THE SEMANTICS OF ERROL, AN ENTITY-RELATIONSHIP, ROLE ORIENTED, QUERY LANGUAGE

by

V.M. Markowitz and Y. Raz

Technical Report #279

June 1983
ABSTRACT

The concepts of the Entity-Relationship Model (ERM) are based on the way in which people perceive information. Accordingly, one would expect that a query language within ERM would have constructs based on the way people communicate, that is, natural language sentences. Such a query language is ERROL (Entity-Relationship, Role Oriented, Query-Language).

A recently reshaped Relational Algebra (RRA) has been proposed as a manipulative part for the ERM, following the choice of the relation as the structural unit of the ERM data-representation level. RRA is used to describe the semantics of the ERROL constructs.
1. INTRODUCTION

The Entity-Relationship Model (ERM) is a popular, semantically oriented database model. It provides a semantic structure description 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 stated by relationships which also are qualified by properties.

Concepts underlying the ERM are presented in [1]; a review may be also found in [3].

Relations are chosen for the data-representational level of the ERM [4]. The attempt of making the construction of complex algebraic expressions to follow patterns of natural sentence combination, is the incentive of reshaping the Relational Algebra. Based on the link between the ERM and the Relational Model established on the data-representation level, a reshaped Relational Algebra (RRA) has been outlined and proposed as the manipulative part of the ERM in [4]. A detailed, formal, presentation of a revised version of this "natural language" oriented Relational Algebra is in preparation.

Recently, a query language, having constructs based on the natural language sentence combination rules, has been proposed in [3]; it is called ERROL. The syntactic constructs of ERROL are briefly reviewed and RRA is used to define their semantics.

The reading of [3] and [4] is essential for the understanding of the following report.
2. THE SEMANTICS OF THE ERROL CONSTRUCTS

We are presenting the semantics of a query language within the Entity-Relationship Model called ERROL - Entity Relationship Role Oriented Query Language. ERROL takes advantage of the possibilities posed by the linguistic analogies of the ERM, by using denotations based on the simple sentences of the schema description, and by using constructs similar to the natural language sentence combination.

Object-set (entity-set or attribute) names and roles are the ERROL identifiers. Most of them are declared in the ER schema over which the queries are expressed. ERROL queries may involve constants (numeric constants, e.g. 2, or strings, e.g. 'ABC'). Roles appear in ERROL prefixed by a single quote (e.g. 'SUPPLIED').

A restriction is an elementary statement of the form "A ≠ c" or "A ≠ B", where c is a constant, A and B are object-sets and ≠ is a comparison operator belonging to the set of comparison operators associated with the v-set, which has to be the same, corresponding to both A and B. Notice that ≠ is a kind of role.

Associations are denoted in ERROL by the skeletons of the paraphrases of the corresponding description sentences [3]. For instance, one of the denotations of the r-set SUPPLY is:

SUPPLIER 'SUPPLYING ITEM DEPARTMENT.

When there is no ambiguity in a reference to an association, the single quote or a role that has not been declared in ERS may be used. This is the case with most properties. For example, the TYPE property of ITEM may be denoted by:

ITEM 'HAVING TYPE, or
ITEM TYPE.
Properties of a same r-set may be embedded in a single denotation. For example, the QTY and PRICE properties of SUPPLY may be denoted by:

SUPPLIER 'SUPPLYING ITEM "TO DEPARTMENT "IN QTY "AT PRICE.

If, for some n-ary (n > 2) r-set, interest is in the association of only part of the involved e-sets, a partial denotation may be used:

DEPARTMENT 'SUPPLIED "BY SUPPLIER.

ERROL accepts comments that may be placed anywhere within a query. A comment is prefixed by double quotation marks.

In the following presentation the ERROL key words are underlined.

All the examples are expressed for a database having the following description:

(REQUEST): DEPARTMENT 'REQUESTING ITEM "IN QTY
(STOCK) : ITEM 'STOCKED "BY SUPPLIER "IN QTY
(SUPPLY) : SUPPLIER 'SUPPLYING DEPARTMENT "WITH ITEM "IN QTY
(DEPARTMENT): DEPARTMENT 'HAS NAME
(ITEM): ITEM 'HAS TYPE
(SUPPLIER): SUPPLIER 'HAS LOCALITY
SUPPLIER 'HAS NAME.

