ON THE CORRECTNESS OF REPRESENTING EXTENDED ENTITY–RELATIONSHIP STRUCTURES IN THE RELATIONAL MODEL

Victor M. Markowitz and Arie Shoshani
Computer Science Research Department
Lawrence Berkeley Laboratory, Berkeley, CA 94720

Abstract. Although the relational representation of *Entity–Relationship* (ER) structures gained extensive coverage, scarce attention has been paid to the issue of *correctness* for such representations. Several mappings have been proposed for the representation of both ER and extended ER (EER) structures by relational schemas. The informal nature of most of these proposals, however, does not allow a precise evaluation of their correctness, nor a comparison of the various mappings. We propose a *canonical* relational representation for EER structures and prove its correctness. We claim that a relational schema represents correctly an EER structure if it has *equivalent* information-capacity with the corresponding canonical representation.

The second problem addressed by this paper is the normalization of relational schemas that represent EER structures. We examine the conditions required by this process and show that ignoring these conditions leads to erroneous analyses and inappropriate design decisions. We show that, under these conditions, the canonical relational representation of any (unrestricted) EER structure has an (information-capacity) equivalent *Boyce-Codd Normal Form* schema.

1. Introduction.

There are two basic approaches to the problem of relational database design. The first approach is the *Universal–Relation* (UR) approach (cf. [2]). The UR approach assumes that all the semantics are captured by dependencies expressed over a universal set of attributes and the user does not need to be aware of how the attributes are grouped into relations. The UR approach lead to the well known *normalization* methodologies for database design ([2], [15], [20]). These methodologies, however, proved to be difficult to use and not entirely reliable, mainly because they required the expression of database semantics only by specifying dependencies between attributes and because the underlying UR assumptions make the attribute naming critical and excessively complex. Furthermore, it has been shown that inter-relation constraints (e.g. referential integrity [6]) must be specified in the relational model in order to insure proper integrity of the database and such constraints are disregarded by normalization.

The difficulties encountered by the relational model in providing a suitable framework for database design lead to the development of new *semantic* data models that have more semantic intuition (cf. [10]). One of the most popular semantic models is the *Entity–Relationship* (ER) model [5]. It is generally agreed that the concepts of *objects* (entities and relationships) and their properties are natural in designing databases. The ER model is widely accepted as a design tool for relational databases: the ER-oriented design consists of the specification of an ER schema which is subsequently mapped into a relational schema. We refer in this paper to the *extended ER* (EER) model [19] which includes, in addition to the basic constructs of the ER model, the generalization and full aggregation abstraction capabilities.

At first glance, it seems reasonable to try and find a mapping of EER schemas directly into relational schemas which are in the highest possible normal form. However, there are several aspects of such a mapping that can be confusing when considered together. Specifically, the correctness of the result of such a mapping is independent of the assignment of names to the relational attributes generated during the mapping. Indeed, correctness here refers to the preservation of the structural semantics of the EER schema. Consider, for example, relationship-set TEACH associating entity-sets FACULTY and COURSE. The structural aspect of mapping TEACH must ensure that the corresponding relational representation reflects the involvement of entity-sets FACULTY and COURSE in TEACH. The second aspect involves assigning names to the relational attributes representing EER attributes. For instance, suppose that entity-set COURSE has an attribute NUMBER and that this attribute is represented by a relational attribute in the relation-scheme corresponding to TEACH. Then this relational attribute can be assigned the name NUMBER, COURSE.NUMBER, TEACH.COURSE.NUMBER, just to name a few reasonable options. Unlike the mapping of EER schemas into relational schemas, normalization relies on the assignment of names to relational attributes and ignoring this dependency can lead to incorrect designs. We give in section 2 several examples that illustrate the issues mentioned above and review some of the existing approaches.

In this paper we examine two issues: (i) what constitutes a *correct* relational schema representation of a given EER schema, and (ii) once a relational schema represents correctly some EER schema, under what conditions can normalization be applied to this relational schema. We discuss the criteria for correctness and propose a canonical relational representation for EER schemas that satisfies these criteria. Consequently, a relational schema is said to represent correctly an EER schema if it has *equivalent* information-capacity with the corresponding canonical representation. Next, we show that the assumptions underlying normalization impose certain restrictions on the assignment of names to the attributes of relational schemas representing EER schemas. Subsequently, we show that any (unrestricted) EER schema can be correctly represented by a *Boyce-Codd Normal Form* (BCNF) schema. Our discussion is independent of a specific strategy for assigning names to relational attributes. The selection of such a strategy could be based on different criteria, such as brevity, clarity of names, and the restrictions mentioned above. We consider the problem of selecting such strategy beyond the scope of this paper and leave it for further investigation.

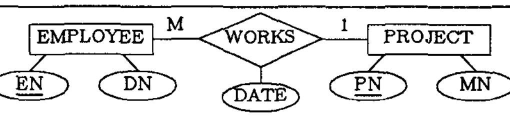
The rest of the paper is organized as follows. In section 3 we sketch the concepts and notations used in this paper. Section 4 contains a brief discussion on EER schemas and their well-definedness. In section 5 we propose a canonical relational representation for EER schemas and prove its correctness. In section 6 we examine the impact of the assumptions underlying normalization on the canonical representation. In section 7 we present the mapping that transforms the canonical representation of any EER schema into an equivalent BCNF schema. Unlike the standard normalization based on functional dependencies, the normalization procedure presented in this section considers inclusion dependencies as well, because the relational schemas that represent EER schemas involve both functional and inclusion dependencies. We conclude with a summary. Due to space limitations we omit or only sketch the proofs. The full version of this paper appears in [18].

2. Relational Representations For EER Structures.

2.1 Correctness Of Relational Representations.

Several mappings have been proposed for the representation of both ER and EER structures by relational schemas (e.g. [5], [11], [19]). The informal nature of most of these proposals, however, does not allow a precise evaluation of their correctness, nor a comparison of the various mappings. Consider, for example, the simple ER structure of figure 1(i) represented by an ER diagram in the usual way (entity-identifiers and relational keys are underlined). Mappings such as that of [19] generate for this ER structure the relational schema of figure 1(ii). Yet another possible representation, following [11], of the same ER structure is the relational schema of figure 1(iii), where every ER object-set is represented by a separate relation-scheme. The natural question is which representation is the *correct* one, and if both are correct how is the *equivalence* of such representations defined. We show in this paper that solutions such as those of figures 1(ii) and 1(iii) are incorrect because they allow the association of states which are inconsistent with respect to the ER semantics. Consider the following simple example which illustrates the violation of an integrity constraint implied by the ER structure. The relational schema of figure 1(ii) can be associated with state $s = \{r_{EMPLOYEE} = [e1\ d1\ p1\ y1],\ r_{PROJECT} = [p2\ e2]\}$. Although s is consistent with respect to the specified keys, it is inconsistent with respect to the semantics of the ER structure: tuple (e1 d1 p1 y1) represents a relationship of relationship-set WORKS which associates a project entity, p1 , that is not represented in $r_{PROJECT}$. This can be fixed by adding to the relational schema of figure 1(ii) an inclusion dependency, namely EMPLOYEE[PN] ⊆ PROJECT[PN] , which will not allow states like s , but will also not allow the insertion of employees which are not associated with some project, contrary to the semantics of the corresponding ER structure. This, in turn, can be fixed by allowing *null* values for attribute PN in relation EMPLOYEE. The main problem with this solution is that the introduction of null values significantly

(i)



(ii)	EMPLOYEE	(EN,DN,PN,DATE)
	PROJECT	(PN,MN)
(iii)	EMPLOYEE	(EN,DN)
	WORKS	(EN,PN,DATE)
	PROJECT	(PN,MN)
(iv)	WORKS [EN]	⊆ EMPLOYEE [EN]
	WORKS [PN]	⊆ PROJECT [PN]

Abbreviations: EN=EMPLOYEE NAME, DN=DEPARTMENT NUMBER, MN=MANAGER NAME, PN=PROJECT NUMBER

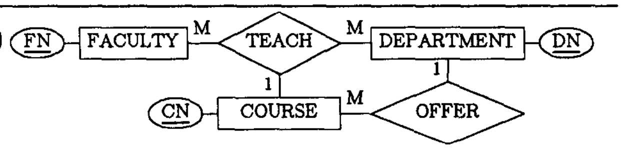
Fig.1 Relational Representations for an ER Structure.

complicates the representation. Thus, additional constraints should be satisfied in order to allow states that are consistent with the semantics of the ER structure. In the schema of figure 1(ii), for example, since DATE characterizes relationship-set WORKS, the corresponding homonym relational attribute should be constrained to have a null value whenever PN has a null value in relation EMPLOYEE. Moreover, the introduction of nulls requires the definition of fairly complex update procedures (see [1] for a discussion on this topic).

