Semantic Definition of a Subset of the Structured Query Language (SQL)

Peter J. Hoogenboom

UUCS-91-026

Department of Computer Science
University of Utah
Salt Lake City, UT 84112, USA

December 9, 1991

Abstract

SQL is a relational database definition and manipulation language. Portions of the manipulation language are readily described in terms of relational algebra. The semantics of a subset of the SQL `select` *statement is described. The* `select` *statement allows the user to query the database. The* `select` *statement is shown to be equivalent to a series of relational and set operations. The semantics are described in terms of abstract data types for relation schemes, tuples, and relations. Certain forms of the union or intersection of two* `select` *statements are shown to have equivalent single* `select` *statement forms.*

Contents

1 Introduction

The Structured Query Language (SQL) consists of a set of commands and utilities for defining and manipulating a relational database. A *relational database* is viewed by the user as a collection of *relations* or *tables.* In fact, these two terms are often used interchangeably.

A table is an unordered collection of *rows*, or *tuples.* Each element of a row is generally referred to as an *attribute* or *column.* The *degree* or *arity* of a table is the number of columns in the table. The *cardinality* of a table is the number of tuples in the table.

One of the first implementations of SQL was in the early 1980's as an interface to a relational database management system called System/R [2, 7]. In 1986, the American National Standards Institute (ANSI) adopted a standard syntax and English language semantics[1].

2 The SQL `select` Statement

The subset of SQL to be considered here is the `select` statement. This subset of SQL allows the user to make queries about a previously defined database (a set of relations).

2.1 Abstract Syntax

$\Psi \in$ Query	$\varepsilon \in$ Expression	$\xi \in$ Identifier	$I \in$ Integer
$\Sigma \in$ Selection	$\Pi \in$ Predicate	$S \in$ String	$R \in$ Real
$\Gamma \in$ Table Expression	$\Upsilon \in$ Table List	$C \in$ Column	$B \in$ Boolean

$$
\begin{array}{lcl}
\Psi & \Rightarrow & \texttt{select}\ \Theta \mid \Psi\ \texttt{union}\ \Psi \mid \Psi\ \texttt{intersection}\ \Psi \\
\Theta & \Rightarrow & \Sigma\ \Gamma \\
\Sigma & \Rightarrow & * \mid \varepsilon \mid \varepsilon,\ \Sigma \\
\varepsilon & \Rightarrow & \varepsilon+\varepsilon \mid \varepsilon-\varepsilon \mid \varepsilon*\varepsilon \mid \varepsilon/\varepsilon \mid +\varepsilon \mid -\varepsilon \mid (\varepsilon) \mid R \mid I \mid \xi \mid S \mid C \\
 & \Rightarrow & \varepsilon\ \texttt{or}\ \varepsilon \mid \varepsilon\ \texttt{and}\ \varepsilon \mid \texttt{not}\ \varepsilon \mid \Pi \mid (\ \varepsilon\ ) \\
\Gamma & \Rightarrow & \texttt{from}\ \Upsilon \mid \texttt{from}\ \Upsilon\ \texttt{where}\ \varepsilon \\
\Upsilon & \Rightarrow & \xi \mid \xi\ \xi \mid \Upsilon,\ \Upsilon \\
C & \Rightarrow & \xi \mid \xi\ .\ \xi \\
\Pi & \Rightarrow & \varepsilon\ \Omega\ \varepsilon \\
\Omega & \Rightarrow & = \mid <> \mid < \mid > \mid <= \mid >=
\end{array}
$$

2.2 Concrete Syntax

The concrete syntax is taken from the ANSI standard [1] and is given in Table 1 and Table 2. Any deviations from the SQL standard syntax are to remove portions of syntax unrelated to the `select` statement and to remove portions of `select` statement syntax that will not be implemented.

<table>
<tr><td><uppercase></td><td>⇒</td><td>A| B| C| ··· | X| Y| Z</td></tr>
<tr><td><lowercase></td><td>⇒</td><td>a| b| c| ··· | x| y| z</td></tr>
<tr><td><digit></td><td>⇒</td><td>1| 2| 3| ··· | 8| 9| 0</td></tr>
<tr><td><letter></td><td>⇒</td><td><uppercase> | <lowercase></td></tr>
<tr><td><character></td><td>⇒</td><td><letter> | <digit></td></tr>
<tr><td><ident></td><td>⇒</td><td><letter> [{ _ | <letter> | <digit> }*]</td></tr>
<tr><td><literal></td><td>⇒</td><td><character string lit> | <numeric lit></td></tr>
<tr><td><character string lit></td><td>⇒</td><td>'<nonquote character>*'</td></tr>
<tr><td><nonquote character>[a]</td><td>⇒</td><td></td></tr>
<tr><td><newline>[b]</td><td>⇒</td><td></td></tr>
<tr><td><numeric lit></td><td>⇒</td><td><exact numeric lit> | <approx numeric lit></td></tr>
<tr><td><exact numeric lit></td><td>⇒</td><td>[+ | -] <unsigned int> [[.] <unsigned int>]</td></tr>
<tr><td><approx numeric lit></td><td>⇒</td><td><exact numeric lit>E [+ | -] <digit>+</td></tr>
<tr><td><unsigned int></td><td>⇒</td><td><digit>*</td></tr>
<tr><td><correl name></td><td>⇒</td><td><ident></td></tr>
<tr><td><table name></td><td>⇒</td><td><ident></td></tr>
<tr><td><column></td><td>⇒</td><td>[<table name>. | <correl name>.] <ident></td></tr>
<tr><td><table reference></td><td>⇒</td><td><table name> [<correl name>]</td></tr>
<tr><td><value spec></td><td>⇒</td><td><ident> | <literal></td></tr>
<tr><td><comparison></td><td>⇒</td><td>= | <> | < | > | <= | >=</td></tr>
</table>

[a] A <nonquote character> is any character except the single quote character ('). This is as specified by the ANSI SQL Standard.

[b] The ANSI SQL Standard states that the <newline> character is implementation dependent. For this implementation, the <newline> character is the linefeed character.

Table 1: Common Elements Used in the Query Statement Syntax (adapted from the ANSI SQL Standard)

<table>
<tr><td><query></td><td>⇒</td><td>select <select list> <table expr></td></tr>
<tr><td></td><td>⇒</td><td><query> [union | intersection] <query></td></tr>
<tr><td><select list></td><td>⇒</td><td><expn> [{, <expn> }*] | *</td></tr>
<tr><td><expn></td><td>⇒</td><td><term> | <expn> + <term> | <expn> - <term></td></tr>
<tr><td><term></td><td>⇒</td><td><factor> | <term> * <factor> | <term> / <factor></td></tr>
<tr><td><factor></td><td>⇒</td><td>[+ | -] <primary></td></tr>
<tr><td><primary></td><td>⇒</td><td><value spec> | <column> | (<expn>)</td></tr>
<tr><td><table expr></td><td>⇒</td><td><from> [<where>]</td></tr>
<tr><td><from></td><td>⇒</td><td>from <table reference> [{, <table reference> }*]</td></tr>
<tr><td><where></td><td>⇒</td><td>where <search cond></td></tr>
<tr><td><search cond></td><td>⇒</td><td><bool term> | <search cond> or <bool term></td></tr>
<tr><td><bool term></td><td>⇒</td><td><bool factor> | <bool term> and <bool factor></td></tr>
<tr><td><bool factor></td><td>⇒</td><td>[not] <bool primary></td></tr>
<tr><td><bool primary></td><td>⇒</td><td><predicate> | (<search cond>)</td></tr>
<tr><td><predicate></td><td>⇒</td><td><expn> <comparison> <expn></td></tr>
</table>

Table 2: Query Statement Syntax (adapted from the ANSI SQL Standard **select** statement syntax)

2.3 Notation

The notation used in Table 1 and Table 2 is BNF (Backus Normal Form) with the following extensions:

1. Square brackets ([...]) indicate optional elements.
2. An element E repeated zero or more times is denoted as E^*. An element E repeated one or more times is denoted as E^+.
3. Curly braces ({ ... }) indicate sequences of elements.

3 Semantic Definition

3.1 Semantic Techniques

Standard semantic techniques using search and backtrack elements similar to Prolog were proposed for this project. Further study of the semantics of the SQL select statement indicated that the queries specified by the select statement were more readily described in terms of relational algebra. The relational algebra is described in terms of set operations (union, intersection, complex product) and several relational operations (selection, projection, division) [4].

For example, a typical SQL query using the select statement has the following form:

```
select < select list >
from < table list >
where < search cond >
```

This query has a straightforward translation to relational algebra and set operations:

$$\pi_{<select\ list>}(\sigma_{<search\ cond>}(\times(<table\ list>)))$$

where π is projection, σ is selection, and $\times$ is the complex product. These operations will be defined more formally later.

With this in mind, it seems natural to define the semantics of SQL query statements in terms of these operations. This approach is desirable for several reasons. First, it makes the semantics more understandable because the meaning is stated in the well-defined terms of relational algebra. In addition, proving statements about the SQL query language should prove to be easier because proofs can draw upon a sound theoretical background and a large body of research on relational algebra. Finally, in this approach several abstract data types will be defined (relations, tuples, schemes). Abstracting these elements of the language allows the semanticist to prove statements about the abstract data types separate from the semantics of the language.

3.2 Informal Semantics

For the purposes of demonstrating the semantics of SQL queries, suppose there exists a database of hosts (computer systems) and processes. Also suppose there is a hosts relation and a process relation. Portions of these relations are shown in Figures 3 and 4.

A query such as: "`select name,load1,status from host`" would provide a table with 3 columns: name, load1, and status. The where clause allows qualification of the rows in the resulting table. For example,

```
select name, status, load1
from host                                  (1)
where load1 > 1.0 and name <> 'cs'
```

would produce Table 5, a 3-column table with all hosts whose `load1` attribute is greater than 1.0 and whose name is not "cs".

<query>. A <query> operates on tables and produces a (possibly empty) table as a result. More complex queries can be formed by using the `union` or `intersection` operators. The `union` operator takes 2 tables, T_1 and T_2, and produces a new table containing the rows of both T_1 and T_2. Duplicate rows in the resulting table are removed. The `intersection` operator takes 2 tables, T_1 and T_2, and produces a new table containing the rows that are in both T_1 and T_2.

Name	Status	Users	Load1	Load5	Load15	Processes
acme	down	0	0.00	0.00	0.00	()
asylum	up	5	1.24	1.24	1.27	(16107,19811,20170)
cs	up	25	1.34	1.00	0.92	(9834,12111,12112)
hellgate	up	0	1.70	0.92	0.70	(4112)
peruvian	up	28	1.63	1.11	1.30	(25,28,31,100,298)
jaguar	up	3	0.14	0.26	0.34	(2178)
jensen	up	1	0.43	0.13	0.04	(431,455)
shafer	up	1	0.09	0.10	0.04	(19928,19929,20050)

Table 3: Host Relation

Name	PID	PPID	User	Mem	CPU	Size	RSS	Host
rshd	25	298	root	0.3	0.0	100	60	peruvian
csh	28	25	hoogen	0.3	20.0	176	80	peruvian
ps	31	28	hoogen	0.9	18.9	252	228	peruvian
emacs	2178	2177	allen	2.5	0.6	996	700	jaguar
mail	431	430	cruse	0.0	0.0	208	36	jensen
xterm	455	454	yih	2.1	0.0	196	112	jensen
xcalc	12112	12111	zeleznik	0.8	0.0	208	204	cs
csh	9834	212	starkey	0.4	0.0	184	92	cs

Table 4: Process Relation

Name	Status	Load1
asylum	up	1.24
hellgate	up	1.70
peruvian	up	1.63

Table 5: Result of query 1.

