MODIFYING THE RELATIONAL ALGEBRA FOR USE IN AN ENTITY-RELATIONSHIP ENVIRONMENT

by,

V.M. Markowitz and Y. Raz

Technical Report #263

February 1983
ABSTRACT

The Entity-Relationship Model is mostly accepted only as an early stage enterprise analysis tool, mainly because it lacks manipulative power. Since a link between the Entity-Relationship Model and the Relational Model may be established by choosing the relation to be the structural unit of the data-representational level of the Entity-Relationship Model, this model could benefit by somehow inheriting the Relational Algebra. The Relational Algebra operators cannot be used without modification in an Entity-Relationship environment, however, without endangering the respective semantic structure.

This paper proposes as manipulative part for the Entity-Relationship Model the Relational Algebra operators adapted to the structural constraints of an Entity-Relationship environment. The relation of these modified operators to the natural language sentence construction is investigated. It is shown that just as the Entity-Relationship Model concepts are based on the way people perceive information, the above operators bear analogies to the way people communicate, that is, natural language. As such, these operators may be used as a basis for query languages, over the Entity-Relationship Model, having constructs close to the natural language ones.
of the real world relevant to a specific enterprise, close to the way people perceive information: entities are qualified by their properties and interactions between entities are expressed by relationships which also are qualified by properties.

Concepts underlying the ERM are reviewed in Section 2, in a modified version of the original proposal of [Chen].

Relations are chosen for the data-representational level of the ERM. The Relational Model (RM) is briefly reviewed in Section 3. An extended, compared to its original definition [Codd72], Relational Algebra (RA) is presented as the manipulative part of the RM.

The great disadvantage of semantic data models, ERM included, is their lack of manipulative power [Codd82]. In Section 4 the RA is considered as the basis for deriving a manipulative part for the ERM, because of its link, established on its data-representational level, with the RM.

In Section 5 the linguistic analogies of the above operators are analyzed.

The paper concludes by drawing some conclusions in Section 6.

2. THE ENTITY-RELATIONSHIP SCHEMA

The ERM views an enterprise as consisting of entities and relationships. An entity is a part of the enterprise representing an object, material or abstract, which can be distinguished from its environment and considered atomic, i.e. not divisible because no part of it will be of interest in that specific environment.

Entities are grouped into entity-sets (e-sets), which have unique names and a type which formalize their time independent aspect. An entity-subset (e-subset) groups together entities from some e-set. We will be concerned with groupings resulting only from manipulations that will be defined below.
\[ R_i = \{[e_1, \ldots, e_n] | e_j \in E_j, j = 1:n\}, \]

where \([e_1, \ldots, e_n]\) represents a relationship.

Not all \(E_j\)'s have to be distinct, hence their ordering is significant. The ordering condition may be replaced by the appending, to every e-set involved in a r-set, of a role asserting the function played by the e-set in that r-set.

A value-set (v-set) groups together values of a same type (in the same sense as for types of e-sets) and have associated a set of binary comparison operators, that can be applied between any two elements of the v-set.

A property of an e-set or r-set is defined by a total function from the respective set to a value-set. Requiring the function to be total means that all the elements of some set share the same properties; the existence of an element implies the existence of values for all the properties of the set it belongs to. Note that the property is an association in the sense the r-set is an association. The range of a property, at a given time, is called attribute, having an attribute-name.

An attribute-instance exists, by definition, only when coupled with some entity or relationship. It may be viewed however, as an entity whose identity could independently become of interest. Then the attribute is perceived as an e-subset of the v-set viewed as e-set, while the corresponding property is perceived as r-set.

A key of an e-set is a list of properties such that any value from the Cartesian product of the corresponding value-sets is always guaranteed to be associated with zero or one entity in the e-set, and no proper sublist of the key satisfies the above condition. An e-set may have any number, zero included, of keys. One of the keys could be chosen to identify, permanently and uniquely, the entities of the e-set in the system. Their use as identifiers is troublesome because they carry information and are subject to change. Therefore, for every e-set a
the surrogate attribute. The surrogate is information-free and protected from the user, who may do no more than cause the system to generate or delete a surrogate. Two surrogates are equal if, and only if, they denote the same entity in the perceived world of entities.

An entity is described by an entity-representation, consisting of the list of all its attribute-instances plus its surrogate. Similarly, a relationship is described by a relationship-representation, consisting of the complete list of all its attribute-instances plus the surrogates of all the participating entities. An entity- or relationship-representation may be viewed as embedding property-representations, each consisting of an attribute-instance plus the, common to all, surrogate or list of surrogates, of the respective entity or relationship.

The concept of r-set can be extended to include the possibility of defining it on a single e-set. Accordingly, every e-set may be associated with a r-set defined on it, and having all its properties. As a consequence of connecting all the properties to r-sets, all the data will be concentrated in r-sets.

Besides the explicitly defined r-sets, there are so called implicit r-sets [Kent], defined by some derivation sequence involving at the initial stage explicit r-sets, e-sets, or both. In the following discussion r-set shall refer to both explicit and implicit r-sets.