2.1 The GET-CLAUSE

The target elements of a query are stated in the GET-CLAUSE, which is headed by the key-word GET and consists of a list of one or more sublists separated by semicolons. A sublist may contain:

(a) an e-set, for instance:

GET SUPPLIER;

(b) several attributes belonging to an e-set, possibly followed by the roles corresponding to the respective properties, separated by commas; the last attribute is part of a full property denotation, including the corresponding e-set, for instance:

GET NAME 'OF DEPARTMENT;
(c) several attributes belonging to an r-set, appearing in an embedded property-denotation of properties belonging to an r-set for instance:

```
GET QTY 'SUPPLIED ITEM "BY SUPPLIER "TO DEPARTMENT;.
```

Every e-set of a sublist (a) or (b), and at least one e-set of a sublist (c), will participate in the qualification phrase (TIS-CLAUSE; see below).

An ERROL query defines a derived r-set, q, and its GET-CLAUSE asserts the σ-scheme of q, S(q): every e-set appearing in the GET-CLAUSE has a corresponding specification in S(q); the c-symbols are the reference identifiers suffixing the denotations (see below), if there are any, or default, system provided, c-symbols. For example, the σ-scheme implied by

```
GET ITEM; DEPARTMENT!X
```

is

```
S(q) = {ITEM:#1; DEPARTMENT: X}.
```

**2.2 The TIS-CLAUSE**

A query generally has, beside the GET-CLAUSE, a statement asserting the required associations connecting the target e-sets. This statement is contained in a TIS-CLAUSE. A TIS-CLAUSE is delimited by the key-word TIS and a semicolon. It is based on the simple association denotations deducible from the ERS, which are combined in a complex qualification phrase. The last semicolon/string of semicolons in a query statement is/are replaced by a period.

**2.2.1 Referencing**

E-set correlation is accomplished with the help of identifiers, preceded by an "!" sign, suffixing the e-set denotations. For example:

```
DEPARTMENT:!DY, EMPLOYEE!EX.
```
The following implicit referencing is used, when possible, in order to keep the referencing limited:

(a) a same e-set in the GET-CLAUSE may be correlated by its simple, unreferenced, denotation;
(b) textual contiguity (see below); and
(c) an e-set suffixed by a single "!" in the TIS-CLAUSE refers to the corresponding e-set of the GET-CLAUSE.

The TIS-CLAUSE of an ERROL query, q, consists of a combination of r-set denotations. An r-set, R, represented by a relation rR, may be referenced several times within q, and each of these references, which we shall call r-set-occurrence, is represented by a relation-occurrence rR^j. The correlation-assessment mapping associated with a relation-occurrence rR^j, that is the o-scheme of rR^j, is straightforwardly deduced from the r-set-occurrence denotation; the c-symbols are the reference identifiers, or default (system provided), for implicit references.

In the sequel we shall use the following notations:

\[ xE \] - denotes a, possibly correlated, object-set whose name is E;
\[ 'R_{ij} \] - denotes the role of \( xE_i \) in r-set Rj;
\[ rR^j \] - denotes a relation-occurrence (j) representing the r-set-occurrence \( R^j \) which is referring to the r-set R;
* - marks operational-referencing; and

\[ xE_1 'R_{1j} \ldots xE_k \] - is a denotation for the r-set Rj which is defined on \( E_1 \) through \( E_k \), or an embedded property denotation, if some of the object-sets are attributes.

The denotation of an r-set-occurrence, \( R^j \), may not include the whole denotation of R, that is, can refer only to part of the object-sets.
two r-set denotations.

LOCALITY 'OF SUPPLIER 'SUPPLYING ITEM!