The columns of the tables T_1 and T_2 must be of the same types. Thus, the i^{th} column named in the query that produced T_1 must be the same type as the i^{th} column named in the query that produced T_2.

<select list>. The <select list> is used to select specific attributes (columns) from tables. If all columns of a table are desired, the * character may be specified. Otherwise, all columns to be in the result are specified separated by commas. The columns named in the <select list> must unambiguously identify the desired column. If 2 tables in a query have the same column names, the column name must be prepended with a <correl name> to unambiguously identify the column. For example, this query involves the **name** column of the host and process tables:

```
select H.name,P.name from host H, process P
where H.name = 'asylum' and P.name = 'csh'
```

<from>. The <from> is used to specify the tables to be considered in the query. If the query involves comparing 2 or more instances of the same table, then a <correl name> can be specified to uniquely identify each instance. The <correl name> acts as a temporary and local name for the table. For example,

```
select C.pid, P.pid, P.name
from process P, process C
where P.pid = C.ppid and C.name = 'emacs'
```

will locate all processes with the name 'emacs'. A new table will be constructed containing the name of each of these processes, the process id, and name of their parent processes.

<where>, <search cond>, and <predicate>. The <where> clause is used to specify relationships that must exist among the columns in the result. The relationships are specified as a <search cond>. For a given row that is being considered by a query, the row is said to *qualify* if the <search cond> of the <where> clause evaluates to a truth value of *true*. Otherwise, the row is *disqualified.*

The <predicate> and <search cond> clauses allow the user to specify the conditions that qualify or disqualify a row from the result of a query. These conditions are evaluated to a truth value. For a given row, a *true* value will qualify the row for the resulting table. A *false* or *unknown* value will disqualify the row. Two or more <predicate> clauses can be connected by the boolean operators **and** and **or** to form more complex <search cond> clauses. The <predicate> clause is used to specify a comparison between values.

<comparison>. The comparison operators are the binary operators that appear in many programming languages. Numeric and character string values can be compared.

Comparison of numeric values is according to their algebraic values. Comparison of character strings is accomplished by comparing characters in the same ordinal position. If the strings are of different lengths, then the comparison is performed with the shorter string extended with spaces to the length of the longer string.

If a <predicate> is of the form "`a` <comparison> `b`", then the result of <predicate> is either *true*, *false*, or *unknown*. The result is *unknown* if either `a` or `b` is *null*. The SQL Standard specifies that nothing, not even another null value, should be equal to the *null* value.

3.3 The Relational Model

The relational model of databases was first described by Codd [3] in 1970. Since then, it has been extensively studied. This section provides a brief overview of the relational model. Formal terminology and concepts introduced here are from Yang [9] and Delobel and Adiba [5] and will be used in subsequent sections.

Attributes

A relational database is composed of a set of *attributes* $A_1, A_2, \ldots A_n$. Each attribute is composed of a single data type. The *domain* or *value-set* of each attribute A_j is written $DOM(A_j)$. If S is a subset of all possible attributes, we write $DOM(S) = \bigcup_{A_k \in S} A_k$.

Tuples

The rows in a relation are referred to as *tuples*. Formally, let X be a subset of the set of all attributes in the database. Let $DOM(X)$ be the domain of X. Let μ: $X \rightarrow DOM(X)$ be a function defined as follows:

$$\mu = \{(A_{i_1}, a_1), \ldots, (A_{i_k}, a_k)\}$$

Each A_{i_j} for $1 \leq j \leq k$ is an attribute in X and an argument to the function μ. Each a_j for $\leq j \leq k$ is a value in $DOM(A_{i_j})$. Thus, $\mu(A_{i_j}) = a_j$. The function μ is called a *tuple over X*.

If $Y \subseteq X$, then the *Y-value(s)* of μ is written

$$\mu[Y] = \{(A_m, a_m) \mid a_m = \mu[A_m], 1 \leq m \leq \mid Y \mid, A_m \in X\}$$

All possible tuples over X is the *complex product* of each domain A_{i_j}. The complex product of domains is defined as:

$$\begin{aligned} TUP(X) &= \times(DOM(A_{i_1}), DOM(A_{i_2}), \ldots, DOM(A_{i_k})) \\ &= \left\{a_1 a_2 \ldots a_k \mid a_j \in DOM(A_{i_j}), j = 1, 2, \ldots, k\right\} \end{aligned}$$

Thus, each tuple μ over X is an element of $TUP(X)$.

Relation Schemes

The scheme of a relation is also known as the *intention* of the relation. The scheme of a relation is defined as a subset of all possible attributes of the database. Thus, a relation over S is a subset of $TUP(S)$. The *cardinality* of a scheme is the number of attributes in the scheme.

Relations

A relation is defined in terms of its scheme. Thus, a *relation over* S is a relation with scheme S. The *arity* of a relation over S is equal to the cardinality of S. The *cardinality* of a relation is equal to the number of tuples belonging to that relation.

Since the relation domain is the key component of the query language, several important operations on relations are discussed here.

Union. The union of two relations R_1 and R_2 is a relation R_3 whose tuples are also in R_1 or R_2 (or both). Formally,

$$R_1 \cup R_2 = \{\mu \mid \mu \in R_1 \; or \; \mu \in R_2\}$$

This operation is allowed only when R_1 and R_2 are *union-compatible.* Union-compatibility is defined as the following property:

Definition 1 *Two relations R_1 and R_2 are union-compatible if their schemes are union-compatible.*

Definition 2 *Two Schemes A and B are union-compatible if there exists a bijection such that for each $A_j \in A$ there exists exactly one $B_k \in B$ for which $DOM(A_j) = DOM(B_k)$.*

Note that the union of any relation R and the empty relation Φ is the relation R.

$$R \cup \Phi = R$$

Complex Product. The *complex product* is very similar to the cartesian product except the ordering of the attributes in a complex product is insignificant. The cartesian product of two relations R_1 and R_2 is denoted $R_1 \times R_2$ and produces all possible pairs of tuples (μ_{R_1}, μ_{R_2}). Rather than a pair of tuples, the complex product produces the concatenation of the tuples.

The complex product of two relations R_1 and R_2 is denoted $R_1 * R_2$ and is defined as

$$R_1 * R_2 = \{\mu \mid \mu_{R_1}\mu_{R_2} \; and \; (\mu_{R_1}, \mu_{R_2}) \in R_1 \times R_2\}$$

Intersection. The intersection of two relations R_1 and R_2 is a relation T whose tuples are in R_1 and R_2. Formally,

$$R_1 \cap R_2 = \{\mu \mid \mu \in R_1 \, and \, \mu \in R_2\}$$

This operation is allowed only when R_1 and R_2 are union-compatible.

Note that the intersection of any relation R and the empty relation Φ is the empty relation.

$$R \cap \Phi = \Phi$$

Projection. The projection operation takes a relation and produces a relation whose attributes are a subset of the original relation's attributes. Formally, if R is a relation over S and X is a subset of S, then the *projection* of R onto X, written as $\pi_X(R)$ is a relation over X.

$$\pi_X(R) = \{\mu[X] \mid X \subseteq S \, and \, \mu \in R\}$$

Selection. The selection operation allows certain tuples to be selected from a relation. The condition for selection is specified in a *formula* which is defined inductively as [9]:

1. $F_1 \Omega F_2$, $F_1 \Omega c$, and $c \Omega F_1$ are formulas where F_1 and F_2 are compatible attributes, c is a constant in $DOM(F_1)$, and Ω is an arithmetic operator in $\{=, \neq, <, \leq, >, \geq\}$.
2. If F_1 and F_2 are formulas, then $F_1 \, and \, F_2$, $F_1 \, or \, F_2$, $not \, F_1$, and $not \, F_2$ are formulas.
3. Nothing else is a formula.

The *selection* of a relation R under a formula F is a subset of R consisting of all the tuples of R that satisfy F. It is written as:

$$\sigma_F(R) = \{\mu \mid \mu \in R \, and \, \mu \, satisfies \, F\}$$

If F is the *null formula*, then $\sigma_F(R) = R$.

3.4 Formal Semantics

Semantic Algebras

Primitive domains to be used are stated here without further explanation. These domains are as stated in Schmidt [8].

Integer numbers. $i \in \mathcal{Z} = \text{Int}$

Floating Point numbers. $v \in \Re = \text{Real}$

Character Strings. $s \in \mathcal{C} = \text{String}$

Boolean Values. $s \in Tr = \text{Tr}$

Identifiers. $id \in Id = \text{Identifier}$

Lists. $L \in D^*$. As specified in Example 3.9, pp 43–44 of Schmidt [8].

DenotableValue. $dv \in DenotableValue = StorableValue + Identifier + Error$

StorableValue. $sv \in StorableValue = NULL + Int + Real + String + Tr + Error$

Store. $st \in Store = Identifier \rightarrow Relation$

This is similar to that specified in Figure 7.1, pg. 140 of Schmidt except the only data type that is storable is a relation.

$$newstore : Store$$
$$newstore = \lambda i.error$$

$$access : Identifier \rightarrow Store \rightarrow Store$$
$$access = \lambda i.\lambda r.\lambda s.s(i)$$

$$update : Identifier \rightarrow Relation \rightarrow Store \rightarrow Store$$
$$update = \lambda i.\lambda r.\lambda s.(\lambda i'.i \text{ } equals \text{ } i' \rightarrow r \, [] \, s(i'))$$

Answer. $a \in Answer = (Relation * State) + Error$

State. $state \in State = Store$

Query Continuation. $qc \in QCont = Relation * State \rightarrow Answer$

Expression List Continuation.
$elc \in ELCont = ExpList * Relation * State \rightarrow Answer$

Expression List. $el \in ExpList = Expression + (Expression * ExpList)$

Type. $t \in Type = Int + Real + String + Tr + Null + TypeError$

Abstract Data Types

Scheme. $s \in Scheme = (Identifier{*}Type{*}Scheme)+EmptyScheme+SchemeError$

$createScheme : Scheme$
$\quad createScheme = EmptyScheme$

$add\_toScheme : Scheme \rightarrow Identifier \rightarrow Type \rightarrow Scheme$
$\quad add\_toScheme = \lambda s.\lambda i.\lambda t.(i, t, s)$

$id\_type\_inScheme : Scheme \rightarrow Identifier \rightarrow Type$

$$\begin{aligned} id\_type\_inScheme = &\\ &\lambda(i,t,s).\lambda i'.(i \; equals \; i') \rightarrow t\\ &\quad [\!] \, (s \; equals \; EmptyScheme) \rightarrow TypeError\\ &\quad [\!] \, id\_type\_inScheme \; s \; i' \end{aligned}$$

$concatScheme : Scheme \rightarrow Scheme \rightarrow Scheme$

$$\begin{aligned} concatScheme = &\\ &\lambda(i,t,s).\lambda(i',t',s').((eqScheme \; s \; EmptyScheme) \; and\\ &\qquad (eqScheme \; s' \; EmptyScheme))\\ &\qquad\quad \rightarrow (i, t, (i', t', EmptyScheme))\\ &\quad [\!] \, (eqScheme \; s \; EmptyScheme) \rightarrow (i, t, (i', t', s'))\\ &\quad [\!] \, (i, t, (concatScheme \; s \; (i', t', s'))) \end{aligned}$$

$eqScheme : Scheme \rightarrow Scheme \rightarrow Tr$

$$\begin{aligned} eqScheme = \lambda(i,t,s).\lambda(i',t',s').&\\ &((i \; equals \; i') \; and \; (t \; equals \; t') \; and\\ &\; (s \; equals \; EmptyScheme) \; and\\ &\; (s' \; equals \; EmptyScheme)) \rightarrow true\\ &[\!] \, ((i \; equals \; i') \; and\\ &\quad (t \; equals \; t') \; and\\ &\quad (eqScheme \; s \; s')) \rightarrow true\\ &[\!] \, false \end{aligned}$$