Over a relationship-set may be defined correspondences; we shall restrict the correspondence concept of [San] to the following: given a r-set \( R = \{[e_1, \ldots, e_n] \mid e_i \in E_i, i = 1:n\} \) defined on the e-sets \( E_1, \ldots, E_n \), a correspondence, denoted by

\[
(E_1, \ldots, E_{k-1}, E_k, E_{k+1}, \ldots, E_n) \to \{E_k\}
\]

associates every tuple \([e_1, \ldots, e_{k-1}, e_{k+1}, \ldots, e_n]\), for which there is some \( e_k \) in \( E_k \) such that \([e_1, \ldots, e_{k-1}, e_k, e_{k+1}, \ldots, e_n]\) is in \( R \), with the set of all the entities \( e_s \) of \( E_k \), such that \([e_1, \ldots, e_{k-1}, e_s, e_{k+1}, \ldots, e_n]\) is in \( R \). An element of \((E_1, \ldots, E_n)\) is called the indexing component (i-component) and its corresponding indexed set (i-set) is an element of \( \{E_i\} \).
functions. Depending on the arity of the i-component, the property would be an entity-property or a relationship-property.

For a given enterprise the complete set of established e-sets, r-sets, properties and attributes, together with the value-sets form the **Entity-Relationship Schema** (ERS), also known as the enterprise schema.

3. THE DATA REPRESENTATION LEVEL OF ERM

The ERS describes the enterprise structure. At the data representation level, the entities are represented by the entity-representations and the relationships are represented by relationship-representations. All are lists of attribute-instances, some of them being surrogates.

We choose to organize these data in relations, which are simple structures and benefit from being the basis of an advanced data model, the **Relational Model** (RM). We shall briefly review its structural and manipulative parts, following [Pir] and [Ull].

A relational database consists of a set of relations defined on a set, \( T = \{A_1, \ldots, A_n\} \), of attributes (not in the sense of the ERM). An attribute \( A_i \) is associated with a domain \( \text{DOM}(A_i) \), the set of its possible values, uniquely identifying the role played by the domain in the relation.

A relation \( r \), defined on the set of attributes \( T_1 \subseteq T \), is a subset of the cross-product:

\[
\text{DOM}(A_{i_1}) \times \ldots \times \text{DOM}(A_{i_k})
\]

where \( T_1 = \{A_{i_1}, \ldots, A_{i_k}\} \).

The relation may be viewed as a table with its columns labelled by the attribute names and the rows, called **tuples**, representing elements in the cross-product subset.
are said to be union-compatible (u-c) if their relational attributes are in a one-to-one correspondence, such that corresponding attributes are associated with the same domain.

Relations are manipulated by Relational Algebra (RA) operators. Let \( r(T_1) \) and \( r(T_2) \) be two relation schemes, such that

\[
T_1 = \{A_1, \ldots, A_k\} \quad \text{and} \quad T_2 = \{B_1, \ldots, B_k\};
\]

\( t_i \) denote a tuple of a relation;

\( t_i \cdot t_j \) denote a tuple of a relation obtained by the concatenation of tuples \( t_i \) and \( t_j \); and

\( t_i[Y] \) denote a tuple containing components of \( t_i \) corresponding to the elements of \( Y \subset T_1 \).

(0) Attribute Renaming

Before applying any algebraic operator, it is possible to specify attribute correspondences by renaming the attributes of any relation. The rename operator \( [P \rightarrow r] \) is of the form:

\[
\rho(r, M), \quad \text{where} \quad r \text{ denotes a relation having the scheme } r(T),
\]

\[
T = \{A_1, \ldots, A_n\}, \quad \text{and} \quad M \text{ is a one-to-one, not necessarily total, functional mapping } \{(A_i \mapsto B_j)\} \text{ with } A_i \in T
\]

and \( B_j \in T', T' = \{B_1, \ldots, B_m\} \).

As a result, \( r \) will have renamed attributes, such that every image of \( A_i \) under \( M \), \( B_j \), will be associated with \( \text{DOM} (A_i) \). Consequently, the result of an algebraic operation will directly inherit the attributes from possibly renamed operands, and, on its turn, may be subjected to renaming and appear as operand in another operation.

(1) Union, Intersection, Difference

Given two u-c relations, \( r \) and \( s \),

\[
\begin{align*}
\quad r \cup s &= \{ t \mid t \in r \text{ or } t \in s \}; \\
\quad r \cap s &= \{ t \mid t \in r \text{ and } t \in s \};
\end{align*}
\]

and
\( r \times s = \{ t \mid t = t_1 t_2, t_1 \in r \text{ and } t_2 \in s \} \).

