Documenting commutative diagrams of relationships to eliminate sources of redundancy in relational data design - Part Two - Logical to Physical Mathematically

John Cartmell

August 18, 2016

DRAFT

1 Introduction

According to the functional view of data, the content of a database instance can be described as a collection of sets and functions in that (i) for each entity type a there is defined in the database instance a set E_a of entities of type a; (ii) for each, possibly optional, many-one relationship $a \overset{r}{\ni\!\!-} b$ there is defined a possibly partial function $f_r : E_a \to E_b$. In this view, an instance of such a relationship r is defined to be a pair of entities e, e' such that $f_r(e) = e'$. Without loss of generality there can be assumed a single set V of all values that potentially might be held in columns of tables, such as all possible texts, numerics, booleans and so on, so that for each attribute $attr$ of entity type et there is defined in the database instance a function $f_{attr} : E_{et} \to V$.

In relational data modelling, each row of data is uniquely distinguishable from the values of a specific set of columns said to comprise the primary key to the data whereas in logical entity relationship (ER) modelling each entity is distinguishable from the values of a specific set of attributes taken in combination with a specific set of relationships with other entities[1].

From this starting position we provide a set of general definitions of *ER schema*, *ER schema instance*, and *ER model* so that from the definition of *ER model* we capture the notion of a database schema and all its envisaged usages (to a meta-mathematician the ER schema notion equates to a *theory* of some kind and an ER model to a theory and all its instances i.e. all its models[2]). We define the conditions for an ER model to be purely *logical* in the sense used in the term *logical data design* and, in contrast, the conditions for an ER model to be *physical*. The definitions are such that a *physical ER model* is pretty much the same thing as a relational database schema. We define the first-cut Chen mapping for generating a first cut physical ER model from a logical ER model and then develop this definition in a way that reduces redundancy in the generated physical model by taking account of commuting and near commuting diagrams of relationships in the logical model and thereby establish

[1] Whichever methodology is followed the goal is to achieve for the database instances the logical principal of identity of indiscernibles.

[2] This is my first and last usage of the term *model* with the meaning the term has in mathematical logic; for the remainder of this paper it will have the meaning as used in data modelling.

a revised Chen mapping $\mathcal{X}$ so that for any logical ER model $\mathcal{M}$, $\mathcal{X}(\mathcal{M})$ is a physical ER model. Finally we define what it is for a logical model to be well-formulated and prove that if $\mathcal{M}$ is a well-formulated logical ER model then the generated physical ER model $\mathcal{X}(\mathcal{M})$ is in Boyce-Codd normal form (BCNF).

2 Definition of ER model

The functional view of data summarised above taken with the requirement of specifying the attributes and relationships from which entitites may be identified suggests a mathematical definition of an ER-schema as follows:

Definition An *ER-schema* is a directed graph having the following additional structure:

(i) a distinguished node v for which there are no outgoing edges and which represents the type of all scalar values

(ii) a distinguished subset of edges representing identifying edges.

If $\mathcal{M}$ is an ER-schema (or an ER-model which, as we define below, includes an ER-schema) then the nodes of $\mathcal{M}$ other than v we say are entity types and we denote by $\mathcal{M}^E_a$, the set of edges leaving entity type a.

The set $\mathcal{M}^A_a$ of attributes of an entity type a is defined as the set of edges that have a as source and v as destination. The set $\mathcal{M}^R_a$ of outgoing relationships of an entity type a is defined as the set of edges having a as source and having destinations other than v. Therefore for all entity types a:

$$\mathcal{M}^E_a = \mathcal{M}^A_a \cup \mathcal{M}^R_a$$

That subset of outgoing relationships of a that are also in the distinguished set of identifying edges is said be the set of identifying relationships of a and is denoted $\mathcal{M}^{iR}_a$.

That subset of those attributes of a that are also in the distinguished set of identifying edges is said to be the set of identifying attributes of a and is denoted $\mathcal{M}^{iA}_a$.

The set of all outgoing identifying edges from a node a will be denoted κ_a.

