Transforming Nonlinear Recursion into Linear Recursion

by

Yannis E. Ioannidis and Eugene Wong
University of Wisconsin, Madison
and University of California, Berkeley

1. Introduction

Consider a function-free and constant free Horn clause [Gall 78]

$$Q_i(\underline{x}^{(1)}) \ \wedge \ , \cdots , \ Q_n(\underline{x}^{(k)}) \ \rightarrow \ P(\underline{x}^{(0)}) \tag{1}$$

where for each i, $\underline{x}^{(i)}$ is a subset of some fixed set of variables $(x_1, x_2, \cdots, x_n)$. We say the formula is *recursive* if at least one term on the left hand side of (1) has the form:

$$Q_i = P \quad \text{and } \underline{x}^{(i)} \neq \underline{x}^{(0)} \tag{2}$$

The recursion is said to be *linear* if (2) holds for exactly one i, and *non-linear* (more precisely, *multilinear*) otherwise. The case of *bilinearity* where (2) holds for exactly two values of i is also of special interest.

In [Ioan 86] we presented an algebraic formulation for linear recursion. In this paper we generalize the algebraic formulation in several significant ways. First, the case of several mutually dependent recursive formulas is considered. Second, we show that all multilinear recursions can be reduced to bilinear mutual recursions. Third, we give an algebraic formulation for bilinear mutual recursion, and obtain a sufficient condition for these to be equivalent to linear ones. Finally, by suitably embedding the structure in a linear algebra, we obtain a finite and tractable test for this equivalence. The bottom line to all this is a set of usable tests to determine whether a general recursion is equivalent to a linear recursion. Although the problem of equivalence to linear recursion has been studied before [Zhan 87], our results are at once simpler and more general.

2. General Recursions

A formula of the form (1) can be restated in terms of relations as follows:

$$P \supset f(Q_1, \cdots, Q_k) \tag{3}$$

where $P, Q_1, \cdots, Q_k$, are relations and f is a function. If (1) is recursive, it is useful to isolate the P's on the right hand side, and write

$$P \supset f(P, Q) \tag{4}$$

where Q denotes the set of terms in $(Q_1, \cdots, Q_k)$ distinct from P.

A natural multidimensional generalization to (4) is the following:

$$P_i \supset f_i(P, Q) \quad , \quad i = 1, 2, \cdots, m, \tag{5}$$

where $P = (P_1, P_2, \cdots, P_m)$ is a set of relations that represent the unknowns, and $Q = (Q_1, \cdots, Q_n)$ is a set of known relations. The system of inequalities (5) is to be solved with a set of "initial conditions"

$$P_i \supset R_i \quad , \quad i = 1, 2, \cdots, m \tag{6}$$

where $R = (R_1, R_2, \cdots, R_m)$ are given. The minimal solution to (5) and (6) is given by the minimal solution of the equation

$$P_i = f_i(P, Q) \cup R_i \quad , \quad i = 1, 2, \cdots, m \tag{7}$$

Recursions of this form with $m > 1$ are sometimes referred to as *mutual recursions*. There is no loss of generality in assuming that there is only one formula for each i in (7), since if there are several, these can be replaced by their union.

3. An Algebraic Formulation of Mutual Recursion

Let $\mathbf{R}_i$ denote the collection of all relations having the same domains as P_i, so that $P \in \mathbf{R} = \underset{i}{\pi} \mathbf{R}_i$. We can define *addition* $\oplus$ on $\mathbf{R}$ by

$$P \oplus P' = \left(P_1 \cup P'_1,\ P_2 \cup P'_2,\ P_m \cup P'_m\right)$$

A function f: $\mathbf{R} \rightarrow \mathbf{R}$ is said to be *linear* if it is

additive: $f(P \oplus P') = f(P) \oplus f(P')$

and

homogeneous: $f(\phi) = \phi$

A recursive equation of the form (7) is said to be *linear* if for a given Q, $f(P, Q)$ is linear in P. If (7) is linear, we write it as

$$P = A(Q) P \oplus R \tag{8}$$

It is trivial to show that A(Q) is a matrix, i.e.,

$$\left[A(Q)\,P\right]_i = \sum_{j=1}^{m} A_{ij}(Q)\,P_j$$

where Σ denotes summation with respect to $\oplus$,

$$A_{ij}(Q)P_j = \left[A(Q)\,(\phi, \phi, \cdots, P_j, \cdots, \phi)\right]_i$$

and $A_{ij}(Q)$ are linear operators on R_j.