(3) **Projection**
\[ r[Y] = \{ t[Y] \mid t \in r, Y \subseteq T \} \]

(4) **Selection**
\[ r / F = \{ t \mid t \in r \text{ and } F \text{ is true when every attribute occurrence } A_j \text{ in } F \text{ is replaced by } t[A_j] \}, \text{ with } F \text{ a Boolean combination of atomic comparisons of the form } (A_i \theta c) \text{ or } (A_i \theta A_j), \text{ where } c \text{ is a constant and } A_i, A_j \text{ are domain-compatible.} \]

(5) **\( \theta \)-Join**
\[ r / \theta B / s = \{ t \mid t = t_1 t_2, t_1 \in r, t_2 \in s \text{ and } t[A] \theta t[B] \}, \text{ where } (A, B) \text{ are domain-compatible.} \]

(6) **Natural Join**
\[ r \bowtie s = \{ t \mid t = t_1 t_2, t_1 \in r, t_2 \in s \text{ and } t_1 [T^*] = t_2 [T^*] \}, \text{ where } T^* = T_1 \cap T_2. \]

(7) **Quotient**
\[ r \div s = \{ t \mid t = t_1 t_2, t_1 \in r[F_1 - X] \text{ and } t_2 \in s[T_2 - X] \text{ and } (\forall t_2 x \in s \Rightarrow \exists t_1 x \in r) \}, \text{ where } X = T_1 \cap T_2. \]

(8) **The Aggregate - Function Operator (\( \Phi \))**
Let \( r(T) \) be a relation-scheme and \( T = X \cup Y \cup A_k \), where \( X \) and \( Y \) are disjoint subsets of \( T \) not including the single attribute \( A_k \).

An aggregate-function (af) manipulation of the relation \( r \) results in a relation \( r' \) whose relation-scheme is \( r'(T') \) where \( T' = X \cup D_k \), \( D_k \) is a computed (by the aggregate-function) attribute whose instances are the values of the af [Klug].
The aggregate-function operator (\( \lambda \)) partitions its input on the attributes \( X \), applies \( af \) to each partition and outputs the \( X \)-value plus the \( af \)-value for each partition.

We are interested in a class of aggregate functions including \( \text{COUNT}, \text{SUM}, \text{MAX} \) and \( \text{MIN} \). Note that for \( \text{COUNT} \), \( Y \) has to be empty.

For every r-set \( R \) of an ERS a relationship-relation, \( rR \), denoted as \( s \)-relation, is established. Its tuples are the relationship-representations of the elements of \( R \). The relation-scheme of \( rR \) includes the name of \( R \) and a set of entity- and attribute-specifications of the form \( V/R: X \). For an entity-specification \( V \) denotes both the e-set name and the name of the corresponding surrogate r-set, \( R \) denotes the e-set role in \( R \) and \( X \) is a correlating-variable (c-var). The c-var has a referential function, of correlating the various appearances of a same entity in different associations. For an attribute-specification \( V \) denotes the name of its associated v-set, \( R \) is the role of the attribute in \( R \) and \( X \) is null. The existence of \( X \) distinguishes the e-sets from attributes. Its lack reflects the strictly local significance of the attributes - that of characterizing a certain r-set or e-set.

The set of all the s-relations corresponding to some ERS is called the Entity-Relationship Relations Set (ERRS).

4. ADAPTING THE RA FOR AN ENTITY-RELATIONSHIP ENVIRONMENT

The ERM has been defined without a manipulative part. By establishing a link with the relational model it is possible to have the ERM inherit the RA, as suggested by [Codd82]. The linking is accomplished by choosing the relation as the structural unit for the data-representation level of the ERM; As the ERRS relations are constrained by the ERM structure, the RA operators cannot be applied directly on them. The adaptation of the RA operators to preserve the ERM structure will provide the operators for the manipulative part of the ERM.
- \( R_j \) denotes a \( r \)-set \( j \);
- \( rR_j \) denotes its corresponding \( s \)-relation;
- \( E_{ijk} \) stands for the entity specification \( E_i / R_j: X_k \) of the \( rR_j \) relation-scheme, where \( E_i \) is an e-set involved in \( R_j \);
- \( A_{kj} \) stands for the attribute specification \( A_k / R_j \) of the \( rR_j \) relation-scheme, where \( A_k \) is an attribute of \( R_j \);
- \( R_j.s \) denotes the set of all the \( E_{ijk} \)'s corresponding to \( rR_j \);
- \( sR_j \) denotes the projection of \( rR_j \) on \( R_j.s \).

For \( s \)-relations the union-compatibility property is defined as follows: two \( s \)-relations are u-c if they represent \( r \)-sets that are defined on the same set of e-sets and in their \( s \)-relation schemes corresponding e-sets have the same entity-specifications. Similarly, two attributes are said to be attribute-compatible (a-c) if they are associated with the same value-set.

(0) Entity-Specification Renaming

Similarly to the \( \rho \) RA operator, the entity-specification renaming \( s \)-operator assures the correlation of same entities appearing in the context of different associations and/or the definition of roles for new \( s \)-relations.

Given \( rR_j \), representing a \( r \)-set \( R_j \), the entity-specification renaming \( s \)-operator has the form:

\[ \rho^S(rR_j, M), \text{ where } M \text{ is a one-to-one, not necessarily total, functional mapping } \{(E_i / R_j \cdot X \rightarrow E_i / R_j \cdot X')\} \text{ affecting roles, correlating-variables or both.} \]

The results of the \( s \)-operations defined below will directly inherit the entity-specifications from their operands, and it is assumed that \( \rho^S \) precedes, when necessary, an \( s \)-operation.
- $\Theta$ denotes the RA operators: union, difference or intersection, and
- $\Theta^s$ denotes the $s$ operators, $s$-union, $s$-difference or $s$-intersection.