Following the same reasoning as above, we can conclude that the relational schema of figure 1(iii) is missing the inclusion dependencies of figure 1(iv). Indeed, it can be shown that the schema of figure 1(iii) together with the inclusion dependencies of figure 1(iv) provide a correct representation for the ER structure of figure 1(i). We still could ask whether this is the only correct representation and how is a correct representation recognized. For example, following [11], both the relational schemas of figures 1(ii) and 1(iii) can represent the ER schema of figure 1(i), and they are equivalent. The problem is that equivalence is defined in [11] without taking inclusion dependencies into account and under the stringent *Pure Universal Instance Assumption* [16]. In this paper we define correctness and equivalence for the relational representations of EER schemas which avoid these problems.

2.2 Assumptions Underlying Normalization.

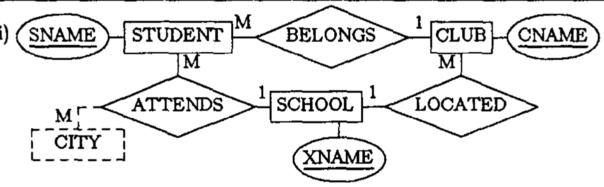
The ER and EER-oriented design of relational schemas was not intended to replace entirely the UR-oriented design, but rather to complement it. The design methodology proposed in [19], for instance, combines the EER-oriented design with relational normalization: relational schemas that represent EER structures are analyzed in order to establish their normal form and, if necessary, are further normalized. The combination of the two design approaches, however, is not simple and could be confusing. Indeed, different authors reach contradictory results concerning the possibility of such a combination, namely that it can be achieved (i) by first modifying the ER structures [3]; (ii) without restricting the allowed ER structures [11]; and (iii) only by restricting the allowed ER structures [1]. We refer below to the last two conclusions. The mapping of [11] allows the representation of multiple relationship-sets involving the same entity-sets. Thus, following [11], the ER structure of figure 2(i) would be represented by the relational schema of figure 2(ii). This representation, however, does not satisfy one of the UR assumptions, namely the

(i) 

(ii)	FACULTY	(FN)		
	COURSE	(CN)	OFFER	(CN, DN)
	DEPARTMENT	(DN)	TEACH	(FN, DN, CN)

Abbreviations: CN=COURSE NUMBER, DN=DEPARTMENT NAME, FN=FACULTY NAME

Fig.2 Relational Representations for ER Structures.

(i) 

(ii) STUDENT	(SNAME)		BELONGS	(SNAME, CNAME)
CLUB	(CNAME)		ATTENDS	(SNAME, XNAME)
SCHOOL	(XNAME)		LOCATED	(CNAME, XNAME)

Fig.3 Relational Representation for an ER Structure.

Unique Role Assumption (URA) [16]. URA constraints a set of attributes to represent at most one ER object-set. In the relational schema of figure 2(ii), for instance, the association of the attribute set {CN,DN} with both the OFFER and TEACH relations implies under URA that TEACH is a relationship-set that associates FACULTY entities with OFFER relationships, as shown later in figure 4. Consequently, the schema of figure 2(ii) distorts the semantics of the ER structure of figure 2(i) where relationship-sets TEACH and OFFER are independent. Unlike [11], [1] contends that only restricted ER structures (whose ER diagrams have no undirected cycles) can be represented by relational schemas that satisfy URA. This means, for instance, that structures such as that shown in figure 2(i) cannot be represented by relational schemas that satisfy URA. In fact, this restriction can be avoided. We show in this paper that the EER-oriented and UR-oriented design approaches can be combined without either restricting or modifying the original EER schemas, but only by selecting for them appropriate relational representations.

Satisfying the UR assumptions that underlie normalization should not be taken lightly, since an inaccuracy, no matter how small, leads to erroneous conclusions. For example, based on the mapping of [11], [12] concludes that in order to guarantee BCNF for relational schemas that represent ER structures, multiple relationship-sets between the same entity-sets should be disallowed. Thus, the relational schema of figure 2(ii) does indeed contain a relation (TEACH) which is not in BCNF apparently because in the ER schema of figure 2(i) entity-sets COURSE and DEPARTMENT are involved together in different relationship-sets. However, since the mapping of [11] does not satisfy URA, this result is based on an incorrect analysis. In this paper we show that the canonical relational representation of any EER structure has an (information-capacity) equivalent BCNF schema, and therefore there is no need to restrict the EER structures in order to guarantee BCNF.

Disregarding the UR assumptions also leads to erroneous analyses. For example, suppose that the relational schema of figure 3(ii) represents the ER structure of figure 3(i). Under the UR assumptions the key dependencies in this schema imply that there are two ways of deriving the school associated with some student and both ways must give the *same* value. In other words, under these assumptions the relational schema of figure 3(ii) carry the implied constraint that some student can attend only the school in which the club he belongs to is located. This conclusion is reached in [19] for a similar ER structure, but without the mention of the UR assumptions on which such a conclusion must be based. Furthermore, this conclusion clearly misinterprets the ER structure in which all relationship-sets are considered independent, as correctly asserted elsewhere in [19]. At fault in this case is the mismatch of the ER-oriented design with normalization: while the assumptions underlying normalization restrict the association of attributes with multiple relation-schemes, these restrictions are ignored in [19] when ER schemas are mapped into relational schemas. Consider an additional example of the effects of such a mismatch. Suppose that the relationship-set ATTENDS of figure 3(i) is extended with the association, with a *many* cardinality, of the additional entity-set CITY, as shown in figure 3(i). Then the extended relationship-set is represented by relation-scheme ATTENDS'(SNAME,CITY,XNAME) which replaces the relation-scheme ATTENDS in the relational schema of figure 3(ii). A simple examination reveals that ATTENDS' is not even in Second Normal Form [20] because of the derived partial dependency of XNAME on SNAME and, apparently, is a candidate for decomposition. Here, disregarding the UR assumptions leads to a wrong design decision.

3. Preliminary Definitions and Notations.

We use in this paper some basic graph-theoretical concepts (e.g. see [8]). We denote by $G = (V, H)$ a directed graph (***digraph***) with set of vertices V and set of edges H, and by $v_i \rightarrow v_j$ a directed edge, h, from vertex v_i to vertex v_j; h is said to be ***incident*** from v_i to v_j. The ***underlying*** (undirected) graph of a digraph results from the digraph by ignoring the edge directions. An undirected ***path*** from (***start***) vertex v_{i_0} to (***end***) vertex v_{i_m} is a sequence of alternating vertices and edges, $v_{i_0} h_{j_1} v_{i_1} \ldots h_{j_m} v_{i_m}$, such that h_{j_k} is incident from (to) $v_{i_{k-1}}$ to (from) v_{i_k}, $1 \leq k \leq m$. A ***cycle*** is a path whose start and end vertices are the same. A path is called ***simple*** if a vertex appears on it at most once. A path (cycle) is said to be directed if all the edges on the path have the same direction and the first edge is incident *from* the start vertex.

Next, we sketch the relational concepts needed in this paper. Details can be found in any textbook (e.g. [15], [20]) for the basic concepts and in [7] for the theory of inclusion dependencies. We use letters from the beginning of the alphabet to denote attributes and letters from the end of the alphabet to denote sets of attributes. A sequence of attributes (e.g. ABC) denotes the set containing these attributes and a sequence of sets (e.g. XY) denotes the union of these sets. We denote by t a tuple, and by $t[W]$ the sub-tuple of t corresponding to the attribute set W.