So that we can define the characteristics of κ_a as a set of identifying properties for entitites of type a we need the following definition:

Definition If s is a set and if $f_{i,1\leq i\leq n}$ is a family of partial functions, $f_i : s \to s_i$ for some sets $s_{i,1\leq i\leq n}$, then we will say that the family of functions $f_{i,1\leq i\leq n}$, is *jointly invertible* if the partial function $\langle f_1, \ldots f_n\rangle : s \to s_1 \times \ldots \times s_n$ is invertible i.e. iff there is a partial function $inv_{\langle f_1,\ldots f_n\rangle} : s_1 \times \ldots \times s_n \to s$ such that (i) for all $x \in s$, $inv_{\langle f_1,\ldots f_n\rangle}(\langle f_1(x), \ldots f_2(x)\rangle) = x$ and (ii) if $y \in s_1 \times \ldots \times s_n$ and $y \notin img(\langle f_1, \ldots f_n\rangle)$ then $inv_{\langle f_1,\ldots f_n\rangle}(y)$ is undefined.

which we then use to define the notion of a database instance as follows:

Definition A *database instance* of an ER schema is a set of entities E_a for each node a of the graph of the schema and a partial function $E_r : E_a \to E_b$ for each edge of the graph $r : a \to b$ such that for each entity type a the family of functions $E_{r, r\in\mathcal{M}^{iE}_a}$, is jointly invertible.

It follows that in every database instance E, for every entity type a there is a function $inv_{E_{\kappa_a}}$ that represents navigation to an entity from an identifying set of related entities or attributes. In a physical model this will equate to keyed lookup.

Without change to the underlying concept then we can say that each ER schema comes equipped with a multi-edge I_a for every entity type a such that if the outgoing identifying edges of a are $k_i : a \to a_i$, for $1 \leq i \leq n$ then the multi-edge has source nodes $\langle a_1, ...a_n \rangle$ and destination node a.
A simple navigation path over an ER model is a sequence of n edges: $et_0 \overset{r_1}{\ni\!\!-} et_1 \overset{r_2}{\ni\!\!-} et_2 ... \overset{r_n}{\ni\!\!-} et_n$. et_0 is said to be the source of the path and et_n is said to be the destination of the path.
We extend this definition to take account of navigation along the multi-edges. To do so we define the set of navigation paths recursively:

(i) Each edge $f : a \to b$ is a navigation path.

(ii) The empty sequence $\langle\rangle : a \to a$ is a navigation path for every entity type a.

(iii) $\langle p, f \rangle : a \to c$ is a navigation path if p is a navigation path $p : a \to b$ and f is an edge $p : b \to c$

(iv) $\langle p_1, ...p_n, I_b \rangle : a \to b$ is a navigation path for all entity types b such that $I_b : \langle b_1, ...b_n \rangle$ and where for each i, $1 \leq i \leq n$, p_i is a path, $p_i : a \to b_i$.

For any database instance E we can extend the definition of E_f, for edges f, so that to every path p, $p : a \to b$, we have defined a function $E_p : E_a \to E_b$. From the initial definition of E_f that applies to edges the definition proceeds recursively as follows:

(i) For each entity type a, $E_{\langle\rangle} : E_a \to E_a$ is defined to be the identity function.

(ii) if p is a navigation path $p : a \to b$ and f is an edge $p : b \to c$ then $E_{\langle p, f \rangle}$ is is defined to be the functional composition $E_p \circ E_f$.

(iii) for all entity types b such that $I_b : \langle b_1, ...b_n \rangle \to b$ and where for each i, $1 \leq i \leq n$, p_i is a path, $p_i : a \to b_i$, $E_{\langle p_1, ...p_n, I_b \rangle}$ is defined to be $\langle E_{p1}, ...E_{p_n} \rangle \circ inv_{E_{\kappa_b}}$.

If r and s are paths both having source a and destination b then we will say $r \leq s$ iff in all instances E, for all entities $e \in E_a$, if $E_r(e)$ is defined then $E_s(e)$ is defined and $E_r(e) = E_s(e)$.

If r and s are paths both having source a and destination b then we will say $r \simeq s$ iff $r \leq s$ and $s \leq r$.

With these definitions, the (meta-relationship) $\leq$ is a partial order on the classes of equivalent paths.

For paths r and s we define $r < s$ to be equivalent to $r \leq s$ and not $r \simeq s$.

Definition An *ER model* is an ER schema and a set of database instances of the schema.

If p is a path within an ER model $\mathcal{M}$ then say that the path is *explicitly represented* wrt the model iff it is equivalent to a simple path.

We generalise the relational data model concept of a candidate key as follows:

Definition A family of paths $p_i : a \to a_i$ within a model $\mathcal{M}$ is said to be *jointly monomorphic* iff in all instances E, the family of functions $E_{p_i, 1 \le i \le n}$ is jointly invertible.

Consider that the various database normal forms (3NF, BCNF, 4NF, 5NF and the like) each prescribe that a database schema be complete in some way as a description of the facts of its instances[3] and observe in particular that BCNF can be paraphrased as saying that those relationships (i.e. functional dependencies) that exist in the data ought to be *represented* in the schema. These considerations motivate the definitions which now follow and conclude with the definition of a *well-formulated* entity model. This definition generalises that of a relational schema being in Boyce-Codd Normal Form (BCNF).

