathtt{null}_f$, and the arrow $f : E \longrightarrow A$ will be replaced by an arrow, called $f^+ : E \longrightarrow A_f$. We denote the result of this change of $\mathbb{E}$ by $\mathbb{E}^+_R$.

More precisely,

Definition 4.1 *Let $\mathbb{E}$ be an EA sketch and R an $\mathbb{E}$-independent set. Define $\mathbb{E}^+_R = (G^+, \mathbf{D}^+, L^+, C^+)$ as follows:*

- *the nodes of G^+ are the nodes of G together with, for each arrow $E \xrightarrow{f} A$ in R, a new node A_f*
- *the edges of G^+ are the edges of G* not *in R together with, for each $E \xrightarrow{f} A$ in R, a new edge $E \xrightarrow{f^+} A_f$, and for each new node A_f, edges $A \xrightarrow{i_{A_f}} A_f$ and $1 \xrightarrow{\mathtt{null}_f} A_f$*
- $\mathbf{D}^+ = \mathbf{D}$
- $L^+ = L$
- *C^+ is the union of C and for each arrow $E \xrightarrow{f} A$ in R, a new cocone*

$$\begin{tikzcd} A \arrow[r, "i_f"] & A_f & 1 \arrow[l, "\mathtt{null}_f"'] \end{tikzcd}$$

Notice that we have made no change to the sketch data model methodology as described elsewhere. Indeed the sketch $\mathbb{E}^+_R$ is just an EA sketch. What we are doing here amounts to the "special values" idea described by Date [18], Chapter 18.

Notice an advantage of this first approach: We could specify more than one type of incomplete information simply by analogously adding more than one `null` value to an attribute. For example, the `Phone` attribute of a `Person` entity may be unknown because that information is not yet available, or because the person refuses to provide the information. Such a distinction could be encoded with this first approach by adding elements that indicate the type of incomplete information.

5 The lift monad and sketch data models

A commonly used construction for dealing with partially defined functions is the *lift monad*. The natural domain of this monad is the (2-)category of ordered objects and order-preserving arrows. When the base category is **set**, to an ordered set X, the lift monad construction assigns the ordered set $X_\perp$ whose elements are those of X together with a new bottom element $\perp$ satisfying $\perp \leq x$ for every element x of X. The (order-preserving) inclusion of X in $X_\perp$ and the collapse of two bottom elements provide a monad structure on $(-)_\perp : \mathbf{ord} \longrightarrow \mathbf{ord}$ whose algebras $\mathbf{ord}^\perp$ are ordered sets with a bottom element. Morphisms preserve the bottom element.

In applications in computer science, the interpretation of $\perp$ is often "undefined". With this in mind we call an arrow in $\mathbf{ord}^\perp$ *fully defined* if the inverse image of $\perp$ is $\perp$, that is, no non-bottom element goes to the bottom. When the base category is a topos other than **set**, it is appropriate to consider the "partial map classifier" and the resulting "constructive lift monad" studied by Kock [31].

When restricted to the discrete order on a set X, $X_\perp$ may be thought of as a "flat" order with a bottom element adjoined. We denote the category of such orders by $\mathbf{set}^\perp$ and call its objects "lifted sets". Note that $\mathbf{set}^\perp$ is the full subcategory determined by objects in the image of the functor

$$\mathbf{set} \longrightarrow \mathbf{ord} \longrightarrow \mathbf{ord}^\perp$$

where the value of the functor from **set** to **ord** at a set X is X with the discrete order. We denote this image by $L : \mathbf{set} \longrightarrow \mathbf{set}^\perp$. An arrow in $\mathbf{set}^\perp$ is fully defined if and only if it is Lf for some arrow $f \in \mathbf{set}$. We denote by $V : \mathbf{set}^\perp \longrightarrow \mathbf{set}$ the functor whose value at a lifted set X is the set of elements of X (including $\perp$). The set of non-bottom elements of a lifted set X is denoted ϕX. For a fully defined arrow $f : X \longrightarrow Y$ of $\mathbf{set}^\perp$ we denote by ϕf the restriction of Vf to ϕX so $\phi f : \phi X \longrightarrow \phi Y$. Thus ϕ is functorial and colimit preserving on the image of L in $\mathbf{set}_0^\perp$ and inverse to L there.

Write $\mathbf{set}_0^\perp$ for the full subcategory of $\mathbf{set}^\perp$ determined by $\mathbf{set}_0$. Our second approach to nulls envisages taking certain "models" in $\mathbf{S} = \mathbf{set}_0^\perp$. Some care is required here as $\mathbf{set}_0^\perp$ is *not* a lextensive category. Indeed, there is a zero object $0 = L\emptyset$, so any sum of terminal objects in $\mathbf{set}_0^\perp$ is 0. However $\mathbf{set}_0^\perp$ does have sums. They are obtained by summing the "non-bottom" elements and identifying bottoms, so any finite lifted set is isomorphic to a sum of the object $L1$.