Equation (8) is a full generalization of the algebraic formulation given in [Ioan 86] for linear recursions involving a single relation ($m = 1$). In the abstract form, the general case for $m \neq 1$ is no different, and it is straightforward to prove that the unique minimal solution of (8) is given by

$$P = A^*(Q)\,R$$

where

$$A^* = \sum_{k=0}^{\infty} A^k(Q)\,R$$

The powers A^k are expressible in terms of a product defined by composition, i.e.,

$$(A \cdot B)\,P = A(BP)$$

To make all this explicit, consider an example

$$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} A_{11}\ P_1 \ \oplus & R_1 \\ A_{21}\ P_1 \ \oplus\ A_{22}\,P_{22} & \oplus\ R_2 \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \oplus \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$$

It is easy to see that

$$\begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}^k = \begin{bmatrix} A_{11}^k & 0 \\ C_k & A_{22}^k \end{bmatrix}$$

where C_k satisfies

$$C_k = A_{21} A_{11}^{k-1} \oplus A_{22} C_{k-1}$$

Since $C_0 = 0$, we have

$$\sum_{k=0}^{\infty} C_k = A_{21} A_{11}^* \oplus A_{22} \sum_{k=0}^{\infty} C_k$$

which yields

$$\sum_{k=0}^{\infty} C_k = A_{22}^* A_{21} A_{11}^*$$

Hence,

$$\sum_{k=0}^{\infty} \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}^k = \begin{bmatrix} A_{11}^* & 0 \\ A_{22}^* A_{21} A_{11}^* & A_{22}^* \end{bmatrix}$$

4. Bilinear Recursion

A function $g: \mathbf{R} \times \mathbf{R} \rightarrow \mathbf{R}$ is said to be *bilinear* if $g(P, P')$ is linear in P for each P', and linear in P' for each P. The recursion (7) is said to be *bilinear* if we can write

$$f(P, Q) = g(P, P; Q)$$

where for a given Q, g is bilinear. Henceforth we shall suppress the dependence of f on Q (since Q is a constant) and write (7) as

$$P = f(P) \oplus R \tag{9}$$

A recursion consisting of only linear and bilinear terms can now be written as

$$P = AP \oplus g(P, P) \oplus R \tag{10}$$

We note that we can always eliminate the linear term by writing

$$\begin{aligned} P &= A^* g(P, P) \oplus A^* R \\ &= g'(P, P) \oplus R' \end{aligned}$$

where $g' = A^* g$ is once again a bilinear function. Therefore, we shall focus on pure bilinear recursions of the form

$$P = g(P, P) \oplus R \tag{11}$$

Bilinear recursion, in the vector form as we have defined it, is generic in the sense that all multilinear recursions can be reduced to a vector bilinear recursion. For example, consider a Horn clause:

$$P(\underline{x}^{(1)}) \wedge P(\underline{x}^{(2)}) \wedge, \cdots, \wedge P(\underline{x}^{(m)}) \wedge Q_1(\underline{y}^{(1)}) \wedge,$$
$$\cdots \wedge Q_k(\underline{y}^{(k)}) \quad \rightarrow P(\underline{x}^{(0)})$$

Without loss of generality, this can always be reexpressed in relational form as

$$P = Q \ltimes P^m \oplus R$$

where P^m denotes the m-fold cartesian product of P with itself, and $\ltimes$ denotes semijoin. We can now write

$$P_1 = Q \ltimes P$$

$$P_2 = P_1 \ltimes P$$

$$\vdots$$

$$P_{m-1} = P_{m-2} \ltimes P$$

$$P = P_{m-1} \ltimes P \oplus R$$

which is bilinear. Because of its generic character, we can confine our consideration to vector-bilinear recursion. All Horn-clause derived recursions can be treated in this way.

Equation (11) can be solved by iteration as follows: Set

$$P_{m+1} = g(P_m, P_m) \oplus R \tag{12}$$

with $P_0 = R$. Because g is bilinear, $\{ P_m \}$ is an increasing sequence since

$$P_m \supset P_{m-1} \Longrightarrow g(P_m, P_m) \supset g(P_{m-1}, P_{m-1})$$
$$\Longrightarrow P_{m+1} \supset P_m$$

and $P_1 = R \oplus g(R, R) \supset P_0 = R$. It follows that $\lim_m P_m = P$ solves (11).

