Algebraic semantics of ER-models from the standpoint of map calculus. Part I: Static view.[1]

Eugenio G. Omodeo[2]
Dipartimento di Matematica Pura ed Applicata
Università degli Studi di L'Aquila
L'Aquila, Italy
omodeo@univaq.it

Ernst-Erich Doberkat
Department of Computer Science
Universität Dortmund
Dortmund, Germany
eed@LS10.de

Spring 2001

[1]This is a preliminary version. The final version will be published in *Electronic Notes in Theoretical Computer Science*, URL: www.elsevier.nl/locate/entcs

[2]This research was in part supported through a travel grant from the *Deutsche Akademischer Austauschdienst*

Abstract

Entity-Relationship modeling is a popular technique for data modeling. Despite its popularity and wide spread use, it lacks a firm semantic foundation. We propose a translation of an ER-model into mapalgebra, suggesting that mapalgebra does provide suitable mechanisms for establishing a formal semantics of entity-relationship modeling. This report deals with the techniques necessary for the translation and provides a static view of an ER-model in its mapalgebraic disguise.

1 Goals of the present study

The structure of complex data may be specified through an Entity-Relationship model. This technique for modeling is rather popular, in part because it permits the visualization of the relationship between data, making the structure of their interrelationship easily grasped. The technique is based on Codd's seminal paper [3], it is available in many variants, and it is one of the ancestors of the popular design method *UML*, see e.g. [11].
Despite its popularity, this approach to data modeling lacks a firm formal foundation: it appeals rather to the intuition of a modeler, neglecting the necessity of formally arguing about an ER-model. There are some proposals for a formal semantics of various versions of the model (section 7 provides a brief overview) which are mainly based on algebraic formalisms like algebraic specifications. A semantics based purely on a calculus coming from logics has not been investigated yet to the authors' knowledge.
Using the map calculus [1] as a formalization of set theory without variables [14] is intuitively appealing: The naive semantics given to an ER-model is essentially set based — entities are modeled through sets, relations through sets of n-ary tuples, and attributes as functions. It is this naive approach which is formalized here.
Hence the language of map calculus (*mapalgebra*, as we will call it in brief) will serve as the target language into which to translate an Entity-Relationship model. Is this translation always possible? does it necessarily presuppose that some unduly artificial restrictions are imposed on a modeling language which —as it stands— has an undebatable practical value? will, perhaps, the limitations of mapalgebra force us into some richer target language?
We will carry out the translation at two levels. In the first place, since the primary role of ER-modeling is to bridle the dynamics of a database within invariable boundaries, we will manage to state the very same constraints in a different and simpler-minded formalism. Only this part of the translation constitutes object of the present paper. Then —in a subsequent paper— we will head to a specification of the abstract data types associated with an ER-model. Mapalgebra lacks explicit means to describe change, and anyway the ADTs are left implicit by the Entity-Relationship language too; nevertheless, we will strive to clarify the semantics of database transactions by formulating how the new state enforced by an update operation relates to the state immediately before it and by showing that updates never cause permanent instability w.r.t. the statically fixed constraints.

At least one benefit should ensue from a systematic translation of ER-models into mapalgebra— to the extent to which this will demonstrate feasible: we will be forced to apply Okkam's razor to the notions involved, so that e.g. the distinctions between relations and entities and between relations and attributes will fade away. Mapalgebra, in fact, was not designed to treat objects of different types, and can only take dyadic relations into account.

Our reference for the language of ER-models is [4] (see, e.g., [15] for an extensive treatment); however, since the full variety of ER-models cannot be accommodated very easily into our framework, we leave out of consideration attributes on relations, at least momentarily.
Our reference on map calculus is [6] (the most fundamental reference of all is [14]), and we briefly report on it in Sec.2. Let us recall here that map calculus, grown out of Boole's "algebra of thought" just a few decades after the latter had been conceived (cf. [12, 13]), is an enriched form of it. Today it looks like a simplified version of Codd's relation algebra which also happens to be a denominator common to a number of taxonomic languages proposed for knowledge representation by several authors. Map calculus can hence be viewed both as

an assembly language for knowledge representation and as a common ancestor of the many systems of algebraic logic available today.

2 Map calculus in a nutshell

Very much like any logical formalism, map calculus consists of a symbolic language, an intended semantics, a collection of logical axiom schemes (which, according to the intended semantics, are valid, i.e. true in any legal interpretation), and a collection of inference rules. Making use of a formalism means describing a privileged *universe* $\mathcal{U}$ *of discourse* (or perhaps a variety of universes of interest) by means of *proper axioms* stated in the symbolic language, in order to then be able to derive —by means of the inference rules— new facts which necessarily follow from the axioms. More axioms means fewer interpretations, and therefore a larger collection of derivable consequences; as an extreme case, in absence of proper axioms, only valid sentences —which are a bit too vacuous to be really useful— are derivable.
Map calculus was designed to ease reasoning about dyadic relations —MAPS, as we will call them— over an unspecified, yet fixed, universe $\mathcal{U}$. Its language is entirely equational, and ground (i.e., devoid of individual variables); we feel therefore authorized to concentrate mainly on syntax and intended semantics, to then summarize the logical axioms and inference rules simply by a table, cf. Figure 2. After illustrating the use of mapalgebra by a few tiny examples, in Sec.3 we will carry out in mapalgebra more challenging specification tasks, which will bring us closer to a crisp semantics of ER-modeling.

2.1 Syntax and semantics of mapalgebra

Mapalgebra consists of *map equalities* $Q{=}R$, where Q and R are *map expressions*:

Definition 1 MAP EXPRESSIONS *are all terms of the following signature:*

symbol :	$\emptyset$	$\mathbb{1}$	ι	r_i	$\cap$	$\triangle$	;	$\smile$	$\overline{}$	$-$	$\cup$
degree :	0	0	0	0	2	2	2	1	1	2	2
priority :					5	3	6	7		2	2

(Of these, $\cap, \triangle, ;, -, \cup$ will be used as left-associative infix operators, $^{\smile}$ as a postfix operator, and $^{-}$ as a line topping its argument.)
We assume a countable infinity $\mathsf{r}_0, \mathsf{r}_1, \mathsf{r}_2, \ldots$ of MAP LETTERS to be available.