Notation If $X_1, \dots X_n$ are sets and if $J = \{i_1, \dots i_j\} \subseteq \{1, \dots n\}$ then denote by P_J the projection function :

$$P_J : X_1 \times X_2 \times \dots X_n \to X_{i_1} \times X_{i_2} \times \dots X_{i_j}$$

i.e. the function given by:

$$P_J(\langle x_1, \dots x_n \rangle) = \langle x_{i_1}, \dots x_{i_j} \rangle.$$

Definition If $\mathcal{M}$ is an entity model, if $b_1, \dots b_n$ and c are entity types of model $\mathcal{M}$ and if f_E is a family of functions such that in every instance E of $\mathcal{M}$:

$$f_E : E_{b_1} \times \dots \times E_{b_n} \to E_c$$

then

- the family of functions f_E is said to be *reducible* to a family of functions:

  $$g_E : E_{b_{i_1}} \times \dots \times E_{b_{i_j}} \to E_c$$

  for some $J = \{i_1, \dots i_j\} \subseteq \{1, \dots n\}$, iff in all instances E:

  $$f_E = P_J \circ g_E$$

- the family of functions f_E is said to be *irreducible* iff there is no proper subset $J = \{i_1, \dots i_j\} \subset \{1, \dots n\}$, and no family of functions $g_E : E_{b_{i_1}}, \dots E_{b_{i_j}} \to E_c$ such that f_E is reducible to g_E.

Definition A tuple of simple paths $\langle p_1, \dots p_n \rangle$ is said to be an *identifying tuple with respect to an entity type a* iff it is in the set of tuples defined recursively as follows:

(i) the empty tuple $\langle\rangle$ is an identifying tuple with respect to a,

(ii) if k_i, $1 \le i \le n$ is the set of all identifying outgoing edges of a then $\langle\langle k_1 \rangle, \dots \langle k_n \rangle\rangle$ is an identifying tuple with respect to a,

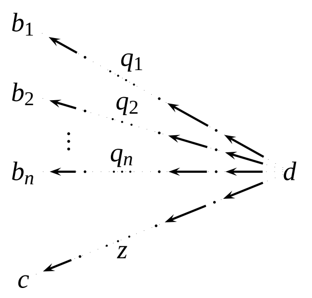
(iii) if $\langle p_1, \dots p_n \rangle$ is an identifying tuple with respect to a and if for some i, $1 \le i \le n$, the destination of p_i is b and if k_j, $1 \le j \le m$ is the set of all identifying outgoing edges of b then $\langle p_1, \dots p_{i-1}, \langle p_i, k_1 \rangle \dots \langle p_i, k_m \rangle, p_{i+1}, \dots p_n \rangle$ is an identifying tuple with respect to a.

Definition If $\mathcal{M}$ is an entity model, if a and b are entity types of $\mathcal{M}$ and if $\langle q_1, \dots q_n \rangle$ is an identifying tuple with respect to b where for each i, $q_i : b \to b_i$, if $f_i : a \to b_i$, for each i, $1 \le i \le n$, is a tuple of edges of $\mathcal{M}$ then say that $\langle f_1, \dots f_n \rangle$ *references b with respect to* $\langle q_1, \dots q_n \rangle$ iff in all instances E of $\mathcal{M}$, $img(E_{\langle f_1, \dots f_n \rangle}) \subseteq img(E_{\langle q_1, \dots q_n \rangle})$.

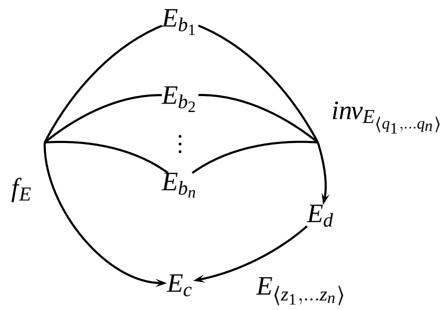
[3] Essentially because being good as a schema is to be a good theory and a good theory is one that is a good fit to the facts.