A ***relational schema*** is a pair (R, Δ) where R is a set of relation-schemes and Δ is a set of dependencies over R. We consider relational schemas which are associated with set of dependencies $\Delta = F \cup I$, where F and I denote sets of functional and inclusion dependencies, respectively. A *relation-scheme* is a named set of attributes, $R_i(X_i)$, where R_i is the relation-scheme name and X_i denotes the associated set of attributes. On the semantic level, every attribute is assigned a *domain*, and a *relation* (value) r_i, is assigned to every relation-scheme, $R_i(X_i)$. A *database state* associated with (R, Δ) is defined as $r = <D_1,...,D_m, r_1...r_k>$, where D_k denotes a domain and r_i is equal to a subset of the cross-product of the domains corresponding to the attributes of $R_i(X_i)$. Two attributes are said to be *compatible* if they are associated with the same domain, and two sets of attributes, X and Y, are said to be *compatible* iff there exists a one-to-one correspondence of compatible attributes between X and Y.

Let $R_i(X_i)$ be a relation-scheme associated with relation r_i. The *projection* of r_i on a subset of X_i, W, is denoted $r_i[W]$, and generates a relation associated with attribute set W, that is equal to $\{ t[W] \mid t \in r_i \}$. The *renaming* of r_i on an attribute set Y that is compatible with X_i, is denoted *rename* $(r_i, X_i \leftarrow Y)$, and generates a relation associated with attribute set Y, that is equal to $\{ t' \mid t \in r_i, t'[Y] = t[X_i]\}$ (see [15]). Let $R_i(X_i)$ and $R_j(X_j)$ be two relation-schemes associated with relations r_i and r_j, respectively; the *natural join* of r_i and r_j is denoted $r_i \bowtie r_j$, and generates a relation associated with attribute set $X_i X_j$, that is equal to $\{t \mid t[X_i] \in r_i, t[X_j] \in r_j\}$.

Let $R_i(X_i)$ be a relation-scheme associated with relation r_i. A *functional dependency* over R_i is a statement of the form R_i: $Y \rightarrow Z$ where Y and Z are subsets of X_i and R_i is called a *tag*; R_i: $Y \rightarrow Z$ is *satisfied* by r_i iff for any two tuples of r_i, t and t', $t[Y] = t'[Y]$ implies $t[Z] = t'[Z]$.

Let $R_i(X_i)$ and $R_j(X_j)$ be two relation-schemes associated with relations r_i and r_j, respectively. An *inclusion dependency* is a statement of the form $R_i[Y] \subseteq R_j[Z]$, where Y and Z are subsets of X_i and X_j, respectively, and Y and Z are compatible; $R_i[Y] \subseteq R_j[Z]$ is *satisfied* by r_i and r_j iff $r_i[Y] \subseteq r_j[Z]$. The attributes involved in the left-hand side of an inclusion dependency are called *foreign* attributes. The set of inclusion dependencies I over the relation-schemes of R can be represented graphically by the following *inclusion dependency digraph*: $G_I = (V, H)$, where $V = R$, and $R_i \rightarrow R_j \in H$ iff $R_i[Y] \subseteq R_j[Z] \in I$. A set of inclusion dependencies I is said to be *acyclic* iff G_I is acyclic.

Let r be some state associated with schema (R, Δ); r is said to be *consistent* if it satisfies the dependencies of Δ. Given a set of dependencies Δ, a dependency δ is said to be *implied* by Δ if every state that satisfies Δ also satisfies δ. The set of dependencies implied by Δ is called the *closure* of Δ and is denoted Δ^+. A *superkey* associated with R_i is a subset of X_i, K_i, such that $R_i : K_i \rightarrow X_i$ is implied by Δ; K_i is called a *key* iff no proper subset of K_i is also a superkey of R_i. A relation-scheme can be associated with several *candidate keys* from which one *primary key* is chosen. If $R_i[Y] \subseteq R_j[Z]$ is an inclusion dependency and Z is the primary key of R_j then Y is called a *foreign key* of R_i *referencing* R_j.

4. Extended Entity–Relationship Structures.

The concepts of the basic *Entity-Relationship* (ER) model, (*entity, relationship, entity–set, relationship–set, value–set, attribute, entity–identifier, weak* entity-set, relationship *cardinality, role*) have been defined originally in [5] and have been repeatedly reviewed since then (e.g. see [19]). For the sake of brevity, we omit the definitions of these concepts and sketch below only the additional concepts of the *extended ER* (EER) model considered in this paper. There is some confusion as to what constitutes a well-defined EER structure. Since we deal in this paper with the representation of such structures, we refer below to their well-definedness and some assumptions concerning their specification. For details see [18]. We refer commonly to entities and relationships as *objects*.

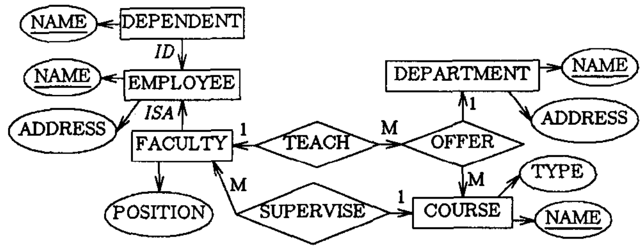


Fig.4 EER Diagram Example (*identifiers are underlined*).

Unlike the basic ER model of [5] the EER model has two additional abstraction capabilities, generalization and full aggregation. *Generalization* is an abstraction mechanism that views a set of entity-sets as a single *generic* entity-set. The inverse of generalization is called *specialization*. A specialization entity-set *inherits* the attributes of all its generic entity-sets, including the entity-identifier. An entity-set which is not specified as the specialization of other entity-sets is called *generalization–source*. In the basic ER model the *aggregation* construct takes three forms: (i) the aggregation of a collection of attributes into an entity-set; (ii) the aggregation of a collection of attributes and the entity-identifiers of several existing entity-sets into a weak entity-set; and (iii) the aggregation of two or more entity-sets into a relationship-set. The basic ER model falls short of providing the *full* capability of aggregation by disallowing relationship-sets to be aggregated further. What is needed in order to provide this capability is simply to allow relationship-sets (as well as entity-sets) to be involved in relationship-sets.

EER-schemas are expressible in a graphic form called *EER diagram* (EERD). Entity-sets, relationship-sets, and attributes, are represented graphically by rectangles, diamonds, and ellipses, respectively. Every vertex is labeled by the name of the represented object-set or attribute. Edges that represent ID-dependencies and generalizations are labeled with *ID* and *ISA* labels, respectively. Roles can be represented by edge labels in the corresponding EER diagram. Unlike in [5], we define the EER diagram as a *directed graph*. The ER diagrams of the basic ER model have an implied directionality, from the entity-set vertices to the attribute vertices and from the relationship-set vertices to the entity-set vertices. In the EER model, however, where entity-sets can be related to other entity-sets by generalization or aggregation, and relationship-sets can be related to other relationship-sets by aggregation, this implied directionality is not enough and the edges of the EER diagrams must explicitly represent the direction of generalizations and aggregations.

An example of an EER diagram for an EER structure is shown in figure 4, where EMPLOYEE, COURSE, and DEPARTMENT, are independent entity-sets, DEPENDENT is a weak entity-set, FACULTY is a specialization of EMPLOYEE, OFFER represents the courses offered by departments (each course is offered by at most one department), SUPERVISE represents the assignment of faculties to supervise courses (each faculty can supervise at most one course), and TEACH represents the assignment of faculties to teach offered courses (each course is taught by at most one faculty).