As above, we want nulls to have no effect on entities in models. Let $C = (v_C, I_C : B_C \longrightarrow G)$ be a cone in an EA sketch $\mathbb{E}$ and let M be a functor $C(\mathbb{E}) \longrightarrow \mathbf{set}_0^{\perp}$. Then MI_C is a diagram $B_C \longrightarrow \mathbf{set}_0^{\perp}$. If all the arrows $MI_C f : MI_C B \longrightarrow MI_C B'$ are fully defined then we denote by $\xi(C,M)$ the limit of the diagram ϕMI_C in $\mathbf{set}_0$ (including the projections). Now we make the following definition.

Definition 5.1 *Let $\mathbb{E} = (G, \mathbf{D}, \mathcal{L}, \mathcal{C})$ be an EA sketch and $\mathcal{R}$ an $\mathbb{E}$-independent set. An $\mathcal{R}$* p-model *M of $\mathbb{E}$ is a graph morphism from G to the underlying graph of the category $\mathbf{set}_0^{\perp}$ such that*

$\mathcal{R}$ p-1 the images of pairs of paths in $\mathbf{D}$ have equal composites in $\mathbf{set}_0^{\perp}$;

$\mathcal{R}$ p-2 the image of any non-$\mathcal{R}$ arrow is fully defined;

$\mathcal{R}$ p-3 the image $M1$ of the vertex 1 of the empty cone is $L1$;

$\mathcal{R}$ p-4 each cone C in $\mathcal{L}$ has image $L\xi(C,M)$;

$\mathcal{R}$ p-5 finite sum cocones in $\mathcal{C}$ have images which are finite sum cocones in $\mathbf{set}_0^{\perp}$.

The idea here is that, as noted in Section 3, the value of an entity to attribute arrow in $\mathcal{R}$ may be null (or undefined), but other arrows and exactness requirements among them should be fully defined. By the first item, $\mathcal{R}$ p-models are functors $M : C(\mathbb{E}) \longrightarrow \mathbf{set}_0^{\perp}$, and by the third and fourth items they are *not* models of $\mathbb{E}$ in $\mathbf{set}_0^{\perp}$. For example, the lift of 1 is not the terminal object in $\mathbf{set}_0^{\perp}$, and for sets A, B the product of $L(A)$ and $L(B)$ in $\mathbf{set}_0^{\perp}$ is *not* generally $L(A \times B)$.

Definition 5.2 *A* homomorphism of $\mathcal{R}$ p-models *is a natural transformation, all of whose components are fully defined. We denote the category of $\mathcal{R}$ p-models of an EA sketch $\mathbb{E}$ by $\mathcal{R}$ p-*Mod$(\mathbb{E})$.

Notice the requirement that the components be fully defined.

We show that in appropriate circumstances the first two approaches yield the same models.

Theorem 5.3 *Let $\mathbb{E}$ be an EA sketch and let $\mathcal{R}$ be an $\mathbb{E}$-independent set. There is an equivalence of categories*

$$\mathrm{Mod}(\mathbb{E}_{\mathcal{R}}^{+}) \simeq \mathcal{R}\, p\text{-}\mathrm{Mod}(\mathbb{E})$$

Proof. We begin by defining a functor $\Psi : \mathrm{Mod}(\mathbb{E}_{\mathcal{R}}^{+}) \longrightarrow \mathcal{R}$ p-Mod$(\mathbb{E})$. Let M be in Mod$(\mathbb{E}_{\mathcal{R}}^{+})$. On nodes we set $(\Psi M)(X) = L(M(X))$. On edges f not in $\mathcal{R}$ we set $\Psi M(f) = LM(f)$. On edges $f : E \longrightarrow A$ in $\mathcal{R}$ we set

$$\Psi M(f)(x) = \begin{cases} \perp & \text{if } x = \perp \\ LM(f^+)(x) & \text{if } x \in M(E) \text{ and } M(f^+)(x) \neq \mathtt{null}_f \\ \perp & \text{if } x \in M(E) \text{ and } M(f^+)(x) = \mathtt{null}_f \end{cases}$$

We need to show that ΨM is functorial, that it is in $\mathcal{R}$ p-Mod$(\mathbb{E})$ and that Ψ is functorial.

To see that ΨM is functorial we need to show that it respects the commutative diagrams. Because L is functorial this is clear for commutative diagrams involving only non-R arrows. This follows by functoriality of L and M, since the assumptions on R ensure that only non-R arrows occur in diagrams of $\mathbf{D}$.

To see that ΨM is in R p-Mod($\mathbb{E}$), note that R p-1 has just been shown and R p-2 and R p-3 follow by the definition of ΨM. R p-5 follows by the definition of ΨM and the fact that L preserves sums. Every cone C in L appears unchanged in $\mathbb{E}_R^+$, so M carries them to limit cones in $\mathbf{set}_0$. Furthermore, because each cone C contains no R arrows, the definition of ΨM ensures that ΨM carries it to $L\xi(C,M)$. Thus ΨM is in R p-Mod($\mathbb{E}$).