For an *interpretation* of mapalgebra one must indicate a nonempty $\mathcal{U}$, and assign a subset $\mathsf{r}_i^{\Im}$ of the Cartesian square $\mathcal{U}^2 =_{\mathrm{Den}} \mathcal{U} \times \mathcal{U}$ to each map letter r_i. Then each map expression P comes to designate, thanks to the rules below, a specific map $P^{\Im}$ (any equality $Q{=}R$ between map expressions turns out, accordingly, to be either true or false):

$$\emptyset^{\Im} =_{\mathrm{Den}} \emptyset, \qquad \mathbb{1}^{\Im} =_{\mathrm{Den}} \mathcal{U}^2, \qquad \iota^{\Im} =_{\mathrm{Den}} \{\langle a,a\rangle \mid a \text{ in } \mathcal{U}\};$$
$$(Q{\cap}R)^{\Im} =_{\mathrm{Den}} \{\langle a,b\rangle \in Q^{\Im} \mid \langle a,b\rangle \in R^{\Im}\};$$
$$(Q\triangle R)^{\Im} =_{\mathrm{Den}} \{\langle a,b\rangle \in \mathcal{U}^2 \mid \langle a,b\rangle \in Q^{\Im} \text{ if and only if } \langle a,b\rangle \notin R^{\Im}\};$$
$$(Q;R)^{\Im} =_{\mathrm{Den}} \{\langle a,b\rangle \in \mathcal{U}^2 \mid \text{there is a } c \text{ in } \mathcal{U} \text{ for which } \langle a,c\rangle \in Q^{\Im} \text{ and } \langle c,b\rangle \in R^{\Im}\};$$
$$(Q^{\smile})^{\Im} =_{\mathrm{Den}} \{\langle b,a\rangle \in \mathcal{U}^2 \mid \langle a,b\rangle \in Q^{\Im}\}.$$

Of the operators and constants in the signature of mapalgebra, only a few deserve being regarded as *primitive* constructs; indeed, we choose to regard as *derived* constructs the following ones, as well as others that we may add to the signature from time to time:[1]

[1]The priorities of $\times, \bigcap, \bigcup, \sum$ follow those of $;, \cap, \cup, \triangle$, and they are 6.5, 5.5, 2.5, 3.5.

$\mathsf{NonVoid}(R)$	$\equiv_{\mathrm{Den}}$	$\mathsf{Total}(\mathbb{1};R)$
$\mathsf{Func}(R)$	$\equiv_{\mathrm{Den}}$	$\mathsf{Coll}(R^{\smile};R)$
$\mathsf{Snglt}(R)$	$\equiv_{\mathrm{Den}}$	$\mathsf{NonVoid}(R)\ \&\ \mathsf{Func}(\mathbb{1};R)\ \&\ \mathsf{Func}(R^{\smile})$
$\mathsf{Const}(R)$	$\equiv_{\mathrm{Den}}$	$\mathsf{Snglt}(R)\ \&\ \mathsf{Coll}(R)$
$\mathsf{Dangl}(D)$	$\equiv_{\mathrm{Den}}$	$\mathsf{Coll}(D\,;\mathbb{1}\,;D)$
$\mathsf{Proj}(L,R)$	$\equiv_{\mathrm{Den}}$	$\begin{cases} \mathsf{Func}(L)\ \&\ \mathsf{Func}(R)\ \&\ L;\mathbb{1}=R;\mathbb{1} \\ \quad \&\ \mathbb{1}=L^{\smile};R\ \&\ \mathsf{Coll}(L\,;L^{\smile}\cap R;R^{\smile}) \end{cases}$

Figure 1: Non-empty map, single-valued map, singleton maps, dangling-value map, and two conjugated projections

$\overline{P}$	$\equiv_{\mathrm{Den}}$	$P\triangle\mathbb{1}$	$\bigcap_{i=1}^{n} P_i$	$\equiv_{\mathrm{Den}}$	$P_1\cap\cdots\cap P_n$
$P-Q$	$\equiv_{\mathrm{Den}}$	$P\cap\overline{Q}$	$\bigcup_{i=1}^{n} P_i$	$\equiv_{\mathrm{Den}}$	$P_1\cup\cdots\cup P_n$
$P\cup Q$	$\equiv_{\mathrm{Den}}$	$\overline{\overline{P}-Q}$	$\sum_{i=1}^{n} P_i$	$\equiv_{\mathrm{Den}}$	$P_1\triangle\cdots\triangle P_n$
$\mathsf{dom}(P)$	$\equiv_{\mathrm{Den}}$	$P;\mathbb{1}\cap\iota$	$\bigtimes_{i=1}^{n} P_i$	$\equiv_{\mathrm{Den}}$	$P_1;\cdots;P_n$
$\mathsf{img}(P)$	$\equiv_{\mathrm{Den}}$	$\mathbb{1};P\cap\iota$			

(to be intended as follows when $n=0$:

$$\bigcap_{i=1}^{0} P_i \equiv_{\mathrm{Den}} \mathbb{1}, \quad \bigcup_{i=1}^{0} P_i \equiv_{\mathrm{Den}} \emptyset, \quad \sum_{i=1}^{0} P_i \equiv_{\mathrm{Den}} \emptyset, \quad \bigtimes_{i=1}^{0} P_i \equiv_{\mathrm{Den}} \iota).$$

The interpretation of mapalgebra obviously extends to the new constructs; e.g.,[2]

$$\mathsf{dom}(P)^{\Im} =_{\mathrm{Den}} \{\,\langle a,a\rangle \in \mathcal{U}^2\ :\ \text{there are } b\text{s in } \mathcal{U} \text{ for which } \langle a,b\rangle \in P^{\Im}\,\}\,.$$

Through *macros*, we can also define shortening notation for map equalities that follow certain patterns, e.g.,

$P\subseteq Q$	$\equiv_{\mathrm{Den}}$	$P-Q=\emptyset$
$P=Q\ \&\ R=S$	$\equiv_{\mathrm{Den}}$	$\emptyset=P\triangle Q\cup R\triangle S$
$\&_{i=1}^{n}\ P_i=Q_i$	$\equiv_{\mathrm{Den}}$	$\emptyset=\bigcup_{i=1}^{n} P_i\triangle Q_i$
$P\subseteq Q\subseteq R$	$\equiv_{\mathrm{Den}}$	$P\subseteq Q\ \&\ Q\subseteq R$
$\mathsf{Coll}(P)$	$\equiv_{\mathrm{Den}}$	$P\subseteq\iota$
$\mathsf{Total}(P)$	$\equiv_{\mathrm{Den}}$	$P;\mathbb{1}=\mathbb{1}$

so that $\mathsf{Coll}(E)$ means "$E^{\Im}$ consists of pairs of the form $\langle a,a\rangle$" (hence E represents a collection of entities rather than a genuine map), and $\mathsf{Total}(P)$ states that for all a in $\mathcal{U}$ there is at least one pair $\langle a,b\rangle$ in $P^{\Im}$.