Given two $u$-c $s$-relations $rR_j$ and $rR_i$ then

$$rR_k = rR_j \Theta^s rR_i \triangleleft sR_j \Theta^s sR_i.$$  

$mR_k$ is also $u$-c with $rR_j$ and $rR_i$ and represents a $r$-set defined
on the same $e$-sets as $R_j$ and $R_i$ and has no properties.

(2) **$s$-Cartesian Product**

The $s$-Cartesian product operator can be defined as follows.

Given two $s$-relations $rR_i$ and $rR_j$:

$$rR_k = rR_i \ast^s rR_j \triangleleft sR_i \ast^s sR_j.$$  

$mR_k$ represents a $r$-set $R_k$ defined on all the $e$-sets involved in
$R_i$ or $R_j$.

(3) **$s$-Projection**

Given a $s$-relation $rR_j$ and $R.s_k \subseteq R_j.s$, the projection of
$rR_j$ on $R.s_k$ is defined as:

$$rR_k = rR_j [R.s_k]^s \triangleleft rR_j [R.s_k].$$  

$mR_k$ represents a new $r$-set, $R_k$, defined on the $e$-sets corresponding
to the elements of $R.s_k$.

It is possible to view an attribute as an $e$-subset of its associated
$v$-set viewed as an $e$-set. This $e$-set will be associated by a new $r$-set
with the $e$-set(s) characterized by the respective attribute. The percep-
tion of attributes as $e$-subsets is the semantical essence of pro-
jection's involving attributes.

$$rR_e = rR_j [R.s_k A_m]^s \triangleleft rR_j [R.s_k A_m].$$  

$mR_e$ represents a new $r$-set $R_e$ defined on the $e$-sets corresponding
in the elements of $R.s_k$ and to $A_m$. The relation-scheme of $rR_e$ includes,
besides the entity-specifications corresponding to the elements of $R.s_k$, 

Technion - Computer Science Department - Technical Report CS0263 - 1983
or entity specification belonging to the s-relation.

Given an s-relation rR and F over rR:

\[ rR' = rR / F \Delta (rR/F)[R.s]. \]

rR' represents a r-set, R', defined on the same e-sets as R.

(5) S- Join

The S-join embeds a Cartesian product followed by selection, where the formula F involves only the comparison of two a-c attributes, each belonging to one of the joined relations.

Given two s-relations rR_i and rR_j and two attributes, A_1 belonging to R_i and A_2 belonging to R_j:

\[ rR_k = rR_i / A_1 \times A_2 / s rR_j \Delta (rR_i/A_1 \times A_2/rR_j)[R_i.s,R_j.s]. \]

rR_k represents a r-set, R_k, defined on all the e-sets involved in R_i or R_j.

(6) S-Natural Join

The second join s-operator is defined as follows.

Given two s-relations rR_i and rR_j and a set S, such that S = R_i.s \cap R_j.s,

\[ rR_k = rR_i \bowtie S rR_j \Delta S R_i \bowtie S R_j. \]

rR_k represents a r-set, R_k, that is defined on all the e-sets, corresponding to all the elements belonging to (R_i.s \cup R_j.s - S).

The correspondence leading to the correlation reflected by S may be established through suitable renamings undergone by rR_i and rR_j prior to the s-natural join.

The natural join is based on, and expresses, correlation. In this sense a correspondence of correlating-variables, established if necessary by renaming, of the operators of a binary s-operator specifies an embedded natural-join.

Technion - Computer Science Department - Technical Report CS0263 - 1983
Natural join may be embedded, besides the Cartesian product, in θ-join and in quotient. Its embedding in union, intersection and difference is a prerequisite of the respective operations, expressed by the u-c condition. The operational implication of an embedded natural-join is the selection-projection part of an explicit natural-join.

(7) S-Quotient

Given two s-relations rR_i and rR_j, and a set D such that,

\[ D = R_i . s \cap R_j . s, \]

the s-quotient of rR_i and rR_j on D is defined as:

\[ rR_k = s^{R_i} rR_j = sR_i : sR_j. \]

rR_k represents a r-set, R_k, defined on the e-sets corresponding to the elements belonging to \((R_i . s \cup R_j . s - D)\).

The relaxing of the condition satisfied by D, such that

\[ D \subseteq R_i . s \cap R_j . s, \]

makes possible the embedding of s-natural join in s-quotient. The join, on \(R_i . s \cap R_j . s - D\), is implied by suitable correlating-variable correspondences, while the s-quotient now requires the explicit specification of D:

\[ rR_k = rR_i : D \overset{S}{\sim} rR_j. \]

(8) S-Aggregate Function Operator

The s-operator counterpart of the aggregate-function operator may be applied only on s-relations that represent n-ary (n > 1) r-sets. Two cases of applying \(S^S\) may be distinguished:

(a) af is \(\text{COUNT}\) and its target is an e-set;

let \(rR_j\) represent an n-ary (n > 1) r-set and \(E \in R_i . s,\)

\[ X = R_i . s - E. \]

Then:

\[ rR^* = S^S_{\text{COUNT}} < X; \ E > (rR_j) \overset{\triangle}{=} S^S_{\text{COUNT}} < X; \ E > (sR_j). \]
$A_k$ be the target attribute belonging to $R_1$, such that
$R_1'.s \subseteq R_j'.s$ and $\chi \subseteq R_j'.s$ but $\chi \notin R_1'.s$.

