ON THE EXPRESSIVE POWER OF THE EXTENDED RELATIONAL ALGEBRA FOR THE UNNORMALIZED RELATIONAL MODEL

Dirk Van Gucht
Computer Science Department
Indiana University
Bloomington, Indiana 47405
vgucht@indiana CSNET

1 Introduction

The original relational database model proposed by Codd permitted complex structures to be entries in a component of a tuple [Cod] However, Codd recommended that only atomic data values be permitted The relational model with this restriction to *first normal form* (1NF) has gained wide acceptance

One of the earliest suggestions that 1NF was too restrictive came from Makinouchi in 1977 [Mak] While his treatment was fairly informal, he showed that relaxing the 1NF restriction could more faithfully model some database applications Similar suggestions were later made by other researchers [AB, Ban, CT, FA, FVT, Gon, JS, KTT, Kor, KV, Mac, OMO, RKS1, SP, TF, V]

Jaeschke and Schek introduced the NEST and UNNEST operators to restructure relations from 1NF to unnormalized form [JS] While they considered nesting only over single attributes, they established some important algebraic results Thomas and Fischer generalized the model to allow multilevel relational structures and introduced an extended relational algebra for this data model They also established many of the results in [JS] for the more generalized model More recently, Schek and Scholl described a similar model [SS] Based on these models, various researchers studied properties of unnormalized relational databases [DKA, FSTV, FT, FV1, FV2, FV3, FVT, FA, Gir, Kor, JS, KTT, OY, RKB, RKS1, RKS2, Sche, Scho, SP, SS, Tho, TF, V, VF1, VF2, VF3]

In this paper we address the question of the expressive power of the extended relational algebra of the unnormalized relational model In particular, we will show that the extended relational algebra is complete in the sense of Bancilhon and Paredaens [Ban, Par] This notion of completeness is called *BP-completeness*, a notion introduced by Chandra and Harel [CH]† In essence, a query language, be it specified as an algebra or as a logic, is BP-complete if one can show that

1 For every database d and every query E of the language, $E(d)$ remains invariant under the transformations which leave d invariant, i e , if ψ is a transformation such that $\psi(d) = d$, then we also have $\psi(E(d)) = E(d)$ This property is called the *BP-boundedness* of the language

† We note that Aho and Ullman addressed related issues [AH]

2 For every database d and every database d', if the set of transformations which leave d' invariant is contained in the set of transformations which leave d invariant, there exists an query E of the language such that $d' = E(d)$ This property is called the *BP-expressiveness* of the language

It was an observation by Imielinski [Imi], questioning the semantics of the NEST and UNNEST operators, which motivated us to study this problem The fact that we are able to generalize the result, that the relational algebra (or calculus) is BP-complete, to the extended relational algebra precisely describes the semantic power of these new operators in the context of the unnormalized relational model In addition, this result yields confidence in the naturalness of the extended relational algebra as a (basic) query language for the unnormalized relational model

2 The Unnormalized Relational Database Model

In this section, we describe an adaptation of the unnormalized relational database model of Thomas and Fischer [TF] This model consists of two components

a) a data description language,

b) a data manipulation language the extended relational algebra

The data description language defines a database as a set of unnormalized relational structures This is in contrast with the classical relational model where a database is defined as a set of normalized (or flat) relations

The mechanism used for manipulating structures is the extended relational algebra, it consists of the following operators

a) the classical relational operators extended to structures union ($\cup$), difference ($-$), selection (σ), projection (Π), and cartesian product ($\times$),

b) two restructuring operators NEST and UNNEST

2 1 Schemes and Structures

In order to formalize the notion of a structure, we need the auxiliary concept of a scheme Intuitively, a scheme specifies the format of a structure