As shown in Figure 1, macros are a device through which one can specify in mapalgebra quite significant properties of maps. Thus $\mathsf{Func}(P)$ means "$P^{\Im}$ is a partial function", $\mathsf{Snglt}(R)$ states that $R^{\Im}$ consists of a single pair, $\mathsf{Dangl}(D)$ states that either $D^{\Im}$ is empty or it has the form $\{\langle d,d\rangle\}$, etc.

In particular, by postulating $\mathsf{Proj}(\lambda,\varrho)$ one requires $\lambda^{\Im},\varrho^{\Im}$ to be *conjugated projections*, i.e. functions defined on the same subset $\mathcal{P}$ of $\mathcal{U}$ so that

- for all a,b in $\mathcal{U}$, there is a p in $\mathcal{P}$ s.t. $\lambda^{\Im}(p)=a$ and $\varrho^{\Im}(p)=b$;
- if p,q in $\mathcal{P}$ are distinct, they cannot satisfy $\lambda^{\Im}(p)=\lambda^{\Im}(q)$ together with $\varrho^{\Im}(p)=\varrho^{\Im}(q)$.

Otherwise stated, $p\mapsto\langle\lambda^{\Im}(p),\varrho^{\Im}(p)\rangle$ is an injection from the subset $\mathcal{P}$ of $\mathcal{U}$ onto $\mathcal{U}^2$, which implies that $\mathcal{U}$ is either singleton or infinite.

[2]One often indicates as the domain of a map $\mathbf{P}$ any set which *includes* the actual set of first components of pairs in $\mathbf{P}$. With our definition, though, $\mathsf{dom}(P)$ represents what might be called the *tight domain* of $P^{\Im}$.

$$
\begin{array}{rcl}
P \mathbin{\triangle} P & = & \emptyset \\
P \mathbin{\triangle} (Q \mathbin{\triangle} P) & = & Q \\
R \cap Q \mathbin{\triangle} R \cap P & = & (P \mathbin{\triangle} Q) \cap R \\
P \cap P & = & P \\
1\!\!1 \cap P & = & P \\
(P \star_1 Q) \star_1 R & = & P \star_1 (Q \star_1 R) \\
\iota ; P & = & P \\
P^{\smile\smile} & = & P \\
(P \star_2 Q)^{\smile} & = & Q^{\smile} \star_2 P^{\smile} \\
(P \mathbin{\triangle} Q) ; R & \subseteq & Q ; R \cup P ; R
\end{array}
$$

From $P ; Q \cap R = \emptyset$ derive $P^{\smile} ; R \cap Q = \emptyset$

From $P \subseteq Q$ derive $P ; R \subseteq Q ; R$

Substitution laws for equals (cf. [8])

$\star_1 \in \{\triangle, \cap, ;\}$ and $\star_2 \in \{\cap, ;\}$

Figure 2: Logical axioms and inference rules for the map calculus

3 Some basic specifications in mapalgebra

Quite often, only an intuitive characterization of the semantics of Entity-Relationship modeling is found in the literature. This paper aims at clarifying it via a systematic translation of ER-diagrams into mapalgebra. In view of the precise semantics of the latter (cf. Sec.2), the translation will provide a formal semantics to the former, which we assume the reader to be already conversant with.

Our translation will presuppose that the universe $\mathcal{U}$ has a certain structure, which we must describe by subjecting some of the map letters to appropriate conditions. Only very few map letters need to be reserved for this task; they are $\pi, \lambda, \varrho, \upsilon, \varepsilon$, where we make the identifications $\pi \equiv_{\mathrm{Den}} \mathsf{r}_0$, $\lambda \equiv_{\mathrm{Den}} \mathsf{r}_1$, $\varrho \equiv_{\mathrm{Den}} \mathsf{r}_2$, $\upsilon \equiv_{\mathrm{Den}} \mathsf{r}_3$, and $\varepsilon \equiv_{\mathrm{Den}} \mathsf{r}_4$ for the sake of definiteness. What conditions must be met by the maps which these symbols designate will be seen in subsections 3.1 and 3.2.

The translation will only operate on *well-founded* ER-diagrams—*ER-models*, as we call them; not even on all of them (at least directly), because in order to ease the translation phase we will place before it a preprocessing phase which brings any given ER-model into a suitably normalized form.

The identifiers of the ER-model will become map expressions under the translation. To make things natural, we will translate them into (or, even more simply, identify them with distinct) map letters. However, three kinds of identifiers are used in ER-modeling whereas mapalgebra has letters of only one kind; therefore, we must be able to express the distinction between ER-entities, ER-relations, and ER-attributes by means of map equalities that constrain them differently. Part of the task of clarifying this distinction is carried out in subsection 3.3 below, while the tasks of specifying normalization and translation algorithms are postponed to later sections.

3.1 Flat tuples

Tuples of length > 2 forcibly enter into the study of database systems, if only because the operation of inserting an entity e into a database often causes the simultaneous assignment of a tuple of values to the attributes of e. To cope with this while avoiding the complications that would result from the treatment of relations of arity greater than 2, we want the universe $\mathcal{U}$ of discourse to include $\mathcal{A}^*$—viz., the set of all finite-length sequences whose components are entities drawn from $\mathcal{A}$ and are in some sense "atomic". We will assume for simplicity that $\mathcal{U} = \mathcal{A} \cup \mathcal{A}^*$ and $\mathcal{A} \cap \mathcal{A}^* = \emptyset$, and, to avoid triviality, that $\mathcal{A} \neq \emptyset$. We indicate by ϵ the null tuple in $\mathcal{A}^*$, and put $\mathcal{A}^+ =_{\mathrm{Den}} \mathcal{A}^* \setminus \{\epsilon\}$.
Two operations on non-null tuples $t_0 t_1 \cdots t_m$ are essential, namely *head isolation* —which determines the first component t_0 of any given such tuple— and *tail extraction*—which determines the sub-tuple $t_1 \cdots t_m$ resulting from removal of the first component. Let us designate these operations by λ, ϱ, in view of the affinity between their properties and the ones of projections, discussed at the end of Sec.2. Moreover, let us represent by υ and ε the collection $\mathcal{A}$ of atomic entities and the null tuple: more precisely, $\upsilon^{\Im} = \{\langle a, b\rangle \in \mathcal{A}^2 \mid a = b\}$ and $\varepsilon^{\Im} = \{\langle \epsilon, \epsilon \rangle\}$ in our intended interpretation $\Im$.