Then:
$$rR' = \exists \sigma' < X; A > (rR_j) \begin{array}{c} \triangle \end{array} \exists \sigma < X; A > (sR_j \bigtriangledown rR_1[R_1'.s,A]).$$

In both cases $rR'$ represents a new r-set, $R'$, defined on the e-sets corresponding to the elements of $X$, and having a derived property whose range is a computed attribute $A_C$ associated with a suitable v-set.

Notice that if all the v-sets are perceived as associated with e-sets, then all the s-relations represent "pure" r-sets, that is all $rR = sR$, and all the above s-operations are slightly changed RA operations. Since such a transformation is always possible, it is evident that the expressive power of the above set of s-operators is equivalent to that of the RA.

These s-operators are meant, however, to handle the attributes differently from the e-sets. The dependence of the attributes, on the e-sets or r-sets they are characterizing, is reflected in the s-operations by their strictly local use. A global use of an attribute means a change of perception and has to be explicitly expressed through a suitable s-projection. Note that such a local use of the attributes is not, in most cases, restrictive. Thus, a general RA selection can always be split in elementary selections affecting only simple (not derived) r-sets: such a selection is equivalent to a combination of union, intersection and difference operations having as operands relations that are the result of elementary selections which, then, can be commuted with any other operator. [U1].
The ERM views the enterprise as consisting of entities, having properties, interacting through relationships, which also may have properties. We have no intention of discussing to what extent this view is restrictive or "unnatural". We will try to point out the kind of natural language sentences which fit this view in order to tailor, for the ERS elements, denotations that would suit a manipulation language approaching the natural way of communication, the natural language.

Entities are, by definition, atomic in the sense that their decomposition is of no interest in the given enterprise. In the same sense, the attribute-instances are also atomic. Both e-sets and attributes are denoted by names that are, when possible, "real-world" noun names. Properties and r-sets describe associations among elements of e-sets and attributes. They can be expressed by simple predicative, natural language, sentences. Such a sentence contains a predicate and some object terms denoting e-sets or attributes.

We choose to denote an association (r-set or property) by the skeleton (including only the predicate and the object terms) of the participial form of the corresponding description sentence. Since a sentence has different paraphrases, accordingly an association will have different denotations. An hypothetic SUPPLY r-set, for instance, could be expressed (the denotations are in uppercase letters and the roles are prefixed by a single quotation mark) by:

SUPPLIER is 'SUPPLYING ITEM to DEPARTMENT ;
DEPARTMENT is 'SUPPLIED with ITEM by SUPPLIER ; and
ITEM is 'SUPPLIED to DEPARTMENT by SUPPLIER .

Notice that what differentiates the paraphrases is which of the terms is in the subject position, while the ordering of the terms in the object position is irrelevant.

The denotation of a property belonging to an e-set or r-set is similarly derived. It includes the denotation of the set the property belongs to, and an attribute denotation. An hypothetic COLOR property of some e-set ITEM, for instance, could be denoted by:

COLOR 'OF ITEM ; and
ITEM 'HAVING COLOR .
the subject position, called role. It relaxes, in most cases, the ordering of e-sets condition, but not as completely as does the role as defined in Section 2. The different r-set denotations provided by the paraphrases of the corresponding description sentence will be useful in traversing the r-set in any direction in the process of deriving implicit r-sets and constructing their corresponding complex description sentences. Such sentences would correspond to complex roles.

Since the ERM concepts are based on the way in which people perceive information, it is natural to expect that a manipulative part of the ERM would have analogies with the way people communicate, that is, natural language sentences.

Besides denotations already given, we shall use elementary statements of the form:

"A \wedge c" or "A \wedge B\wedge c",

where A and B are attribute denotations,

where A and B are attribute denotations,
c is a constant and
\(\theta\) is some comparison operator.

We shall call such a statement a restriction.

We use the following additional notations:

- \(E_i\) denotes an e-set i, while \(X_i\) denotes a correlating-variable coupled with \(E_i\);
- \(A_k\) denotes an attribute \(k\);
- \(R_{ij}\) denotes the role of \(E_i\) or \(A_i\) in a r-set \(j\);
- \(E_{i_1} : X_{i_1} R_{i_1 j} E_{i_2} : X_{i_2} \ldots E_{i_m} : X_{i_m}\) is one of the denotations of a r-set \(j\);
- \(E_{i_1} : X_{i_1} R_{i_1 j} \ldots E_{i_m} : X_{i_m} A_k\) is one of the denotation of a property \(k\) of some r-set \(j\).
with the help of so-called reference words such as this, that, etc. Notice that an implicit correlation is provided by textual contiguity.