Next suppose $\alpha : M \longrightarrow N$ is a morphism of Mod($\mathbb{E}_R^+$). We define the natural transformation $\Psi\alpha$ by $\Psi\alpha_X = L\alpha_X$. To check for naturality, suppose f is an arrow of $\mathbb{E}$. If f is not in R then naturality is immediate since L is a functor. If on the other hand $f : E \longrightarrow A$ is an element of R then consider

$$\begin{tikzcd} LME \arrow[r, "L\alpha_E"] \arrow[d, "(\Psi M)f"'] & LNE \arrow[d, "(\Psi N)f"] \\ LMA \arrow[r, "L\alpha_A"'] & LNA \end{tikzcd}$$

and check for commutativity under each of the three cases that appear in the definition of $\Psi M(f)$. Let $x \in LME$. If $x = \perp$ then the square commutes since $\Psi M(f)$, $\Psi N(f)$, $L\alpha_E$ and $L\alpha_A$ all preserve $\perp$. Similarly, in the third case $M(f^+)(x) = M\mathtt{null}_f$ and the naturality of α ensures that $N(f^+)(\alpha_E(x)) = N\mathtt{null}_f$ and again $\perp$ is preserved. For the remaining (second) case of the definition $\Psi M(f)(x) = LM(f^+)(x)$, and naturality follows since L is functorial. Finally, note that all components are fully defined and so $\Psi\alpha$ is indeed a morphism in R p-Mod($\mathbb{E}$) and Ψ preserves composition of natural transformations, so it is a functor.

Next we define $\Phi : R$ p-Mod($\mathbb{E}$) $\longrightarrow$ Mod($\mathbb{E}_R^+$). Let N be in R p-Mod($\mathbb{E}$). We need to define a model $\Phi N : \mathbb{E}_R^+ \longrightarrow \mathbf{set}_0$. On nodes we set

$$(\Phi N)(X) = \begin{cases} \phi N(X) & \text{if } X \text{ is a node in } \mathbb{E} \\ VNA & \text{if } X = A_f \text{ for some } f \text{ in } R \end{cases}$$

On edges f we set

$$(\Phi N)(f) = \begin{cases} \phi N f & \text{if } f \text{ is non } R \\ \phi NE \hookrightarrow VNE \xrightarrow{VNf} VNA & \text{if } f = g^+ : E \longrightarrow A_g \text{ for } g \text{ in } R \\ \phi NA \hookrightarrow VNA & \text{if } f = i_g \text{ for } g \text{ in } R \\ 1 \xrightarrow{\perp} VNA & \text{if } f = \mathtt{null}_g \text{ for } g \text{ in } R \end{cases}$$

We need to show that ΦN is functorial, that it is a model of $\mathbb{E}_R^+$ and that Φ is functorial.

Commutative diagrams in $\mathbf{D}^+$ are preserved by ΦN since only non R arrows are involved, N is functorial and ϕ takes a fully defined commutative diagram in

$\mathbf{set}_0^{\perp}$ to a commutative diagram of functions in $\mathbf{set}_0$. Cones are preserved because none of their arrows are from R, condition R p-4 ensures that N sends them to L of a limit cone in $\mathbf{set}_0$ and ΦN is defined by applying ϕ which returns them to the limit cones in $\mathbf{set}_0$. Similarly, N sends cocones in $\mathbb{E}$ to sum diagrams in $\mathbf{set}_0^{\perp}$ and by the construction of sums in $\mathbf{set}_0^{\perp}$, ϕ carries them to sum diagrams in $\mathbf{set}_0$. Furthermore, the definition of ΦN on their injections guarantees that the cocones $A \xrightarrow{i_f} A_f \xleftarrow{\mathtt{null}_f} 1$ are preserved. Thus ΦN is a model of $\mathbb{E}_R^+$.

Next suppose $\alpha : M \longrightarrow N$ is a morphism of R p-Mod$(\mathbb{E})$. We define the natural transformation $\Phi\alpha$ by

$$\Phi\alpha_X = \begin{cases} \phi\alpha_X & \text{if } X \text{ is a node of } \mathbb{E} \\ V\alpha A & \text{if } X = A_f \text{ for } f \text{ in } R \end{cases}$$

Note that the first case is well-defined by the fully-definedness condition on morphisms of R p-Mod$(\mathbb{E})$. To check for naturality, suppose f is an arrow of $\mathbb{E}_R^+$. If f is in $\mathbb{E}$ then naturality is immediate since the naturality square is ϕ applied to a naturality square for α. If $f = g^+$ for $g : E \longrightarrow A$ in R, consider

$$\begin{tikzcd}
\phi ME \arrow[r, "\phi\alpha_E"] \arrow[d, hook] & \phi NE \arrow[d, hook] \\
VME \arrow[r, "V\alpha_E"] \arrow[d, "VMg"'] & VNE \arrow[d, "VNg"] \\
VMA \arrow[r, "V\alpha_A"'] & VNA
\end{tikzcd}$$

Both squares in the diagram clearly commute and the vertical composites define ΦMg and ΦNg. Finally, since α_A is fully defined and being an arrow in $\mathbf{set}_0^{\perp}$ preserves $\perp$, the naturality squares for i_g and $\mathtt{null}_g$ commute for g in R. Clearly Φ preserves composition of natural transformations, so it is a functor.