We can state that

- ε designates a singleton and diagonal map, by the condition $\mathsf{Const}(\varepsilon)$;
- υ represents a non-empty collection $\mathcal{A}$ of entities distinct from the entity, ϵ, represented by ε, by means of the conditions $\mathsf{Coll}(\upsilon)$, $\mathsf{NonVoid}(\upsilon)$, $\varepsilon \cap \upsilon = \emptyset$;
- λ, ϱ designate functions $\boldsymbol{\lambda} = \lambda^{\Im}$ and $\boldsymbol{\varrho} = \varrho^{\Im}$ whose common domain $\mathcal{A}^+$ is the complement of $\mathcal{A} \cup \{\epsilon\}$ in $\mathcal{U}$, by means of the conditions
$$\mathsf{Func}(\lambda),\ \mathsf{Func}(\varrho),\ \text{and}\ \lambda \,;\, \mathbb{1} = \varrho \,;\, \mathbb{1} = \overline{(\upsilon \cup \varepsilon) \,;\, \mathbb{1}};$$
- $\boldsymbol{\varrho}(p)$ belongs to $\mathcal{A}^*$ for all p, by means of the condition $\upsilon \cap \mathbb{1} \,;\, \varrho = \emptyset$;
- for all a in $\mathcal{A}$ and all q in $\mathcal{A}^*$ there is a tuple p with $\boldsymbol{\lambda}(p) = a$ and $\boldsymbol{\varrho}(p) = q$, by means of the condition $\upsilon \,;\, \overline{\mathbb{1} \,;\, \upsilon} = \lambda^{\smile} \,;\, \varrho$;
- the function $p \mapsto \langle \boldsymbol{\lambda}(p), \boldsymbol{\varrho}(p)\rangle$ is injective, by the condition
$$\mathsf{Coll}(\lambda \,;\, \lambda^{\smile} \cap \varrho \,;\, \varrho^{\smile}).$$

To sum all these conditions up, let us introduce the notation
$$\begin{array}{rcl}\mathsf{HdTl}(L,R,Y,E) & \equiv_{\mathrm{Den}} & \mathsf{Func}(L)\ \ \&\ \ \mathsf{Func}(R)\ \ \&\ \ \mathsf{Const}(E)\\ & & \&\ \ \mathsf{Coll}(Y)\ \ \&\ \ \mathsf{NonVoid}(Y)\ \ \&\ \ E \cap Y = \emptyset\\ & & \&\ \ Y \cap \mathbb{1} \,;\, R = \emptyset\ \ \&\ \ L \,;\, \mathbb{1} = R \,;\, \mathbb{1} = \overline{(E \cup Y) \,;\, \mathbb{1}}\\ & & \&\ \ Y \,;\, \overline{\mathbb{1} \,;\, Y} = L^{\smile} \,;\, R\ \ \&\ \ \mathsf{Coll}(L \,;\, L^{\smile} \cap R \,;\, R^{\smile}),\end{array}$$
so that we can concisely state everything by the single requirement $\mathsf{Hdtl}(\lambda, \varrho, \upsilon, \varepsilon)$.
After noticing that
$$\left(\bigtimes_{j=1}^{i-1} \varrho \right) \,;\, \lambda$$
designates the operation of extracting the i-th component of a tuple, let us also provide a handy characterization of h-tuples:
$$\begin{array}{rcl} i^{\mathsf{th}}(L,R) & \equiv_{\mathrm{Den}} & \bigtimes_{j=1}^{i-1} R \,;\, L\\ h\text{-}\mathsf{tuples}(R) & \equiv_{\mathrm{Den}} & \mathsf{img}(R) \cap \mathsf{dom}(\, h^{\mathsf{th}}(R,R)\,) - \mathsf{dom}(\,(h+1)^{\mathsf{th}}(R,R)\,).\end{array}$$
Thus, under assumption that $\mathsf{Hdtl}(\lambda, \varrho, \upsilon, \varepsilon)$, the map expression $h\text{-}\mathsf{tuples}(\varrho)$ represents the collection of all tuples $t_1 \cdots t_h$ in $\mathcal{A}^*$ for any natural number h.

3.2 Place-holders

Let $\mathbf{r}_1, \ldots, \mathbf{r}_n$ be certain maps whose properties must, for the purposes of some application, be stated formally. To take into account the fact that other maps may enter into play too, let us designate the $\mathbf{r}_i$s by the map letters $\mathsf{r}_{s+1}, \ldots, \mathsf{r}_{s+n}$ (so as to reserve some "free space" at the beginning for special map letters). As discussed above, we are assuming that our universe of discourse is the disjoint union $\mathcal{U} = \mathcal{A} \cup \mathcal{A}^*$ of a collection of atomic entities and a collection of tuples; moreover, for the sake of simplicity, we assume that $\mathbf{r}_1, \ldots, \mathbf{r}_n \subseteq \mathcal{A}^2$.
Placeholders are special elements of $\mathcal{U}$ which occur at most once in $\mathbf{r}_1, \ldots, \mathbf{r}_n$; moreover, no pair $\langle *, \star \rangle$ both of whose components are placeholders can belong to any $\mathbf{r}_i$. These assumptions reflect the possibility that, at least occasionally, when a pair $\langle a_0, a_1 \rangle$ is about being inserted into a relation in a database, the value of either component a_b may still be unknown; nevertheless, a provisional value which bears no information can temporarily occupy the place of the missing value: performing the insertion would simply be a waste of effort if both components were unknown.
Let us utilize $\pi \equiv_{\text{Den}} \mathsf{r}_0$ to state the properties of placeholders. It is understood that $\mathsf{Coll}(\pi)$—what we are representing here by a map is in fact a collection of entities. Thus, for each r_{s+i}, the requirement that no placeholder occurs twice as the first component of a pair in $\mathbf{r}_i$ can be stated as follows:

$$\pi \cap \big((\mathsf{r}_{s+i} \cap (\mathsf{r}_{s+i} \,;\, \overline{\iota}))\,;\, \mathbb{1}\big) \;=\; \emptyset \qquad (i = 1, \ldots, n)\ .$$

The similar requirement that for each distinct pair $\mathsf{r}_{s+i}, \mathsf{r}_{s+j}$ no placeholder occurs both in $\mathbf{r}_i$ and in $\mathbf{r}_j$ as the first component of a pair gets translated even more simply:

$$\pi \cap (\mathsf{r}_{s+i} \,;\, \mathbb{1}) \cap (\mathsf{r}_{s+j} \,;\, \mathbb{1}) \;=\; \emptyset \qquad (i, j = 1, \ldots, n \text{ and } i < j\ )\ .$$

Then comes the requirement that for no pair $\mathsf{r}_{s+i}, \mathsf{r}_{s+j}$ a placeholder can occur both as a first component in $\mathbf{r}_i$ and as a second component in $\mathbf{r}_j$:

$$\pi \cap (\mathsf{r}_{s+i} \,;\, \mathbb{1}) \cap (\mathsf{r}_{s+j}^{\smile} \,;\, \mathbb{1}) \;=\; \emptyset \qquad (i, j = 1, \ldots, n)\ .$$