The referencing is assured by the correlating-variables explicitly set by the entity-specification renaming s-operator. For a given r-set \( R_j \) we will be interested in the set of the e-set denotations referenced outside the frame of the \( R_j \) denotation, in some expression \( Q : S^Q_{R_j} \).

For any two r-sets, \( R_i \) and \( R_j \), \( S^C_{R_i, R_j} \) denotes the set of mutual references, i.e.: \( S^C_{R_i, R_j} = S^Q_{R_i} \cap S^Q_{R_j} \). Remark that \( S^C_{R_i, R_j} \) implicitly expresses the natural-join embedded in binary s-operations, excepting s-natural join, taking \( rR_i \) and \( rR_j \) as operands.

5.2 Partially Denoted Relationship-Sets

When the denotation for a r-set \( R \) does not include the denotations of all the e-sets participating in \( R \), this is called a partially denoted r-set and the derivation of a new r-set, \( R' \), is implied. \( R' \) is represented by \( rR' \), which is obtained by the s-projection of the \( rR \) that represents \( R \):

\[
rR' = rR[S^Q_{R}]^S
\]

For instance, let \( R \) be denoted by

DEPARTMENT 'SUPPLIED by SUPPLIER with ITEM,

then

DEPARTMENT 'SUPPLIED by SUPPLIER

denotes \( R' \) represented by \( rR' = rR[DEPARTMENT, SUPPLIER]^S \).

5.3 Relativization

Natural language sentences can be combined by restrictive relativization. Roughly this means that a sequence of sentences is connected such that any two neighbouring sentences are chained on an object-term, raised from an object position in the first sentence,
Similarly, association denotations can be chained in order to express certain queries.

Let \( R_j \) and \( R_i \) be two r-sets denoted by \( E_m : X \ \mathcal{R}_m \) \( \ldots \) \( E_k : X_k \) and \( E_k : X \ \mathcal{R}_k \) \( \ldots \) \( E_n : X_n \) respectively.

Their chaining on \( E_k \) by relativization is expressed by:

\[
E_m : X \ \mathcal{R}_m \ldots E_k : X \ \mathcal{R}_k \ldots E_n : X_n
\]

and implies the derivation of a r-set \( R' \) represented by \( rR' \):

\[
rR_X = rR_j \bigtriangleup S_{R_j,R_i}^G \ rR_i ;
\]

\[
rR' = rR_X[S_{R_X}^{Q_S}]^s ,
\]

where \( rR_X \) represents an intermediary derived r-set \( R_X \).

For instance, if \( R_i \) is denoted by \( \text{DEPARTMENT}:D \ '\text{REQUESTING ITEM} \) and by \( R_j \) is denoted by \( \text{ITEM} \ '\text{SUPPLIED to DEPARTMENT}:D \),

then \( \text{DEPARTMENT}:D \ '\text{REQUESTING ITEM} \ '\text{SUPPLIED to DEPARTMENT}:D \)

is denoting a r-set \( R' \) represented by \( rR' \):

\[
rR' = rR_i \bigtriangleup S_{R_i,R_j}^G \ rR_j ,
\]

where \( S_{R_i,R_j}^G = \{\text{DEPARTMENT}, \text{ITEM}\} \)

Similarly, let two properties be denoted by

\[
E_m : X \ \mathcal{R}_m \ldots A_k \ \text{and} \ A_n \ \mathcal{R}_n \ldots E_k : X_k
\]

where \( A_k \) and \( A_j \) are a-c attributes. The chaining of the two properties is expressed as

\[
E_m : X \ \mathcal{R}_m \ldots A_k \ \mathcal{R}_n \ldots A_j \ \mathcal{R}_k \ldots E_k : X_k
\]

and implies the derivation of \( R' \) represented by \( rR' \):
Notice that \( S^c_{R_i,R_j} \) could imply an embedding natural-join. If the second property is replaced by a constant \( c \), the chaining is interpreted as:

\[
rR' = rR_i / S_{A_k} \otimes c.
\]

Another form of relativization occurs when the connecting relative pronoun does not stand for the subject of the second sentence in a chain pair and is called closed relative clause.

The chaining by closed relative clause of the above \( r \)-sets \( R_i \) and \( R_j \) is expressed as:

\[
E_m : X_m \ R_{mj} \ ... \ E_k : X_k \ * \ E_n : X_n \ R_{ni} \ ... \ E_k : X_k
\]

and is evaluated as for simple relativization. The only difference is that while for relativization \( S^c_{R_i,R_j} \) contains at least one element, now it may be empty, reducing the natural-join to Cartesian-product.

5.4 Negation

Let \( R_j \) be a \( r \)-set denoted by \( E_k : X_k \ R_{kj} \ ... \ E_m : X_m \).