Finally, we show that $\Psi\Phi(N) \cong N$ and $\Phi\Psi(M) \cong M$. On nodes both are straightforward from the definitions of Ψ and Φ. Indeed, for nodes of $\mathbb{E}$ Ψ adds a bottom element to the value while Φ strips this away. Similarly on R attributes, null elements are exchanged for bottom elements. Extending this to arrows is straightforward. □

It might seem that Definition 5.1 is rather *ad hoc*, so we provide an alternative definition as follows. The new definition is motivated by the fact that the fully-definedness of the components of morphisms of R p-models suggests that they might be viewed as $\mathbf{set}_0$ valued models of some subsketch.

We denote by $\mathbb{E}$-R the EA sketch resulting when the arrows in R are deleted from $\mathbb{E}$, and by J the inclusion sketch morphism $\mathbb{E}\text{-}R \xrightarrow{J} \mathbb{E}$.

Definition 5.4 *An R-partial model of $\mathbb{E}$ is a functor $M : C(\mathbb{E}) \longrightarrow \mathbf{set}_0^{\perp}$ such that*

$$C(\mathbb{E}\text{-}R) \xrightarrow{J} C(\mathbb{E}) \xrightarrow{M} \mathbf{set}_0^{\perp}$$

factors through L as a model of $\mathbb{E}$-R in $\mathbf{set}_0$.

Proposition 5.5 *If R is $\mathbb{E}$-independent then a functor $M : C(\mathbb{E}) \longrightarrow \mathbf{set}_0$ is an R p-model if and only if it is an R -partial model.*

Proof. Suppose M is an R -partial model. Since MJ factors through L as

$$C(\mathbb{E}\text{-}R) \xrightarrow{\overline{M}} \mathbf{set}_0 \xrightarrow{L} \mathbf{set}_0^{\perp}$$

say, and L preserves finite sums and finite connected limits we see that for M:

R p-1 holds since M is a functor and there are no R arrows in $\mathbf{D}$

R p-2 holds by the factorization using $\overline{M}$. Indeed M of any non-R arrow is fully defined, being L of a $\mathbf{set}_0$ arrow

R p-3 holds since $M1 = MJ1 = L\overline{M}1 = L1$, the last equality because $\overline{M}$ is a model and the first since J is a morphism of sketches

R p-4 For any cone in L, R p-4 holds trivially using that $\overline{M}$ is a model.

R p-5 holds since L preserves sums (being a left adjoint).

In the other direction, let M be an R p-model and γ be a non-R arrow in G. Then by R p-2, $M\gamma = MJ\gamma = Lf$ where f is some $\mathbf{set}_0$ arrow, and define $\overline{M}\gamma = f$. For nodes X of G define $\overline{M}X$ to be the unique K in $\mathbf{set}_0$ such that $LK = MX$. Thus $MJX = MX = L\overline{M}X$. Notice that L is bijective on objects and faithful since $\mathbf{set}_0^{\perp}$ is defined to be the free $(-)+1$ algebras. Thus $MJ = L\overline{M}$ and it remains to show that $\overline{M}$ is a model. That $\overline{M}$ preserves commutative diagrams and sums follows since L is faithful and preserves sums. Preservation of limit cones follows by R p-3 and R p-4. □

In conclusion we remark at this point that while adding bottom values at first seems an attractive addition, the modification to the notion of model that is introduced in order to provide a reasonable semantics is undesirable. Moreover, there is, for the basic case of merely lifted sets, no change in the expressive capacity of the models from that obtained by simply adding null elements to attributes.

Recall that the query language for a sketch data model $\mathbb{E}$ is the classifying category, usually denoted $Q(\mathbb{E})$. Notice that the query language $Q(\mathbb{E})$ obtained from $\mathbb{E}$ is all that we naturally have available in the case of lifted set models, but $Q(\mathbb{E})$ *need not have the universal property with respect to R p-models that it has as a classifying category with respect to models,* and consequently there may be no canonical evaluation of a query in $Q(\mathbb{E})$ for an R p-model M.

6 Implementing partial arrows in the sketch data model

Our third approach requires more serious modification of the basic EA sketch. Like the first approach it does not require variation of the notion of model. It shares with the lifted set approach the idea that nulls are missing information rather than special values.

The idea here is that entity to attribute arrows with missing information should be implemented as partial arrows. Recall that a partial arrow from X to Y is a pair $\langle i, f\rangle$ where $i : X_0 \longrightarrow X$ is a part of X (a monic arrow with codomain X), and $f : X_0 \longrightarrow Y$ is arbitrary.