Definition If $\mathcal{M}$ is an entity model and if $b_1, \ldots b_n$ and c are entity types within $\mathcal{M}$ and if f_E is a family of functions such that for each instance E of $\mathcal{M}$, $f_E : E_{b_1} \times \ldots \times E_{b_n} \to E_c$, then the family of functions f_E is *represented* in the ER model iff either

(i) the family f_E is irreducible and there exists an entity type d and an identifying tuple of simple paths with respect to d, $\langle q1, \ldots q_n \rangle$, such that for each $q_i : d \to b_i$ and a a simple path $z = \langle z_1, \ldots z_l \rangle$ such that $z : d \to c$, for some $l \geq 0$ as here:



where z_1 not identifying and such that in all instances E, $inv_{E_{\langle q_1, \ldots q_n \rangle}} \circ E_{\langle z_1, \ldots z_l \rangle} = f_E$



or

(ii) the family f_E is reducible to an irreducible family g_E and the family g_E is represented in the model.

Remark For any entity model $\mathcal{M}$ and for any type b of $\mathcal{M}$ the family of identity functions on entities of type b :

$$id_{E_b} : E_b \to E_b$$

is represented because we can choose both $q : b \to b$ and and $z : b \to b$ to be the empty path $\langle \rangle$; then we have:

$$\begin{aligned} inv_{E_{\langle q_1, \ldots q_n \rangle}} \circ E_{\langle z_1, \ldots z_l \rangle} &= inv_{E_{\langle \rangle}} \circ E_{\langle \rangle} \\ &= id_{E_b}^{-1} \circ id_{E_b} \\ &= id_{E_b} \end{aligned}$$

as required.

Remark For any entity model $\mathcal{M}$, for any $n \geq 1$, for any tuple of types $b_1, \ldots b_n$ and for any i, $1 \leq i \leq n$, if in any instance E of $\mathcal{M}$, p_{i_E} is the i'th projection function:

$$p_{i_E} : E_{b_1} \times \ldots \times E_{b_n} \to E_{b_i}$$

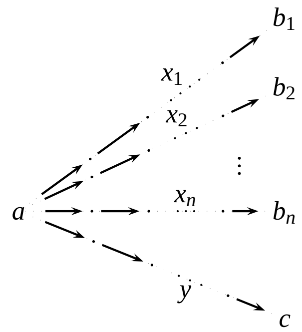
then the family of functions p_{i_E} are represented in model $\mathcal{M}$. This is because this family of functions is reducible to the family of identify functions on E_{b_i} and this family is represented as previously remarked.

Definition An ER model $\mathcal{M}$ is *well-formulated* iff

(i) for each entity type a, there is no proper subset I of the set of identifying edges $\mathcal{M}_a^{iE}$ that is jointly monomorphic

(ii) for all entity types a and for all entity types b with identifying outgoing edges $k_{i,1 \leq i \leq n}$ where for each i, $k_i : b \to b_i$, for each family of edges $f_i : a \to b_i$ such that $\langle f_1, \ldots f_n \rangle$ references b with respect to $\langle k_1, \ldots k_n \rangle$, the path $\langle f_1, \ldots f_n, I_b \rangle$ is explicitly represented. Note that from this condition it follows that for all entity types a and for all entity types b, for all identifying tuples $\langle q_1, \ldots q_n \rangle$ with respect to b, where for each i, $q_i : b \to b_i$, for each family of edges $f_i : a \to b_i$ such that $\langle f_1, \ldots f_n \rangle$ references b with respect to $\langle q_1, \ldots q_n \rangle$, the path $\langle f_1, \ldots f_n, I_{\langle q_1, \ldots q_n \rangle} \rangle$ is explicitly represented.

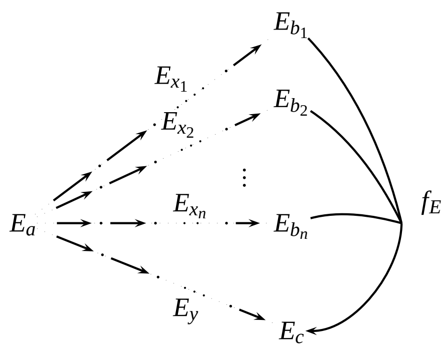
(iii) if for some $n \geq 1$, a, $b_{i,1 \leq i \leq n}$, and c are entity types and if $x_{i,1 \leq i \leq n}$, and y are simple paths such that for each i, $x_i : a \to b_i$, and such that $y : a \to c$ as shown here:



then if in each instance E there exists a unique function $f_E : E_{b_1} \times E_{b_n} \to E_c$

such that domain of $f_E \subseteq img(E_{\langle x_1, \ldots x_n \rangle})$ and for each i, $1 \leq i \leq n$, $E_{\langle x_1, .. x_n \rangle} \circ f_E = E_y$ and the family of functions f_E is irreducible then either in every instance E, $E_{\langle x_1, .. x_n \rangle}$ is invertible or else the family of functions f_E are represented in the model by an entity type d, and an identifying tuple of simple paths with respect to d, $q_1, \ldots q_n$, z such that $\langle x_1, .. x_n \rangle$ references d with respect to $q_1, \ldots q_n$ and from which it follows (from note in clause (ii)) that there is a simple path $p : a \to d$ such that in all instances E,

$$E_{\langle x_1, .. x_n \rangle} \circ inv_{E_{\langle q_1, \ldots q_n \rangle}} = E_p$$