Then comes the requirement that placeholders cannot appear together in the same pair:

$$\mathsf{r}_{s+i} \cap (\pi \,;\, \mathbb{1}) \cap (\mathbb{1} \,;\, \pi) \;=\; \emptyset \qquad (i = 1, \ldots, n)\ .$$

Next comes the requirement that if $*$ is a placeholder and $a \neq *$, then neither the two pairs $\langle *, b \rangle$, $\langle a, b \rangle$ nor the two pairs $\langle b, * \rangle$, $\langle b, a \rangle$ can belong together to the same $\mathbf{r}_i$ for any b:

$$\big((\pi \,;\, \mathsf{r}_{s+i}) \cap (\overline{\iota} \,;\, \mathsf{r}_{s+i}) \big) \cup \big((\mathsf{r}_{s+i} \,;\, \pi) \cap (\mathsf{r}_{s+i} \,;\, \overline{\iota}) \big) \;=\; \emptyset \qquad (i = 1, \ldots, n)\ .$$

Finally, if υ represents the collection $\mathcal{A}$ of all atomic entities (in a universe $\mathcal{U}$ which also comprises such compound objects as tuples), then it seems natural to require that $\pi \subseteq \upsilon$.

To condense all these requirements into a single condition

$$\mathsf{PIH}(\pi, \{\mathsf{r}_{s+1}, \ldots, \mathsf{r}_{s+n}\}, \upsilon),$$

it suffices to put[3]

$$\begin{array}{lcl} \mathsf{PIH}(P, \{R_1, \ldots, R_n\}, Y) & \equiv_{\text{Den}} & \emptyset \;=\; \bigcup_{i=1}^{n} \Big(\bigcup_{j=i+1}^{n} P \cap R_i \,;\, \mathbb{1} \cap\ R_j \,;\, \mathbb{1} \\ & & \qquad\qquad \cup\, P \cap (R_i \cap R_i \,;\, \overline{\iota}) \,;\, \mathbb{1} \\ & & \qquad\qquad \cup\, R_i \cap P \,;\, \mathbb{1} \cap \mathbb{1} \,;\, P \\ & & \qquad\qquad \cup\, P \,;\, R_i \cap \overline{\iota} \,;\, R_i \cup R_i \,;\, P \cap R_i \,;\, \overline{\iota} \\ & & \qquad\qquad \cup\ \bigcup_{j=i}^{n} P \cap R_i \,;\, \mathbb{1} \cap R_j^{\smile} \,;\, \mathbb{1} \Big) \\ & & \&\ \ P \subseteq Y\ . \end{array}$$

[3] We are now getting rid of redundant parentheses occurring in the preceding map equalities.

3.3 Attributes, keys, and database entities

Suppose that C and $\lambda, \varrho, \upsilon, \varepsilon$ meet the conditions $\mathsf{Coll}(C)$ and $\mathsf{HdTl}(\lambda, \varrho, \upsilon, \varepsilon)$ specified above, and let $\mathbf{C}$ be a collection of atomic entities, named C. Thinking of an attribute on $\mathbf{C}$ just as a function partially defined on it whose values are atomic, we could specify that α designates such an attribute $\boldsymbol{\alpha}$ simply by stating

$$\mathsf{Func}(\alpha) \ \& \ \mathsf{dom}(\alpha) \subseteq C \ \& \ \mathsf{img}(A) \subseteq \upsilon.$$

For technical reasons, we prefer $\boldsymbol{\alpha}$ to assign a value, perhaps a "dummy" one, to every entity e in $\mathbf{C}$. Assuming that at most one dummy, d_α, is needed for α (different, as a general rule, from the dummies of other attributes, and also distinct from place-holders that may be used to represent unknown values), let δ_α designate either $\emptyset$ or $\{\langle d_\alpha, d_\alpha \rangle\}$ depending on whether d_α is or is not exploited at some time; accordingly, we require that $\mathsf{Attr}(\alpha, C, \delta_\alpha, \upsilon)$ is met, where

$$\begin{array}{lll} \mathsf{Attr}(A, C, D, Y) & \equiv_{\mathrm{Den}} & \mathsf{Func}(A) \ \ \& \ \ \mathsf{dom}(A) = C \\ & & \quad \& \ \ \mathsf{Dangl}(D) \ \ \& \ \ D \subseteq \mathsf{img}(A) \subseteq Y. \end{array}$$

In the case of a *mandatory* attribute β, the condition becomes $\mathsf{Attr}(\beta, C, \emptyset, \upsilon)$, which is equivalent to

$$\mathsf{Func}(\beta) \ \ \& \ \ \mathsf{dom}(\beta) = C \ \ \& \ \ \mathsf{img}(\beta) \subseteq \upsilon.$$

When $\alpha_0, \dots, \alpha_k$ are distinct attributes which form a key of C, the conditions $\boldsymbol{\alpha}_0(c) = a_0, \dots, \boldsymbol{\alpha}_k(c) = a_k$ identify at most one entity c in $\mathbf{C}$ for any $(k+1)$-tuple of values a_i. Moreover such values are mandatory, in the strong sense that none of them can ever be dummy or can be superseded by a placeholder indicating that it is temporarily unknown. This can be stated as

$$\mathsf{Key}(\{\alpha_0, \dots, \alpha_k\}, \pi) \ \ \& \ \ \&_{i=0}^{k} \mathsf{Attr}(\alpha_i, C, \emptyset, \upsilon),$$

where π designates the collection of all placeholders (cf. Sec.3.2 above) and

$$\begin{array}{lll} \mathsf{Key}(\{A_0, \dots, A_k\}, P) & \equiv_{\mathrm{Den}} & \mathsf{Func}\big(\bigcap_{i=0}^{k} (i+1)^{\mathsf{th}}(\lambda, \varrho) \ ; A_i^{\smile} \big) \\ & & \qquad \& \ \ \&_{i=0}^{k} P \cap \mathsf{img}(A_i) = \emptyset \, . \end{array}$$

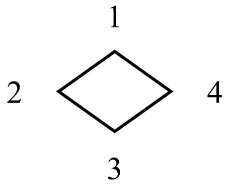
A class of entities of the same type, once an ordering $A_1, \dots, A_h$ of its distinct attributes has been fixed, can be represented as a function F such that $\mathsf{Ens}(F, \pi, \lambda, \varrho, \upsilon, \delta_{A_1}, \dots, \delta_{A_h})$, where

$$\begin{array}{ll} \mathsf{Ens}(F, P, L, R, Y, D_1, \dots, D_h) & \equiv_{\mathrm{Den}} \\ & \mathsf{dom}(F) \cap (\, P \cup \mathsf{img}(R)\,) = \emptyset \\ & \ \& \ \ \mathsf{img}(F) \subseteq h\mathsf{-tuples}(R) \\ & \ \& \ \ \&_{i=1}^{h} \mathsf{Attr}\big(F \ ; i^{\mathsf{th}}(L, R), \mathsf{dom}(F), \ D_i \ , \ Y\big) \, . \end{array}$$

Notice that the condition $\mathsf{Func}(F)$ needs not to be made explicit here, as it holds in consequence of the requirements $\mathsf{img}(F) \subseteq h\text{-}\mathsf{tuples}(\varrho)$ and $\&_{i=1}^{h} \mathsf{Func}(F; i^{\mathsf{th}}(\lambda, \varrho))$.