GET NAME 'OF DEPARTMENT TIS 'REQUESTING ITEM 'STOCKED by some supplier"

Chaining is based on the natural language sentence combination by relativization. The connection is accomplished through the last e-set of an association-denotation, where it was in an object position, by raising it to the subject position, of the following association-denotation. A first query example will illustrate the chaining of two r-set denotations.

(ERROL 1) "Find the names of departments requesting some item stocked by some supplier"

GET NAME 'OF DEPARTMENT TIS 'REQUESTING ITEM 'STOCKED 'BY SUPPLIER.

Restrictions always appear chained with association denotations. For instance TYPE = '1' is relevant when included in ITEM 'HAVING TYPE = '1'.

A longer chaining is accomplished when the place of the constant is taken in the restriction by another object-set denotation.

(ERROL 2) "Find the items which are stocked by suppliers located in localities of suppliers supplying these items"

GET ITEM TIS 'STOCKED 'BY SUPPLIER 'HAVING LOCALITY = LOCALITY 'OF SUPPLIER 'SUPPLYING ITEM!.

The chaining may also be accomplished by closed relative clause relativization where the object-subject contiguity is broken. The break is expressed in ERROL by a pseudo-role, indicated by "'*"."
The closed-relative-clause chaining has the same interpretation as above, but, unlike the chaining by relativization, the set of mutual references may be empty.

Yet another type of chaining is through restriction, when the evaluation includes an RRA 0-join. Let $R^i_1$ and $R^i_2$ be two...
r-set-occurrences and \( xE_k \theta xE_n \) a restriction. The chaining of \( R_1^j \) and \( R_2^j \) by this restriction is expressed by

\[ xE_m^1 \ldots xE_k \theta xE_n^1 \ldots xE_p \]

and represented by

\[ rR' = R_1^j[xE_k \theta xE_n]rR_2^j \]

The RRA \( \theta \)-join may imply \( \epsilon \)-join, \( \epsilon \)-projection or both.

The query (ERROL 2) is evaluated as follows:

1. \( S(q) = \{ITEM:#1\} \)
   \[ S(rR_1) = \{ITEM:#1; SUPPLIER:#2\} \]
   \[ S(rR_2) = \{SUPPLIER:#2; LOCALITY*\} \]
   \[ S(rR_2') = \{SUPPLIER:#4; LOCALITY*\} \]
   \[ S(rR_2^1) = \{SUPPLIER:#4; ITEM:#1\} \]

2. \( rR' = R_2^1[LOCALITY=LOCALITY] rR_2^2 \)
   \[ S(rR') = \{SUPPLIER:#2; SUPPLIER:#4\} \]

3. \( rR'' = R_1^1 \times rR', \text{ where } M(rR_1,rR') = \{SUPPLIER:#2\} \)
   \[ S(rR'') = \{ITEM:#1; SUPPLIER:#4\} \]

4. \( rR''' = rR'' \times rR_3^1, \text{ where } M(rR'',rR_3^1) = \{SUPPLIER:#4; ITEM:#1\} \)
   \[ S(rR''') = \{ITEM:#1\} \text{ and } q = rR'''. \]

Notice the correlation-assessment stage; the c-symbols are all implicit.

Whenever the restriction is of the form \( xE \theta c \), the evaluation involves RRA selection; thus,

\[ xE_m^1 \ldots xE_k \theta c \] is represented by

\[ rR' = R_1^j[xE_k \theta c]. \]
2.2.3 Negation

An r-set, R, has a complementary r-set, that groups the "possible" relationships that are not elements of R. All the "possible" relationships of an r-set R are obtained by the Cartesian product of the e-sets on which the r-set is defined. The complementary r-set of an r-set R is denoted by the negated denotation of R.

(ERROL 4) "Get the pairs of suppliers and departments, such that the department is not requesting items stocked by the respective supplier"

GET SUPPLIER; DEPARTMENT TIS NOT 'REQUESTING ITEM 'STOCKED 'BY SUPPLIER!.

The ERROL negation is evaluated through the RRA not operation.