3 Definitions Of Logical and Physical Entity Models

3.1 Preliminaries

Definition An *equi-join condition* between two entity types a and b is defined to be a sequence of pairs of attributes of a respectively b i.e it is for some n, $n \geq 1$ a sequence of n pairings of attributes of a, respectively, b, i.e. a function $\sigma : N_n \rightarrow \mathcal{M}_a^A \times \mathcal{M}_b^A$.

If σ is a equi-join condition between two entity types a and b then we will denote the i'th pairing of attributes as $\sigma_{i,1}, \sigma_{i,2}$. Thus we have that $\sigma_{i,1} \in \mathcal{M}_a^A$ and $\sigma_{i,2} \in \mathcal{M}_b^A$.

If σ is a equi-join condition within a schema s and if E is an instance of s then denote by E_σ the many-valued function from E_a to E_b defined by $\sigma(e) = \{e' \in E_b : \forall i \in N_n, \sigma_{i,1}(e) = \sigma_{i,2}(e')\}$.

Definition An equi-join condition σ between two entity types a and b is defined to be an *inclusion dependency* iff the set $E_\sigma(e)$ is non-empty, for all instances E of s and for all $e \in E_a$.

By the *domain* of a join condition σ in an instance E we shall mean the set $\{e \in E_a \| \forall i, 1 \leq i \leq n, \sigma_{i,1(e)}$is defined$\}$.

Definition An include dependency σ between two entity types a and b is *referential*[4] iff the set $E_\sigma(e)$ is a singleton set, for all instances E and for all e in the domain of σ .

3.2 Definition of Logical ER Model

Definition A well-formulated ER model is *purely-logical* iff it also satisfies: (i) there are no edges r such that there is a simple path p which does not include r in its definition and such that $r \simeq p$ and (ii) there are no non-trivial referential inclusion dependencies.

We say that an ER model is a *logical ER model* iff it is purely logical.

3.3 Definition of Physical ER Model

Definition A *physical ER model* is a well formulated ER model that also satisfies: (i) all identifying edges are attributes and (ii) for each relationship r there is navigational path p containing only attributes such that $r \simeq p$.

4 First Cut Chen Transformation

The transformation described by Chen provides a first cut transformation from a logical model to a physical model recursively.

[4]Also called a referential constraint or a foreign key constraint. Oracle Database Concepts Documentation: *If any column of a composite foreign key is null, then the non-null portions of the key do not have to match any corresponding portion of a parent key.*

If $\mathcal{M}$ is a model then in the Chen transformed model $\mathcal{X}_0(\mathcal{M})$ the attributes of an entity type a are the attributes of a in the model $\mathcal{M}$, plus additional 'physical' attributes implementing outgoing relationships :

$$\mathcal{X}_0(\mathcal{M})_a^A = \mathcal{M}_a^A \cup \mathcal{X}_0(\mathcal{M})_a^{A+}$$

where :

$$\mathcal{X}_0(\mathcal{M})_a^{A+} = \sum_{r \in \mathcal{M}_a^R} \mathcal{X}_0(\mathcal{M})_{dst(r)}^{iA}$$

In this definition, $\mathcal{X}_0(\mathcal{M})_{dst(r)}^{iA}$ denotes the subset of identifying attributes of the destination of a relationship r in the transformed model. These are the identifying attributes of the original model plus the 'physical' attributes which implement *identifying* relationships.

$$\mathcal{X}_0(\mathcal{M})_a^{iA} = \mathcal{M}_a^{iA} \cup \mathcal{X}_0(\mathcal{M})_a^{iA+}$$

where:

$$\mathcal{X}_0(\mathcal{M})_a^{iA+} = \sum_{r \in \mathcal{M}_a^{iR}} \mathcal{X}_0(\mathcal{M})_{dst(r)}^{iA}$$

What these recursive definitions express is that the attributes of the physical model are those of the logical model plus simple paths $\langle r_0, r_1, ...r_n, a\rangle$ where $n \geq 0$, where for $i \geq 1$, r_i is itself an identifying relationship and where a is an identifying attribute. Such an attribute $\langle r_0, ...r_n, a\rangle$ is an identifying iff r_0 is identifying.