$unionCompatible : Scheme \rightarrow Scheme \rightarrow Tr$

$$\begin{aligned} unionCompatible = \lambda(i,t,s).\lambda(i',t',s').&\\ &((t \; equals \; t') \; and \; (eqScheme \; s \; EmptyScheme) \; and\\ &\; (eqScheme \; s' \; EmptyScheme)) \rightarrow true\\ &[\!] \, ((t \; equals \; t') \; and\\ &\quad (unionCompatible \; s \; s')) \rightarrow true\\ &[\!] \, false \end{aligned}$$

$cardinality : Scheme \rightarrow Int$

$$cardinality = \lambda(i,t,s).s\ eqScheme\ EmptyScheme \rightarrow 0\ []\ 1 + cardinality(s)$$

Tuple. $t \in Tuple = (Identifier{*}StorableValue{*}Tuple) + EmptyTuple + TupleError$

$createTuple : Tuple$
$$createTuple = EmptyTuple$$

$add\_toTuple : Tuple \rightarrow Identifier \rightarrow Tuple$
$$add\_toTuple = \lambda t.\lambda i.(i, inStorableValue(NULL), t)$$

$updateTuple : Identifier \rightarrow StorableValue \rightarrow Tuple \rightarrow Tuple$
$$\begin{aligned} &updateTuple = \\ &\lambda i'.\lambda sv'.\lambda(i,sv,t).(i\ equals\ i') \rightarrow (i, sv', t) \\ &\qquad []\ (eqTuple\ t\ EmptyTuple) \rightarrow tupleError \\ &\qquad []\ (updateTuple\ i'\ sv'\ t) \end{aligned}$$

$concatTuple : Tuple \rightarrow Tuple \rightarrow Tuple$
$$\begin{aligned} concatTuple = \lambda(i,sv,t).\lambda(i',sv',t').&((eqTuple\ t\ EmptyTuple)\ and \\ &(eqTuple\ t'\ EmptyTuple)) \\ &\rightarrow (i, sv, (i', sv', EmptyTuple)) \\ &[]\ (eqTuple\ t\ EmptyTuple) \rightarrow (i, sv, (i', sv', t')) \\ &[]\ (i, sv, (concatTuple\ t\ (i', sv', t'))) \end{aligned}$$

$eqTuple : Tuple \rightarrow Tuple \rightarrow Tr$
$$\begin{aligned} eqTuple = \lambda(i,sv,t).\lambda(i',sv',t').&((i\ equals\ i')\ and\ (sv\ equals\ sv')\ and \\ &(t\ equals\ EmptyTuple)\ and \\ &(t'\ equals\ EmptyTuple)) \rightarrow true \\ &[]\ ((i\ equals\ i')\ and \\ &(sv\ equals\ sv')\ and \\ &(eqTuple\ t\ t')) \rightarrow true \\ &[]\ false \end{aligned}$$

$arity : Tuple \rightarrow Int$
$$arity = \lambda(i,sv,t).(t\ eqTuple\ EmptyTuple) \rightarrow 0\ []\ 1 + arity(t)$$

Relation. $r \in Relation = (Scheme \times Tuple\ list) + RelationError$

$createRelation : Relation$
$$createRelation = (createScheme(), [\,])$$

$getRelationScheme : Relation \rightarrow Scheme$
$$getRelationScheme = \lambda(s, tupleList).s$$

$getRelationTuples : Relation \rightarrow Tuplelist$

$$getRelationTuples = \lambda(s, tupleList).tupleList$$

$arity : Relation \rightarrow Int$

$$arity = \lambda r.(cardinality(getRelationSchemer))$$

$cardinality : Relation \rightarrow Int$

$$cardinality = \lambda r.(length(getRelationTuplesr))$$

$updateRelationScheme : Relation \rightarrow Scheme \rightarrow Relation$

$$updateRelationScheme = \lambda(s, tL).\lambda s'.(s', tL)$$

$addTupleToRelation : Tuple \rightarrow Relation \rightarrow Relation$

$$addTupleToRelation = \lambda t.\lambda(s, tL).(s, (t\ cons\ tL))$$

$memberOfRelation : Tuple \rightarrow Relation \rightarrow Tr$

$$\begin{aligned} memberOfRelation = \lambda t.\lambda(s, tL).null(tL) &\rightarrow false \\ &[]\,(eqTuple\ t\ hd(tL)) \rightarrow true \\ &[]\,(memberOfRelation\ t\ tl(tL)) \end{aligned}$$

$intersection : Relation \rightarrow Relation \rightarrow Relation$

$$intersection = \lambda r.\lambda r'.(intersection\_aux\ (createRelation())\ r\ r')$$

$intersection\_aux : Relation \rightarrow Relation \rightarrow Relation \rightarrow Relation$

$$\begin{aligned} &intersection\_aux = \\ &\quad \lambda rslt.\lambda(s, tL).\lambda(s', tL').(tL'equals[\,]) \rightarrow rslt \\ &\qquad []\,(tLequals[\,]) \rightarrow rslt \\ &\qquad []\,(memberOfRelationhd(tL)r') \\ &\qquad\quad \rightarrow (intersection\_aux \\ &\qquad\qquad (addTupleToRelationhd(tL)rslt) \\ &\qquad\qquad (s, tl(tL)) \\ &\qquad\qquad (s', tL')) \\ &\qquad []\,(intersection\_auxrslt(s, tl(tL))(s', tL')) \end{aligned}$$

$selection : Relation \rightarrow Expression \rightarrow Relation$

$$\begin{aligned} &selection = \\ &\quad \lambda r.\lambda e.(selection\_aux \\ &\qquad (updateRelationScheme\ (createRelation())\ (getRelationScheme\ r)) \\ &\qquad r \\ &\qquad e) \end{aligned}$$

```
selection_aux : Relation → Relation → Expression → Relation
      selection_aux =
          λrslt.λ(s, tList).λe.(tL'equals[ ]) → rslt
                              [] (let r = E[e] hd(tList)
                                 in (cases r of
                                       isBoolean(tv) →
                                          (tv →(selection_aux
                                                  (addTupleToRelation
                                                      hd(tList) rslt)
                                                  (s, tl(tList))
                                                  e)
                                          [] (selection_aux rslt (s, tl(tList)) e))
                                       [] RelationError)
                                 end)
```

The remaining operations on relations are similar or composed of the operations shown above. The union and complex product operations are very similar to intersection and projection is very similar to selection.

3.5 Valuation Functions

```
Q:Query → QCont → Answer
  Q[Θ qop Ψ] qc =
    S[Θ] λ (r,s).(cases qop of
                  "union" →
                      (cases r of
                          (_,_) → (let qc' = Q[Ψ] λ (r',s').
                                              (cases r' of
                                                  (_,_) → qc ((union r r'),s')
                                                  [] RelationError → Error)
                                   in qc' ((createRelation()),s) end)
                          [] RelationError → Error)
                  [] "intersection" →
                      (cases r of
                          (_,_) → (let qc' = Q[Ψ] λ (r',s').
                                              (cases r' of
                                                  (_,_) → qc ((intersection r r'),s')
                                                  [] RelationError → Error)
                                   in qc' ((createRelation()),s) end)
                          [] RelationError → Error)
                  [] RelationError → Error)
  Q[Θ] qc = S[Θ] qc

S:Statement → QCont → QCont
```

```
  S[* Γ] qc = λ(r',s').(let qc' = TE[Γ] qc in qc' (r',s') end)

  S[ε,Σ Γ]qc =
  TE[Γ] Sel[ε,Σ] λ(eL,r,s). qc ((projection r eL),s)

Sel:Selection → ELCont → QCont
    Sel[ε] elc = λ (r,s).elc (ε,r,s)

    Sel[ε,Σ] elc =
        Sel[Σ] λ (eL,r,s).elc ((ε,eL),r,s)

TE:TableExp → QCont → QCont
    TE[from Υ] qc = TL[Υ] qc

    TE[from Υ where ε] qc =
        TL[Υ] λ(r,s).(cases r of
                             (_,_) → qc ((selection r ε),s)
                             [] Error

TL:TableList → QCont → QCont
    TL[ξ] qc = T[ξ] λ(r,s).qc (r,s)
    TL[ξ,Γ] qc =
    T[ξ] λ(r,s).(let qc' =TL[Γ] qc
                 In qc' (r,s) end)

T:Table → QCont → QCont
  T[ξ] qc =
   λ (r,store).(cases r of
                   (_,_) →
                      (let r' = (access ξ store)
                       in (cases r' of
                                 (_,_) → qc ((cartesianProduct r r'),store)
                                 [] RelationError → Error)
                       end)
                   RelationError → Error)

  T[ξ.ξ'] qc =
   λ (r,store).(cases r of
                   (_,_) →
                      (let r' = (access ξ store)
                       in (cases r' of
                                 (_,_) → qc ((cartesianProduct r r'),store)
                                 [] RelationError → Error)
```

```
                                end)
                            RelationError → Error)
```

Expressions are evaluated only with respect to a particular tuple of a relation. The tuple (and its Scheme) provide all necessary information in order to evaluate an expression. The expression valuation function evaluates the expression to its normal form.

```
E:Expression → Tuple → Scheme → Expression
  E[ε1 + ε2] t s =
    (let ε1' = E[ε1] t s
     in (let val ε2' = E[ε1] t s
         in (cases ε1' of
               isColumn(col) →
                   (cases ε2' of
                      isColumn(col') →
                          ((id_type_inScheme s C[col])
                           =
                           (id_type_inScheme s C[col']))
                           →
                              (let rslt = ((accessTuple t C[col]) +
                                           (accessTuple t C[col']))
                               in (cases rslt of
                                      isInt(i) → inExpression(i)
                                      [] isReal(v) → inExpression(r)
                                      [] ExpError)
                               end)
                           [] ExpError
                      [] isInt(i) →
                           ((id_type_inScheme s C[col])
                            = INT)
                           →
                               inExpression((accessTuple t C[col]) + i)
                      [] isReal(v) →
                           ((id_type_inScheme s C[col])
                            = REAL)
                           →
                                inExpression((accessTuple t C[col]) + v))
               [] isInt(i) →
                   (cases ε2' of
                      isColumn(col') →
                          (INT
                           =
                          (id_type_inScheme s C[col']))
```

```
                    →
                       (let rslt = (i + (accessTuple t C[col']))
                        in (cases rslt of
                              isInt(i') → inExpression(i')
                              [] ExpError)
                        end)
                    [] ExpError
               [] isInt(i') →
                    inExpression(i + i')
               [] ExpError)
          [] isReal(v) →
               (cases ε2' of
                  isColumn(col') →
                    (REAL
                     =
                     (id_type_inScheme s C[col']))
                     →
                       (let rslt = (v + (accessTuple t C[col']))
                        in (cases rslt of
                              isReal(v') → inExpression(v')
                              [] ExpError)
                        end)
                     [] ExpError
                  [] isReal(v') →
                       inExpression(v + v')
                  [] ExpError)
          [] ExpError)
     end)
  end)
```

Other operations (boolean, relational, arithmetic) are completed in a similar fashion. The base cases in expression evaluation are as follows:

```
E [S] t s = inExpression(S)
E [I] t s = inExpression(I)
E [R] t s = inExpression(R)
E [B] t s = inExpression(B)
E [NULL] t s = inExpression(NULL)
E[C] t s =
  (let rslt = (accessTuple t C[C])
   in (cases rslt of
         isInt(i) → inExpression(i)
         [] isReal(v) → inExpression(v)
```

```
        [] isString(s) → inExpression(s)
        [] isBoolean(tv) → inExpression(tv)
        [] isNull() → inExpression(NULL)
        [] ExpError)
end)
```

4 Equivalence of Queries

A given query can be expressed in more than one way. In particular, the **union** and **intersection** of two queries on the same relations has an equivalent single query form. Furthermore, the single query form frequently involves less typing.