An appropriate construction for a sketch that implements this idea follows:

Definition 6.1 *Let $\mathbb{E}$ be an EA sketch and R a set of $\mathbb{E}$-independent arrows. Define $\mathbb{E}_R^{\sim} = (\widetilde{G}, \widetilde{\mathbf{D}}, \widetilde{L}, \widetilde{C})$ as follows:*

- *the nodes of $\widetilde{G}$ are the nodes of G and for each $E \xrightarrow{f} A$ in R, new nodes E_f and $\overline{E}_f$*
- *the edges of $\widetilde{G}$ are the edges of G not in R, and for each $E \xrightarrow{f} A$ in R, new edges $E_f \xrightarrow{m_f} A$, $E_f \xrightarrow{i_f} E$ and $\overline{E}_f \xrightarrow{j_f} E$*
- $\widetilde{\mathbf{D}} = \mathbf{D}$
- $\widetilde{L} = L$
- *$\widetilde{C}$ is the union of C and for each $E \xrightarrow{f} A$ in R, a new cocone*

$$E_f \xrightarrow{i_f} E \xleftarrow{j_f} \overline{E}_f$$

Theorem 6.2 *Let $\mathbb{E}$ be an EA sketch. Let R be an $\mathbb{E}$-independent set. There is an equivalence of categories:*

$$\mathrm{Mod}(\mathbb{E}_R^{+}) \simeq \mathrm{Mod}(\mathbb{E}_R^{\sim})$$

Proof. We begin by defining a functor $\Psi : \mathrm{Mod}(\mathbb{E}_R^{+}) \longrightarrow \mathrm{Mod}(\mathbb{E}_R^{\sim})$. Let M be in $\mathrm{Mod}(\mathbb{E}_R^{+})$. On nodes we define $(\Psi M)X = MX$ if X is a node of $\mathbb{E}$, and if $X = E_f$ or $X = \overline{E}_f$ we define $(\Psi M)(X)$ so that both squares in the following diagram are pullbacks, and thus the rows are sum diagrams:

$$\begin{tikzcd}
\Psi M(E_f) \arrow[r, "\Psi M(i_f)"] \arrow[d, "\Psi M(m_f)"'] & M(E) \arrow[d, "M(f^+)"] & \Psi M(\overline{E}_f) \arrow[l, "\Psi M(j_f)"'] \arrow[d] \\
MA \arrow[r, "M(i_{A_f})"'] & M(A_f) & M1 \arrow[l, "M(null_A)"]
\end{tikzcd}$$

On edges f of G that are not in R we define $\Psi M(f)$ to be $M(f)$. On the edges $E_f \xrightarrow{m_f} A$, $E_f \xrightarrow{i_f} E$ and $\overline{E}_f \xrightarrow{j_f} E$ the effect of ΨM is defined by the diagram above.

We need to show that ΨM is functorial, that it is an $\mathbb{E}_R^{\sim}$-model and that Ψ is functorial.

To see that ΨM is functorial we need to show that it respects the commutative diagrams of $\mathbb{E}_R^{\sim}$. Because M is functorial this is clear for commutative diagrams involving only non-R arrows, but the assumptions on R and the construction of $\mathbb{E}_R^{\sim}$ ensure that only such diagrams occur in $\mathbb{E}_R^{\sim}$.

Trivially ΨM send cones in $\widetilde{L}$ to limit cones (after all the cones are the same in all three of $\mathbb{E}, \mathbb{E}_R^{+}$ and $\mathbb{E}_R^{\sim}$). Similarly for cocones in C. The extra cocones in $\widetilde{C}$ are sent to sum diagrams in $\mathbf{set}_0$ as noted above. Thus ΨM is a model.

Next suppose $\alpha : M \longrightarrow N$ is a morphism of $\mathrm{Mod}(\mathbb{E}_R^{+})$. We define the natural transformation $\Psi\alpha$ on nodes of $\mathbb{E}$ by $\Psi\alpha_X = \alpha_X$. For the nodes E_f, $\Psi\alpha_{E_f}$ is defined as shown in the diagram, noting that the front face is a pullback and that the commutativity of the back, right and bottom faces ensures that the outside of the diagram commutes.

$$\begin{tikzcd}
(\Psi M)E_f \arrow[rr, "(\Psi M)i_f"] \arrow[dr, dashed, "(\Psi\alpha)_{E_f}"] \arrow[dd, "(\Psi M)m_f"'] & & ME \arrow[dr, "\alpha_E"] \arrow[dd, "Mf^+" near start] & \\
 & (\Psi N)E_f \arrow[rr, "(\Psi N)i_f" near start] \arrow[dd, "(\Psi N)m_f" near end] & & NE \arrow[dd, "Nf^+"] \\
MA \arrow[rr, "M(i_{A_f})" near start] \arrow[dr, "\alpha_A"'] & & MA_f \arrow[dr, "\alpha_{A_f}"] & \\
 & NA \arrow[rr, "Ni_{A_f}"'] & & NA_f
\end{tikzcd}$$

For the nodes $\overline{E}_f$ a similar argument defines $\Psi\alpha_{\overline{E}_f}$.

These definitions make $\Psi\alpha$ a natural transformation since, for arrows in $\mathbb{E}$, α is already natural, while for the arrows m_f and i_f the naturality squares are the left and top faces of the diagram above. Similarly for j_f. Furthermore, Ψ preserves composition of natural transformations, so it is a functor.