Let $R^j$ be an r-set-occurrence denoted by $xE_{m1} \ldots xE_{mk}$. Then $xE_{m1} \text{TIS NOT} \ldots xE_{mk}$ is represented by

$$rR' = -orR^j.$$  

The query (ERROL 4), for instance, is evaluated as follows:

(1) $S(q) = \{\text{SUPPLIER:#1; DEPARTMENT:#2}\}$

$S(rR_1^1) = \{\text{DEPARTMENT:#2; ITEM:#3}\}$

$S(rR_2^1) = \{\text{ITEM:#3; SUPPLIER:#1}\}$

(2) $rR' = rR_1^1 \times rR_2^1$, $M(rR_1^1, rR_2^1) = \{\text{ITEM:#3}\}$

$S(rR') = \{\text{DEPARTMENT:#2; SUPPLIER:#1}\}$

(3) $rR'' = -rR'$

$S(rR'') = S(rR')$, and $q = rR''$.

2.2.4 Branching

Branching is based on the natural language sentence combination by coordination; the coordination connectors are AND and OR. In order to enforce a certain order of evaluation brackets may be used.
(ERROL 5) "Get the departments requesting items that are stocked by suppliers from LONDON, supplied with items of type 1".

\[ \text{GET DEPARTMENT TIS 'REQUESTING ITEM 'STOCKED 'BY SUPPLIER 'HAVING LOCALITY='LONDON' AND 'SUPPLIED 'WITH ITEM 'HAVING TYPE='l}. \]

The OR-branching in multi-target queries where more than one sublist are in the GET-CLAUSE, needs references to all the e-sets of the GET-CLAUSE, in every branch in order to be unambiguous; otherwise the branches are perceived as implicitly bordered, through the pseudo-role, with all the unreferenced target e-sets.

The AND-branching is semantically equivalent to chaining by relativization. Accordingly, its evaluation involves RRA product.

Let \( R_{1}^{j} \) and \( R_{2}^{i} \) be two r-set occurrences. Their AND branching expressed by

\[ xE_{k} \text{ TIS 'R}_{k1} \ldots xE_{m} \text{ AND 'R}_{k2} \ldots xE_{n} \text{ is represented by} \]

\[ rR' = rR_{1}^{j} \times rR_{2}^{i}. \]

The OR-branching evaluation involves RRA bordered union. If in the above branching we would replace AND with OR, the result would be represented by \( rR' = rR_{1}^{j} \cup rR_{2}^{i} \). Whenever \( rR_{1}^{j} \) and \( rR_{2}^{i} \) are union-compatible, the bordered-union reduces to ordinary union.

2.2.5 Nesting

A branching may have as pivot any inner element of a TIS-CLAUSE, not necessarily the leading subject. In this case the respective element will head an inner TIS-CLAUSE

(ERROL 6) "Get the departments requesting item that are either supplied to them or having type 1".

\[ \text{GET DEPARTMENT TIS 'REQUESTING ITEM TIS 'SUPPLIED 'TO DEPARTMENT! OR 'HAVING TYPE='1}. \]
An inner TIS-CLAUSE contains chains starting with roles corresponding to the pivot subject (e-set in the above example), like the general TIS-CLAUSE. While the general TIS-CLAUSE corresponds to an e-set, inner TIS-CLAUSEs may qualify attributes or sets as well.

The nesting types mentioned above may enforce a certain, parenthesis-like, order of evaluation, which we shall exemplify by outlining the evaluation of query (ERROL 6):

1. \( S(q) = \{\text{DEPARTMENT:#1} \} \)
   \( S(r_{R1}^1) = \{\text{DEPARTMENT:#1; ITEM:#2} \} \)
   \( S(r_{R2}^1) = \{\text{ITEM:#2; DEPARTMENT:#1} \} \)
   \( S(r_{R3}^1) = \{\text{ITEM:#2; TYPE}^* \} \)
2. \( r_{R1}' = r_{R3}^1 \text{[TYPE='1']} \), \( S(r_{R1}') = \{\text{ITEM:#2} \} \)
3. \( r_{R2}'' = r_{R2}^1 \cup r_{R1}' \), \( M(r_{R2}^1, r_{R1}') = \{\text{ITEM:#2} \} \)
   \( S(r_{R2}'') = \{\text{ITEM:#2; DEPARTMENT:#1} \} \)
4. \( r_{R2}''' = r_{R1}^1 \times r_{R2}'' \), \( M(r_{R1}^1, r_{R2}'') = S(r_{R2}'') \)
   \( S(r_{R2}''') = \{\text{DEPARTMENT:#1} \}, \text{ and } q = r_{R2}'''. \)

Notice that stages (2) and (3) must precede (4).