4 An ER-model is given...

In this initial study, let us convene that every diamond represents a *dyadic* relation —a map in our terminology— and that no diamond bears any attributes.
We need to regard one parameter of a relation as first, and the other one as second, which we do according to the numbering stipulation



Uniqueness of names is assumed to its fullest extent; viz., we are forbidding synonymy not only between (entity-)boxes, between diamonds and between ovals, but also between a box and a diamond etc.

We assume the IsA-graph to be acyclic; i.e.,

$$\mathsf{IsA}^{\smile} \cap \bigtimes_{i=1}^{n} \mathsf{IsA} \;=\; \emptyset$$

must hold for all $n \geq 0$. Moreover, we assume that at most one IsA-edge exits from each box:

$$\mathsf{IsA}^{\smile} \,;\, \mathsf{IsA} \;\subseteq\; \boldsymbol{\iota}_{\text{boxes}}.$$

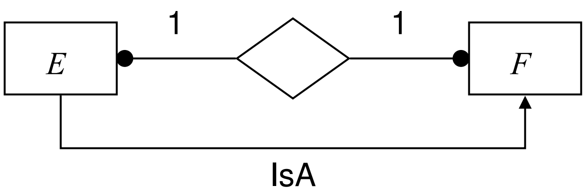
The latter assumption reflects two facts: on the one hand, we are taking into account *single inheritance* only; on the other hand, we are *forbidding shortcuts* in inheritance chains, by requiring that

$$\mathsf{IsA} \cap \bigtimes_{i=1}^{n+2} \mathsf{IsA} = \emptyset$$

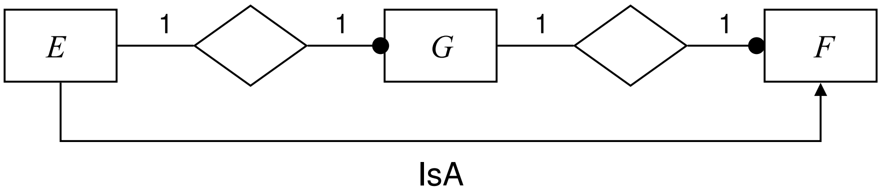
holds for all $n \geq 0$.
Since an acyclic function defined on a finite set (the set of boxes in our case) cannot be total, the IsA-graph will have nodes of outdegree 0. Such “sink” nodes —which represent objects of maximal genericity— are usually called *roots* in the literature on the object-oriented paradigm. Clearly, there would be no loss of generality in assuming that there is *exactly one root*.

4.1 Digression on a little anomaly that may be caused by IsA

An IsA-edge connecting box E to box F indicates that the set $\mathbf{E}$ of entities designated by E is included at all times in the one, $\mathbf{F}$, designated by F. Unless this inclusion could be satisfied, at least occasionally, as a strict inclusion, distinguishing between E and F would hardly make any sense, and the ER-model would be somehow redundant.
Notice, however, that the following situation implies that E and F designate the same entity:



(Indeed, $\mathbf{F} = \mathbf{E}$ ensues from $|\mathbf{E}| = |\mathbf{F}|$ and $\mathbf{E} \subseteq \mathbf{F}$ when $\mathbf{F}$ is known to be finite).
Generalizations of this situation are easily found, e.g.

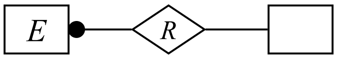


(where $\mathbf{F} = \mathbf{E}$ ensues from $|\mathbf{F}| \leq |\mathbf{G}| \leq |\mathbf{E}|$ and $\mathbf{E} \subseteq \mathbf{F}$ in view of the finiteness of $\mathbf{F}$—note that $\mathbf{G} = \mathbf{E}$ is not entailed, though!). Admitting slightly defective ER-models of the above kind would cause some marginal conceptual difficulties in the part of this research which regards 'dynamics', momentarily postponed to another paper. However, since we have the reasonable expectation that any similar anomaly can be revealed by a simple semantic check, so as to be then corrected, we assume that this check is available to us and we do not digress any further on this issue.

5 Normalization of an ER-model

We will present below a simple preprocessing algorithm that converts any given ER-model (devoid of relation-attributes) into one that better fits our translation purposes.
Thanks to our normalization process, *tight domain* and *image* of each map —in the sense to be explained now— will be at hand in an ER-model $\mathcal{M}$; this is to say, they will be endowed with an explicit name.

Definition 2 *We call E the* TIGHT DOMAIN *of R in $\mathcal{M}$ if $\mathcal{M}$ requires that*

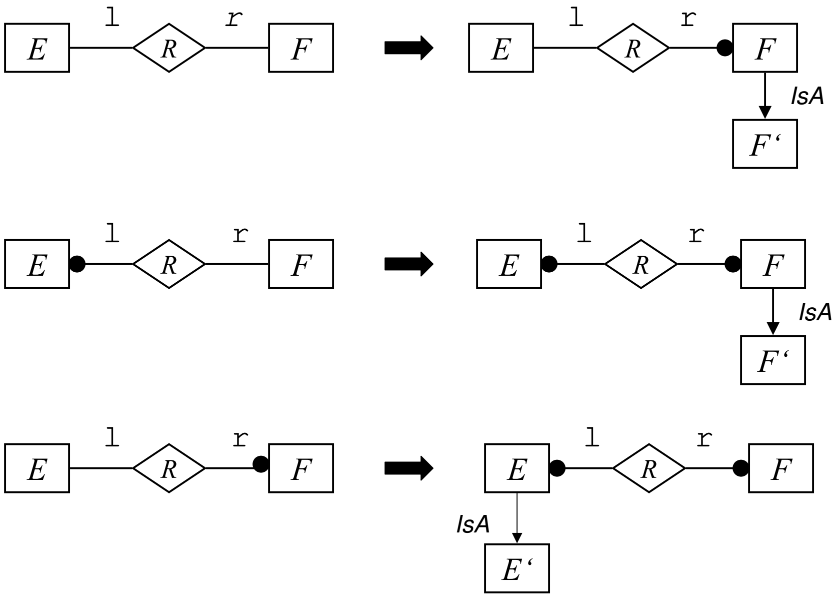


The definition of TIGHT CODOMAIN*, or briefly* IMAGE*, is symmetric.*

Here is the preprocessing algorithm:

Phase 0: Detect all pairs E, F with $E \not\equiv F$ such that, due to the IsA-anomaly, E and F must designate the same class of entities. For every such pair, coalesce E with F.