Suppose $\Psi_{1,F_1}(R_1, R_2, \ldots, R_n)$ and $\Psi_{2,F_2}(R_1, R_2, \ldots, R_n)$ are two queries over relations $R_1, R_2, \ldots, R_n$. Also, suppose these queries involve a selection operation using F_1 and F_2, respectively.

The **union** and **intersection** operations are only defined for relations that are union-compatible. Thus, $\Psi_{1,F_1}(R_1, R_2, \ldots, R_n)$ and $\Psi_{2,F_2}(R_1, R_2, \ldots, R_n)$ must be union-compatible.

Union. The union of these two queries is the relation containing the tuples that satisfy F_1 or F_2 (from the definition of union of two relations). The definition of a formula for selection says that if F_1 and F_2 are formulas, then F_1 and F_2 joined by the boolean **or** operation is also a formula. Let $F_3 = F_1$ **or** F_2. The syntax for the **select** statement allows a query of the form $\Psi_{3,F_3}(R_1, R_2, \ldots, R_n)$.

Intersection. Showing that there exists a single-query form of the intersection of two union-compatible queries over the same relations is analogous to showing the same property for union.

5 Conclusions

The semantics of a portion of the SQL select statement has been described. The description demonstrates the combination of standard semantics style which was used to describe how an SQL query is transformed into a series of operations on relations, and abstract data types used to describe the relational database model. The resulting semantics are operational in style because the semantics of a query are described in terms of well-defined operations on the abstract data types. This makes correctness and other types of proofs easier because the correctness of the abstract data types can be dealt with separately.

In the process of defining the semantics of the SQL select statement, several observations were made:

- Initially, the model developed for how a query "works" required the use of an environment (in a reference like **host H**, **H** would be a local variable) and store (to store relations). Further investigation into what a query actually does revealed a much simpler series of operations than was first anticipated. The resulting semantics do not require an environment. Expanding the syntax in any way to handle more of the SQL syntax would require the use of an environment, however. This could easily be done by making a $State = Environment * Store$ rather than $State = Store$ as it is now.

- Portions of the chosen syntax are unnecessary. In particular, the **union** and **intersection** of two queries have equivalent single query forms involving the **or** and **and** boolean operations.

References

[1] AMERICAN NATIONAL STANDARDS INSTITUTE. *X3.135-1989: Database Language — SQL with Integrity Enhancement.* New York, NY, 1989.

[2] BLASGEN, M. W., ET AL. System R: An architectural overview. *IBM Systems Journal 20*, 1 (1981), 41–62.

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

[4] DATE, C. J. *An Introduction to Database Systems*, 2nd ed. Addison-Wesley Publishing Company, Reading, MA, 1977.

[5] DELOBEL, C., AND ADIBA, M. *Relational Database Systems.* Elsevier Science Publishers, Amsterdam, The Netherlands, 1985.

[6] KORTH, H. F., AND SILBERSCHATZ, A. *An Introduction to Database Systems.* McGraw-Hill Book Company, NewYork, NY, 1986.

[7] SANDBERG, G. A primer on relational database concepts. *IBM Systems Journal 20*, 1 (1981), 23–40.

[8] SCHMIDT, D. A. *Denotational Semantics: A Methodology for Language Development.* Wm. C. Brown Publishers, Dubuque, Iowa, 1988.

[9] YANG, C. *Relational Databases.* Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

TECH REPORTS

UNIVERSITY OF UTAH - COMPUTER SCIENCE DEPARTMENT
SALT LAKE CITY, UTAH 84112

Febraury 13, 1992

* = Thesis (Masters)
** = Dissertation (PhD)
NA = Not Available

UUCS-92-001	Akella, Venkatesh and Ganesh Gopalakrishnan. "Specification and Validation of Control-Intensive Integrated Circuits in hopCP. Xeroxing Cost: $1.85 (37 pgs)
UUCS-92-002	Akella, Venkatesh and Ganesh Gopaladrishnan. "CFSIM: A Concurrent Compiled Code Simulator for hopCP." Xeroxing Cost: "$1.00 (20 pgs)
UUCS-92-003 **	Cohen, Michael F. "Spacetime Control of Linked Figures." Xeroxing Cost: $5.85 (117 pgs)
UUCS-92-004	Gopalakrishnan, Ganesh, Brunavd, Erik and Nick Michell. "A Correctness Criterion For Asynchronous Circuit Validation And Optimization." Xeroxing Cost: $.80 (16 pgs)
UUCS-92-005	Liebchen, Armin and Ganesh Gopalakrishnan. "Synamic Reordering of High Latency Transactions in Time-Warp Simulation Using A Modified Micropipeline." Xeroxing Cost: $.85 (17 pgs)

UUCS-91-001	Gershon, Elber and Elaine Cohen. "Error Bounded Variable Distance Offset Operator for Free From Curves and Surfaces." Xeroxing Cost: $.80 (16 pgs)
UUCS-91-002	Starkey, Mike. "A Lisp-based OCCAM Interpreter." Xeroxing Cost: $.65 (13 pgs)
UUCS-91-003	Starkey, Mike and T.M. Carter. " Transforming Disfigured And Disoriented Areas Into Routable Switchboxes." Xeroxing Cost: $.35 (7 pgs)
UUCS-91-004	Starkey, Mike and T.M. Carter. "Switchbox Routing By Pattern Matching." Xeroxing Cost: $.60 (12 pgs)
UUCS-91-005 *	Ming, John Charles. "A Multistrategy Machine Learning Approach To Object Recognition." Xeroxing Cost: $13.05 (261 pgs)
UUCS-91-006 *	Sohrt, Wolfgang. "Interaction With Constraints In Three-Dimensional Modeling." Xeroxing Cost: $5.80 (116 pgs)
UUCS-91-007 *	Susswein, Steven Y. "Parallel Path Consistency." Xeroxing Cost: $3.45 (69 pgs)
UUCS-91-008	Soller, Jerome B. and Neil E. Cotter. "Generality Vs. Speed Of Convergence In The Cart-Pole Balancer." Xeroxing Cost: $.80 (16 pgs)
UUCS-91-009	Soller, Jerome B. and Neil E. Cotter. "Prejudicial Search and Backprop." Xeroxing Cost: $.20 (4 pgs)

UUCS-91-010 Susswein, Steven Y., Henderson, Thomas C., Zachary, Joseph L., Hansen, Chuck, Hinker, Paul and Gary C. Marsden. "Parallel Path Consistency." Xeroxing Cost: $1.00 (20 pgs)

UUCS-91-011 * Starkey, Michael. "An Automated Wire Routing Kernel." Xeroxing Cost: $4.10 (72 pgs.)

UUCS-91-012 Akella, Venkatesh and Ganesh Gopalakrishnan. "Hierarchical Action Refinement: A Methodology For Compiling Asynchronous Circuits from a Concurrent HDL." Xeroxing Cost: $1.05 (21 pgs)

UUCS-91-013 Brunvand, Erik. "A Cell Set for Self-Timed Design using Actel FPGAs." Xeroxing Cost: $1.70 (34 pgs)

UUCS-91-014 Smith, Jeffrey Kent. "SEQWARP: A Low-Cost Linear Systolic Array for Biological Sequence Comparison." Xeroxing Cost: $3.90 (78 pgs)

UUCS-91-015 Gopalakrishnan, Ganesh and Richard Fujimoto. "Design and Verification of the Roolback Ship using HOP: A Case Study of Formal Methods Applied to Hardware Design." Xeroxing Cost: $2.30 (46 pgs)

UUCS-91-016 Akella, Venkatesh and Ganesh Gopalakrishnan. "From Process-Oriented Functional Specifications to Efficient Asynchronous Circuits." Xeroxing Cost: $.40 (8 pgs)

UUCS-91-017 Bracha, Gilad and Gary Lindstrom. "Modularity Meets Inheritance." Xeroxing Cost: $.90 (18 pgs)

UUCS-91-018 Akella, Venkatesh and Gahesh Gopalakrishan. "Static Analysis Techniques for the Synthesis of Efficient Asynchronous Circuits." Xeroxing Cost: $1.20 (24 pgs)

UUCS-91-019 ** Lee, Hyo Jong. "Numerical Weather Prediction Model for Distributed Computing Systems. Xeroxing Cost: $4.05 (81 pgs)

UUCS-91-020 George, Lal and Gary Lindstrom. "Using a Functional Language And Graph Reduction To Program Multiprocessor Machines. Xeroxing Cost: $

UUCS-91-021 Akella, Venkatesh and Ganesh Gopalakrishnan. "hopCP: A Concurrent Hardware Description Language." Xeroxing Cost: $1.70 (34 pgs)

UUCS-91-022 Jain, Prabhat, Ganesh Gkopalakrishnan and Prabhakar Kudva. "Verification of Regular Arrays by Symobolic Simulation." Xeroxing Cost: $.80 (16 pgs)

UUCS-91-023 Jain, Prabhat and Ganesh Gopalakrishnan. "Efficient Symbolic Simulation Based Verification Using The Parametric Form of Boolean Expressions." Xeroxing Cost: $1.25 (25 pgs)

UUCS-91-025 Hoogenboom, Peter J.. "System Performance Advisor User Guide." Xeroxing Cost: $2.20 (44 pgs)

UUCS-91-026 Hoogenboom, Peter. J.. "Semantic Definition of a Subset of the Structured Query Language (SQL)." Xeroxing Cost: $1.05 (21 pgs)

UUCS-91-027 Short, Wolfgang. "Interaction With Constraints in Threee-Dimensional Modeling." Xeroxing Cost: $5.70 (11 pgs)

UUCS-91-028 ** — Mecklenburg, Robert William. "Towards A Language Independent Object System." Xeroxing Cost: $9.00 (180 pgs)

UUCS-91-029 ** — Swanson, Mark R.. "Concurrent Scheme, A Language for Concurrent Symbolic Computing." Xeroxing Cost: $5.70 (114 pgs)

UUCS-91-030 ** — Lee, Hyo Jong. "Numerical Weather Predication Model." Xeroxing Cost: $4.00 (80 pgs)

UUCS-91-031 * — Susswein, Steven Y.. "Parallel Path Consistency." Xeroxing Cost: $3.45 (69 pgs)

UUCS-91-032 * — Paik, Karen Lynn. "Trivariate B-Splines." Xeroxing Cost: $3.00 (60 pgs)

UUCS-90-002 — George, Lal. "A Scheduling Strategy for Shared Memory Multi-processors." Xeroxing Costs: $.60 (12 pgs)

UUCS-90-003 — Bruderlin, Beat. "Detecting Ambiguities: An Optimistic Approach to Robustness Problems in Computational Geometry." Xeroxing Costs: $1.20 (24 pgs)

UUCS-90-004 — Bruderlin, Beat. "Robust Regularized Set Operations on Polyhedra." Xeroxing Cost: $1.45 (29 pgs)

UUCS-90-005 — Bruderlin, Beat. "DI-A Portable, Object Oriented User Interface Toolkit for Modlula-2 Applications." Xeroxing Cost: $2.10 (42 pgs)

UUCS-90-010 — Akella, Venkatesh. "hopCP: Language Definition, Semantics and Examples." Xeroxing Cost: $1.25 (25 pgs)