2.2.6 Set Expressions

The basis to express universal quantification is the correspondence (see [3]). In ERROL the correspondence is denoted by the association whose elements form the i-component followed by the key-word SET and the denotation corresponding to the i-set.

(ERROL 7) "Get the suppliers stocking at least all the items requested by some department"

GET SUPPLIER TIS 'STOCKING SET ITEM CONTAINS SET ITEM 'REQUESTED "BY DEPARTMENT."
Sets may be compared by set operators (EQ, IN, CONTAINS, NOT EQ, NOT IN, NOT CONTAINS), with other compatible sets, i.e. sets containing objects of a same type.

Any element of the correspondence may be qualified.

(ERROL 8) "Get the departments supplied at least by SAXON and PEUGEOT suppliers"

GET DEPARTMENT TIS 'SUPPLIED 'BY SET SUPPLIER CONTAINS SET SUPPLIER TIS 'HAVING NAME TIS='SAXON' OR='PEUGEOT'.

The i-set may also be qualified, provided all the chains in the respective TIS-CLAUSE are headed by set comparison-operators.

A correspondence may be expressed over a derived r-set. A new role, not declared in the ERS for associations connecting the e-sets referred to by this correspondence, declares the intention of defining a derived r-set. The actual definition is included in the TIS-CLAUSE qualifying the i-set element (not the i-set as a whole).

(ERROL 9) "Get the departments for which all the suppliers located in LONDON, stock some item requested by them"

GET DEPARTMENT TIS 'HAVING SET SUPPLIER TIS 'STOCKING ITEM 'REQUESTED 'BY DEPARTMENT!; CONTAINS SET SUPPLIER TIS 'HAVING LOCALITY='LONDON'.

In order to make the derived r-set definition unambiguous, the referencing inside the TIS-CLAUSE of the i-set element has to follow two rules:

(a) every e-set of the i-component has to be referenced at least once, and

(b) all the other references have to be local to this TIS-CLAUSE.

Set expressions may be nested, one inside another, the only restriction is the satisfaction of the above referencing rules.
A statement of the form \( \text{SET} \ x \in \text{operation} \ \text{SET} \ x \in \) resembles the restriction, therefore it will be called set-restriction. The same as for restriction, in a set-restriction the two components must be compatible, which means that the sets consist of objects of a same type. Similar to the restrictions, set-restrictions can be integrated in chains and branches. The division-set consists of the i-set element alone.

Let two correspondences be expressed over the r-set occurrences \( R_j^1 \) and \( R_i^1 \), and \( \text{SET} \ x \in k \ \text{CONTAINS SET} \ x \in k \) be a set-restriction; the chaining of these correspondences by the set-restriction, is expressed by:

\[
x \in_p R_{p1} \ldots \ \text{SET} \ x \in_k \ \text{CONTAINS SET} \ x \in_k R_{m2} \ldots x \in_n
\]

and is represented by

\[
rR' = rR_1^j \upsilon rR_2^i,
\]
with the division-set \( D(rR_1^j, rR_2^i) \) consisting of \( x \in k \).