5. Equivalence to Linear Recursion

In this section our objective is to discover sufficient conditions under which a bilinear recursion (11) is equivalent to some linear recursion.

The bilinear function g is said to be *associative* if

$$g\left(g(a, b), c\right) = g\left(a, g(b, c)\right)$$

Associativity of g is a simple, albeit strong, condition to ensure linear equivalence. We defer showing its sufficiency for the moment, except to note that the most familiar example of all recursions, viz., the ancestor example, can be expressed in bilinear form as

$$\text{Ancestor} = \text{Parent} \oplus \pi_{14} \left(\text{Ancestor} \overset{2=3}{\bowtie} \text{Ancestor}\right)$$

or equivalently in a linear form

$$\text{Ancestor} = \text{Parent} \oplus \pi_{14} \left(\text{Parent} \overset{2=3}{\bowtie} \text{Ancestor}\right)$$

The equivalence here is due to the associativity of the bilinear function

$$g(Q, R) = \pi_{14} \left(Q \overset{2=3}{\bowtie} R\right)$$

Linear-equivalence is ensured by a condition weaker than associativity, namely, power-associativity. A bilinear g can be viewed as a multiplication, and we can define *power* as follows:

$$a^1 = a\ , \quad a^{m+1} = g(a, a^m) \tag{13}$$

We say g is *power-subassociative* if

$$g(a^m, a^n) \subseteq a^{m+n} \quad \text{for all} \quad m, n \tag{14}$$

and *power-associative* if (14) holds with equality. Now, if g is power-subassociative, then (12) yields a simple formula for P_m, viz.,

$$P_m = \sum_{k=0}^{2^m} R^k \tag{15}$$

To verify this, we need only to compute

$$g\left(\sum_{k=1}^{2^m} R^k \ , \ \sum_{k=1}^{2^m} R^k\right) = \sum_{k,l=1}^{2^m} g\left(R^k , R^l\right)$$

$$\sum_{k,l=1}^{2^m} R^{k+l} = \sum_{k=2}^{2^{m+1}} R^k$$

Equation (15) implies that the solution of (11) is given by

$$P = \sum_{k=1}^{\infty} R^k \tag{16}$$

For a fixed Q, let G(Q) be the linear operator defined by

$$G(Q)R = g(Q, R) \tag{17}$$

Then, we can write $R^{m+1} = G^m(R)R$ and

$$P = G^*(R)R \tag{18}$$

which can be restated as follows:

Theorem 1. Let g be power-subassociative. Thus (11) is equivalent to the linear recursion

$$P = G(R)\,P \oplus R \tag{19}$$

Corollary: (19) holds if g is associative.

proof of corollary: We shall show that associativity implies power-associativity, whence (19) follows *a fortiori*. Write

$$g(a^m , a^n) = g\left(g(a , a^{m-1}) \ , \ a^n\right)$$

If g is associative, we have

$$g\left(g(a, a^{m-1}), a^n\right) = g\left(a, g(a^{m-1}, a^n)\right)$$

and power associativity now follows by induction. q.e.d.

For an example consider a set of mutual recursions used in [Ceri 87]:

$$P_1 = Q_1 \oplus P_1 \circ P_3 \oplus P_2 \tag{20}$$

$$P_2 = P_1 \circ P_3 \oplus Q_2$$

$$P_3 = P_3 \circ R \oplus Q_3$$

where all relations are binary. The operator $\circ$ is defined by

$$A \circ B = \pi_{14} \left(A \overset{2=3}{\bowtie} B\right)$$

We can recast the example in the form of (10) by identifying

$$AP = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & B(R) \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} \qquad \left(B(R)\,Q = Q \circ R\right)$$

and

$$g(Q, P) = \begin{bmatrix} P_1 \circ Q_3 \\ P_1 \circ Q_3 \\ 0 \end{bmatrix}$$

We can now transform (20) into the form of (11) as

$$P = g(P, P) \oplus A^*(R)\,Q$$

where A^* is given by

$$A^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & B^*(R) \end{bmatrix}$$

and g remains the same as before.