The definition of $\mathcal{X}_0(\mathcal{M})_a^{A+}$ can be reformulated in this way:

$$\mathcal{X}_0(\mathcal{M})_a^{A+} = \sum_{n\geq 0} \sum_{r_0 \in \mathcal{M}_a^R} \sum_{r_1 \in \mathcal{M}_{dst(r_0)}^{iR}} \cdots \sum_{r_n \in \mathcal{M}_{dst(r_{n-1})}^{iR}} \mathcal{M}_{dst(r_n)}^{iA}$$

and the definition of the subset $\mathcal{X}_0(\mathcal{M})_a^{iA+}$ can similarly be reformulated:

$$\mathcal{X}_0(\mathcal{M})_a^{iA+} = \sum_{n\geq 0} \sum_{r_0 \in \mathcal{M}_a^{iR}} \sum_{r_1 \in \mathcal{M}_{dst(r_0)}^{iR}} \cdots \sum_{r_n \in \mathcal{M}_{dst(r_{n-1})}^{iR}} \mathcal{M}_{dst(r_n)}^{iA}$$

These reformulated definitions are the starting point for the definitions that follow.

5 Chi Transform - a Revised Chen Transformation

To correct the Chen transformation we take note of equivalent paths so as not to introduce redundant attributes.

Say that a path $\langle r_0, r_1, ...r_n\rangle \in \mathcal{X}_0(\mathcal{M})_a^{A+}$ *is subsumed by* a simple path $\langle s_0, s_1, ...s_m\rangle$ iff $m \geq 1$ and either:

(i) $\langle r_0, r_1, ...r_n\rangle \simeq \langle s_0, s_1, ...s_m\rangle$ and for some j, $j > 1$, s_j is not identifying.

or:

(ii) $\langle r_0, r_1, ...r_n\rangle < \langle s_0, s_1, ...s_m\rangle$ and $r_0 \neq s_0$.

We define $\mathcal{X}_1(\mathcal{M})_a^{A+}$ to be the subset of $\mathcal{X}_0(\mathcal{M})_a^{A+}$ consisting of those paths for which there are no paths that subsume them.

We define the Chi transformed model $\mathcal{X}(\mathcal{M})$ by:

$$\mathcal{X}(\mathcal{M})_a^A = \mathcal{M}_a^A \cup \mathcal{X}_2(\mathcal{M})_a^{A+}$$

where $\mathcal{X}_2(\mathcal{M})_a^{A+}$ is the set of equivalence classes of paths in $\mathcal{X}_1(\mathcal{M})_a^{A+}$ with respect to the $\simeq$ equivalence relation.

and by:

$$\mathcal{X}(\mathcal{M})_a^{iA} = \mathcal{M}_a^A \cup \mathcal{X}_2(\mathcal{M})_a^{iA+}$$

where[5]:

$$\begin{aligned}\mathcal{X}_2(\mathcal{M})_a^{iA+} = \{C \in \mathcal{X}_2(\mathcal{M})_a^{A+} \mid & \text{ there exists } \langle s_0, s_1, \ldots s_m\rangle \in C \text{ such that either } s_0 \text{ is identifying or} \\ & \text{there exists } \langle r_0, r_1, \ldots r_n\rangle \in \mathcal{X}_0(\mathcal{M})_a^{iA+} \text{ and a simple path } \langle s'_1, \ldots s'_{m'}\rangle \text{ such that} \\ & r_0 \text{ is identifying and } \langle r_0, r_1, \ldots r_n\rangle \text{ is subsumed by } \langle s_0, s'_1, \ldots s'_m\rangle\}\end{aligned} \quad (1)$$

6 Boyce-Codd Normal Form

One measure of the goodness of a physical model is whether it satisfies the well-formedness condition know as Boyce Codd Normal Form. Written in the terminology we are using here it can be expressed as follows: a physical ER model is in Boyce Codd Normal Form (BCNF) iff for all entity types a, for all attributes $x_1, \ldots x_n$ and y, $n \geq 1$, if in all instances E, there exists a unique n-ary partial function f such that $E_{<x_1,\ldots x_n>} \circ f = E_y$ then either y is x_i for some i or else in all instances E, $E_{<x_1,\ldots x_n>}$ is invertible.