Let Ω denote the *universe* of attributes A *scheme* S is a directed rooted tree for which the leaf nodes are distinct elements of Ω The root node will be denoted $\mathbf{R}(S)$ The set of non-leaf (*interior*) nodes of S is denoted $\mathrm{int}(S)$ The set of leaf nodes (*attributes*) of S is denoted by $\mathrm{att}(S)$ The set of all interior nodes which define proper *subschemes* of S is called $\mathrm{sub}(S)$ Clearly, $\mathrm{sub}(S) = \mathrm{int}(S) - \{\mathbf{R}(S)\}$ The set of interior nodes which have only leaf nodes as their children is called the set $\mathrm{frn}(S)$ of *frontier* nodes of S If $\mathrm{frn}(S) = \{\mathbf{R}(S)\}$ we will call S a *flat scheme* If the notions att, int, sub, and frn are applied to a node $M \in \mathrm{int}(S)$, they will be applied to the subtree $\mathbf{T}(M)$ of S having M as root Thus, $\mathrm{att}(M)$ means $\mathrm{att}(\mathbf{T}(M))$, etc

For a node $M \in S$, the set of children of M, denoted by $\mathbf{C}(M)$, is partitioned into attributes $\mathbf{A}(M) = \mathbf{C}(M) \cap \mathrm{att}(S)$ and into *higher order* nodes $\mathbf{H}(M) = \mathbf{C}(M) \cap \mathrm{int}(S)$ If the functions $\mathbf{C}$, $\mathbf{A}$ and $\mathbf{H}$ are applied to a scheme S, they will be applied to the root $\mathbf{R}(S)$ Thus, $\mathbf{C}(S)$ means $\mathbf{C}(\mathbf{R}(S))$, etc If we label a node Y^*, where Y denotes a set of nodes of S, we will always mean $\mathbf{C}(Y^*) = Y$ Finally, we say that two nodes $M, N \in \mathrm{int}(S)$ are *incomparable* in S if and only if neither is a descendant of the other

We are now ready to define structures A *structure* s over the scheme S is defined recursively as follows

1 if S is a flat scheme then s is a finite set ot tuples, where a tuple t is a mapping with domain $\mathbf{C}(S)$ (in the case of a flat scheme $\mathbf{C}(S) = \mathbf{A}(S) = \mathrm{att}(S)$) such that, for each $N \in \mathbf{C}(S)$, $t(N) \in \mathrm{dom}(N)$

2 otherwise s is a finite set of tuples where a tuple t is a mapping with domain $\mathbf{C}(S)$ such that, for each $N \in \mathbf{H}(N)$, $t(N)$ is a nonempty structure over $\mathbf{T}(N)$ and for each $N \in \mathbf{A}(N)$, $t(N) \in \mathrm{dom}(N)$

If S is a flat scheme, a structure over S is called a *flat relation* or simply a *relation*

A *database* is a finite set of structures

It is interesting to compare schemes with *formats* as defined by Hull and Yap in the Format Model [EY] Schemes are special formats in the following sense

1 the concept of *generalization* in formats is not considered in the definition of schemes, and

2 *set formation* is always done whenever we do *aggregation*

One can interpret schemes as formats without generalization nodes which are in *normal form* as defined in the Format Model

2 2 The Extended Relational Algebra

In order to query, reorganize or analyze structures, the URDM is equiped with a set of algebraic operators The operators are divided into two groups The first group of operators consists of the classical relational operators extended in the natural way to structures union, cartesian product, selection†, and projection (cf [FT, TF]) The second group consists of the NEST and UNNEST operators, which are important and natural restructuring operators NEST and UNNEST were first defined by Jaeschke and Schek [JS] and further analyzed by [FSTV, FT, FV1, FV2, Gir, KTT, RKS1, RKS2, Sche, Scho, SS, Tho, V, VF1, VF2] In this section we will define the NEST and UNNEST operations

First, let t be a tuple over scheme S and let $Y \subset \mathbf{C}(S)$ The *Y-portion* of t, denoted $t[Y]$, is the restriction of the tuple t to Y‡

Now let s be a structure over scheme S and let $Y \subset \mathbf{C}(S)$ Then $\mathrm{NEST}_Y(s)$ is a structure s^ν over scheme S^ν, where S^ν is obtained by inserting a new child Y^* of $\mathbf{R}(S)$ into S and making all of the members of Y children of Y^* in S^ν Hence $\mathbf{C}(S^\nu) = (\mathbf{C}(S) - Y) \cup \{Y^*\}$ and $\mathbf{C}(Y^*) = Y$ A tuple $t^\nu \in s^\nu$ if and only if there exists a tuple $t \in s$ such that