Now, we shall briefly review certain restrictions concerning the EER structures. A detailed discussion can be found in [18]. The restrictions concerning the specification of generalization structures are also discussed in the survey of [10]. The first restriction refers to disallowing ***directed*** cycles in EER diagrams. While directed cycles involving aggregations are disallowed because the associated structures are devoid of any real-world semantics (e.g. directed cycles connecting weak entity-sets) other cycles indicate the presence of redundant structures (e.g. directed cycles connecting specialization entity-sets). Specialization entity-sets are restricted to have unique generalization-sources and are considered as weak entity-sets iff their generalization-sources are specified as weak entity-sets. The final restriction concerns the ***names*** used for the specification of EER structures. For a given EER structure, object-sets must have unique ***global*** names, attributes must have unique ***local*** names among the attributes associated with the same object-set, and the roles must have unique names among the multiple roles of some object-set in another object-set. Attributes are globally identified by ***global*** names consisting of their local names prefixed by the names of their associated object-sets.

We conclude this section by discussing the ***navigational*** semantics of an EER structure [13]. Let G_{ER} be an EER diagram. A ***navigation—path*** in G_{ER} consists of a simple path between two vertices of the underlying (undirected) graph of G_{ER}. In the EER diagram of figure 4, for example, there are two navigation-paths between FACULTY and DEPARTMENT: via TEACH and OFFER, and via SUPERVISE, COURSE, and OFFER. A navigation-path between two object-set vertices, O_i and O_j, represents a meaningful association between the objects of O_i and O_j [13]. We assume the ***Distinct Meaning Path Assumption (DMPA)*** which states that distinct navigation-paths between any two object-set vertices of an EER diagram represent distinct associations between the respective object-sets. In particular, multiple relationship-sets associating common object-sets have ***distinct meanings***. The two navigation-paths above, for example, represent associations with clearly different meanings. Note that unlike the unique path assumption of [13], DMPA does not restrict the EER structures, but rather states that it is incorrect to assume that distinct navigation-paths represent the same association.

5. Canonical Representations For EER Structures.

As mentioned in the introduction, the mapping of EER schemas into relational schemas has two aspects: (i) to generate a relational schema that represents the structural semantics of the EER schema; and (ii) to assign names to the attributes generated in (i). In this section we are interested in characterizing only the structural aspect of a correct representation which is independent of any specific name assignment for relational attributes. We propose a mapping, called **Crep** (***C****anonical* ***rep****resentation*), which is provably correct. Since **Crep** is independent of a particular attribute naming mechanism, we refer to the resulting relational schema as the ***canonical*** relational representation. As will be shown in the next section, in order to make the result of **Crep** compatible with normalization, the relational attributes must satisfy additional constraints. We define **Crep** in the following three subsections which treat the independent entity-set, aggregation, and generalization constructs. While it is sufficient to represent these constructs using relation-schemes and inclusion dependencies, we also use functional dependencies to represent the identifiers and relationship cardinalities in EER schemas. In order to avoid a specific name assignment for relational attributes, we use in our examples (internal) ***symbols*** for referencing attributes, such as A_{4_2} (representing the second attribute of relation-scheme R_4).

5.1 Independent Entity–Set.

EER value-sets are represented straightforwardly by relational domains. The relational model has an aggregation mechanism, for aggregating attributes to form relation-schemes, similar to the aggregation of EER attributes that defines independent (i.e. neither weak nor specialization) entity-sets. Thus, an independent entity-set, E_i, is straightforwardly represented by a relation-scheme, $R_i(X_i)$, such that the following condition holds:

(E1) X_i is in a one-to-one correspondence with the EER attributes of E_i and the domain associated with an attribute of X_i represents the value-set of the corresponding attribute of E_i.

Relation-scheme R_i is associated with functional dependency $R_i: Z_i \rightarrow (X_i - Z_i)$, where Z_i is the subset of X_i that corresponds to the identifier of E_i.

For example, the three independent entity-sets of the EER structure in figure 4 are represented by the relation-schemes of figure 5(i) through figure 5(iii).

5.2 Object–Set Aggregation.

Unlike the aggregation of EER attributes, the object-set aggregation has no analog relational construct. Let object-set O_i be the aggregation of object-sets O_{i_j}, $1 \leq j \leq m$, and let each object-set O_{i_j} be represented by relation-scheme $R_{i_j}(Y_{i_j})$, $1 \leq j \leq m$, respectively. Then object-set O_i is represented by relation-scheme $R_i(X_i)$ together with inclusion dependencies $R_i[X_{i_j}] \subseteq R_{i_j}[Y_{i_j}]$, $1 \leq j \leq m$, where X_i is the union of two disjoint sets of attributes, X'_i and X''_i, defined below:

(A1) X'_i is in a one-to-one correspondence with the EER attributes of O_i, where the correspondence is specified as in (E1) above;

(A2) $X''_i = \bigcup_{j=1}^{m} X_{i_j}$, is a set of foreign attributes, where attribute sets X_{i_j}, $1 \leq j \leq m$, are pairwise disjoint, and every attribute set X_{i_j} is in a one-to-one correspondence with Y_{i_j}, $1 \leq j \leq m$, such that the domain associated with an attribute of X_{i_j} is equal to the domain of the corresponding attribute of Y_{i_j}.

Relation-scheme R_i is associated with functional dependency $R_i: Z_i X''_i \rightarrow (X_i - Z_i X''_i)$, where Z_i is the subset of X_i that corresponds to the identifier of O_i. If O_i is a relationship-set then for every object-set that is involved in O_i with cardinality ***one***, O_{i_j}, R_i is associated with the additional functional dependency $R_i: (X''_i - X_{i_j}) \rightarrow X_{i_j}$, where X_{i_j} is defined as above.

For example, the four aggregations of the EER structure in figure 4 are represented as shown in figure 5(v) through figure 5(viii). The weak entity-set DEPENDENT, for instance, is represented by relation-scheme $R_5(X_5)$, where $X'_5 = A_{5_1}$ and $X''_5 = X_{5_1} = A_{5_2}A_{5_3}$. The relationship-set OFFER is represented by relation-scheme $R_6(X_6)$, where

X'_6 is empty and X''_6 consists of two disjoint subsets, $X_{6_2} = A_{6_1}A_{6_2}$ and $X_{6_3} = A_{6_3}A_{6_4}$. Note that all the EER attributes of the object-sets that are involved in an aggregation have correspondents in the relation-scheme representing the aggregate object-set. This is caused by the lack of any (relational) key information at this stage; keys are computed in a latter stage and redundant attributes are removed by a normalization process.

5.3 Generalization.

Like object-set aggregation, generalization also lacks an analog relational construct. Let entity-set E_i be the specialization of entity-sets E_{i_j}, $1 \leq j \leq m$, and let E_s be the generalization-source of E_i (recall that specializations have unique generalization-sources). The set of *inherited* attributes of E_i consists of the attributes associated with all the generalizations of E_i. Let E_s be represented by relation-scheme $R_s(Y_s)$ and each entity-set E_{i_j} be represented by relation-scheme $R_{i_j}(Y_{i_j})$, $1 \leq j \leq m$, respectively. Then entity-set E_i is represented by relation-scheme $R_i(X_i)$ together with inclusion dependencies $R_i[X_{i_j}] \subseteq R_{i_j}[Y_{i_j}]$, $1 \leq j \leq m$, where X_i is the union of two disjoint sets of attributes, X'_i and X''_i, defined below:

(G1) X'_i is in a one-to-one correspondence with the set consisting of the EER attributes and the inherited attributes of E_i, where the correspondence is specified as in (E1) above;

(G2) X''_i is in a one-to-one correspondence with $Y''_s \subseteq Y_s$, where Y''_s is defined as in (A2) above, such that the domain associated with an attribute of X''_i is equal to the domain of the corresponding attribute of Y''_s;

(G3) each set of foreign attributes, X_{i_j}, includes X''_i, and is in a one-to-one correspondence of compatible attributes with Y_{i_j}, $1 \leq j \leq m$, such that corresponding attributes of X_{i_j} and Y_{i_j} result from the mapping of either the same EER attribute or correspond to the same attribute of Y''_s.

Note that there exists a unique subset of X_i, X_{i_s}, which is in a one-to-one correspondence of compatible attributes with Y_s. Relation-scheme R_i is associated with functional dependency $R_i : X_{i_s} \rightarrow (X_i - X_{i_s})$.