A negation of \( R_j \), expressed as

\[
E_k : X_k \ \text{NOT} \ R_{kj} \ ... \ E_m : X_m
\]

implies the derivation of the \( r \)-set \( R' \) represented by \( rR' \):

\[
rR_X = rR_i^s \ ... \ rR_i^m
\]

\[
rR' = rR_X^{s} \ rR_j.
\]

First, \( rR_X \) represents a \( r \)-set grouping all the possible associations among the entities from the \( e \)-sets on which \( R_j \) is defined, represented by \( rR_i^k \) through \( rR_i^m \). Then, \( rR' \) represents the \( r \)-set grouping the
represented by $rR'$:

$$rR_X = \text{DEPARTMENT} \ast^S \text{ITEM};$$

$$rR' = rR_X -^S \text{REQUEST},$$

where \text{DEPARTMENT} and \text{ITEM} are the \text{s-relations} representing the respective \text{e-sets}.

5.5 Coordination

Yet another method of combining natural language sentences is coordination, meaning the connecting of several sentences with a common subject. It may be perceived as a branching on the common subject. In the same way we can coordinate association denotations or restrictions by using the logical connectors \text{AND} and \text{OR}.

\text{AND} branching is semantically similar to chaining and is evaluated through the \text{s-natural join}. Let two \text{r-sets} $R_j$ and $R_1$ be denoted $E_k$: $X_k$ $\mathcal{R}_j$ ... $E_i$: $X_i$ and $E_k$: $X_k$ $\mathcal{R}_1$ ... $E_m$: $X_m$ respectively.

Their \text{AND} branching on $E_k$ is expressed as:

$$E_k$: $X_k$ ($\mathcal{R}_j$ ... $E_i$: $X_i$ \text{AND} $\mathcal{R}_1$ ... $E_m$: $X_m$).

This implies the derivation of the \text{r-set} $R'$ represented by $rR'$:

$$rR_X = rR_j \bigcirc^S rR_1;$$

$$rR' = rR_X[S^Q_{rR_1}]^S.$$

Notice that when $S^C_{rR_1} = S^Q_{rR_1} = S^Q_{rR_1}$, the \text{s-natural join} above reduces to the \text{s-intersection}. In this case, the combined \text{AND NOT} branching is evaluated straightforwardly through the \text{s-difference}:

$$rR' = rR_j -^S rR_1.$$
represented by \( rR' \):

\[
rR' = rR_j \cup^S rR_1.
\]

When the above condition does not hold, a different interpretation must be given to the branching, as follows:

- we note by \( S^*_{R_j R_1} = S^Q_{R_j} \cup S^Q_{R_1} \);
  \[
  S^1_{R_j} = S^*_{R_j R_1} - S^Q_{R_j},
  \]
  \[
  S^1_{R_1} = S^*_{R_j R_1} - S^Q_{R_1}.
  \]

\[
rR'_j = rR_j \ast^S rR_{1_1} \ast^S \cdots \ast^S rR_{1_m}, \text{ where } rR_{1_1} \text{ through } rR_{1_m}
\]
represent e-sets whose denotations are in \( S^1_{R_j} \);

\[
rR'_1 = rR_{1_1} \ast^S rR_{k_1} \ast^S \cdots \ast^S rR_{k_n}, \text{ where } rR_{k_1} \text{ through } rR_{k_n}
\]
represent e-sets whose denotations are in \( S^1_{R_1} \);

- we have constructed \( rR'_j \) and \( rR'_1 \) such that \( S^Q_{R_j} = S^Q_{R_1} = S^*_{R_j R_1} \)
and \( R' \) is represented by

\[
rR' = rR'_j \cup^S rR'_1.
\]

Thus, for instance, DEPARTMENT ('REQUESTING ITEM OR 'SUPPLIED by SUPPLIER) denotes the r-set \( R' \) represented by \( rR' \):

\[
rR'_j = \text{REQUEST} \ast^S \text{SUPPLIER} ;
\]
\[
rR'_e = \text{SUPPLY} \ast^S \text{ITEM} ;
\]
\[
rR' = rR'_j \cup^S rR'_e,
\]

where SUPPLIER and ITEM are the s-relations representing the respective e-sets.
"only", etc. We won't be concerned with the natural language quantifiers, that may be ambiguous, but with their specification in a more precise way, through set expressions.

Thus, "all" is translated into "set-contains-set", and "only" into "set-equal-set". The less inclusive "at most" becomes "set-included in-set", and "at least" becomes "set-contains-set".

In ERM, the basis for set expressions is provided by correspondence. It will be indicated by a modified denotation of the r-set over which it is expressed, with the modification being the prefixing of the i-set denotation by SET. Thus, let \( R_j \) be a r-set denoted by

\[
E_k : X_k R_{kj} \ldots E_i.
\]

A correspondence over \( R_j \), having \( E_i \) as the i-set, is denoted by \( E_k : X_k R_{kj} \ldots \text{SET } E_i \). Similarly, a correspondence over \( R_1 \), denoted by \( E_m : X_m R_{ml} \ldots \text{E_j having } E_j \text{ as the i-set, is denoted by} \)

\[
E_m : X_m R_{ml} \ldots \text{SET } E_j.
\]

The chaining of the two correspondences is possible when \( E_i \) and \( E_j \) denote the same e-set, and can be expressed as:

\[
E_k : X_k R_{kj} \ldots \text{SET } E_i \text{ SOP SET } E_j R_{jl} \ldots E_m : X_m
\]

where SOP is a set-comparison operator.