1 $t^\nu[\mathbf{C}(S) - Y] = t[\mathbf{C}(S) - Y]$ and

2 $t^\nu(Y^*) = \Pi_Y(\{t' \in s \mid t'[\mathbf{C}(S) - Y] = t[\mathbf{C}(S) - Y]\})$

Thus, the tuples of s which agree outside of Y are merged into a single tuple of s^ν with a set-valued entry in the Y^* position

Let s be a structure with scheme S and let $M \in \mathbf{H}(S)$ Then $\mathrm{UNNEST}_M(s)$ is a structure s^μ over scheme S^μ, where S^μ is obtained by deleting M from S and making the children of M into children of $\mathbf{R}(S)$, i e , $\mathbf{C}(S^\mu) = (\mathbf{C}(S) - \{M\}) \cup \mathbf{C}(M)$ A tuple $t^\mu \in s^\mu$ if and only if there exists a tuple $t \in s$ such that

1 $t^\mu[\mathbf{C}(S) - \{M\}] = t[\mathbf{C}(S) - \{M\}]$ and

2 $t^\mu[\mathbf{C}(M)] \in t(M)$

2 3 Dependencies in the URDM

The definitions of dependencies are straightforward generalizations of the classical ones Let S be a scheme and X and Y subsets of $\mathbf{C}(S)$ Then $X \rightarrow Y$ denotes a *functional dependency* (FD) A structure s over S satisfies the FD $X \rightarrow Y$ if and only if for any two tuples t_1 and $t_2 \in s$ such that $t_1[X] = t_2[X]$, we have $t_1[Y] = t_2[Y]$ The set X is a *key* for s if and only if s satisfies the FD $X \rightarrow \mathbf{C}(S)$

The notions of *multivalued dependencies*, *join dependencies* and other dependencies are similarly extended to the URDM We omit details

2 4 Basic Results

In this section we present some of the basic algebraic properties of the NEST and UNNEST operators used in this paper These results are due to Jaeschke and Schek [JS], Fischer and Thomas [FT, TF], and Van Gucht and Fischer [V, VF2]

Lemma 1 [JS, FT] Let s be a structure over scheme S and $Y \subset \mathbf{C}(S)$ Then

$$\mathrm{UNNEST}_{Y^*}(\mathrm{NEST}_Y(s)) = s$$

This lemma states that UNNEST is the left inverse of NEST, hence no information is lost after nesting However unnesting followed by renesting does not always preserve a structure In the following lemma we characterize when the UNNEST operator produces lossless restructuring

Lemma 2 [TF] Let s be a structure over scheme S and let $Y^* \in \mathbf{H}(S)$ Then $s = \mathrm{NEST}_Y(\mathrm{UNNEST}_{Y^*}(s))$ if and only if s satisfies the FD $(\mathbf{C}(S) - Y^*) \rightarrow Y^*$

---

† The select operator comes in two kinds let s be a structure over scheme S and let N, $M \in \mathbf{C}(S)$, then

1 $\sigma_{N=M}(s) = \{t \in s \mid t(N) = t(M)\}$ and

2 $\sigma_{N \neq M}(s) = \{t \in s \mid t(N) \neq t(M)\}$

This enables us not to consider the difference operator as a primitive operator

‡ Remember that a tuple is defined as a mapping which enables us to use the concept of the restriction of a mapping For a single node $N \in \mathbf{C}(S)$, $t[N]$ is a tuple with one component and $t(N)$ is the value of the entry in the N-position of t

Two NEST operations need not commute [TF, FV1, VF1] However, two UNNEST operations will commute

Lemma 3 [JS,FT] Let s be a structure over scheme S and let $M, N \in \mathbf{H}(S)$ Then,

$$\text{UNNEST}_M(\text{UNNEST}_N(s)) = \text{UNNEST}_N(\text{UNNEST}_M(s))$$

Since the UNNEST operator commutes, we can extend its definition to cover unnesting over a set of nodes Let S be a scheme and $Q(S) \subset \text{sub}(S)$ We say that $Q(S)$ is an *unnesting set* of S if and only if it satisfies the following property if $M \in Q(S)$, then all the ancestors of M in S, except for $\mathbf{R}(S)$, are also in $Q(S)$ Now let s be a structure over S and $Q(S)$ be an unnesting set of S Then $\text{UNNEST}_{Q(S)}(s)$ is defined as the structure s, if $Q(S) = \emptyset$, otherwise as the structure

$$\text{UNNEST}_{M_1}(\text{UNNEST}_{M_2}(\quad (\text{UNNEST}_{M_k}(s)) \quad))$$

where the sequence $(M_1, \quad , M_k)$ satisfies the conditions

1 $\{M_1, \quad , M_k\} = Q(S)$ and
2 for each pair of indices i and j, $1 \leq i < j \leq k$, either M_i is a descendant of M_j in S or M_i and M_j are incomparable nodes in S