For example, the generalization in the EER structure of figure 4 is represented as shown in figure 5(iv), where $X'_4 = A_{4_1}A_{4_2}A_{4_3}$, $X_{4_1} = A_{4_2}A_{4_3}$, and X''_4 is empty.

5.4 Correctness.

In a relational database associated with an EER structure, every object of an object-set O_i is represented by a unique tuple, t. Let t_A and t_{ID} be the subtuples of t that consist of the values that correspond to the EER attributes and the identifier of O_i, respectively. Then:

(S1) If O_i is an independent entity-set then tuple t consists only of t_A. For any tuple, t', representing another object of O_i, $t'_{ID} \neq t_{ID}$.

(S2) If O_i is a specialization entity-set then tuple t consists of the concatenation of t_A with tuples t_S and t_H, where t_S represents the corresponding generalization-source entity and t_H consists of the values that correspond to all the inherited EER attributes of O_i except those associated with its generalization-source (which are represented in t_S). For any tuple, t', representing another object of O_i, $t'_S \neq t_S$.

(S3) If O_i is the aggregation of object-sets O_{i_j}, $1 \leq j \leq m$, then tuple t consists of the concatenation of tuples t_A and t_G, where t_G consists of the concatenation of the tuples representing the corresponding aggregated objects. Let $\overline{t}_{ID}$ be the subtuple of t that consists of the concatenation of tuples t_{ID} and t_G, and let t_{G_j} be the subtuple of t_G that does not include the tuple representing in t_G the object of O_{i_j}. For any tuple, t', representing another object of O_i, (i) $\overline{t}'_{ID} \neq \overline{t}_{ID}$; and (ii) if O_i is a relationship-set then for each object-set O_{i_j} with cardinality *one* in O_i, $t'_{G_j} \neq t_{G_j}$.

a.

	Relation : Object–Set	Attribute : ER Attribute	Foreign Attribute : Attribute	Inclusion Dependencies
(i)	$R_1(X_1)$: EMPLOYEE	A_{1_1} : NAME A_{1_2} : ADDRESS		
(ii)	$R_2(X_2)$: DEPARTMENT	A_{2_1} : NAME A_{2_2} : ADDRESS		
(iii)	$R_3(X_3)$: COURSE	A_{3_1} : NAME A_{3_2} : TYPE		
(iv)	$R_4(X_4)$: FACULTY	A_{4_1} : POSITION	$A_{4_2}:A_{1_1}$ $A_{4_3}:A_{1_2}$	$R_4[A_{4_2}A_{4_3}] \subseteq R_1[A_{1_1}A_{1_2}]$
(v)	$R_5(X_5)$: DEPENDENT	A_{5_1} : NAME	$A_{5_2}:A_{1_1}$ $A_{5_3}:A_{1_2}$	$R_5[A_{5_2}A_{5_3}] \subseteq R_1[A_{1_1}A_{1_2}]$
(vi)	$R_6(X_6)$: OFFER		$A_{6_1}:A_{2_1}$ $A_{6_2}:A_{2_2}$	$R_6[A_{6_1}A_{6_2}] \subseteq R_2[A_{2_1}A_{2_1}]$
			$A_{6_3}:A_{3_1}$ $A_{6_4}:A_{3_2}$	$R_6[A_{6_3}A_{6_4}] \subseteq R_3[A_{3_1}A_{3_2}]$
(vii)	$R_7(X_7)$: TEACH		$A_{7_1}:A_{4_1}$ $A_{7_2}:A_{4_2}$ $A_{7_3}:A_{4_3}$	$R_7[A_{7_1}A_{7_2}A_{7_3}] \subseteq R_4[A_{4_1}A_{4_2}A_{4_3}]$
			$A_{7_4}:A_{6_1}$ $A_{7_5}:A_{6_2}$ $A_{7_6}:A_{6_3}$ $A_{7_7}:A_{6_4}$	$R_7[A_{7_4}A_{7_5}A_{7_6}A_{7_7}] \subseteq R_6[A_{6_1}A_{6_2}A_{6_3}A_{6_4}]$
(viii)	$R_8(X_8)$: SUPERVISE		$A_{8_1}:A_{4_1}$ $A_{8_2}:A_{4_2}$ $A_{8_3}:A_{4_3}$	$R_8[A_{8_1}A_{8_2}A_{8_3}] \subseteq R_4[A_{4_1}A_{4_2}A_{4_3}]$
			$A_{8_4}:A_{3_1}$ $A_{8_5}:A_{3_2}$	$R_8[A_{8_4}A_{8_5}] \subseteq R_3[A_{3_1}A_{3_2}]$

b. Functional Dependencies: $R_1 : A_{1_1} \rightarrow A_{1_2}$ $R_3 : A_{3_1} \rightarrow A_{3_2}$ $R_5 : A_{5_1}A_{5_2}A_{5_3} \rightarrow \emptyset$ $R_7 : A_{7_4}A_{7_5}A_{7_6}A_{7_7} \rightarrow A_{7_1}A_{7_2}A_{7_3}$
$R_2 : A_{2_1} \rightarrow A_{2_2}$ $R_4 : A_{4_2}A_{4_3} \rightarrow A_{4_1}$ $R_6 : A_{6_3}A_{6_4} \rightarrow A_{6_1}A_{6_2}$ $R_8 : A_{8_1}A_{8_2}A_{8_3} \rightarrow A_{8_4}A_{8_5}$

Fig.5 Canonical Relational Representation for the EER Structure of Figure 4.

Definition 5.1. A relational schema is said to *represent correctly* an EER schema if any (consistent) database state associated with that relational schema satisfies conditions (S1) through (S3) above. □

Proposition 5.1. Let $(R, I \cup F)$ be the relational schema generated by **Crep** for some well-defined EER structure. Then $(R, I \cup F)$ represents correctly that EER structure.
Proof Sketch . The proof is by induction on the number of steps of **Crep**. □

It is well known in database design that the same data can be structured in different ways, that is, represented by different schemas, provided these schemas have *equivalent information-capacities* [9]. Since **Crep** generates provably correct relational representations for EER structures, any relational representation of a given EER structure should have an *equivalent* information-capacity with the canonical representation of that EER structure. We are not interested, however, in any relational representation for EER structures, but only in those that preserve the EER attribute values by having the relational domains in a one-to-one correspondence with the EER value-sets, and by representing every EER attribute by at least one relational attribute. The information-capacity equivalence of two relational schemas defined below follows [9] and captures the requirement mentioned above.

Definition 5.2. Let (R, Δ) and (R', Δ') be two relational schemas. (R', Δ') is said to *dominate* (R, Δ) if there exist *total* functions ϕ and ϕ' such that:

(1) ϕ maps consistent states of (R, Δ) into consistent states of (R', Δ');

(2) ϕ' maps consistent states of (R', Δ') into consistent states of (R, Δ);

(3) the composition of ϕ followed by ϕ' is the identity on the set of all consistent states of (R, Δ);

(4) For any state r of (R, Δ), ϕ commutes with each permutation of the values in the domains of r; similarly, for any state r' of (R', Δ'), ϕ' commutes with any permutation of the values in the domains of r'.

(R, Δ) and (R', Δ') are said to be *equivalent* if (R, Δ) dominates (R', Δ') and (R', Δ') dominates (R, Δ). □

Informally, a schema (R', Δ') dominates another schema (R, Δ) if (R', Δ') can be associated with more states then (R, Δ), that is, while every legal state associated with (R, Δ) can be exactly reconstructed from its mapping into a state of (R', Δ'), some states associated with (R', Δ'), cannot be exactly reconstructed from their mappings into states of (R, Δ). Condition (4) above ensures that attribute values are preserved by the state mappings (in [9] such mappings are called *generic*), thus reflecting our requirement for the relational representation of EER structures. Now we can define the correctness of relational representations for EER structures as follows.