UUCS-90-011 ** — Elens, Bob. "Sequencing Computational Events in Heterogeneous Distributed Systems." Xeroxing Cost: $4.25 (85 pgs)

UUCS-90-012 — Starkey, Mike and Ali Farhang. "C-RISC: A C Language Reduced Instruction Set Computer." Xeroxing Cost: $.40 (8 pgs)

UUCS-90-013 — Henderson, Thomas C., John Evans, Lane Grayston, Allen Sanderson, Leigh Stoller, and Eliot Weitz. " CSCV: A CAD-Based Vision System." Xeroxing Cost: $2.25 (45 pgs)

UUCS-90-014 — Kessler, Robert R. and Mark R. Swanson. "Concurrent Schemes." Xeroxing Cost: $1.80 (36 pgs)

UUCS-90-015 — Ginosar, Ran, and Nick Michell. "On the Potential of Asynchronous Pipelined Processors." Xeroxing Cost: $.45 (9 pgs)

UUCS-90-016 — Gopalakrishnan, Ganesh, and Prabhat Jain. "Some Recent Asynchronous System Design Methodologies." Xeroxing Cost: $2.65 (53 pgs)

UUCS-90-017 — Evans, John. "A Distributed Object Oriented Graphical Programming System." Xeroxing Cost: $ 5.45 (109 pgs)

UUCS-90-018 — Henderson, Thomas and Patrick Dalton. "Z-infinity: A Framework for Autonomous Agent Behavior Specification and Analysis." Xeroxing Cost: $1.15 (23 pgs)

UUCS-90-019 — Kessler, Robert R. and John D. Evans. "DPOS: A Metalanguage and Programming Environment for Parallel Processors." Xeroxing Cost: $.55 (11 pgs)

UUCS-90-020 Evans, John D. "DPOS Programming Manual." Xeroxing Cost: $1.05 (21 pgs)

UUCS-90-021 Gopalakrishnan, Jain, Akella, Josephson, Kuo. "Combining Verification and Simulation." Xeroxing Cost: $.85 (17 pgs)

UUCS-90-022 * Thingvold,Jeffrey Allyn. "Elastic And Plastic Surfaces For Modeling And Animation." Xeroxing Cost: $5.80 (116 pgs)

UUCS-90-023 Bruderlin, Beat. "An Axiomatic Approach for Solving Geometric Problems Symbolically." Xeroxing Cost: $1.95 (29 pgs)

UUCS-90-024 Bruderlin, Beat. "Interaction With Constraints in 3D Modeling." Xeroxing Cost: $1.00 (20 pgs)

UUCS-90-025 ** Yeh, Dowming. "Static Analysis of Multiparadigm Languages." Xeroxing Cost: $7.90 (158 pgs)

UUCS-90-26 * Livingston, John Barry. "Intersurface Continuity of B-Spline Based Solid Models." Xeroxing Cost: $4.90 (98 pgs)

UUCS-90-027 * Dooley, Debra Lynn. "Computer Illustration of Three Dimensional Sculptured Surfaces." Xeroxing Cost: $4.20 (84 pgs)

UUCS-90-028 * Smith, Glenn Gordon. "Modeling The Motion of Human Systems With Computer Graphics." Xeroxing Cost: $4.65 (93 pgs)

UUCS-89-001 Lee, Hyo Jong. "Automatic Mesh Analysis Technique by Knowledge-Based System." Xeroxing Cost: $2.15 (43 pgs)

UUCS-89-002 Lee, Hyo Jong. "Implementation and Characteristics of Rule-based System for the Finite Element Analysis." Xeroxing Cost: $1.40 (28 pgs)

UUCS-89-003 George, Lal. "An Abstract Machine for Parallel Graph Reduction." Xeroxing Cost: $1.00 (20 pgs)

UUCS-89-004 Carter, Tony. "Cell Matrix Methodologies for Integrated Circuit Design." Xeroxing Cost: $.55 (11 pgs)

UUCS-89-005 Carter, Tony. "Cascade: Hardware for High/Variable Precision Arithmetic." Xeroxing Cost: $.60 (12 pgss)

UUCS-89-006 Carter, Tony. "Cascade: A Hardware Alternative to Bignums." Xeroxing Cost: $1.35 (27 pgs)

UUCS-89-008 Gu, Jun and Smith, Kent. "Structured, Technology Independent VLSI Design." Xeroxing Cost: $2.35 (47 pgs)

UUCS-89-009 Seeley, Donn. "A Tour of the Worm." Xeroxing Cost: $.75 (15 pgs)

UUCS-89-010 Chou, Jin J. and Cohen, Elaine. "Automatic Sculptured Five-Axis Milling with Check Surfaces."

UUCS-89-011 Chou, Jin J. and Cohen, Elaine. "Constant Scallop Height Tool Path Generation."

UUCS-89-012 Henderson, Tom. "Logical Behaviors." Xeroxing Cost: $1.20 (24 pgs)

UUCS-89-013	Carter, Tony. "The Set Theory of Arithmetic Decomposition."
UUCS-89-014	Smith, Kevin. "RIPPLE User's Manual and Reference. Xeroxing Cost: $5.30 (106 pgs)
UUCS-89-016	Carter, Tony and Smith., Kent. "Path-Programmable Logic"
UUCS-89-017	Cohen, Elaine and Chou, Jin. "Computing Offsets and Tool Paths with Voronoi Diagrams."
UUCS-89-019 NA	Cohen, Elaine and Chou, Jin. "Hidden Curve Removal for Untrimmed and Trimmed NURB Surfaces."
UUCS-89-020 **	Loosemore, Sandra. "A Visual Lisp Debugging Environment."
UUCS-89-021 NA	Elaine Cohen and Thingvold, Jeff. "Elastic and Plastic Surfaces for Modeling and Animation."
UUCS-89-022	Mantha, Surya, George, Lal, and Lindstrom, Gary. "Abstract Semantics for Functional Constraint Programming."
UUCS-89-023 *	Banks, Michael Joseph. "A User Interface Model and Tools for Geometric Design." Xeroxing Cost: $4.50 (90 pgs)
UUCS-89-024 **	Loosemore, Sandra Jean. " A Visual Lisp Debugging Environment." Xeroxing Cost: $6.35 (127 pgs)
UUCS-89-025 **	Chou, Jin Joacob. "Numerical Control Milling Machine Toolpath Generation For Regions Bounded By Free From Curves and Surgaces." Xeroxing Cost: $7.75 (155 pgs)
UUCS-89-026 *	"Ma, Kwan-Liu. "TICL: A Type Inference System for Common Lisp." Xeroxing Cost: $4.10 (82 pgs)

UUCS-88-001	Grupen, Roderic A., Henderson, Tom, Biggers, Klaus, and Meek, Sandy. "Task Defined Grasp Force Solutions." Xeroxing Cost: $.40 (8 pages)
UUCS-88-002	Brandt, Richard C., "Building Databases for the Computer-Based Memorization System." Xeroxing Cost: $1.35 (27 pages)
UUCS-88-003	Gopalakrishnan, Ganesh C., Fujimoto, Richard M. HOP: A Process Model for Synchronous Hardware Systems. Xeroxing Cost: $2.65/ 53 pgs.
UUCS-88-004	Carter, Tony M. and Robertson, James E. "Radix-16 Signed-Digit Division." Xeroxing Cost: $.85 (18 pages).
UUCS-88-005 **	Gu, Jun. "Parallel Algorithems and Architectures for very fast AI Search."
UUCS-88-006	Gu, Jun. " An Optimal, Parallel Discrete Relaxation Algorithm and Architecture."
UUCS-88-007	Sikorski, Kris. "Asymptotic Near Optimality of the Bisection Method." Xeroxing Cost: $1.05 (21 pgs).
UUCS-88-008	Bloomentahl, Mark "Approximation of Sweep Surfaces by Tensor Product B-splines."(cost 72 pages, $3.60)
UUCS-88-009	Cobb, James E., "Tiling the Sphere with Rational

Bezer Patches." Xeroxing Cost: $.70 (14 pages).

UUCS-88-010 Henderson, Samal. "An Annoted Bibliography of Multi-Sensor Integration."

UUCS-88-011 Fujimoto, Tsai, Gopalakrishnan. "Design and Evaluation of the Rollback Chip: Special Purpose Hardware for Time Warp." Xeroxing Cost: $1.90/38 pgs.

UUCS-88-012 Gopalakrishnan, Fujimoto, Akella, Mani. "HOP: A Process Model for synchronous Hardware Semantics, and Experiments in Process Composition. Xeroxing cost: $2.35/47 pages.

UUCS-88-013 ** Feng, Hwa-Chung. "A Multiprocessor Implementation of CSP." Xeroxing cost: $5.40/108 pages.

UUCS-88-014 Thompson, Richard. "Design Specifications for Hardware Assisted Rollback Computation." $.75/15 pgs.

UUCS-88-016 Akella, Venkatesh: YAMA: "Yet Another MicroAssembler: Description & User's Guide." Xeroxing Cost $1.00 / 20 pgs.

UUCS-88-018 Ganesh, Fujimoto, Akella, Mani, Kevin Smith. "Specification-Driven Design of Custom Hardware in HOP" $2.20/44 pgs.

UUCS-88-019 Fujimoto, Richard. "The Virtual Time Machine."

UUCS-88-020 ** Shebs, Stanley T. "Implementing Primitive Datatypes from Higher-Level Languages." Xeroxing cost: $8.30/166 pgs.

UUCS-88-021 Fujimoto, Richard M. "Time Warp on a Shared Memory Multiprocessor" Xeroxing Cost: $1.30 / 26 Pgs.

UUCS-88-023 Gatrell, Lance B. "CAD-Based Grasp Synthesis" Xeroxing Cost: $2.55 / 51 pgs.

UUCS-88-024 Gu, Jun and Smith, Kent F. "Fast Structured Design of VLSI Circuits." Xeroxing Cost: $2.45 (49 pgs)

UUCS-88-025 Sosic, Rok and Gu, Jun. "How to search for Millions of Queens". Xeroxing Cost: $.80 (16 pgs)

UUCS-88-026 * Malley, Thomas J.V.. "A Shading Method For Computer Generated Images." Xeroxing Cost: $3.50 (70 pgs)

UUCS-88-027 * Ho, Chih-Cheng. "Computer-Aided Geometric Design Based Three-Dimensional Object Representations For Computer Vision." Xeroxing Cost: $3.95 (79 pgs)

UUCS-88-028 * Peterson, John William. "PRT - A High Quality Image Synthesis System for B-Spline Surfaces." Xeroxing Cost: $4.90 (98 pgs)

UUCS-88-029 ** Samal, Ashok Kumar. "Parallel Split-Level Relaxation." Xeroxing Cost: $6.80 (136 pgs)

UUCS-88-030 * Pourheidri, Mohammad. "Moped (A Portable Debugger)." Xeroxing Cost: $3.05 (61 pgs)

UUCS-88-031 * Elvins, Thomas Todd. "Modeling Systems of Rigid Bodies in Motion Using Dynamics."