Van Gucht and Fischer [V, VF2] show that $\text{UNNEST}_{Q(S)}(s)$ is well-defined

Lemma 4 [V, VF2] Let s be a structure over scheme S and $Q(S)$ an unnesting set of S Then, $\text{UNNEST}_{Q(S)}(s)$ is a well-defined structure

Corollary 1 Let s be a structure over scheme S Then, $\text{UNNEST}_{\text{sub}(S)}(s)$ is a well-defined relation over the flat scheme R, with $\mathbf{C}(R) = \text{att}(S)$

Henceforth we will let r_s denote $\text{UNNEST}_{\text{sub}(S)}(s)$ and we will call r_s the *associated relation* of s Similarly we will denote the flat scheme R such that $\mathbf{C}(R) = \text{att}(S)$ by R_S, thus the structure r_s is defined over the scheme R_S

3 The Expressive Power of the Extended Relational Algebra

In this section, we will prove that the extended relational algebra is BP-complete To establish this result, we will use the formalism of Paredaens [Par] rather than that of Bancilhon [Ban] or Chandra and Harel [CH] In Section 3 1, we review the formalism of [Par] In Section 3 2, we review the results of [Par] related to the BP-completeness of the relational algebra Finally, in Section 3 3, we generalize Paredaens' results to the extended relational algebra of the URDM

3 1 Definitions

Let s be a structure over S The *associated domain* of s, denoted $D(s)$, is $\bigcup_{A \in \mathbf{C}(R_S)} \Pi_{\{A\}}(r_s)$, i e , $D(s)$ consists of all the atomic values appearing in s Let $s_1, \quad , s_n$ be structures over $S_1, \quad , S_n$ respectively The domain of $s_1, \quad , s_n$, denoted $D(s_1, \quad , s_n)$, is $\bigcup_{1 \leq i \leq n} D(s_i)$

The basic information of the structures $s_1, \quad , s_n$ is the set $BI(s_1, \quad , s_n) = \{s_E | s_E = E(s_1, \quad , s_n)$, where E is an expression of the extended relational algebra with operands $s_1, \quad , s_n\}$

Let s be a structure over S and let ψ be a bijection on $D(s)$ We define the extension, ψ^e, of ψ to s recursively as follows

1 if s is a flat relation then $\psi^e(s) = \{\psi \circ t | t \in s\}$ ($\circ$ indicates function composition and recall that t is a mapping from $\mathbf{C}(S)$ to $D(s)$), otherwise
2 $\psi^e(s) = \{\psi^e \circ t | t \in s$, where $\psi^e \circ t(N) = \psi \circ t(N)$ if $N \in \mathbf{A}(S)$ and $\psi^e \circ t(N) = \psi^e(t(N))$ if $N \in \mathbf{H}(S)\}$

The function ψ is called *s-compatible* if and only if for all $t \in s$, $\psi^e \circ t \in s$ The set of all s-compatible functions forms a group under function composition and is called the *cogroup* of s, denoted $CG(s)$ Let $s_1, \quad , s_n$ be structures over the schemes $S_1, \quad , S_n$ respectively A bijection ψ, defined over $D(s_1, \quad , s_n)$, is called *$(s_1, \quad , s_n)$-compatible* if and only if ψ, restricted to $D(s_i)$, is s_i-compatible for every s_i, $1 \leq i \leq n$ The set of these functions also forms a group under function composition and is called the cogroup of $\{s_1, \quad , s_n\}$ and denoted $CG(s_1, \quad , s_n)$

It can easily be shown that $BI(s_1, \quad , s_n) = BI(s_1 \times \quad \times s_n)$ and $CG(s_1, \quad , s_n) = CG(s_1 \times \quad \times s_2)$ which allows us the concentrate in the rest of the paper on the basic information and cogroup of a single structure