Definition 5.3. A relational schema is said to *represent correctly* an EER schema if it has *equivalent* information-capacity with the canonical representation of that EER schema. □

In conclusion, **Crep** generates relational schemas of the form $(R, I \cup F)$, where (R, I) represents the EER structure, and F represents entity identifiers and relationship cardinalities. The functional and inclusion dependencies are used to compute the *keys* associated with the relation-schemes. This computation, which is necessary for analyzing the *normal form* of the relational schema, requires a special framework discussed in the next section.

6. A Framework For Dependency Analysis.

The mapping of EER schemas into relational schemas is followed usually by the computation of keys and, possibly, by normalization. In this section we examine the assumptions underlying key computation and normalization in the well established framework of the *Weak Instance Model* [17]. Satisfying these assumptions leads to three conditions that the relational attributes generated by **Crep** must satisfy. These conditions constrain the way in which the relational attributes can be assigned names, but do not restrict the capability of mapping general EER schemas into relational schemas. We assume below that two attributes are identical iff they are assigned the same *global* name, where global names are used to identify the attributes within the entire relational schema.

The *Weak Instance Model* is an approach to the relational model that provides a framework in which data dependencies can be specified and compared over the entire database rather than single relations. In particular, functional dependencies can, in such a framework, be specified over a universal set of attributes rather than over single relation-schemes, so that they do not need to be associated with a relation-scheme tag, that is, are *untagged*. Given a state r associated with schema (R, Δ), its *weak instance*, u, is a relation associated with the universal set of attributes (which consists of the union of the attribute sets associated with the relation-schemes of R) such that u satisfies Δ and for every relation r_i of r associated with relation-scheme $R_i(X_i)$, r_i is a subset of $u[X_i]$. The *Weak Instance Model* is based on several assumptions. We briefly review below these assumptions and examine their impact on **Crep**. All UR assumptions are surveyed in [16].

A relational attribute, A, generated by **Crep** represents the EER attribute (i) to which it corresponds directly (see (E1), (A1), and (G1) of **Crep**), or (ii) which is represented by the relational attribute to which A corresponds as a foreign attribute (see (A2) and (G3) of **Crep**). The *Universal-Relation Scheme Assumption* requires each attribute to represent a property of the same class of objects in every relation-scheme in which it appears. Accordingly, in a relational schema generated by **Crep**

(U1) each attribute must represent a unique EER attribute.

In the relational schema of figure 5, for example, the attributes that represent EER attribute NAME of entity-set DEPARTMENT (e.g. A_{2_1}, A_{6_1}) must be assigned different names than the attributes that represent EER attribute NAME of entity-set EMPLOYEE (e.g. A_{1_1}, A_{4_2}, A_{7_2}).

The *Unique Role Assumption* (URA) requires every subset W of the universal set of attributes, which consists of more than one attribute, to represent *at most one* basic *association* among the attributes of W. The first aspect of URA refers to W as part of the attribute-set of some relation-scheme: if W appears in more than one relation-scheme then W must represent the same class of objects in all the relation-schemes in which it appears. Consequently, in a relational schema generated by **Crep** two relation-schemes $R_i(X_i)$ and $R_j(X_j)$ that correspond to two distinct object-sets O_i and O_j, respectively, must satisfy the following conditions:

(U2) (i) if $X_i \cap X_j$ consists of more than one attribute, then there exists a relation-scheme $R_k(X_k)$ such that $X_i \cap X_j = X_k$; and (ii) if $X_i \subseteq X_j$ then there exists a relation-scheme $R_k(X_k)$ such that $X_i \subseteq X_k \subseteq X_j$ and O_j is either the aggregation or specialization of the object-set corresponding to $R_k(X_k)$.

Note that conditions (U2.i) and (U2.ii) above correspond to the *association integrity* and *containment condition* of [16], respectively. For example, if in the relational schema of figure 5 attributes A_{8_1}, A_{8_2}, A_{8_3}, A_{8_4} and A_{8_5} are assigned the same names as attributes A_{7_1}, A_{7_2}, A_{7_3}, A_{7_6} and A_{7_7}, respectively, then condition (U2) is not satisfied because, following this assignment, attribute set X_8 is included in X_7 although the corresponding relationship-sets are independent. Similarly, the relational schema of figure 2(ii) (see section 2) does not satisfy condition (U2) because the attribute set of OFFER is included in the attribute set of TEACH, although the corresponding relationship-sets are independent; condition (U2) is satisfied, if, for instance, attribute CN of OFFER is appropriately renamed.

In general, the *basic association* represented by some attribute set W refers to the projection on W of the join of a set of relations. The corresponding join expression is based on a *join-path* which consists of a sequence of relation-schemes of the form $R_{i_1}(X_{i_1}),\ldots, R_{i_m}(X_{i_m})$, where W is included in the union of attribute sets X_{i_j}, $1 \leq j \leq m$, every relation-scheme appears only once in the sequence, and the intersection of the attribute sets associated with any two neighbor relation-schemes in the sequence, R_{i_j} and $R_{i_{j+1}}$, $1 \leq j < m$, is nonempty. As noted in [16], if multiple join-paths can be associated with a given set of attributes, then URA implies the additional *One-Flavor Assumption* (OFA). OFA requires all the join-paths that can be associated with some attribute set to represent the same *flavor* of relationship (see [16] or [17] for details). Let R be the set of relation-schemes generated by **Crep** and let R be associated with the following *attribute digraph*: $G_A = (V, H)$, where $V = R$, and $R_i \rightarrow R_j \in H$ iff $X_j \subseteq X_i$, and $\nexists\, R_k(X_k) \in R$ such that $X_j \subseteq X_k \subseteq X_i$. We show below that, in order to comply with OFA, a relational schema generated by **Crep** must satisfy the following condition:

(U3) the associated attribute digraph has an acyclic underlying graph (i.e. is free of undirected cycles).

For example, the relational schema of figure 3(ii) (see section 2) does not satisfy condition (U3) because the associated attribute digraph contains an undirected cycle; condition (U3) is satisfied, if, for instance, attribute XNAME of relation ATTENDS is appropriately renamed.

Proposition 6.1. Let $(R, I \cup F)$ be a relational schema generated by **Crep**, that satisfies conditions (U1) and (U2). Then R satisfies OFA only if condition (U3) is satisfied.

Proof Sketch . We use the following notations: G_A and G_I denote the attribute and inclusion dependency digraphs associated with $(R, I \cup F)$, respectively; G_{ER} and $\overline{G}_{ER}$ denote the EER diagram represented by **Crep** and the subgraph of G_{ER} induced by the object-set vertices, respectively. The proof is based on the following claims.

Claim 1. Digraph G_A is a subgraph of digraph G_I.

Claim 2. Digraphs G_I and $\overline{G}_{ER}$ are isomorphic.

Claim 3. Any join-path over R is represented by a simple path in G_A.

The elements of the association represented by a navigation-path of G_{ER} are called *traversals* [13]. By claims 1 and 2, every path of G_A is isomorphic to a navigation-path of G_{ER}. Let α be a navigation-path in G_{ER}. Following the definition of traversals of [13], it can be shown that the traversals corresponding to α are tuples in the relation generated by the join corresponding to the join-path of G_A that is isomorphic to α. By claim 3, every join-path is represented by some path of G_A, which, in turn, is isomorphic to some navigation-path of G_{ER}. Consequently, the *Distinct Meaning Path Assumption* of the EER model (see section 4) implies that for a given set of attributes distinct join-paths represent different relationship *flavors*. Consequently, in order to satisfy OFA G_A must be free of undirected cycles. □

7. Normalization.

The framework discussed in the previous section allow us to analyze the normal form of relational schemas representing EER schemas. First we examine how the keys, necessary for establishing normal forms, are computed for these relational schemas.

Proposition 7.1. Let $(R, I \cup F)$ be generated by **Crep** and let $\overline{F}$ be the closure of F under the following rule: from $R_i[XYZ] \subseteq R_j[X'Y'Z']$ and $R_j : X' \rightarrow Y'$ derive $R_i : X \rightarrow Y$. Then any functional dependency σ belongs to $\overline{F}^+$ iff σ belongs to $(I \cup F)^+$.