UUCS-88-032 * Feng, Hwa-chung. "A Multiprocessor Implementation of CSP." Xeroxing Cost: $5.40 (108 pgs)

UUCS-87-002 Jacobs, Steven R. and Smith, Kent F., "PPL Quick Reference Guide (CMOS)." Xeroxing Cost: $2.70 (54 pages)

UUCS-87-004 Thomas, Spencer W., "Set Operation on Sculptured Solids" Xeroxing Cost: $1.55 (31 pages)

UUCS-87-006 Henderson, Thomas C., Allen, Peter, Cox, Ingemar, Mitiche, Amar, Durrant-Whyte, Hugh, and Snyder, Wes. "Workshop on Multisensor Integration in Manufacturing Automation." Xeroxing Cost: $1.65 (33 pages)

UUCS-87-007 Thune, Nils and Bhanu, Bir, "CAOS An Approach to Robot Control." Xeroxing Cost: $2.30 (46 pages)

UUCS-87-008 Thune, Mari and Bhanu, Bir, "Implementation of a Hierarchical Control System on a BBN Butterfly Multiprocessor: Initial Studies and Results." Xeroxing Cost: $3.70 (74 pages)

UUCS-87-010 Grupen, Roderic A. and Henderson, Thomas C., "High-Level Planning for Dextrous Manipulation." Xeroxing Cost: $1.15 (13 pages)

UUCS-87-011 Grupen, Roderic A. and Henderson, Thomas C. "Apparent Symmetries in Range Data." Xeroxing Cost: $0.45 (9 pages)

UUCS-87-012 Tinker, Peter, "Managing Large Address Spaces Effectively on the Butterfly." Xeroxing Cost: $0.55 (11 pages)

UUCS-87-013 Henderson, Thomas C. and Hansen, Chuck, "An Approach to Three-Dimensional Scene Databases." Xeroxing Cost: $1.05 (21 pages)

UUCS-87-014 Peterson, John W., "Distributed Computation for Computer Animation." Xeroxing Cost: $0.65 (13 pages)

UUCS-87-015 * Yost, Jeffrey B., "Computational Fluid Dynamics for Realistic Image Synthesis." Xeroxing Cost: $5.80 (116 pages)

UUCS-87-016 Hansen, Chuck, and Henderson, Thomas C., "CAGD-Based Computer Vision" Xeroxing Cost: $1.30 (26 pages

UUCS-87-017 Samal, Ashok, and Henderson, Thomas C., "Performance of Arc Consistency Algorithms on the CRAY." Xeroxing Cost: $1.45 (29 pages)

UUCS-87-018 Lyche, Tom, "A Note on the Oslo Algorithm."

UUCS-87-019 Gopalakrishnan, Ganesh, "Implementing Functional Programs Using Mutable Abstract Types." Xeroxing Cost: $1.05 (21 pages)

UUCS-87-020 Samal, Ashok, and Henderson, Tom, "IKS: Image Kernel System User's Manual." Xeroxing Cost: $0.65 (15 pages)

UUCS-87-021 Fujimoto, Richard M. and Feng, Hwa-chung, "A Shared Memory Algorithm and Proof for the Alternative Construct in CSP." Xeroxing Cost: $1.75 (35 pages)

UUCS-87-022 Fujimoto, Richard M. and Campbell, William B., "Efficient Instruction Level Simulation of Computers." Xeroxing Cost: $1.20 (24 pages)

UUCS-87-023 Fujimoto, Richard M. and Swope, Steven, "Optimal Performance of Distributed Simulation Programs." Xeroxing Cost: $.75 (15 pages)

UUCS-87-024 Gopalakrishnan, Ganesh, "A Compositional Model for Synchronous VLSI Systems." Xeroxing Cost: $1.05 (21 pages)

UUCS-87-025 Fujimoto, Richard M. and Tsai, Jya-Jang and Gopalakrishnan, Ganesh, "The Roll Back Chip: Hardware Support for Distributed Simulation Using Time Warp." Xeroxing Cost: $1.40 (28 pages)

UCS-87-026a Fujimoti, Richard M., "Performance Measurements of Distributed Simulation Strategies." Xeroxing Cost: $1.40 (28 pages)

UUCS-87-027 Lindstrom, Gary and Bage, Goran, "Combinator Evaluation of Functional Programs with Logical Variables." Xeroxing Cost: $1.30 (26 pages)

UUCS-87-028 Henderson, T., et al., "Multisensor Knowledge Systems: Interpreting 3-D Structure." $1.90 (38 pages)

UUCS-87-029 Grupen, R.A., et al., "A Control Paradigm for General Purpose Manipulation Systems." $1.05 (21 pages)

UUCS-87-030 Henderson, Tom, Weitz, Eliot, "Knowledge-Based 2-D Vision System Synthesis." Xeroxing Cost: $3.40 (68 pages)

UUCS-87-031 * Muehle, Eric George. "FBOBS: A Merger of Two Knowledge Representation Paradigms." Xeroxing Cost: $4.90 (98 pgs)

UUCS-86-001 Swope, Steven M. and Fujimoto, Richard M., "SIMON II KERNEL Reference Manual (21 May 1987)" Xeroxing Cost: $5.65 (113 pages)

UUCS-86-002 Boult, T. and Sikorski, K., "Complexity of Computing Topological Degree of Lipschitz Functions in N Dimensions" Xeroxing Cost: $1.15 (23 pages)

UUCS-86-003 Jacobs, Steven R. and Smith, Kent F., "PPL Quick Reference Guide(NMOS)" (Revised 1-12-87) Xeroxing Cost: $3.50 (70 pages)

UUCS-86-004 Jacobs, Steven R. and Smith, Kent F., "SIMPPL User's Guide" (Revised 1-12-87) Xeroxing Cost: $1.55 (31 pages)

UUCS-86-005 Jacobs, Steven R., Smith, Kent F., Smith, Jeff, and Smith, Kevin,"PPL Design Examples (NMOS30 Version)" (12-12-85) Xeroxing Cost: $4.65 (93 pages)

UUCS-86-006 Jacobs, Steven R. and Smith, Kent F., "Tiler User's Guide" (Revised 1-12-87) Xeroxing Cost: $1.50 (30 pages)

UUCS-86-007 Grupen, Roderic A. and Henderson, Thomas C., "A Survey of Dextrous Manipulation." Xeroxing Cost: $2.30 (46 pages)

UUCS-86-008 Wang, Wei, Gu, Jun, and Henderson, Thomas C., "A Systolic Array of Implementation of a Discrete Relaxation Algorithm Using Path Programmable Logic." Xeroxing Cost: $2.80 (56 pages)

UUCS-86-009 Rajopadhye, Sanjay V., Purushothaman, S., and Fujimoto Richard "On Synthesizing Systolic Arrays from Recurrence Equations with Linear Dependencies." Xeroxing Cost: $0.80 (16 pages)

UUCS-86-010 Carter, Tony M., "Ada to Silicon Compiler Study" Xeroxing Cost: $8.15 (163 pages)

UUCS-86-011 Rajopadhye, Sanjay V. and Furimoto, Richard, "Systolic Array Synthesis by Static Analysis of Program Dependencies." Xeroxing Cost: $0.60 (12 pages)

UUCS-86-012 Grodstein, Joel, "Sisyphus User's Manual." Xeroxing Cost: $1.00 (20 pages)

UUCS-86-113 Hansen, Chuck and Henderson, Tom, "UTAH Range Database." Xeroxing Cost: $1.20 (24 pages)

UUCS-86-114 Henderson, Thomas C. and Hansen, Chuck, "Multisensor Knowledge Systems." Xeroxing Cost: $1.05 (21 pages)

UUCS-86-115 Samal, Ashok and Henderson, Tom, "Combining Symbolic and Numeric Computation of the CRAY." Xeroxing Cost: $1.05 (21 pages)

UUCS-86-116 Gu, Jun, Wang, Wei, and Henderson, Thomas C., "A Consistent Labeling Architecture." Xeroxing Cost: $1.65 (53 pgs)

UUCS-86-117 Gopalakrishnan, Ganesh C., "From Algebraic Specifications to Correct VLSI Systems." Xeroxing Cost: $13.80 (276 pages)

UUCS-86-118 * Grodstein, Joel. "Sisyphus - An Environment for Simulation." Xeroxing Cost: $4.90 (98 pgs)

UUCS-86-119 ** Mueller, Timothy Irwin. "Geometric Modelling With Nultiv Ariate B -Splines." Xeroxing Cost: $6.80 (136 pgs)

UUCS-86-120 * Knell, Jeffrey Kent. "Resource Allocation In An Experimental Lisp Compiler. Xeroxing Cost: $4.75 (95 pgs)

UUCS-85-001 Rajopadhye, Sanjay and Panangaden, Prakash, "Verification of Systolic Arrays: A Stream Functional Approach." Xeroxing Cost: $1.15 (23 pages)

UUCS-85-002 * Seppi, Larry, "Implementation of SCMAID: A Microprogrammer's Assistant for Concurrent Soft Controlled Microarchitectures." Xeroxing Cost: $7.20 (144 pages)

UUCS-85-101 Mohr, Roger, and Henderson, Thomas C., "ARC and Path Consistency Revisited". Xeroxing Cost: $0.55 (11 pages)

UUCS-85-102 Henderson, Tom, Hansen, Chuck, Grupen, Rod, and Bhanu, Bir, "The Synthesis of Visual Recognition Strategies" Xeroxing Cost: $0.45 (9 pages)

UUCS-85-103 Sikorski, K., "Optimal Solution of Nonlinear Equations" Xeroxing Cost: $0.95 (19 pages)

UUCS-85-104 Cohen, Elaine, "A New Local Basis for Designing with Tensioned Splines" Xeroxing Cost: $2.35 (47 pages)

UUCS-85-105 Keller, Robert M., "Rediflow Architecture Prospectus" (date of last revision: 5 April 1986) Xeroxing Cost: $1.50 (30 pages)

UUCS-85-110 Parvin, Bahram, and Bhanu, Bir, " Segmentation of Complex Outdoor Scenes" Xeroxing Cost: $1.25 (25 pages)

UUCS-85-111 Bhanu, Bir and Ming, John C., "Recognition of 2-D Occluded Objects and their Manipulation by PUMA 560 Robot" Xeroxing Cost: $4.55 (91 pages)

UUCS-85-112 Bhanu, Bir, Ho, C.C., and Henderson, T., "3-D Model Building for Computer Vision" Xeroxing Cost: $1.05 (21 pages)

UUCS-85-113 Henderson, Tom and Samal, Ashok, "2-D Scene Analysis Using Split-Level Relaxation" Xeroxing Cost: $1.50 (30 pages)

UUCS-85-114 Bhanu, B., Lee, S.K., Ho, C.C., and Henderson, T.C, "Range Data Processing: Representation of Surfaces by Edges" Xeroxing Cost: $0.75 (15 pages)

UUCS-85-115 Tom Henderson, Chuck Hansen, Ashok Samal, C.C. Ho. and Bir Bhanu, "CAGD Based 3-D Visual Recognition" Xeroxing Cost: $0.90 (18 pages)

UUCS-85-116 Daniel S. Janik, MD, MPH, Thomas C. Henderson, PhD., John L. Lyon, MD, Ed Sharp, and August L. Jung, MD, "The Importance of Unknowns in Epidemiologic Studies" Xeroxing Cost: $0.45 (9 pages)

UUCS-85-117 Gu, Jun and Smith, Kent F., "Fully Custom Design and Implementation of Optimal Path Programmable Cell Sets Using ZYTREX 2-Micron Metal-Gate CMOS Process"

UUCS-89-118 * Lin, Frank Chung Huei. "Load Balancing and Fault Tolerance in Applicative Systems." Xeroxing Cost: $6.70 (130 pgs)

UUCS-85-119 * O'Dell, Connie Lynn. "Approximating Data With Parametric B-Splines for Computer Aided Design." Xeroxing Cost: $4.60 (93 pgs)

UUCS-85-120 * Donahue, Brian Douglas. "Modeling Complex Objects With Generalized Sweeps." Xeroxing Costs: $3.75 (75 pgs)

UUCS-85-121 * Warren, Lloyd Van. "Geometric Hashing For Processing Complex Scenes." Xeroxing Cost: $4.30 (86 pgs)