The cogroup $CG(s)$ of a structure s can be represented by a flat relation, called the *cogroup relation* let $D(s) = \{d_1, \quad , d_k\}$ The cogroup relation is a relation over the scheme D such that $\mathbf{C}(D) = \{D_1, \quad , D_k\}$, where D_j is a not previously used attribute corresponding to the domain value d_j, for every j, $1 \leq j \leq k$ The cogroup relation is defined

as $\{\psi^t | \psi^t(D_j) = \psi(d_j), 1 \leq j \leq k$ where $\psi \in CG(s)\}$†

3 2 Paredaens' Results for Flat Relations

In this section, we review the main results of [Par]

Lemma 5 [Par] Let r, r_1, and r_2 be three flat relations over schemes R, R_1, and R_2 respectively, with $CG(r) \subset CG(r_1)$ and $CG(r) \subset CG(r_2)$ For every set $X(\neq \emptyset) \subset \mathbf{C}(R_1)$, A, $B \in \mathbf{C}(R_1)$ we have

1 $CG(r) \subset CG(r_1 \cup r_2)$
2 $CG(r) \subset CG(r_1 \times r_2)$
3 $CG(r) \subset CG(\Pi_X(r_1))$
4 $CG(r) \subset CG(\sigma_{A=B}(r_1))$
5 $CG(r) \subset CG(\sigma_{A \neq B}(r_1))$

Lemma 6 [Par] The cogroup relation of a flat relation r over scheme R belongs to the basic information of r, i e , $CG(r) \in BI(r)$

The following algorithm constructs an algebraic expression for $CG(r)$

Suppose r has m tuples
Consider $u = r^m = r \times \quad \times r$ over the scheme U
There is a tuple t in u that contains all the tuples of r
We obtain $CG(r)$ by the following two constructions

(1) **For every** A, $B \in \mathbf{C}(U)$ **do**
if $t(A) = t(B)$ replace u by $\sigma_{A=B}(u)$
else replace u by $\sigma_{A \neq B}(u)$

(2) t, which still belongs to u, contains
all the elements $d_1, \quad , d_k$ of $D(r)$,
say on the attributes $D_1, \quad , D_k \in \mathbf{C}(U)$
$v \leftarrow \Pi_{\{D_1, \quad , D_k\}}(u)$

It can be shown that $v = CG(r)$

The following theorem indicates that the relational algebra is BP-complete

Theorem 1 [Par] Let r_1 and r_2 be relations Then $r_1 \in BI(r_2)$ if and only if $CG(r_2) \subset CG(r_1)$ and $D(r_1) \subset D(r_2)$

Remark 1 As part of the proof of Theorem 1, Paredaens actually shows that for any relation r, $r \in BI(CG(r))$ Since by Lemma 6, $CG(r) \in BI(r)$, we may conclude that $BI(r) = BI(CG(r))$ Furthermore, by Theorem 1, $CG(r) = CG(CG(r))$

† We will identify ψ^t with ψ

3 3 Generalizations of Paredaens' Results to Unnormalized Structures

We begin with a generalization of Lemma 5

Lemma 7 Let s, s_1, and s_2 be three structures over schemes S, S_1, and S_2 respectively, with $CG(s) \subset CG(s_1)$ and $CG(s) \subset CG(s_2)$ For every set $X(\neq \emptyset) \subset \mathbf{C}(S_1)$, M, $N \in \mathbf{C}(S_1)$, $X^* \in \mathbf{H}(S_1)$ we have

1 $CG(s) \subset CG(s_1 \cup s_2)$
2 $CG(s) \subset CG(s_1 \times s_2)$
3 $CG(s) \subset CG(\Pi_X(s_1))$
4 $CG(s) \subset CG(\sigma_{M=N}(s_1))$
5 $CG(s) \subset CG(\sigma_{M \neq N}(s_1))$
6 $CG(s) \subset CG(\mathrm{NEST}_X(s_1))$
7 $CG(s) \subset CG(\mathrm{UNNEST}_{X^*}(s_1))$
8 $CG(s) = CG(\mathrm{NEST}_X(s))$