We can now compute powers as follows:

$$g(a, a^k) = \begin{bmatrix} B^k(a_3) \quad (a_1) \\ B^k(a_3) \quad (a_1) \\ \phi \end{bmatrix}$$

$$g(a^j, a^k) = \phi \quad , \quad j > 1$$

Hence, g is power-subassociative and the example is equivalent to the linear recursion

$$P = P \circ A^*Q \oplus Q$$

The Ceri-Tanca example provides a good illustration of mutual recursion, but it is not a good example for Theorem 1. This is because (20) is only pseudo-nonlinear and can be linearized by first solving the last equation and then substituting the resulting P_3 in the first two equations to get a pair of mutual *linear* recursions in $(P_1 \, P_2)$.

For a second example, consider the following:

$$P_1 = P_1 \circ P_2 \oplus P_1 \circ P_1 \oplus Q_1$$

$$P_2 = P_2 \circ P_1 \oplus P_2 \circ P_2 \oplus Q_2$$

where $\circ$ is defined as in the last example. This pair of equations is already in the form of (11), and the bilinear function g can be chosen to be

$$g(a, b) = a \circ Kb \quad , \; K = [1 \;\; 1]$$

Here,

$$g\left(a, g(b, c)\right) = g\left(g(a, b), c\right) = a \circ Kb \circ Kc$$

so that g is associative and the bilinear recursion is again equivalent to a linear one.

6. Embedding R in a Linear Algebra

The defining condition (14) for power-subassociativity and power-associativity is difficult to verify in general, since it has to be tested for all m and n. Unless it is recursively verifiable as in our example (20), the condition provides no finite test. In this section we shall find a finite test for power-associativity (unfortunately, not for subassociativity). This is done by embedding the operations: addition (= union) and multiplication (= g) in a linear algebra. This embedding also has an independent interest that has yet to be explored.

As before, let $\mathbf{R} = \prod_{i}^{N} \mathbf{R}_i$ and each $\mathbf{R}_i$ is the space of all relations having the same domains. Suppose that the domains of a given $\mathbf{R}_i$ are $D_1, D_2, \ldots, D_{k_1}$. By definition, each $R \in \mathbf{R}_i$ is a subset of $\mathbf{D}_i = \prod_{j=1}^{k_1} D_j$. As such, R can be viewed as a function R: $\mathbf{D}_i \rightarrow \{0, 1\}$, defined by

$$R(t) = 1\,, \quad \text{if the tuple t is in R}$$
$$= 0\,, \quad \text{otherwise}$$

The reason why we want to look at R this way is to be able to embed relations in a richer algebraic structure.

Now, each $\mathbf{R}_i$ corresponds to the space of all functions mapping $\mathbf{D}_i$ into $\{0, 1\}$ and

$$\mathbf{R} = \left\{ \text{all functions} \quad \rho\colon \mathbf{D} \rightarrow \{0, 1\} \right\}$$

We can now extend each $\mathbf{R}_i$ by considering all functions mapping $\mathbf{D}_i$ into $\mathbb{R} = \{\text{reals}\}$. Each such function will be called an *extended* relation. In this way, we can extend $\mathbf{R}$ to

$$\mathbf{R} = \{\text{all functions } \rho : \mathbf{D} \rightarrow \mathbb{R}\}$$

which is well known to be a linear vector space over the field $\mathbb{R}$.

Observe that the addition operation $\oplus$ defined on $\mathbf{R}$ can now be interpreted as a binary operator on $\{0, 1\}$ given by

$$a \oplus b = 1 \quad \text{if} \quad \max(a, b) = 1, a, b, \in \{0, 1\}$$
$$= 0$$

For the extended relation, the appropriate addition is the ordinary addition of real numbers. If we define for real numbers a,

$$|a| = 1\,, \text{ if } a > 0$$

$= 0\,,\quad \text{otherwise}$

Then,

$$|a + b| = |a| \oplus |b| \qquad \text{whenever} \quad a, b > 0$$

Applying this to relations, we can use $|\ \ |$ to map $\overline{\mathbf{R}}$ into $\mathbf{R}$ and have the relationship:

$$|Q + R| = |Q| \oplus |R|\,,\ Q, R \in \mathbf{R}$$

Given a function f: $\mathbf{R} \rightarrow \mathbf{R}$, define an extension $\overline{f}$: $\overline{\mathbf{R}} \rightarrow \overline{\mathbf{R}}$ as follows:

$$\overline{f}(\rho) = \sum_{t \in \mathbf{D}} \rho(t)\, f(1_t) \tag{21}$$

where 1_t is the indicator function

$$1_t(s) = 1 \quad \text{if} \quad s = t$$

$$= 0\,,\quad \text{otherwise}$$

Note that the summation in (21) is with respect to ordinary addition.