UUCS-85-122 * Kitaoka, Shoichi. "KIT: An Experimental Solid Modelling System. Xeroxing Cost: $5.35 (107 pgs)

UUCS-85-123 * Duggan, Gerald Patrick. "Optimizaitons as Transformations: A Compiler Paradigm. Xeroxing Cost: $4.85 (97 pgs)

UUCS-84-001 Shilcrat, Esther, Panangaden, Prakash, and Henderson, Thomas C., "Implementing Multi-Sensor Systems in a Functional Language". Xeroxing Cost: $0.75 (15 pages)

UUCS-84-002 Henderson, Thomas C. and Shilcrat,Esther,"Logical Sensor Systems". Xeroxing Cost: $1.60 (32 pages)

UUCS-84-003 Riesenfeld, Richard F. and Hollaar, Lee A., "Computer Aided Design." Xeroxing Cost: $1.50 (30 pages)

UUCS-84-004 Keller, Robert M. and Panangaden, Prakash, "Semantics of Networks Containing Indeterminate Operators."
Xeroxing Cost: $0.90 (18 pages)

UUCS-84-005 Panangaden, Prakash, "A Category Theoretic Formalism for Abstract Interpretation", Xeroxing Cost: $1.30 (26 pages)

UUCS-84-006 Panangaden, Prakash, Mishra, Prateek, "Abstract Interpretation and Indeterminacy" Xeroxing Cost: $0.90 (18 pages)

UUCS-84-007 Pickett, Forrest B., "Simulation of Cells.
Xeroxing Cost: $4.50 (90 pages)

UUCS-84-008 Riesenfeld, Richard F. and Smith, Kent F., "An Experimental System for Computer Aided Geometric Design."
Xeroxing Cost: $3.40 (68 pages)

UUCS-84-009 ** Thomas, Spencer W., "Modelling Volumes Bounded by B-Spline Surfaces." Xeroxing Cost: $6.25 (125 pages)

UUCS-84-010 Rajopadhye, Sanjay V., "A Formal Basis for Synthesising Systolic Arrays." Xeroxing Cost: $1.75 (35 pages)

UUCS-83-001 Henderson, Thomas C. and Wu So Fai, "Pattern Recognition in a Multi-sensor Environment". Xeroxing Cost: $2.20 (44 pages)

UUCS-83-002 Henderson, Thomas C. and Wu So Fai, "Some Experiments with the 3-D Hough Shape Transform". Xeroxing Cost: $1.45 ((29 pages)

UUCS-83-003 Henderson, Thomas C., Shilcrat, Esther, and Hansen, Charles, "A Fault Tolerant Sensor Scheme." Xeroxing Cost: $0.75 (15 pages)

UTEC-83-026 Organick, Elliott I., et. al., "Transforming an Ada Program Unit to Silicon and Verifying Its Behavior in an Ada Environment: A First Experiment."
Xeroxing Cost: $0.85 (17 pages)

UTEC-83-030 Organick, Elliott, et. al., "Transparent Interface Between Software and Hardware Versions of Ada Compilation Units."
Xeroxing Cost: $1.10 (22 pages)

UTEC-83-040 Subrahmanyam, P.A. and You, J.-H., "FUNLOG = Functions + Logic: A Computational Model Integrating Logic Programming and Functional Programming." Xeroxing Cost: $1.55 (31 pages)

UTEC-83-075 Organick, Elliot I., "Transformation of Ada program Units Into Silicon." Xeroxing Cost: $1.75 (35 pages)

UTEC-83-076 Rajopadhye, Sanjay V. and Subrahmanyam,P.A.,"TRACIS: Transformations on Ada for Circuit Synthesis - A Report on the Methodology for a Silicon Compiler."
Xeroxing Cost: $1.40 (28 pages)

UUCS-83-101 Lindstrom, Gary, "The Key Node Method: A Highly-Parallel Alpha-Beta Algorithm". Xeroxing Cost: $1.20 (24 pages)

UUCS-82-012 Lindstrom, Gary, and Hunt, Frances C., "Consistency and Currency in Functional Databases". Xeroxing Cost: $1.25 (25 pages)

UUCS-82-015 Purushothaman, S., and Subrahmanyam, P. A., "A Model for Call-By-Need and Stream Primitives in CCS". Xeroxing Cost: $0.80

(16 pages)

UUCS-82-016 — Drenan, Lawrence S., and Organick, Elliott I., "ADA to Silicon Transformations: The Outline of a Method". Xeroxing Cost: $1.80 (36 pages)

UTEC-82-020 — Organick, Elliott I., "Transformation of ADA Programs into Silicon". Xeroxing Cost: $3.35 (67 pages)

UTEC-82-066 — Purushothaman S., and Subrahmanyam, P. A., "An Algebraic Formulation of Seitz's Weak Conditions for Self Timed Circuits". Xeroxing Cost: $0.50 (10 pages)

UTEC-82-070 — VLSI Research Group, "The Path Programmable Logic (PPL) User's Manual". Xeroxing Cost: $7.15 (143 pages)

UUCS-82-100 — Keller, Robert M., and Lindstrom, Gary, "Toward Function-Based Distributed Database Systems." Xeroxing Cost: $1.90 (38 pages)

UTEC-82-101 * — Carter, Tony, "ASSASSIN; A Cad System for Self-Timed Control-Unit Design". Xeroxing Cost: $1.40 (28 pages)

UTEC-82-103 — Organick, Elliott I., "Transformation of Ada Program into Silicon." Xeroxing Cost: $1.25 (25 pages)

UTEC-81-018 — Baxter, Brent S., "A Standard Magnetic Tape Format for Digital Image Exchange". Xeroxing Cost: $0.75 (15 pages)

UTEC-81-020 — Boll, Steven F., "Noise Suppression Methods for Robust Speech Processing". Xeroxing Cost: $2.50 (50 pages)

UTEC-80-004 * — Youngberg, James E., "A Constant Percentage Bandwidth Transform for Acoustic Signal Processing". Xeroxing Cost: $6.00 (120 pages)

UTEC-80-058 — Boll, Steven F., "Noise Suppression Methods for Robust Speech Processing". Xeroxing Cost: $1.85 (37 pages)

UUCS-80-101 — Subrahmanyam, P. A., "On Proving the Correctness of Data Type Implementation". Xeroxing Cost: $2.20 (44 pages)

UUCS-80-102 — Barsky, Brian A., and Gram, Christian, "A Description of Several Tools for the Synchronization of Concurrent Processes." Xeroxing Cost: $0.95 (19 pages)

UUCS-80-103 — Gram, Christian, and Organick, Elliott I., "Characteristics of a Functional Programming Language". Xeroxing Cost: $2.40 (48 pages)

UUCS-80-104 — Barsky, Brian A., and Thomas, Spencer W., "Transpline Curve Representation Systems". Xeroxing Cost $1.15 (23 pages)

UUCS-80-105 — Hayes, Alan B. "Stored State Asynchronous Sequential Circuits" Xeroxing Cost: $0.60 (12 pages)

UUCS-80-107 — Subrahmanyam, P. A., "A New Approach to Specifying and Handling Exceptions". Xeroxing Cost: $1.25 (25 pages)

UUCS-80-108 — Davis, A. L., and Denny, W. M., "A Characterization of Parallel Systems". Xeroxing Cost: $2.40 (48 pages)

UUCS-80-109 — Davis, A. L., and Drongowski, P. J., "Data Flow Computers: A Tutorial and Survey". Xeroxing Cost: $4.95 (99 pages)

UUCS-80-111 Weiser, Uri and Davis, Alan L., "Mathematical Representation for VLSI Arrays". Xeroxing Cost: $1.55 (31 pages)

UUCS-80-112 Keller, Robert M., "Semantics and Applications of Function Graphs". Xeroxing Cost: $3.50 (70 pages)

UUCS-80-113 * Petersen, Tracy L., "Acoustic Signal Processing in the Context of a Perceptual Model". Xeroxing Cost: $4.85 (97 pages)

UTEC-79-022 Pulsipher, Dennis C., "Application of Adaptive Noise Cancellation to Noise Reduction in Audio Signals". Xeroxing Cost: $3.50 (70 pages)

UTEC-79-039 Boll, Steven F., "ARPA Semi-Annual Technical Report/Noise Suppression Methods for Robust Speech Processing". Xeroxing Cost: $2.95 (59 pages)

UTEC-79-081 Frost, Richard L., Rushforth, Craig K., and Baxter, Brent S., "High Resolution Astronomical Imaging Through the Turbulent Atmosphere". Xeroxing Cost: $6.45 (129 pages)

UUCS-79-101 Lindstrom, Gary, "Alpha Beta Pruning on Evolving Game Trees". Xeroxing Cost: $2.60 (52 pages)

UTEC-79-102 Boll, Steven F., "Suppression of Acoustic Noise in Speech Using Spectral Subtraction". Xeroxing Cost: $4.25 (85 pages)

UUCS-79-103 Bloomenthal, Jules I., Fish, Russell D., and Riesenfeld, Richard F., "Frame Buffer System Selection" Xeroxing Cost: $0.65 (13 pages)

UUCS-79-104 Forrest, A. R., "On the Rendering of Surfaces". Xeroxing Cost: $ 0.95 (19 pages)

UUCS-79-111 Maxey, Gregory F., and Organick, Elliott I., "CASL - A Language for Automating the Implementation of Computer Architectures". Xeroxing Cost: $1.75 (35 pages)

UUCS-79-114 Boll, Steven F., and Pulsipher, Dennis C., "Suppression of Acoustic Noise in Speech Using Two Microphone Adaptive Noise Cancellation". Xeroxing Cost: $2.70 (54 pages)

UUCS-79-115 Lane, J. M., and Riesenfeld, R. F., "The Application of Total Positivity to Computer Aided Curve and Surface Design". Xeroxing Cost: $4.70 (94 pages)

UUCS-79-116 Lindstrom, Gary, and Soffa, Mary Lou, "Referencing and Retention in Block-Structured Coroutines". Xeroxing Cost: $2.50 (50 pages)

UUCS-79-117 Cohen, Elaine, "Discrete B-Splines and Subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics". Xeroxing Cost: $1.25 (25 pages)

UTEC-79-163 Boll, Steven F., "Noise Suppression Methods for Robust Speech Processing". Xeroxing Cost: $3.30 (66 pages)

UTEC-78-073 Boll, Steven F., "Noise Suppression Methods for Robust Speech Processing". Xeroxing Cost: $5.50 (110 pages)

UUCS-78-102 Keller, Robert M., "An Approach to Determinacy Proofs". Xeroxing Cost: $1.55 (31 pages)

UUCS-78-104 Keller, Robert M., "Sentinels: A Concept for Multiprocess Coordination". Xeroxing Cost: $1.60 (32 pages)

UUCS-78-105 Keller, Robert M., Lindstrom, Gary L., and Patil, Suhas, "An Architecture for a Loosely-Coupled Parallel Processor". Xeroxing Cost: $2.65 (53 pages)

UUCS-78-108 Davis, A. L., "Data Driven Nets: A Maximally Concurrent, Procedural, Parallel Process Representation for Distributed Control Systems". Xeroxing Cost: $2.10 (42 pages)

UUCS-78-111 Boll, Steven F., "A Spectral Subtraction Algorithm for Suppression of Acoustic Noise in Speech". Xeroxing Cost: $1.50 (30 pages)

UTEC-78-135 Evans, David C., "Display of Complex Three Dimensional Finite Element Models". Xeroxing Cost: $5.10 (102 pages)

UTEC-78-204 Boll, Steven F., "Noise Suppression Methods for Robust speech Processing". Xeroxing cost: $4.20 (84 pages)