Next we define $\Phi : \mathrm{Mod}(\mathbb{E}_R^{\sim}) \longrightarrow \mathrm{Mod}(\mathbb{E}_R^{+})$. Let N be in $\mathrm{Mod}(\mathbb{E}_R^{\sim})$. On nodes we set

$$(\Phi N)(X) = \begin{cases} N(X) & \text{if } X \text{ is a node of } \mathbb{E} \\ N(A)+1 & \text{if } X = A_f \text{ is a new attribute} \end{cases}$$

On edges f of G that are not in R we define $\Phi N(f)$ to be $N(f)$. Let the sum in the second case be $\Phi N(A) \xrightarrow{\Phi N(i_{A_f})} \Phi N(A_f) \xleftarrow{\Phi N(\mathtt{null}_A)} \Phi N(1)$. If the edge f is in R, we use the fact that $N(E_f) \xrightarrow{N(i_f)} N(E) \xleftarrow{N(j_f)} N(\overline{E}_f)$ is a sum so that the arrows $N(E_f) \xrightarrow{N(m_f)} N(A) = \Phi N(A) \xrightarrow{\Phi N(i_{A_f})} \Phi N(A_f)$ and $N(\overline{E}_f) \longrightarrow 1 = \Phi N(1) \xrightarrow{\Phi N(\mathtt{null}_A)} \Phi N(A_f)$ define

$$\Phi N(f^+) : \Phi N(E) \cong N(E_f) + N(\overline{E}_f) \longrightarrow \Phi N(A_f)$$

We need to show that ΦN is functorial, that it is a model of $\mathbb{E}_R^{+}$ and that Φ is functorial.

Commutative diagrams in $\mathbf{D}^+$ are preserved by ΦN since only non R arrows are involved and N is functorial. Cones in L^+ are sent to limit cones in $\mathbf{set}_0$ because none of their arrows are from R, and N is a model. Similarly, N sends cocones in $\mathbb{E}$ to sum diagrams in $\mathbf{set}_0$. The only remaining cocones are of the form $A \xrightarrow{i_{A_f}} A_f \xleftarrow{\mathtt{null}_f} 1$ and the definition of $\Phi N(A_f)$ shows that this is sent to a sum diagram as above. Thus ΦN is a model of $\mathbb{E}^+_R$.

Next suppose $\alpha : M \longrightarrow N$ is a morphism of $\mathrm{Mod}(\mathbb{E}^{\sim}_R)$. We define the natural transformation $\Phi\alpha$ on nodes of $\mathbb{E}$ by $\Phi\alpha_X = \alpha_X$. For the nodes A_f, $\Phi\alpha_{A_f}$ is defined as shown on the bottom faces of the diagram below, that is $(\Phi\alpha)_{A_f} = \alpha_A + \alpha_1$.

$$\begin{tikzcd}
ME_f \arrow[rr, "Mi_f"] \arrow[dr, "\alpha_{E_f}"] \arrow[dd, "Mm_f"'] & & \Phi ME \arrow[dr, "(\Phi\alpha)_E"] \arrow[dd, "(\Phi M)f^+"] & & M\overline{E}_f j_f \arrow[ll, "M"'] \arrow[dr, "\alpha_{\overline{E}_f}"] \arrow[dd] & \\
& NE_f \arrow[rr, "Ni_f"] \arrow[dd, "Nm_f"] & & \Phi NE \arrow[dd, "(\Phi N)f^+"] & & N\overline{E}_f \arrow[ll, "Nj_f"'] \arrow[dd] \\
MA \arrow[rr, "(\Phi M)(i_{A_f})"] \arrow[dr, "\alpha_A"'] & & \Phi MA_f \arrow[dr, dashed, "(\Phi\alpha)_{A_f}"] & & M1 \arrow[ll, "\Phi M(\mathtt{null}_f)"'] \arrow[dr, "\alpha_1"] & \\
& NA \arrow[rr, "\Phi Ni_{A_f}"'] & & \Phi NA_f & & N1 \arrow[ll, "\Phi N(\mathtt{null}_f)"]
\end{tikzcd}$$

These definitions make $\Phi\alpha$ a natural transformation since, for arrows in $\mathbb{E}$, α is already natural, while for the arrows i_{A_f} and $\mathtt{null}_f$ the bottom squares of the diagram above show naturality, remembering that $(\Phi\alpha)_A = \alpha_A$ and $(\Phi\alpha)_1 = \alpha_1$. The only remaining arrows are of the form f^+ and the middle vertical square of the diagram is the required naturality square. It commutes because all outside faces of the diagram commute, so the two composites from ΦME to ΦNA_f are both the unique arrow determined by the coproduct in the top line of the diagram. Furthermore, Φ preserves composition of natural transformations, so it is a functor.

Finally, we need that $\Psi\Phi(N) \cong N$ and $\Phi\Psi(M) \cong M$. On nodes both are straightforward from the definitions of Ψ and Φ. Indeed, for nodes and arrows of $\mathbb{E}$ neither Φ nor Ψ makes any change. Ψ adds new entities that model null values in models with partial arrows while Φ models the partial arrows with null values. Similarly on R attributes, null elements are exchanged for partial arrow specifications. $\Box$

7 Effects on queries

As the previous sections have shown, three constructions that extend an EA sketch or its models in order to introduce incomplete information result in equivalent model categories. For example, while the sketches $\mathbb{E}^+_R$ and $\mathbb{E}^{\sim}_R$ are different, they are *Morita equivalent*. In the second approach, the case of the category of R p-models, a different notion of model is used to obtain a category equivalent to the model categories for $\mathbb{E}_R^+$ and $\mathbb{E}_R^\sim$. However, the classifying categories for $\mathbb{E}_R^+$ and $\mathbb{E}_R^\sim$ are clearly different from each other and from the classifying category for $\mathbb{E}$. As noted above, this last may fail to satisfy the classifying category property for R p-models. Hence the query languages that arise depend on the construction used and even to speak formally of the query language for the R p-models may require a new notion which we call *the lift-model classifying category*.