Proof Sketch : (only-if) see proposition 4.1 of [4].

(if) Clearly, $(I \cup F)^+ = (I \cup \overline{F})^+$. Let $\sigma \in (I \cup \overline{F})^+$. We show that $\sigma \in \overline{F}^+$. Assume $\sigma = R_i : Y \rightarrow A$ and $\sigma \notin \overline{F}^+$. Then it can be shown that $\sigma \notin (I \cup \overline{F})^+$ by constructing a state that satisfies $(I \cup \overline{F})$ but does not satisfy σ. □

We show below that the keys of any relation-scheme R_i can be computed using only functional dependencies of $\overline{F}$ associated with R_i. We denote by $\overline{F}_i$ the subset of $\overline{F}$ associated with (having *tag*) R_i, and by $\overline{F}_U$ the set of *untagged* functional dependencies corresponding to $\overline{F}$.

Proposition 7.2. Let $(R, I \cup F)$ be generated by **Crep** and let $R_i(X_i) \in R$ and $YZ \subseteq X_i$. Then $Y \rightarrow Z$ belongs to $\overline{F}_U^+$ iff $Y \rightarrow Z$ belongs to $\overline{F}_i^+$.

Proof Sketch : (only-if) Obvious. (if) The proof is based on condition (U3) concerning the attribute digraph associated with R, discussed in section 6. □

Now the candidate keys can be computed in the usual way and the relation-schemes can be associated with primary keys. Although well known, the result presented below is valid only under the conditions discussed in the previous section.

Proposition 7.3. Let $(R, I \cup F)$ be generated by **Crep**. For every relation-scheme $R_i(X_i)$ of R, let (1) O_i be the object-set corresponding to R_i; (2) Z_i be the subset of X_i corresponding to the identifier of O_i; (3) T_i denote the set of relation-schemes $\{R_j(X_j) \mid R_i[X_{i_j}] \subseteq R_j[X_j] \in I\}$; and (4) W_i be the union of all the *foreign-keys* of R_i, where each foreign-key W_{i_j} is in a one-to-one correspondence with the *primary key* of relation-scheme R_j of T_i.

R_i is associated with keys as follows.

(i) If O_i is an *entity-set* then the primary key is $Z_i W_i$.

(ii) If O_i is a *relationship-set* then if all the cardinalities of the object-sets involved in O_i are *many*, then (1) the primary key is W_i; otherwise (2) the primary key is $(W_i - W_{i_j})$, where W_{i_j} is the *foreign-key* that references the relation-scheme which represents some object-set that has cardinality *one* in O_i. □

Note that for independent entity-sets W_i is empty, for specialization entity-sets Z_i is empty and all foreign-keys are equal, and for weak entity-sets and relationship-sets the corresponding foreign-keys are pairwise disjoint.

Generally, **Crep** generates relational schemas which are not in a high normal form. For example, relation-scheme $R_6(X_6)$ in figure 5, whose primary key is A_{6_3}, is not in Third Normal Form [20] because of the transitive dependency $A_{6_1} \rightarrow A_{6_2}$. In the sequel of this section we define a mapping, called **Norm**, that transforms the relational schemas generated by **Crep** into information-capacity equivalent BCNF schemas. Further we show that **Norm** generates minimal schemas in the sense that all *redundant* attributes are removed, where an attribute is considered redundant if its removal has no effect on the initial information-capacity. Thus, using **Crep** and **Norm** together, the resulting relational schemas are correct, minimal, and in BCNF.

Definition 7.1 (**Norm**). Let $(R, I \cup F)$ be generated by **Crep**. For every relation-scheme $R_i\,(X_i)$ of R let T_i denote the set of relation-schemes $\{\, R_j \mid R_i[X_{i_j}] \subseteq R_j[X_j] \in I \,\}$. A relation-scheme $R_i\,(X_i)$ of R is said to be *norm−ready* if either T_i is empty or all the relation-schemes of T_i have been already mapped by **Norm**.

Norm maps $(R, I \cup F)$ into $(R', I' \cup F')$ by successively mapping *norm−ready* relation-schemes of R, R_i, as follows:

(1) $R_i\,(X_i)$ is mapped into $R'_i\,(X'_i)$, where X'_i is generated by removing from X_i all the foreign attributes which do not belong to any foreign-key of R_i;

(2) every functional dependency of F associated with R_i, $R_i : Z \rightarrow W$, is mapped into functional dependency $R'_i : Z' \rightarrow W'$ of F', such that Z' and W' are generated by removing the attributes of $(X_i - X'_i)$ from Z and W, respectively;

(3) every inclusion dependency of I involving R_i in its left-hand side, $R_i[X_{i_j}] \subseteq R_j[X_j]$, is mapped into inclusion dependency $R'_i[Y] \subseteq R'_j[Z]$ of I', such that Y and Z are generated by removing from X_{i_j} the attributes of $(X_i - X'_i)$, and by removing from X_j the attributes corresponding to $(X_{i_j} - Y)$, respectively;

Norm is associated with two *state mappings*, η and η', where η maps each state r of $(R, I \cup F)$ into a state r' of $(R', I' \cup F')$ and η' maps each state r' of $(R', I' \cup F')$ into a state $\tilde{r}$ of $(R, I \cup F)$, as follows:

η maps every relation r_i of r into $r'_i = r_i\,[\,X'_i\,]$;

η' successively maps *state−ready* relations r'_i of r' into $\tilde{r}_i = r'_i \bowtie_{R_j \in T_i} rename\,(\,\tilde{r}_j\,,\,X_j \leftarrow X_{i_j}\,)$, where r'_i is said to be *state−ready* if either T_i is empty or if the relations of $\tilde{r}$ that are associated with relation-schemes of T_i have been already mapped by η'. □

For example, **Norm** maps the relational schema of figure 5 into the normalized schema of figure 6. Mapping **Norm** is well-defined: it terminates due to the acyclicity of I, and it can be verified that it generates syntactically correct schemas. The following proposition shows that **Norm** generates BCNF relational schemas that have equivalent information-capacity (in the sense of definition 5.2) with the corresponding unnormalized schemas, and that are free of redundant attributes.

Proposition 7.4. Let $(R, I \cup F)$ be generated by **Crep** and let $(R', I' \cup F')$ be the result of applying **Norm** on $(R, I \cup F)$. Then (i) $(R, I \cup F)$ and $(R', I' \cup F')$ have equivalent information-capacities; and (ii) all the relation-schemes of R' are free of redundant attributes and are in BCNF.

Proof Sketch. (i)The proof is based on the following claims.

Claim 1. Let r be a state associated with R which satisfies $(\,I \cup F\,)$. Then $\eta\,(\,r\,)$ satisfies $(\,I' \cup F'\,)$.

Relation−Schemes	Keys	Inclusion Dependencies
$R_1(A_{1_1}, A_{1_2})$	A_{1_1}	
$R_2(A_{2_1}, A_{2_2})$	A_{2_1}	
$R_3(A_{3_1}, A_{3_2})$	A_{3_1}	
$R_4(A_{4_1}, A_{4_2})$	A_{4_2}	$R_4[A_{4_2}] \subseteq R_1[A_{1_1}]$
$R_5(A_{5_1}, A_{5_2})$	$A_{5_1} A_{5_2}$	$R_5[A_{5_2}] \subseteq R_1[A_{1_1}]$
$R_6(A_{6_1}, A_{6_3})$	A_{6_3}	$R_6[A_{6_1}] \subseteq R_2[A_{2_1}]$
		$R_6[A_{6_3}] \subseteq R_3[A_{3_1}]$
$R_7(A_{7_2}, A_{7_6})$	A_{7_6}	$R_7[A_{7_2}] \subseteq R_4[A_{4_2}]$
		$R_7[A_{7_6}] \subseteq R_6[A_{6_3}]$
$R_8(A_{8_2}, A_{8_4})$	A_{8_2}	$R_8[A_{8_2}] \subseteq R_4[A_{4_2}]$
		$R_8[A_{8_4}] \subseteq R_3[A_{3_1}]$

Fig.6 Normalized Representation for the EERD of Fig. 4.