The next lemma simplifies the requirement for showing BCNF to consideration of irreducible families of functions:

Lemma 6.1 *A model $\mathcal{M}$ is in BCNF iff for all entity types a, for all attributes $x_1, \ldots x_n$ and y, $n \geq 1$, in all instances E, there exists a unique n-ary partial function f such that $E_{<x_1,\ldots x_n>} \circ f = E_y$ and if the family of functions f_E is irreducible then either $n = 1$, $x_1 = y$ and $E_f = id_{E_y}$ or else in all instances E, $E_{<x_1,\ldots x_n>}$ is invertible.*

Proof Suppose $\mathcal{M}$ is an ER model and that a is an entity type of $\mathcal{M}$ and that $x_1, \ldots x_n$ and y are attributes of a and suppose that in all instances E of $\mathcal{M}$ there is a unique function $f_E : v \to v$ such that

$$E_{\langle x_1,\ldots x_n\rangle} \circ f_E = E_y$$

Suppose that f_E is reducible to g_E and that g_E is irreducible. We have therefore that, for some J,

$$f_E = P_J \circ g_E$$

and therefore that

$$E_{\langle x_1,\ldots x_n\rangle} \circ P_J \circ g_E = E_y$$

and because

$$E_{\langle x_1,\ldots x_n\rangle} \circ E_{proj_J} = E_{\langle x_{i_1},\ldots x_{i_j}\rangle}$$

it follows that

$$E_{\langle x_{i_1},\ldots x_{i_j}\rangle} \circ g_E = E_y.$$

Since g_E is irreducible it follows from the initial assumption that either $j = 1$ and $x_{i_1} = y$ and y is one of the $x_1, \ldots x_n$ as required or else $E_{\langle x_{i_1},\ldots x_{i_j}\rangle}$ is invertible from which it follows that $E_{\langle x_1,\ldots x_n\rangle}$ is invertible, as required.

[5] In fact this definition needs modifying to deal with cases when an r sequence is subsumed by two distinct s sequences - otherwise too many identifying attribues are generated.

We aim to show:

Theorem

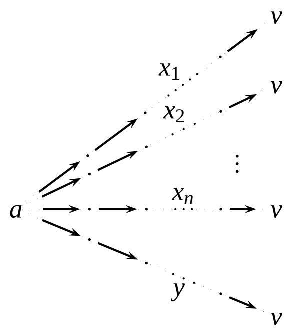
If an ER model $\mathcal{M}$ is well-formulated then the transformed model $\mathcal{X}(\mathcal{M})$ is in Boyce-Codd Normal Form.

Proof

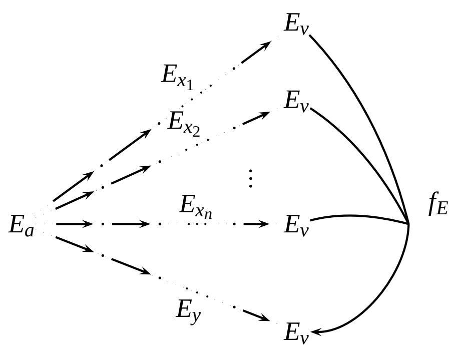
Suppose that $\bar{x}_1,...\bar{x}_n,\bar{y}$ are attributes of the entity type a of model $\mathcal{X}(\mathcal{M})$ suppose that in all instances E of $\mathcal{X}(\mathcal{M})$ there exists a a unique n-ary partial function E_f such that $E_{<\bar{x}_1,...\bar{x}_n>} \circ E_f = E_{\bar{y}}$ we need to show that either $\bar{y}$ is $\bar{x}_i$ for some i or else in all instances E of $\mathcal{X}(\mathcal{M})$, $E_{<\bar{x}_1,...\bar{x}_n>}$ is invertible.

From the definition of $\mathcal{X}$ it follows that for each $\bar{x}_i$ there is a $m_i, m_i \geq 1$ and a simple path $\langle x_{i,1},...x_{i,m_i}\rangle$ in $\mathcal{M}$ where $x_{i,j}$ is identifying, for $j > 1$ and $dest(x_{i,m_i}) = v$ such that either, $m_i = 1$ and $\bar{x}_i = x_{i,1}$ or $m_i > 1$ and $\bar{x}_i = [\langle x_{i,1},...x_{i,m_i}\rangle]$. It follows likewise that for some $m \geq 1$, there is a simple path $\langle y_1,..y_m\rangle$ in $\mathcal{M}$ such that either $m = 1$ and $\bar{y} = y_m$ or $m > 1$ and $\bar{y} = [\langle y_1,..y_m\rangle]$.