Proof These statements can be proved in a straightforward way □

Lemma 7 provides a generalization of Lemma 5 to un normalized structures Generalizing Lemma 6 requires more work Essentially, the problem we face deals with the fact that the cogroup relation $CG(s)$ of a structure s is a flat relation, whereas s is an arbitrary unnormalized structure This will require transforming s into a flat relation, while retaining the expressiveness of s We begin with some technical results

Let s be a structure over scheme S such that $\mathbf{C}(S) = \{A_1, \quad , A_k\}$ Let $X^* \in \mathbf{H}(S)$ Let S^2 be the scheme of the structure $s \times s$, and let us assume that $\mathbf{C}(S^2) = \{A_{1,1}, \quad , A_{1\,k}, A_{2,1}, \quad , A_{2\,k}\}$, with the obvious correspondences between A_j and $A_{1,j}$ and between A_j and $A_{2,j}$, for $1 \leq j \leq k$ Without loss of generality, we assume that $A_{1,1}$ and $A_{2,1}$ correspond to X^* We define

$$\mathrm{TAG}_{X^*}(s) = \Pi_{\{A_{1\,1}, \quad , A_{1\,k}, A_{2\,1}\}}(\sigma_{A_{1\,1}=A_{2\,1}}(s \times s))$$

Clearly $s = \Pi_{\{A_{1\,1}, \quad , A_{1\,k}\}}(\mathrm{TAG}_{X^*}(s))$, which can also be written as $s = \Pi_{\mathbf{C}(S)}(\mathrm{TAG}_{X^*}(s))$ since the set $\{A_{1\,1}, \quad , A_{1\,k}\}$ corresponds to $\mathbf{C}(S)$

We now establish some properties of $\mathrm{TAG}_{X^*}(s)$

Lemma 8 Let s be a structure over S and let $X^* \in \mathbf{H}(S)$ Then

1 $CG(s) = CG(\mathrm{UNNEST}_{X^*}(\mathrm{TAG}_{X^*}(s)))$
2 $BI(s) = BI(\mathrm{UNNEST}_{X^*}(\mathrm{TAG}_{X^*}(s)))$

3 $D(s) = D(\text{UNNEST}_{X^*}(\text{TAG}_{X^*}(s)))$

In order to prove that $CG(s) \in BI(s)$, we already mentioned that we need to flatten s, while retaining its expressiveness Our first step in this process will consist in introducing an intermediate structure, which we will call ALMOSTFLAT(s)

Let $(X_1^*, \quad, X_n^*)$ be an *unnesting sequence* of S, i e ,

1 $\{X_1^*, \quad, X_n^*\} = \text{sub}(S)$, and

2 for each pair of indices i and j, $1 \leq i \leq j \leq n$, either X_i^* is a descendant of X_j^* in S or X_i^* and X_j^* are incomparable nodes in S

We define

$$\text{ALMOSTFLAT}(s) =$$

$$\text{UNNEST}_{X_1^*}(\text{TAG}_{X_1^*}(\quad (\text{UNNEST}_{X_n^*}(\text{TAG}_{X_n^*}(s))) \quad))$$

Notice that we did not parameterize ALMOSTFLAT(s) with the unnesting sequence $(X_1^*, \quad, X_n^*)$ The reason is that in subsequent lemmas or theorems, we do not rely on the choice of the unnesting sequence

We can now formulate a generalization of Lemma 8

Lemma 9 Let s be a structure over scheme S Then

1 $CG(s) = CG(\text{ALMOSTFLAT}(s))$

2 $BI(s) = BI(\text{ALMOSTFLAT}(s))$

3 $D(s) = D(\text{ALMOSTFLAT}(s))$

Proof Each of the statements 1, 2, and 3 follows from Lemma 8 by a simple recursion argument □

In addition to Lemma 9, the structure ALMOSTFLAT(s) has the following fundamental properties

Lemma 10 Let s be a structure over S and let S_f be the scheme of ALMOSTFLAT(s) Then, for any $X^* \in \text{H}(S_f)$ there exists a set $M_{X^*} \subset \text{C}(S_f)$ such that

1 for any tuple $t \in \text{ALMOSTFLAT}(s)$ and any $x \in t(X^*)$, there exists a tuple $v \in \text{ALMOSTFLAT}(s)$ such that $v[M_{X^*}] = x$ and $v(X^*) = t(X^*)$