Claim 2. Let r' be a state associated with R' which satisfies $(\,I' \cup F'\,)$. Then $\eta'(\,r'\,)$ satisfies $(\,I \cup F\,)$.

The proof of the first claim is straightforward and the proof of the second claim is by induction on the number of steps of η'. The last condition of definition 5.2 is obviously satisfied by both η and η'. Following claims 1 and 2, the first two conditions of definition 5.2 are satisfied for both directions of dominance. We must show that the last condition of definition 5.2 is also satisfied.

(a) $\eta'(\,\eta(\,r\,)\,) = r$ follows from the fact that **Norm** preserves the primary keys of the relation-schemes of R and that the joins involved in η' are all on primary keys.

(b) $\eta(\,\eta'(\,r'\,)\,) = r'$. The proof is by induction on the number of steps of η'.

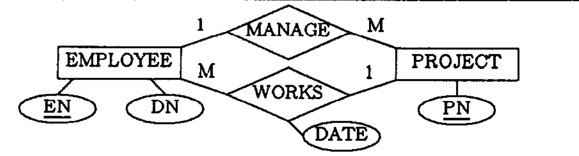
(ii) **Norm** removes only foreign attributes that do not belong to any foreign-key. Suppose that a foreign-key attribute is removed from some $R_i\,(X_i) \in R$. Then it can be shown that the information-capacity of $(R, I \cup F)$ is not preserved. Concerning the normal form of the relation-schemes of R', it can be verified that all functional dependencies implied by $(I' \cup F')$ are key dependencies. □

8. Summary And Open Questions.

We have proposed in this paper a canonical relational representation for EER structures, proved its correctness, and defined the concept of equivalence for relational schemas representing EER structures. Thus, a relational representation of some EER structure is said to be correct if it is equivalent to the canonical representation of that EER structure. We have examined the conditions that must be satisfied in order to make the EER-oriented design compatible with normalization and have shown that every EER structure has a BCNF relational representation. Note that we have left open the issue of a specific name assignment for relational attributes and gave only the constraints that such an assignment must satisfy.

The formalism developed in this paper can be applied to answer additional questions, such as under what conditions are two EER structures equivalent and whether several object-sets can be represented correctly by a single relation. The first question is addressed in [11] where the equivalence is defined for relational representations that do not involve inclusion dependencies and under the stringent *Pure Universal Instance Assumption*. Consider the

(i)



(ii)	EMPLOYEE	(EN,DN)
	WORKS	(EN,PN,DATE)
	MANAGE	(PN,MN)
(iii)	PROJECT	(PN)
(iv)	WORKS [EN]	$\subseteq$ EMPLOYEE [EN]
	WORKS [PN]	$\subseteq$ PROJECT [PN]
	MANAGE [EN]	$\subseteq$ EMPLOYEE [EN]
	MANAGE [PN]	$\subseteq$ PROJECT [PN]

Abbreviations: EN=EMPLOYEE NAME, DN=DEPARTMENT NUMBER, MN=MANAGER NAME, PN=PROJECT NUMBER

Fig.7 Relational Representations for an ER Structure.

following example adapted from [11]. Following [11], the EER structures of figures 1(i) and 7(i) are equivalent because the relational schema of figure 7(ii) is equivalent, in the sense of [11], to the relational schema of figure 1(iii). However, the relational schema of figure 7(ii) represents incorrectly, according to our correctness criteria, the ER structure of figure 7(i). A correct representation for this structure consists of the relation-schemes of figures 7(ii) and 7(iii) together with the inclusion dependencies of figure 7(iv). Since the relational schema of figure 1(iii) is also an incorrect representation for the ER structure of figure 1(i), the equivalence above has no real significance. Indeed, the correct representations for the ER structures of figures 1(i) and 7(i) are not equivalent in the sense defined in this paper. The open question is whether two different EER structures can be non trivially (i.e. without considering renamings) equivalent.

Some ER-oriented methodologies, such as that of [19], use a single relation-scheme for the representation of several object-sets. Such a representation can be viewed as a transformation of the canonical representation which consists of *merging* relation-schemes. As discussed in the introduction, such a transformation involves allowing nulls for certain attributes and requires the specification of additional constraints in order to preserve the semantics of the original schema. The open questions here are (i) what are these additional constraints and (ii) what is their impact on the information-capacity equivalence.

Acknowledgement : This work was supported by the Director, Office of Basic Energy Sciences, Engineering and Geo Sciences Division, of the U.S. Department of Energy, under Contract No. DE-AC03-76SF00098.

References

[1] P. Atzeni and D.S. Parker, "Assumptions in relational database theory", *ACM Symposium on Principles of Database Systems*, 1982, pp. 1-9.

[2] C. Beeri, P.A. Bernstein, and N. Goodman, "A sophisticate's introduction to database normalization theory", *Proc. of the 4th VLDB Conference*, IEEE Computer Society Press, 1978, pp. 113-124.

[3] L.I. Brady, "A universal relation assumption based on entities and relationships" *Proc. of the 4th International Conference on Entity-Relationship Approach*, IEEE Computer Society Press, 1985, pp. 208-215.

[4] M.A. Casanova, R. Fagin, and C.H. Papadimitriou, "Inclusion dependencies and their interaction with functional dependencies", *Journal of Computer and System Sciences* **28**,1 (Feb. 1984), pp. 29-59.

[5] P.P. Chen, "The entity-relationship model- towards a unified view of data", *ACM Trans. on Database Systems* **1**,1 (March 1976), pp. 9-36.

[6] E.F. Codd, "Extending the relational database model to capture more meaning", *ACM Trans. on Database Systems* **4**,4 (Dec 1979), pp. 397-434.

[7] S.S. Cosmadakis and P.C. Kanellakis, "Equational Theories and Database Constraints", in *Proc. of the 17th ACM Symposium on Theory of Computing,* 1985, pp. 273-284.

[8] S. Even, *Graph Algorithms*, Computer Science Press, 1979.

[9] R. Hull, "Relative information capacity of simple relational database schemata", *Proc. of the 3rd ACM Symposium on Principles of Database Systems*, 1984, pp. 97-109.

[10] R. Hull and R. King, "Semantic database modeling: Survey, applications, and research issues", *Computing Surveys* **19**,3 (September 1987), pp. 201-260.

[11] S. Jajodia, P.A. Ng, and F.N. Springsteel, "The problem of equivalence for entity-relationship diagrams", *IEEE Trans. on Software Engineering* **SE-9**,5 (September 1983), pp. 617-630.

[12] S. Jajodia, P.A. Ng, and F.N. Springsteel, "Entity-relationship diagrams which are in BCNF", *International Journal of Computer and Information Sciences* **12**,4 (1983), pp. 269-283.

[13] Y.E. Lien, "On the semantics of the entity-relationship data model", in *Entity-Relationship Approach to System Analysis and Design*, P.P. Chen (ed), North-Holland, Amsterdam, 1980, pp. 155-167.

[15] D. Maier, *The theory of relational databases*, Computer Science Press, 1983.

[16] D. Maier, D. Rozenshtein, and D.S. Warren, "Window functions", *Advances in Computing Research*, vol.3, JAI Press, 1986, pp. 213-246.

[17] D. Maier, J.D. Ullman, and M. Vardi, "On the foundations of the universal relation model", *ACM Trans. on Database Systems* **9**,2 (June 1984), pp. 283-308.

[18] V.M. Markowitz and A. Shoshani, "On the correctness of representing extended entity-relationship structures in the relational model", Technical Report LBL-26389, Computer Science Research Department, Lawrence Berkeley Laboratory, December 1988.

[19] T.J. Teorey, D. Yang, and J.P. Fry, "A logical design methodology for relational databases using the extended entity-relationship model", *Computing Surveys* **18**,2 (June 1986), pp. 197-222.

[20] J.D. Ullman, *Principles of database systems* (second edition), Computer Science Press, 1982.