In the model $\mathcal{M}$ therefore, for each i, $1 \leq i \leq n$, for some m_i, $m_i \geq 1$, we have a path of length m_i which we denote $x_i = \langle x_{i,1},...x_{i,m_i}\rangle$ and for some m, $m \geq 1$ we have a path of length m which we denote $y = \langle y_1,..y_m\rangle$ as shown here:



Each instance E of $\mathcal{M}$ gives rise to an instance $\mathcal{X}(E)$ of $\mathcal{X}(\mathcal{M})$ and from the definition of $\mathcal{X}(E)$ it follows that for every instance E of $\mathcal{M}$ there is a unique function E_f such that $E_{<x_1,...x_n>} \circ E_f = E_y$, as shown here:



From the assumption that the model $\mathcal{M}$ is well-formulated and from condition (iii) of the definition of well-formulated, either $E_{\langle x_1,...x_n\rangle}$ is invertible in every instance E of $\mathcal{M}$ in which case $\mathcal{X}(E)_{\langle \bar{x}_1,...\bar{x}_n\rangle}$ is invertible in every instance E of $\mathcal{M}$ and the proof is completed or else the family of function f_E are represented in the model $\mathcal{M}$. From the definition of a function family being represented it follows that either (i) y is x_i for some i from which it follows that $\bar{y}$ is $\bar{x}_i$ for some i and the proof is complete or (ii) there is an entity type b in $\mathcal{M}$ and an identifying family of simple paths $q_1, \ldots q_n$, $q_i : b \to v$ and a path $z : b \to v$ such that in every instance E of $\mathcal{M}$:

$$inv_{E_{\langle q_1,\ldots q_n\rangle}} \circ E_z = f_E$$

from which it follows that in every instance E of $\mathcal{M}$:

$$E_{\langle x_1,\ldots x_n\rangle} \circ inv_{E_{\langle q_1,\ldots q_n\rangle}} \circ E_z = E_{\langle x_1,\ldots x_n\rangle} \circ f_E$$

and thus, from our initial assumption, that:

$$\forall E \in inst_{\mathcal{M}}, \quad E_{\langle x_1,\ldots x_n\rangle} \circ inv_{E_{\langle q_1,\ldots q_n\rangle}} \circ E_z = E_{\langle y_1,..y_m\rangle} \tag{2}$$

In this case, because $\mathcal{M}$ is well-formulated and from condition (ii) of the definition of well-formulated it follows that there exists a path $\langle p_1, \ldots p_k\rangle : a \to b$, $k \geq 0$, such that:

$$\forall E \in inst_{\mathcal{M}}, \quad \langle E_{x_1}, \ldots E_{x_n}\rangle \circ inv_{E_{\langle q_1,\ldots q_n\rangle}} = E_{\langle p_1,\ldots p_k\rangle} \tag{3}$$

Either $k = 0$ and $\langle q_1, \ldots q_n\rangle = \langle x_1, \ldots x_n\rangle$ in which case $E_{\langle x_1,\ldots x_n\rangle}$ is invertible in every instance E of $\mathcal{M}$ and thus $\mathcal{X}(E)_{\langle \bar{x}_1,\ldots \bar{x}_n\rangle}$ is invertible in every instance $\mathcal{X}(E)$ of $\mathcal{X}(\mathcal{M})$ and the proof is complete or else $k \geq 1$ and it follows from (2) and (3) that:

$$\forall E \in inst_{\mathcal{M}}, \quad E_{\langle p_1,\ldots p_k\rangle} \circ E_{\langle z_1,\ldots z_l\rangle} = E_{\langle y_1,..y_m\rangle} \tag{4}$$

We will show that this leads to a contradiction and so complete the proof. If $m > 1$ it follows from (4) that $p_2, \ldots p_k, z_1, \ldots z_l$ subsume $\langle y_1, ..y_m\rangle$, which implies that $\langle y_1, ..y_m\rangle$ is excluded from $\mathcal{X}_1(\mathcal{M})_a^{A+}$ and thus that $\bar{y} = [\langle y_1, ..y_m\rangle]$ is not an attribute of $\mathcal{X}(\mathcal{M})$ contrary to our initial assumption.

Therefore we must conclude that $m = 1$. In this case we have y_1 an attribute of a in $\mathcal{M}$ and from (4) we have in all instances E of $\mathcal{M}$:

$$E_{y_1} = E_{\langle p_1,\ldots p_k\rangle} \circ E_{\langle z_1,\ldots z_l\rangle} \tag{5}$$

which is to say in all instances E of $\mathcal{M}$:

$$E_{y_1} = E_{\langle p_1,\ldots p_k, z_1,\ldots z_l\rangle} \tag{6}$$

We have shown, therefore, that y_1 is an outgoing edge of a in $\mathcal{M}$ which is equivalent to a simple path of $\mathcal{M}$ of length ≥ 2 which contradicts the initial assumption that the model $\mathcal{M}$ is purely logical and so completes the proof.