2 for any tuple $t \in \text{ALMOSTFLAT}(s)$, $t[M_{X^*}] \in t(X^*)$

Proof Follows from the construction of ALMOSTFLAT(s) and can be shown by induction on $n = |\text{sub}(S)|$ □

We are now ready to generalize Lemma 6 to unnormalized relational structures

Lemma 11 The cogroup relation of a structure s over scheme S belongs to the basic information of s, i e , $CG(s) \in BI(s)$

Proof The proof is divided into several steps

STEP 1

Lemma 9 states that $CG(s) = CG(\text{ALMOSTFLAT}(s))$ and $BI(s) = BI(\text{ALMOSTFLAT}(s))$ which implies that $CG(s) \in BI(s)$ if and only if $CG(\text{ALMOSTFLAT}(s)) \in BI(\text{ALMOSTFLAT}(s))$ In the rest of this proof, we show that indeed $CG(\text{ALMOSTFLAT}(s)) \in BI(\text{ALMOSTFLAT}(s))$

STEP 2

The construction of ALMOSTFLAT(s) and its properties allows us to use an algorithm similar to Paredaens' algorithm to construct an expression of the extended relational algebra for $CG(\text{ALMOSTFLAT}(s))$ starting from ALMOSTFLAT(s)

Paredaens' algorithm takes as input a flat relation r and constructs $CG(r)$ Since ALMOSTFLAT(s) is not in general a flat relation, however, we will have to use a technique for interpreting it as if it was a flat relation This technique consists in interpreting the children of S_f (S_f is the scheme of ALMOSTFLAT(s)) as the attributes of another scheme, denoted S_{flat} More precisely, S_{flat} is defined such that

1 $\text{C}(S_{flat}) = \text{C}(S_f)$, and

2 $\text{A}(S_{flat}) = \text{C}(S_{flat})$, i e , S_{flat} is a flat scheme

We will use the notation FLAT(s) when we interpret ALMOSTFLAT(s) as a flat relation over S_{flat} Now run Paredaens' algorithm on FLAT(s) The algorithm returns an expression of the extended relation algebra for $CG(\text{FLAT}(s))$ Since in general ALMOSTFLAT(s) is not a flat relation, its domain $D(\text{ALMOSTFLAT}(s))$ is a subset of $D(\text{FLAT}(s))$, the domain of FLAT(s) It should therefore be clear that the set $CG(\text{FLAT}(s))$ is not the set of functions we are interested in, we are interested in $CG(\text{ALMOSTFLAT}(s))$ which consists of functions defined over a smaller domain than those in $CG(\text{FLAT}(s))$ We will show, however, that $CG(\text{ALMOSTFLAT}(s))$ is the set of functions obtained by restricting the functions of $CG(\text{FLAT}(s))$ to $D(\text{ALMOSTFLAT}(s))$

NOTATION

In the rest of this proof, we will use the following notation

i m is the number of tuples of FLAT(s) and therefore

also the number of tuples of ALMOSTFLAT(s)

ii t is the tuple selected from FLAT(s)^m which contains all the tuples of FLAT(s) and which is used to drive step 1 of Paredaens' algorithm Since there is a one-to-one correspondence between the tuples of FLAT(s) and the tuples of ALMOSTFLAT(s), we sometimes regard t as a tuple containing all the tuples of ALMOSTFLAT(s)

iii u denotes the structure obtained after running step 1 of Paredaens' algorithm U denotes the scheme of u

STEP 3

The following three properties can be easily verified

1 The tuple t belongs to u

2 For every $\psi \in CG(\text{FLAT}(s))$, we have $\psi(u) = u$

3 For every tuple $v \in u$, v contains all the tuples of FLAT(s) (and therefore also all the tuples of ALMOSTFLAT(s))

STEP 4

Let $N_f \in \mathbf{H}(S_f)$ By the construction of u, we know that there exists m attributes $N_1, \ldots, N_m$ in U corresponding to N_f Let $N \in \{N_1, \ldots, N_m\}$ and let v be a tuple of u Since v contains all the tuples of ALMOSTFLAT(s) (see STEP 3 item 3), there exists a tuple $v_f \in$ ALMOSTFLAT(s) such that $v(N) = v_f(N_f)$ We know, by Lemma 10 that for any N_f, there exists a set $\mathcal{M}_{N_f} \subset \mathbf{C}(S_f)$ which only depends on N_f such that