UTEC-77-017 Stockham, Thomas G., "Sensory Information Processing". Xeroxing Cost: $2.80 (56 pages)

UTEC-77-029 * Faugeras, Olivier Dominique, "Digital Color Image Processing and Psychophysics Within the Framework of a Human Visual Model". Xeroxing Cost: $6.45 (129 pages)

UTEC-77-090 Boll, Steven F., "Noise Suppression for Robust Speech Processing". Xeroxing Cost: $3.90 (78 pages)

UUCS-77-101 Boll, Steven F., "Improving Linear Prediction Analysis of Noisy Speech by Predictive Noise Cancellation". Xeroxing Cost $ .20 (4 pages)

UUCS-77-103 Keller, Robert M., "Denotational Models for Parallel Programs with Indeterminate Operators". Xeroxing Cost: $1.40 (28 pages)

UUCS-77-105 Lindstrom, Gary, "Backtracking in Generalized Control Settings". Xeroxing Cost: $1.60 (32 pages)

UUCS-77-106 Nelson, Sallie S., and Lindstrom, Gary, "CONSIM: A Conversational Simulation Language Implemented Through Interpretive Control Self-Modelling". Xeroxing Cost: $1.20 (24 pages)

UUCS-77-107 Boll, Steven F., "Application of the Saber Method for Improved Spectral Analysis of Noisy Speech". Xeroxing Cost: $2.00 (40 pages)

UUCS-77-108 * Dintelman, Sue Marie Thompson, "Design and Implementation of a Relational Data Base System for a Minicomputer" Xeroxing Cost: $4.65 (93 pages)

UUCS-77-110 Keller, Robert M., "Semantics of Parallel Program Graphs". Xeroxing Cost: $1.25 (25 pages)

UUCS-77-112 Smith, John J., and Smith, Diane C. P., "Integrated Specifications for Abstract Systems".

Xeroxing Cost: $2.50 (50 pages)

UUCS-77-113 Davis, A. L., "The Architecture of DDML: A Recursively Structured Data Driven Machine". Xeroxing Cost: $1.45 (29 pages)

UUCS-77-114 Lindstrom, Gary, "Efficiency in Nondeterministic Control

	Through Non-Forgetful Backtracking". Xeroxing Cost: $1.00 (20 pages)
UTEC-77-117	Stockham, Thomas G., "Sensory Information Processing". (1 January 1976 - 30 June 1976) Xeroxing Cost: $2.85 (57 pages)
UTEC-77-118	Stockham, Thomas G., "Sensory Information Processing". (1 July 1976 - 31 March 1977) Xeroxing Cost: $5.25 (105 pages)
UTEC-77-135 *	McDonald, Lee E., "Image Recognition Using Generalized Correlation". Xeroxing Cost: $3.75 (75 pages)
UTEC-76-008	Stockham, Thomas G., "Sensory Information Processing and Symbolic Computation". Xeroxing Cost: $7.00 (140 pages)
UTEC-76-015 *	Crow, Franklin C., "The Aliasing Problem in Computer-Synthesized Shaded Images". Xeroxing Cost: $5.20 (104 pages)
UTEC-76-209 *	Callahan, Michael Wayne, "Acoustic Signal Processing Based on the Short-Time Spectrum". Xeroxing Cost: $4.25 (85 pages)
UTEC-76-218 *	Newell, Martin Edward, "ARPA Technical Report: The Utilization of Procedure Models in Digital Image Synthesis" Xeroxing Cost: $5.35 (107 pages)
UTEC-76-226	Christiansen, Richard W., "Word Recognition in Continuous Speech Using Linear Prediction Analysis". Xeroxing Cost: $4.30 (86 pages)
UTEC-76-270 *	Fuchs, Henry, "The Automatic Sensing and Analysis of 3-D Surface Points from Visual Scenes". Xeroxing Cost: $4.45 (89 pages)
UTEC-76-271 *	Atashroo, Mohammad A., "Pole-Zero Modeling and its Applications to Speech Processing". Xeroxing Cost: $5.20 (104 pages)

UTEC-75-047 **	Parke, Frederic Ira, "A Parametric Model for Human Faces" Xeroxing Cost: $5.60 (112 pages)
UTEC-75-115	Rom, Raphael Jona, "Image Transmission and Coding Based on Human Vision". Xeroxing Cost: $4.15 (83 pages)
UTEC-75-118 *	Ingebretsen, Robert Bergstrom, "Log Spectral Estimation for Stationary and Nonstationary Processes". Xeroxing Cost: $6.95 (139 pages)
UTEC-75-142	Greer, William H., "Monaural Sensitivity to Dispersion in Impulses and Speech". Xeroxing Cost: $7.50 (150 pages)
UTEC-75-168	Baxter, Brent, "Image Processing in the Human Visual System". Xeroxing Cost: $4.60 (92 pages)

UTEC-74-013	Miller, Neil J., "Removal of Noise from a Voice Signal by Synthesis". Xeroxing Cost: $4.45 (89 pages)
UTEC-74-025 *	Colas, Patrick Baudelaire, "Digital Picture Processing and Psychophysics: A Study of Brightness Perception". Xeroxing Cost: $7.80 (156 pages)
UTEC-74-029 *	Cole, Randolph E., "The Removal of Unknown Image Blurs by Homomorphic Filtering: Xeroxing Cost: $6.40 (128 pages)

UTEC-74-078 **	Vickers, Donald Lee, "Sorceror's Apprentice: Head-mounted Display and Wand". Xeroxing Cost: $4.25 (85 pages)
UTEC-74-091 **	Cannon, Thomas J., "Digital Image Deblurring by Nonlinear Homomorphic Filtering" Xeroxing Cost: $4.75 (95 pages)
UTEC-74-120 **	Clark, James Henry, "3-D Design of Free-Form B-Spline Surfaces". Xeroxing Cost: $4.05 (81 pages)
UTEC-74-133 **	Catmull, Edwin, "A Subdivision Algorithm for Computer Display of Curved Surfaces". Xeroxing Cost: $4.10 (82 pages)
UTEC-74-151	Boll, Steven F., "Selected Methods for Improving synthesis Speech Quality Using Linear Predictive Coding: System Description, Coefficient Smoothing and Streak". Xeroxing Cost: $3.15 (63 pages)
UTEC-73-123 *	Boll, Steven F., "A Prior Digital Speech Analysis". Xeroxing Cost: $5.75 (115 pages)
UTEC-73-124 *	Clark, Becky Jane, "A Design for a Micromodular Self-Timed System". Xeroxing Cost: $1.55 (31 pages)
UTEC-73-125 *	Marcchal, Andre, "Data Structure Techniques in an Implementation of APL". Xeroxing Cost: $6.65 (133 pages)
UTEC-73-126	Riesenfeld, Richard F., "Applications of B-Spline Approximation to Geometric Problems of computer-Aided Design". Xeroxing Cost: $4.65 (93 pages)
UTEC-73-126	Barnhill, Robert E., and Riesenfeld, Richard F., "Computer Aided Geometric Design". Information Pamphlet. Xeroxing Cost: (Free)
UTEC-73-127 *	Wessler, Barry David, "Computer-Assisted Visual Communication. Xeroxing Cost: $5.35 (105 pages)
UTEC-73-128	Debry, Roger K. and Greenfield, Harvey, "Simulation of Arbitrary Shaped Boundaries for Hemodynamic Studies" Xeroxing Cost: $14.00 (280 pages)
UTEC-73-129 *	Phong, Bui Tuong,"Illumination for Computer Generated Images". Xeroxing Cost: $4.80 (96 pages)

UTEC-72-118 *	Dick-Kau Au, Anthony, "A Computer Graphics Approach for Understanding prosthetic Heart Valve Characteristics". Xeroxing Cost: $3.75 (75 pages)
UTEC-72-119 *	Gestreicher, Donald, "Automatic Printed Circuit Board Design". Xeroxing Cost: $6.25 (125 pages)
UTEC-72-120	Parke, Frederic Ira, "Computer Generated Animation of Faces". Xeroxing Cost: $2.80 (56 pages)
UTEC-72-121	Archuleta, Michael, "Hidden Surface Line Drawing Algorithm". Xeroxing Cost: $3.70 (74 pages)
UTEC-72-122 *	Burton, Robert Preece, "Real-Time Measurement of Multiple Three-Dimensional Positions". Xeroxing Cost: $6.25 (125 pages)
March, 1972	Stockham, Thomas G., "Intra-Frame Encoding for Monochrome Images by Means of a Psychophysical Model Based on Nonlinear

Filtering of Multiplied Signals". Xeroxing Cost: (Free)

UTEC-71-113 Gouraud, Henri, "Computer Display of Curved Surfaces". Xeroxing Cost: $4.20 (84 pages)

UTEC-71-114 Bennion, Scott Thomas, "A Method of Solution for Hydrodynamic and Radiation Diffusion as a Multi-Material Problem in One Dimension". Xeroxing Cost: $5.75 (115 pages)

UTEC-71-115 Greenfield, Harvey, and Debry, Roger, "An Application of Computer Graphics: Two Concurrent Investigations within the Medical Field". Xeroxing Cost: $14.90 (298 pages)

UTEC-71-116 * Carey, Thomas A. and Hankley, William, J., "Empirical Modelling of Occurrence of Severe Weather Events". Xeroxing Cost: $4.80 (96 pages)

UTEC-71-117 Ashton, Alan Conway, "Electronics, Music and Computers". Xeroxing Cost: $8.85 (177 pages)

UTEC-70-100 * Mahl, Robert, "An Analytical Approach to Computer Systems Scheduling". Xeroxing Cost: $5.95 (119 pages)

UTEC-70-101 * Watkins, Gary Scott, "A Real Time Visible Surface Algorithm" Xeroxing Cost: $4.75 (95 pages)

UTEC-70-102 Wehrili, Robert, Smith, Max J., Smith, Edward F., "ARCAID, The ARChitect's Computer Graphics AID". Xeroxing Cost: $6.65 (131 pages)

UTEC-70-103 * Withrow, Carol, "A Dynamic Model for Computer-Aided Choreography". Xeroxing Cost: $3.00 (60 pages)

UTEC-70-104 Newman, William M., "An Experimental Display Programming Language for the PDP - 10 Computer". Xeroxing Cost: $2.35 (47 pages)

UTEC-70-105 Newman, William M., Gouraud, Henri, and Oestreicher, Donald R. "A Programmer's Guide to PDP - 1- Euler". Xeroxing Cost: $3.50 (70 pages)

UTEC-70-106 Nelson, Stanley Logan, "Master Schedule Building and the Flexibility Scheduled School". Xeroxing Cost: $3.80 (76 pages)

UTEC-70-107 Templeton, Frederick E., and Hankley, William J., "Optimal Control of a Process with Discrete and Continuous Decision Variables". Xeroxing Cost: $6.15 (123 pages)

UTEC-70-108 Seror, Dennis E., "DCPL, A Distributed Control Programming Language". Xeroxing Cost: $9.15 (183 pages)

UTEC-70-109 Durney, Carl H., "A Computer Graphics Method for Solving Transcendental Equations". Xeroxing Cost: $0.75 (15 pages)

UTEC-70-110 Durney, Carl H., "An Experiment in Using Interactive Computer Graphics in Teaching Transient Transmission-line Theory." Xeroxing Cost: $1.30 (26 pages)

UTEC-70-111 Mahl, Robert, "Visible Suface Algorithms for Quadric Patches". Xeroxing Cost: $1.40 (28 pages)

1968 TR 4-5 Warnock, John E., "A Hidden Line Algorithm for Halftone Picture Representation". Xeroxing Cost: $0.70

(14 pages)