Note that statements of the form "SET - set operation - SET" resembles the restrictions mentioned above, therefore, we shall call them set-restrictions. Set-restrictions, just like restrictions, can be integrated in chains, on branches. We shall discuss only the interpretation of the chaining for SOP = \text{CONTAINS}, as the other cases are similarly treated. Thus,

\[
E_k : X_k R_{kj} \ldots \text{SET } E_i \text{ CONTAINS SET } E_j R_{jl} \ldots E_m : X_m
\]

implies the derivation of the r-set \( R' \) represented by \( rR' \):

\[
rR' = rR_j E_i \triangleright^S rR_1, \text{ with } S'_{R_j, R_1} \text{ possibly implying an embedded natural join};
\]

\[
rR' = rR_j \left(S'_{R_j, R_1} \right)^S.
\]
For instance, $\text{DEPARTMENT} \mid \text{REQUESTING at least ITEM 'SUPPLIED by SUPPLIER}$ has the equivalent set-expression

$\text{DEPARTMENT} \mid \text{REQUESTING SET ITEM CONTAINS SET ITEM 'SUPPLIED by SUPPLIER}$

which denotes the r-set $R'$ represented by $rR'$:

$rR' = \text{REQUEST} \mid \text{SUPPLY ITEM}$

It is worth noting that complex expressions involving quantifiers have a recursive form which can be translated straightforwardly to nested s-operations with s-quotients only over single e-sets (how would you express a general quotient in natural language?). Take, for instance, the following natural language expression:

"Suppliers supplying all the departments, requesting some item, and supplying every of the departments with all the items that department is requesting".

It may be translated to the following set expression:

$\text{SUPPLIER:S SUPPLYING SET DEPARTMENT:D} \mid \text{that is SUPPLIED by SUPPLIER:S with SET ITEM = SET ITEM 'REQUESTED by DEPARTMENT:D')}$

$\text{CONTAINS}$

$\text{SET DEPARTMENT} \mid \text{(that is 'REQUESTING ITEM)}$

which denotes a r-set $R'$ represented by $rR'$:

$rR' = \text{SUPPLY} \mid \text{REQUEST, which also embeds a natural-join ITEM implied by $S^C = \{\text{DEPARTMENT}\};}$

$rR' = rR' \mid \text{DEPARTMENT REQUEST [DEPARTMENT]}^S$. 
There is a type of natural language quantifiers, of the form "at most n", "exactly n", etc. (where \( n \) is a positive integer).

To express them, we shall use the same framework of the correspondence where COUNT replaces SET in the denotation.

Such expressions define implicit (derived) properties. Statements like "COUNT - comparison operator - COUNT" are restrictions in the above sense, and also are integrated in chains or branches.

Let \( R_j \) be a \( r \)-set denoted by \( E_{k \cdot X_k \ R_{k j}} \ldots E_{i} \). Then, \( E_{k \cdot X_k \ R_{k j}} \ldots \text{COUNT} \ E_{i} \) implies the derivation of a \( r \)-set \( rR' \) represented by \( rR' \):

\[
rR' = \text{COUNT}^{S} <X_{i}; E_{i}> (rR_{j}),
\]

where \( X \) includes all the entity-specifications of \( rR_{j} \) besides that of \( E_{i} \).

\( R' \) is defined on the \( r \)-sets corresponding to the entity-specifications of \( X \) and has a new (computed) attribute associated with the set of positive integer numbers. \( R' \) may be involved; therefore, in chainings etc., just as any \( r \)-set.

For instance, DEPARTMENT SUPPLIED BY SUPPLIER with COUNT ITEM implies the derivation of the \( r \)-set \( R' \) defined on DEPARTMENT and SUPPLIER, having an attribute \{COUNT ITEM\} and represented by \( rR' \):

\[
rR' = \text{COUNT}^{S} <\text{DEPARTMENT} ; \text{SUPPLIER} ; \text{ITEM}> (\text{SUPPLY}).
\]

COUNT is only one of the aggregate functions; others, like SUM, MAX, MIN also define implicit properties and are formulated in a way analogous to COUNT.
We have proposed a set of operators, called s-operators, to form the manipulative part of the ERM. The s-operators have been defined with the help of the RA operators on the basis of the choice of the relation as the structural unit of the data-representational level of the ERM. The s-operators depart from their RA counterparts by being sensible to the semantic structure imposed by the ERM. For a flat schema, that is, consisting of e-sets and r-sets with no properties (attributes), they differ from the RA operators only by the extended meaning given to the entity-specification correspondences implying, for some of them, the embedding of natural-join.

The linguistic analogies of the s-operators were investigated. It has been shown that just as the ERM concepts are based on the way people perceive information, there is a close relationship between the application of the s-operators and the natural language sentence combination. As such they could be considered as the basis of a query language over the ERM having constructs close to the natural language ones. Such a query language, called ERROL (Entity-Relationship, Role Oriented, Query Language), was proposed and defined in [MR] and the s-operators were used to describe the semantics of the ERROL constructs [Mar].
REFERENCES