Phase 1 (Factorization): Make all relations both left- and right-total, by repeated use of the rewriting rules below (where $\ell, \mathtt{r} \in \{1, *\}$, and E', F' are fresh names):



6 Renderings of ER-models in mapalgebra

Here is one way of translating into a set of map equalities an ER-model $\mathcal{M}$ resulting from the normalization process presented above:

A. Introduce p.w. distinct symbols $\pi, \lambda, \varrho, \varepsilon$ which also are distinct from υ and from any identifier in $\mathcal{M}$. For each identifier A of an *optional* attribute in $\mathcal{M}$ (this ignores, among others, any attribute which is in a key), introduce a brand new symbol δ_A. Postulate that

$$\mathsf{HdTl}(\lambda, \varrho, \upsilon, \varepsilon) \quad \& \quad \mathsf{PlH}(\pi, \Re, \upsilon)$$

(cf. Sections 3.1 and 3.2), where $\Re$ is the set of all attribute-, and relation-identifiers in $\mathcal{M}$ and of all new symbols δ_A that have been associated with optional attributes.

We are, of course, regarding the elements of $\Re \cup \{\pi, \lambda, \varrho, \upsilon, \varepsilon\}$ —as well as the entity identifiers in $\mathcal{M}$— as symbols drawn from the alphabet of map letters r_i, typed in a different font. This identification with map letters induces an ordering between identifiers which we will tacitly call into play when saying, e.g., “let $A_1, \ldots, A_h$ be all distinct attribute identifiers that refer to the entity class represented by E” (in such a case will assume the function $j \mapsto i_j$ to preserve order, where i_j is the subscript of A_j among map letters).

B. For every IsA-edge $E \rightarrowtail F$ in $\mathcal{M}$, impose that $\mathsf{dom}(E) \subseteq \mathsf{dom}(F)$. For every entity identifier F which has no issuing IsA-edge, impose that $\mathsf{dom}(F) \subseteq \upsilon$.

C. For every entity identifier F, let $A_1, \ldots, A_h$ be all distinct attribute identifiers that refer to F in $\mathcal{M}$. Impose that

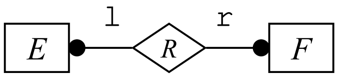
$$\mathsf{Ens}(F, \pi, \lambda, \varrho, \upsilon, \delta_{A_1}, \ldots, \delta_{A_h}) \quad \& \quad \&_{i=0}^{h} A_i = F \,;\, i^{\mathsf{th}}(\lambda, \varrho)$$

(cf. Sec.3.3), where $\delta_{A_i} \equiv_{\mathrm{Den}} \emptyset$ for each mandatory attribute A_i. Moreover, if A_{i_0} together with $A_{i_1}, \ldots, A_{i_k}$ constitutes a key for F, then require that $\mathsf{Key}(\{A_{i_0}, A_{i_1}, \ldots, A_{i_k}\}, \pi)$.

D. For every pair A, B of distinct attributes of $\mathcal{M}$ where A is optional, impose that $\delta_A \cap \mathsf{img}(B) = \emptyset$.

D′. For every pair A, F where A identifies an optional attribute in $\mathcal{M}$ and F identifies an entity class in $\mathcal{M}$, impose that $\delta_A \cap \mathsf{dom}(F) = \emptyset$.

E. For every part



of $\mathcal{M}$, impose that
$$\mathsf{dom}(E) = \mathsf{dom}(R) - \pi \quad \& \quad \mathsf{dom}(F) = \mathsf{img}(R) - \pi \ .$$
Moreover, if $r \equiv 1$, then impose that $\mathsf{Func}(R)$; and, if $\ell \equiv 1$, then impose that $\mathsf{Func}(R^{\smile})$.

Definition 3 *Let $\mathcal{M}^{\mathrm{r}}$ be the conjunction of all map constraints resulting from the above translation algorithm applied to $\mathcal{M}$; moreover, let $\mathcal{M}^{\mathrm{s}}$ and $\mathcal{M}^{\mathrm{w}}$ be the analogous conjunctions that result when the main condition in step E of the translation is either strengthened into*
$$\mathsf{dom}(E) = \mathsf{dom}(R) \quad \& \quad \mathsf{dom}(F) = \mathsf{img}(R) \ .$$
or weakened into
$$\mathsf{dom}(R) - \pi \subseteq \mathsf{dom}(E) \quad \& \quad \mathsf{img}(R) - \pi \subseteq \mathsf{dom}(F) \ .$$
respectively.
Then $\mathcal{M}^{\mathrm{r}}$, $\mathcal{M}^{\mathrm{s}}$, and $\mathcal{M}^{\mathrm{w}}$ are called the RELAXED, STRICT, *and* WEAK COUNTERPART *of $\mathcal{M}$ in mapalgebra.*

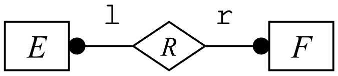
Our rationale for adopting a cautious translation such as $\mathcal{M}^{\mathrm{r}}$ as the official one, in spite of $\mathcal{M}^{\mathrm{s}}$ appearing to be more faithful, will clearly emerge in our next paper of this series (it was only briefly hinted at in our discussion on placeholders in Sec.3.2). In the meantime, the following two lemmas show that even the weakest of the three translations, namely $\mathcal{M}^{\mathrm{w}}$, is in a sense acceptable.