The detailed analysis of differences between the classifying categories will be deferred to a future paper. In the meantime we record here by way of illustration a few examples.

We begin by comparing the classifying category $Q(\mathbb{E})$ for $\mathbb{E}$ with the classifying category $Q(\mathbb{E}_R^+)$ for $\mathbb{E}_R^+$.

We consider first selection queries. The added null elements for $\mathbb{E}_R^+$ attributes mean that we can add queries which refer to null values. For example, we can ask for the `Persons` whose `phone` attribute is null. All of the nulls we have introduced are essentially typed by their attribute, so one classical difficulty with null values, that they are equal independent of type does not arise. As we mentioned earlier, augmenting attributes by additional new values corresponding to different types of unknown information is handled smoothly by $\mathbb{E}_R^+$.

The case of join queries is not so favourable. For example, `Office` entities might have a `phone` attribute and a join with the data in the previous paragraph should indicate pairs `Person, Office` where the `Person`'s `phone` number is that of the `Office`. Unfortunately in this case, there may be many spurious pairs in the join — all those `Person, Office` pairs where the `Person` and the `Office` have unknown `phone` attributes. It is, of course possible using finite sums (allowed by $Q(\mathbb{E}_R^+)$) to obtain the non-null `phones` with a query result called `phone-nn` for example, then join both `Person` and `Office` with this result, and finally join those results with each other.

Projection queries do not apparently pose any particular difficulty in this case.

The construction of $\mathbb{E}_R^+$ from $\mathbb{E}$ is done without adding any new nodes to the graph G of $\mathbb{E}$: only a new element per attribute is added. By contrast, the construction of $\mathbb{E}_R^\sim$ adds many new entities, arrows involving them and monic specifications. Thus we expect stronger effects on the expressivity of the query language. This is indeed the case.

First for selection queries and referring to our examples above. Suppose that the `Person` and `Office` entities have their `phone` attributes expressed by arrows PP and OP respectively. The construction of $\mathbb{E}_R^\sim$ adds new entities called $\texttt{Person}_{PP}$ and $\texttt{Office}_{OP}$ together with new arrows and monic specifications. When these entities are modelled they provide the `Persons` and `Offices` with known `phone` attributes.The join of the new entities over the `phone` attribute now computes the expected pairs, at least of $\texttt{Person}_{PP}$ and $\texttt{Office}_{OP}$ entities, but this result is easily seen as `Person, Office` pairs via the new monic specifications.

Fortunately the construction of $\mathbb{E}_R^\sim$ requires that entities like $\texttt{Person}_{PP}$ be complemented. Had this not been required, as is usually the case in categories of partial morphisms, there would be *no* way of querying on `nulls` at all.

8 Conclusion

The most important conclusion to draw from this work is that, unlike the relational data model, there is, as shown by the first and third approaches to partiality, no need to alter the foundations of the sketch data model in any way in order to support partiality. Notice that both $\mathbb{E}_R^+$ and $\mathbb{E}_R^\sim$ are standard sketch data models.

On the other hand, interestingly, the second approach does require substantial modification of the foundations, needing as it does a new definition of model because $\mathbf{set}_0^\perp$ is not lextensive, and the appropriate notion of classifying category for that approach remains to be developed. It is likely to be some time before a detailed treatment of sketch data models valued in ordered sets (including information systems in the sense of Scott) can be worked out. This is contrary to the expectations of a number of workers who predicted it would be a routine extension of the theory.

The most important results of the paper are the two Morita equivalence theorems. These theorems align precisely the three approaches, under the hypotheses on R, and show that they have equivalent expressive power.

It is also worth noting two smaller points of interest: the requirement that morphisms of R p-models be fully-defined was unexpected, but can be justified in retrospect, and the need to complement the subobject of domain of definition of partial functions was also unexpected. This latter could be avoided, but only by restricting the morphisms of models of the third approach by requiring them to be "cartesianly partial" by which we mean that the naturality squares involving the inclusions of the domains of definition must all be pullbacks.

References

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

[2] K. Baclawski, D. Simovici and W. White. A categorical approach to database semantics. *Mathematical Structures in Computer Science* 4, 147–183, 1994.

[3] F. Borceux. *Handbook of Categorical Algebra 3*. Cambridge University Press, 1994.

[4] P. P. -S. Chen. The Entity-Relationship Model—Toward a Unified View of Data. *ACM TODS* 2, 9–36, 1976.

[5] E. F. Codd and C. J. Date Much ado about nothing. in Date *Relational Database Writings 1991–1994,* Addison-Wesley, 1995.

[6] C. N. G. Dampney and Michael Johnson. TIME Compliant Corporate Data Model Validation. Consultants' report to Telecom Australia, 1991.