The query (ERROL 7), for instance, is evaluated as follows:

1. \( S(q) = \{\text{SUPPLIER} : #1\} \)
   \( S(rR_1^i) = \{\text{SUPPLIER}^* : #1; \ \text{ITEM}^*\} \)
   \( S(rR_2^i) = \{\text{ITEM}^*; \ \text{DEPARTMENT}^*\} \)

2. \( rR' = rR_1^j \upsilon rR_2^i; \ D(rR_1^j, rR_2^i) = \{\text{ITEM}^*\} \)
   \( S(rR') = \{\text{SUPPLIER} : #1\}, \ \text{and} \ q = rR'. \)

When the set-operation is \( \text{IN} \), the expression above is represented by

\[
rR' = rR_1^i \upsilon rR_2^j.
\]

When the set-operation is \( \text{EQ} \), the expression above is represented by

\[
rR' = (rR_1^j \upsilon rR_2^i) \ \text{n} (rR_1^i \upsilon rR_2^j).
\]

Whenever an element of a correspondence is qualified, the respective set-expression involves nesting. In a first case, the qualification refers to a single element. Let \( rR_j^i \) be an r-set-occurrence
involved in:

\[
xe^p_R \ldots \text{SET} \ xe_k \ TIS \ldots ;
\]

where \( xe_k \) is the qualified element. Any set expression of which the above correspondence is part of, is evaluated as follows:

(a) first is evaluated \( rr' \), whose \( \sigma \)-scheme consists only of \( xe_k \), described by the TIS-CLAUSE;

(b) next is evaluated \( rr'' = rr^j \times rr' \) where  \( M(rr^j, rr') = \{ xe_k \} \); and

(c) \( rr'' \) will be the operand of an RRA division.

The query (ERROL 8), for instance, is evaluated as follows:

1. \( S(q) = \{ \text{DEPARTMENT:#1} \} \)
   \( S(rr^1) = \{ \text{DEPARTMENT*:#1}; \text{SUPPLIER*} \} \)
   \( S(rr^2) = \{ \text{SUPPLIER*}; \text{NAME*} \} \)

2. \( rr' = rr^2 \left[ \text{NAME='SAXON' OR NAME='PEUGEOT'} \right] \)
   \( S(rr') = \{ \text{SUPPLIER*} \} \).

3. \( rr'' = rr^1 \div rr', \ D(rr^1, rr') = \{ \text{SUPPLIER*} \} \)
   \( S(rr'') = \{ \text{DEPARTMENT:#1} \} , \text{ and } q = rr'' \).

A more complex case of nesting is when the complete description of a derived r-set, over which the correspondence is expressed is included in the qualification. In such a case the relation \( rr' \), described by the TIS-CLAUSE, is evaluated and \( S(rr') \) consists of the specifications of all the e-sets involved in the derived r-set; \( rr' \) may be part of an RRA division.

The query (ERROL 9), for instance, is evaluated as follows:

1. \( S(q) = \{ \text{DEPARTMENT:#1} \} \)
   \( S(rr^1) = \{ \text{SUPPLIER*}; \text{ITEM:#3} \} \)
   \( S(rr^2) = \{ \text{ITEM:#3}; \text{DEPARTMENT:#1} \} \)
   \( S(rr^3) = \{ \text{SUPPLIER*}; \text{LOCALITY*} \} \)
(2) \( rR' = rR_1^1 \times rR_2^1 \), \( M(rR_1^1,rR_2^1) = \{ \text{ITEM:#3} \} \)

\( S(rR') = \{ \text{SUPPLIER*}; \; \text{DEPARTMENT:#1} \} \)

(\( rR' \) is the derived r-set)

(3) \( rR'' = rR_3^1[\text{LOCALITY='LONDON']} \)

\( S(rR'') = \{ \text{SUPPLIER*} \} \)

(4) \( rR''' = rR' \oplus rR'' \), \( D(rR',rR'') = \{ \text{SUPPLIER*} \} \)

\( S(rR'''') = \{ \text{DEPARTMENT:#1} \} \), and \( q = rR''' \).

2.2.7 Derived Properties (Aggregate Functions)

With the help of the correspondence, it is possible to express derived properties. Such a property associates the i-components elements of a correspondence with values obtained through a computation applied to the i-set sets. These computations are performed by so-called aggregate functions (af): \text{COUNT}, \text{SUM}, \text{MAX}, \text{MIN}. In the correspondence denotation the above key words will replace the key work \text{SET} in order to obtain a derived-property denotation.