Lemma 1 *Let $\Im$ be an interpretation satisfying $\mathcal{M}^{\mathrm{w}}$ so that $\pi^{\Im} \setminus \bigcup_{i \in \mathbb{N} \setminus \{0\}} \mathsf{r}_i^{\Im}$ has infinite cardinality.*
Then there is an interpretation $\Im'$ such that
- *$P^{\Im} \subseteq P^{\Im'}$ for all map letters P;*
- *$P^{\Im'} = P^{\Im}$ for all map letter P* not *acting as a relation identifier in $\mathcal{M}$;*
- *$\Im'$ satisfies $\mathcal{M}^{\mathrm{r}}$.*

Lemma 2 *Let $\Im$ be an interpretation satisfying $\mathcal{M}^{\mathrm{r}}$ so that the domain of discourse of $\Im$ can be decomposed as $\mathcal{U} = \mathcal{A} \cup \mathcal{A}^*$ (where $\mathcal{A} = \{a \in \mathcal{U} \mid \langle a, a \rangle \in \upsilon^{\Im}\}$).*
Then there is an interpretation $\Im'$ over an extended domain $\mathcal{U}' = \mathcal{B} \cup \mathcal{B}^*$ such that
- *$P^{\Im} \subseteq P^{\Im'}$ for all map letters P;*
- *$P^{\Im'} \cap (\mathcal{U} \setminus \pi^{\Im})^2 = P^{\Im} \setminus \pi^2$ for all map letter P;*
- *$\Im'$ satisfies $\mathcal{M}^{\mathrm{s}}$.*

Remark 1 *A variant version of step E (of the relaxed translation, which also happens to be our favorite translation) would fit very well with our purposes too. Here it is:*

E′. *For every part*



of $\mathcal{M}$, *impose that*

$$\begin{array}{rl} E = \lambda^{\smile} \,;(R\,;\mathbb{1} \cap \varrho) & \& \quad F = \lambda^{\smile} \,;(R^{\smile}\,;\mathbb{1} \cap \varrho) \\ & \& \quad \mathsf{img}(R) \cup \mathsf{dom}(R) \subseteq \pi \cup \mathsf{img}(\varrho)\ . \end{array}$$

Moreover, if $\mathrm{r} \equiv 1$, *then impose that* $\mathsf{Func}(R)$; *and, if* $\ell \equiv 1$, *then impose that* $\mathsf{Func}(R^{\smile})$.

Indicating by $\mathcal{M}^{\mathrm{r}'}$ *the result of the new translation, it is easily seen that interpretations of mapalgebra satisfying* $\mathcal{M}^{\mathrm{r}}$ *induce interpretations satisfying* $\mathcal{M}^{\mathrm{r}'}$ *and conversely. This appears more clearly if we replace* R *by a new map letter* R' *in* $\mathcal{M}^{\mathrm{r}'}$ *and then observe that an* R *complying with the old translation and an* R' *complying with the new one correspond to each other as follows:*

$$\begin{array}{rcl} R & = & (\lambda^{\smile} \cup \pi)\,; R'\,; (\lambda \cup \pi)\ , \\ R' & = & \big(\pi \cup \big(\lambda \cap \varrho\,; \bigcap_{i=1}^{n}\ i^{\mathsf{th}}(\lambda, \varrho)\,; A_i^{\smile}\,\big)\big)\,; R \\ & & ;\big(\pi \cup \big(\lambda \cap \varrho\,; \bigcap_{j=1}^{m}\ j^{\mathsf{th}}(\lambda, \varrho)\,; B_j^{\smile}\,\big)\big)^{\smile}\ , \end{array}$$

where $A_1, \ldots, A_n$ *and* $B_1, \ldots, B_m$ *are the respective attributes of* E *and* F.

7 Related Work

This study stands in line with other approaches to provide a firmly based semantics for ER-models. They are mainly based on algebraic modeling techniques and capitalize on the semantic framework that come with them. Hettler [9] gives a translation of these models into the specification language SPECTRUM, essentailly modeling entities as records with attributes as entries, but not taking inheritance into account. The report [2] transforms an ER-diagram into an attributed graph signature, and the integrity constraints into first-order logic formulas. The ER-models considered do not take inheritance explicitly into account. The paper focusses on the dynamic aspects — viz., transactions – through homomorphisms, hence showing how transactions may be caught through an algebraic framework. The formal semantics of an extended ER-model is investigated in [7, 10] from a database point of view, proposing the semantics of a database signature as the set of all interpretations; this work does not mention algebraic specifications explicitly. In [4] it is shown how to generate an algebraic specification from an ER-model, hereby carrying the model based semantics of such a specification over to ER-models. [5] generalizes this approach somewhat by proposing the colimit of a diagram extracted from an ER-model as the categorial semantics of the model.

References

[1] D. Cantone, E. G. Omodeo, and A. Policriti. *Set Theory for Computing*. Springer-Verlag, 2001. in print.

[2] I. Claßen, M. Löwe, S. Waßerroth, and J. Wortmann. Static and dynamic semantics of entity-relationship models based on algebraic methods. Technical report, Department of Computer Science, Technical University, Berlin, 1994.

[3] E. F. Codd. A relational model for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[4] E.-E. Doberkat. Generating an algebraic specification from an ER-model. *International Journal of Software Engineering and Knowledge Engineering*, 7(4):525–552, 1997.

[5] E.-E. Doberkat. The categorial semantics of ER-models. Technical report, Chair for Software Technology, University of Dortmund, 1999.

[6] A. Formisano, E.G. Omodeo, and M. Temperini. Goals and benchmarks for automated map reasoning. *Journal of Symbolic Computation*, 29(2):259–297, 2000.

[7] M. Gogolla and U. Hohenstein. Towards a semantic view of an extended entity-relationship model. *ACM Transactions on Database Systems*, 16:369–416, 1991.

[8] D. Gries and F.B. Schneider. *A logical approach to discrete math.* Texts and Monographs in Computer Science. Springer-Verlag, 1994.

[9] R. Hettler. Zur Übersetzung von E/R-Schemata nach SPECTRUM. Technical Report TUM I-9333, Technical University, Munich, 1993.

[10] U. Hohenstein. *Formale Semantik eines erweiterten Entity-Relationship Modells.* B. G. Teubner, Stuttgart und Leipzig, 1993.

[11] M. Page-Jones. *Fundamentals of Object-Orientied Design in UML.* Dorset House Publishing & Addison-Wesley, New York and Boston, 2000.

[12] E. Schröder. *Vorlesungen über die Algebra der Logik (exakte Logik)*, volume 2.1. B. Teubner, 1891. Reprinted by Chelsea Publishing Co., New York, 1966.

[13] E. Schröder. *Vorlesungen über die Algebra der Logik (exakte Logik)*, volume 3, Algebra und Logic der Relative, part 1. B. Teubner, Leipzig, 1895. Reprinted by Chelsea Publishing Co., New York, 1966.

[14] A. Tarski and S. Givant. *A formalization of set theory without variables*, volume 41 of *Colloquium Publications.* American Mathematical Society, 1987.

[15] B. Thalheim. *Entity-Relationship Modeling: Foundations of Database Technology.* Springer-Verlag, 2000.