[7] C. N. G. Dampney and Michael Johnson. Fibrations and the DoH Data Model. Consultants' report to NSW Department of Health, 1999.

[8] C. N. G. Dampney and Michael Johnson. A formal method for enterprise interoperability: A case study in a major health informatics information system. *Proceedings of the Thirteenth International Conference on Software and Systems Engineering*, CNAM Paris, vol 3, 12-5, 1–6, 2000.

[9] C. N. G. Dampney and Michael Johnson. Half-duplex interoperations for cooperating information systems. In Advances in Concurrent Engineering, IN-1, 7pp, International Institute of Concurrent Engineering, ISBN 09710461-0-7, 2001.

[10] C. N. G. Dampney, Michael Johnson and G. M. McGrath. Audit and Enhancement of the Caltex Information Strategy Planning (CISP) Project. Consultants' report to Caltex Oil Australia, 1994.

[11] C. N. G. Dampney, Michael Johnson, and Robert Rosebrugh. View Updates in a Semantic Data Model Paradigm. Proceedings of the Twelfth Australasian Database Conference ADC2001, 29–36, IEEE Press, 2001.

[12] C.N.G Dampney, Graham Pegler and Michael Johnson. Harmonising Health Information Models - a critical analysis of current practice. Proceedings of the Health Informatics Conference 2001, 8pp. ISBN: 0 9585370 8 9

[13] C. J. Date. NOT is not "Not"! in Date *Relational Database Writings 1985–1989,* Addison-Wesley, 1990.

[14] C. J. Date. EXISTS is not "Exists"! (Some logical flaws in SQL). in Date *Relational Database Writings 1985–1989,* Addison-Wesley, 1990.

[15] C. J. Date. Watch out for outer join. in Date and Darwen *Relational Database Writings 1989–1991,* Addison-Wesley, 1992.

[16] C. J. Date. Oh no not Nulls again. in Date and Darwen *Relational Database Writings 1989–1991,* Addison-Wesley, 1992.

[17] C. J. Date. Faults and defaults (in five parts). in Date, Darwen and McGoveran *Relational Database Writings 1994–1997,* Addison-Wesley, 1998.

[18] C. J. Date. *Introduction to Database Systems*. Addison-Wesley, seventh edition, 2000.

[19] D. Dey and S. Sarkar. A probabilistic relational model and algebra. *ACM TODS* 21, 1996.

[20] 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.

[21] Zinovy Diskin and Boris Cadish. Variable set semantics for generalised sketches: Why ER is more object oriented than OO. In *Data and Knowledge Engineering,* 2000.

[22] C. Galindo-Legaria and A. Rosenthal. Outerjoin simplification and reordering for query optimization. *ACM TODS,* 22, 1997.

[23] P. Goel and B. Iyer. SQL query optimization reordering for a general class of queries. In *Proceedings of ADDM SIGMOD,* 1996.

[24] M. Gogolla and U. Hohenstein. Towards a semantic view of an extended entity-relationship model. *ACM TODS*, 16, 369–416, 1991.

[25] A. Islam and W. Phoa. Categorical models of relational databases I: Fibrational formulation, schema integration. Proceedings of TACS94. Eds M. Hagiya and J. C. Mitchell. *Lecture Notes in Computer Science* 789, 618–641, 1994.

[26] Michael Johnson and C. N. G. Dampney. On the value of commutative diagrams in information modelling. In Algebraic Methodology and Software Technology, *Springer Workshops in Computing*, 1994.

[27] Michael Johnson and Robert Rosebrugh. View updatability based on the models of a formal specification. *Proceedings of FME 2001*, 534–549, *Lecture Notes in Computer Science* 2021, 2001.

[28] Michael Johnson and Robert Rosebrugh. Sketch Data Models, Relational Schema and Data Specifications. *ENTCS*, Volume 61, 6, 1–13, 2002.

[29] Michael Johnson and Robert Rosebrugh. Database interoperability through state based logical data independence. *International Journal of Computer Applications in Technology*, 16, number 2-3, (2003) 97-102.

[30] Michael Johnson, Robert Rosebrugh, and R. J. Wood. Entity-relationship models and sketches. *Theory and Applications of Categories* 10(2002), 94–112.

[31] Anders Kock. Algebras for the partial map classifier monad. *Lecture Notes in Mathematics* 1488(1991), 262–278.

[32] E. Lippe and A ter Hofstede. A category theoretical approach to conceptual data modelling. *RAIRO Theoretical Informatics and Applications*, 30:31–79, 1996.

[33] K. Liu and R. Sunderraman. Indefinite and maybe information in relational databases. *ACM TODS*, 15, 1990.

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

[35] F. Piessens and Eric Steegmans. Selective Attribute Elimination for Categorical Data Specifications. Proceedings of the 6th International AMAST 424-436, *Lecture Notes in Computer Science* 1349, 1997.

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

[37] G. Southon, C. Sauer, and C. N. G. Dampney. Lessons from a failed information systems initiative: issues for complex organisations. *International Journal of Medical Informatics*, Elsevier Science, 1999.

[38] C. Tuijn and M. Gyssens. CGOOD, a categorical graph-oriented object data model. it Theoretical Computer Science 160, 217-239, 1996.