The function $\overline{f}$ defined by (21) is linear (on the vector space $\overline{\mathbf{R}}$) whether f is linear or not. (Observe that in (21) only the values of f on single tuples (i.e., 1_t) are used.) However, f is recoverable from $\overline{f}$ if and only if f is linear.

Proposition 1.

For any $A \in \mathbf{D}$, let $1_A \in \mathbf{R} \subset \overline{\mathbf{R}}$ denote its indicator function, i.e. $1_A(t) = 1$, $t \in A$, $= 0$ otherwise. Then,

$$f(1_A) = |\overline{f}(1_A)| \tag{22}$$

if and only if f is linear.

Proof: First, suppose f is linear. Then,

$$f(1_A) = \sum_{t \in A} f(1_t)$$

while

$$\overline{f}(1_A) = \sum_{t \in A} f(1_t)$$

and (22) follows:

Conversely, suppose (22) holds. Then

$$f(1_A) = |\sum_{t \in A} f(1_t)| \tag{23}$$

$$= \sum_{t \in A} f(1_t)$$

which implies that f is linear. q.e.d.

For a linear f we shall continue to write it as $f(\rho) = A\rho$, and similarly for its extension,

$$\overline{f}(\rho) = \overline{A}\rho$$

We now have the following relationship between A and $\overline{A}$.

Proposition 2.

For all integers k and all $R \in \overline{\mathbf{R}}$

$$|\overline{A}^k R| = A^k |R| \tag{23}$$

Proof: We use induction and first verify that the relationship is an identity for $k = 0$. Next, assume the relationship to be true for $k \leq m$. Then,

$$|\overline{A}^{m+1} R| = |\overline{A}\,\overline{A}^m R| = |\sum_{s \in D} (\overline{A}^m R)(s) A1_s|$$

$$= |\sum_{s \in D} |\overline{A}^m R|(s) A1_s|$$

$$= |\sum_{s \in D} (A^m |R|)(s) A1_s|$$

$$= |A \sum_{s \in D} (A^m |R|)(s) 1_s| = |AA^m R|$$

$$= |A^{m+1} R|$$

q.e.d.

Proposition 2 implies that A^* can be computed from any positive power series of $\overline{A}$, e.g.,

$$A^* = |e^{\overline{A}}| \tag{24}$$

or

$$A^* = |(I - \alpha \overline{A})^{-1}|, \quad \alpha > 0$$

Thus far, we have not found a way to exploit the connection to computational advantage. However, some results are more easily seen in terms of $\overline{A}$. For example, suppose $A = B \oplus C$ where B and C commute. Thus, $\overline{A} = \overline{B} + \overline{C}$, and $\overline{B}$ and $\overline{C}$ also commute. Hence, (24) immediately implies that

$$(B \oplus C)^* = B^* C^* \quad \text{wherever B and C commute.}$$

Given a function g on $\mathbf{R} \times \mathbf{R}$, we can extend it as follows.

$$\overline{g}(\rho, \sigma) = \sum_{s,t} \rho(s) \ \sigma(t)\, g(1_s, 1_t) \tag{25}$$

As in the case of (21), $\overline{g}$ is bilinear whether g is bilinear or not, but g and $\overline{g}$ are isomorphic iff g is bilinear as is indicated as follows.

Proposition 2.

The following relationship holds if and only if g is bilinear

$$g(1_A, 1_B) = |\overline{g}(1_A, 1_B)| \tag{26}$$

The proof is similar to that of Proposition 1 and will be omitted.