\text{COUNT} is applied directly to sets of entities and returns the number of entities in these sets.

(ERROL 10) "Get the items that are stocked by at least 3 suppliers".

\text{GET ITEM TIS 'STOCKED "BY COUNT SUPPLIER \geq 3."}

\text{SUM} processes attribute-instances corresponding to a property, such that is takes into account any duplicate attribute-instance.

(ERROL 11) "Get the departments supplied with a total quantity of items greater than \text{22000}".

\text{GET DEPARTMENT TIS 'SUPPLIED "WITH SUM QTY > 22000."}

\text{MAX} and \text{MIN} are similar to \text{COUNT}.

Derived properties are expressible not only with the help of correspondences over explicit r-sets, but also with correspondences...
over implicit (derived) r-sets. The denotation and the referencing rules are similar to those for set expressions based on correspondences over derived r-sets.

A derived property, precisely as an explicit one, may be chained with other properties, whether they are derived or not, and a new attribute may be involved in restrictions or may be qualified by a TIS-CLAUSE, the same as an explicit one.

A derived property may, in fact, appear in any place in which an explicit one is used, including the GET-CLAUSE, as illustrated by the following example.

(ERROL 12) "Get the name of the department together with the number of the items they request".

The aggregate-function expressions define new attributes and are involved in restrictions. Let \( R_j \) be the r-set-occurrence involved in the aggregate-function expression

\[
x_{E_p} \cdot R_p \ldots \text{COUNT} x_{E_k};
\]

it describes a derived e-set represented by

\[
rR' = \varphi_{\text{COUNT}} <C; \{x_{E_k}\}> (rR_j),
\]

where \( C \) is the af-component and \( \{x_{E_k}\} \) is the af-argument and \( S(rR') \) includes, besides \( C \), the specification of the new attribute.

The query (ERROL 10), for instance, is evaluated as follows:

1. \( S(q) = \{\text{ITEM:#1}\} \)
   \( S(rR_1^1) = \{\text{ITEM*:#1}; \text{SUPPLIER*}\} \)

2. \( rR' = \varphi_{\text{COUNT}} <\{\text{ITEM:#1}\}; \{\text{SUPPLIER*}\}> (rR_1^1) \)
   \( S(rR') = \{\text{ITEM:#1}; A*\} \) where \( A \) is the new attribute.

3. \( rR'' = rR'[A > 3] \)
   \( S(rR'') = \{\text{ITEM:#1}\}, \text{and } q = rR''. \)
When the \text{SUM} aggregate-function is used, a reference to the r-set characterized by the corresponding attribute, has to be embedded in the respective expression, since \text{SUM} is applied on property-representations.

Thus, the query (ERROL 11) is evaluated as follows:

(1) \( S(q) = \{\text{DEPARTMENT} : \#1\} \)
\[ S(rR_1^1) = \{\text{DEPARTMENT}^* : \#1; \text{ITEM}^*; \text{QTY}^*\} \]

(2) \( rR' = \sum_{\text{SUM}} \langle \text{DEPARTMENT} : \#1; \{\text{QTY}^*\} \rangle (rR_1^1) \)
\[ S(rR') = \{\text{DEPARTMENT} : \#1; A^*\} \]

(3) \( rR'' = rR'[A^* > 22000] \)
\[ S(rR'') = \{\text{DEPARTMENT} : \#1\} , \text{and} \ q = rR'' \]

More complex af-expressions may involve nesting; the evaluation of such expressions is similar to that of nested set-expressions.
3. CONCLUSION

We have given the semantics of the ERROL basic constructs with the help of a reshaped Relational Algebra (RRA). Since both ERROL and RRA have a close relationship with natural-language sentence combination rules (see [3] and [4]), the straightforward use of the RRA operators in expressing the semantics of the ERROL constructs is not surprising.

We haven't intended to give a full semantic definition of ERROL. A more complete and, possibly, more precise semantic definition of ERROL, is in preparation.
REFERENCES