The bilinear function $\overline{g}$ on the linear vector space $\overline{\mathbf{R}}$ defines a multiplication that distributes over addition, i.e.,

$$\overline{g}(a + b, c) = \overline{g}(a, c) + \overline{g}(b, c) \tag{27}$$

$$\overline{g}(a, b + c) = \overline{g}(a, b) + \overline{g}(a, c)$$

By definition, the pair $(\overline{\mathbf{R}}, \overline{g})$ is a *linear algebra.*

The functions g and $\overline{g}$ define powers on $\mathbf{R}$ and $\overline{\mathbf{R}}$ respectively, as follows:

$$a^{(k)} = g(a, a^{(k-1)}) \ , \ a^{(1)} = a \ , a \in \mathbf{R}$$

$$a^k = \overline{g}(a, a^{k-1}) \ , \ a^1 = a \ , a \in \overline{\mathbf{R}}$$

Equation (27) implies that

$$|a^k| = |a|^{(k)}$$

The concept of power associativity can now be extended to g. We say a bilinear function g on $\overline{\mathbf{R}} \times \overline{\mathbf{R}}$ is *power associative (subassociative)* if

$$\bar{g}(a^k, a^j) = a^{k+j} \quad (\leq a^{k+j})$$

Proposition 3.

A bilinear function g on $\mathbf{R} \times \mathbf{R}$ is power associative (subassociative) if its extension $\bar{g}$ is power associative (subassociative).

Proof: Suppose $\bar{g}$ is power associative, then for $a \in \mathbf{R}$

$$g(a^{(k)}, a^{(j)}) = |\bar{g}(a^{(k)}, a^{(j)})|$$

$$= |\bar{g}(a^k, a^j)| = |a^{k+j}| = a^{k+j}$$

For subassociativity, we only need to replace the next-to-last equality by $\leq$.

q.e.d.

Theorem 2. A sufficient condition for g to be power associative is the following.

$$\bar{g}(a^k, a^j) = a^{k+j} \quad \text{for } k + j \leq 4 \tag{28}$$

Proof: The proof is based on the fact that $\bar{g}$, being a bilinear function on a linear algebra with characteristic $\neq 2$, is power associative if and only if (28) holds [Scha 66]. The desired result of Theorem 2 then follows from Proposition 3. q.e.d

Note: The important thing about (28) is that it need only be tested for a small number of k and j. This provides an easy-to-test sufficient condition for a bilinear recursion to be equivalent to a linear one.

7. Conclusion

In this paper, we continue to develop an algebraic theory of recursive inference begun in [Ioan 86]. Here, our focus is on bilinear recursions. These are of special interest for two major reasons: (1) If extended to the vector case, bilinear recursions suffice to represent all multilinear recursions, hence all recursions arising from Horn clauses. (2) Under suitable conditions, bilinear recursions are equivalent to linear recursions. Thus, a theory of linear equivalence for bilinear recursion encompasses all multilinear recursions.

With addition defined by set union, and multiplication defined by a bilinear function, a space of relations becomes a semiring, and the problem of linear equivalence has a simple algebraic statement which readily yields a simple sufficient condition, viz., power subassociativity. This condition is new, and both simpler and more general than anything previously discovered. Furthermore, by embedding relations in a richer structure (linear algebra), we obtain a sufficient condition that is even easier to test than power subassociativity.

In extending the algebraic formulations to vectors of relations, we not only succeed in transforming multilinear recursions into bilinear recursions, but also accommodate "mutual recursions" automatically in the process. For linear mutual recursions, the algebraic formulation results in a theory of matrix algebra that greatly facilitates manipulation.

We note that, thus far, the results of the algebraic approach that we have undertaken are theoretical, not computational. However, efficient algorithms nearly always require both a general theory and special properties of the problem at hand. From this point of view, the next logical direction is to examine the details of relational operations (e.g., projection, join, selection) in light of the algebraic theory. For example, what properties would render a combination of joins and projections power subassociative? We are in the process of undertaking such an investigation.

References

[Gall 78] H. Gallaire and J. Minker, *Logic and Databases,* Plenum Press, New York, 1978.

[Zhan 87] W. Zhang and C.T. Yu, " A Necessary Condition for a Doubly Recursive Role to be Equivalent to a Linear Recursive Role," *Proc. ACM-SIGMOD Conference,* May 1987, pp. 345-356.

[Ioan 86] Y.E. Ioannidis and E. Wong, "An Algebraic Approach to Recursive Inference," *Proc. First International Conference on Expert Systems,* April 1986, pp. 209-223.

[Ceri 87] S. Ceri and L. Tanca, "Optimization of Systems of Algebraic Equations for Evaluating Datalog Queries," *Proc. 1987 VLDB Conference,* Brighton, England, pp. 31-41.

[Scha 86] R.D. Schafer, *An Introduction to Nonassociative Algebras,* Academic Press, New York, 1966.