1 for any tuple $v_f \in$ ALMOSTFLAT(s) and any $x_f \in v_f(N_f)$, there exists a tuple $w_f \in$ ALMOSTFLAT(s) such that $w_f[\mathcal{M}_{N_f}] = x_f$ and $w_f(N_f) = v_f(N_f)$

2 for any tuple $v_f \in$ ALMOSTFLAT(s), $v_f[\mathcal{M}_{N_f}] \in v_f(N_f)$

Now let $v \in u$ Since v contains all the tuples of ALMOSTFLAT(s), we can interpret items 1 and 2 also in the following way Let $\mathcal{M}_{N_1}, \ldots, \mathcal{M}_{N_m}$ be the subsets of $\mathbf{C}(U)$ corresponding to $\mathcal{M}_{N_f}$ We have that for any $N \in \{N_1, \ldots, N_m\}$

1' For every $x \in v(N)$, there exists a $N_x \in \{N_1, \ldots, N_m\}$ such that $v[\mathcal{M}_{N_x}] = x$ and $v(N) = v(N_x)$

2' $v[\mathcal{M}_N] \in v(N)$

STEP 5

Let $\psi \in CG(\text{FLAT}(s))$, $v \in u$, and $N \in \{N_1, \ldots, N_m\}$ By STEP 3 item 2, we know that $\psi \circ v \in u$ We claim that

$$\psi \circ v(N) = \{\psi \circ x | x \in v(N)\}$$

We first show that $\{\psi \circ x | x \in v(N)\} \subset \psi \circ v(N)$ Let $x \in v(N)$ By STEP 4 (item 1'), there exists $N_x \in \{N_1, \ldots, N_m\}$ such that $v[\mathcal{M}_{N_x}] = x$ and $v(N) = v(N_x)$ Hence $\psi \circ v[\mathcal{M}_{N_x}] = \psi \circ x$ and $\psi \circ v(N) = \psi \circ v(N_x)$ Since $\psi \circ v \in u$ and $N_x \in \{N_1, \ldots, N_m\}$, we have by STEP 4 (item 2') $\psi \circ v[\mathcal{M}_{N_x}] \in \psi \circ v(N_x)$ Hence $\psi \circ x \in \psi \circ v(N)$ Therefore $\{\psi \circ x | x \in v(N)\} \subset \psi \circ v(N)$

We now show that $\psi \circ v(N) \subset \{\psi \circ x | x \in v(N)\}$ Since $CG(\text{FLAT}(s))$ is a group, ψ has an inverse ψ^{-1} Since $\psi \circ v \in u$, a similar argument as the one made in the previous paragraph can be made for ψ^{-1} and the tuple $\psi \circ v$ It follows that $\{\psi^{-1} \circ y | y \in \psi \circ v(N)\} \subset \psi^{-1} \circ \psi \circ v(N)$, which is of course equal to $v(N)$ This further implies that $\{\psi \circ \psi^{-1} \circ y | y \in \psi \circ v(N)\} \subset \{\psi \circ x | x \in v(N)\}$ Hence $\{y | y \in \psi \circ v(N)\}$ which is equal to $\psi \circ v(N) \subset \{\psi \circ x | x \in v(N)\}$

STEP 6

We are now in a position to construct $CG(\text{ALMOSTFLAT}(s))$ Consider the tuple t (see NOTATION) Since $D(\text{ALMOSTFLAT}(s)) \subset D(\text{FLAT}(s))$, there exists a set of attributes $\mathcal{E} \subset \mathbf{C}(U)$ such that

1 For any element $d \in D(\text{ALMOSTFLAT}(s))$, there exists a unique element $D \in \mathcal{E}$ such that $t(D) = d$

2 For any element $D \in \mathcal{E}$, $t(D) \in D(\text{ALMOSTFLAT}(s))$

We can show, using STEP 5, that $CG(\text{ALMOSTFLAT}(s)) = \Pi_{\mathcal{E}}(CG(\text{FLAT}(s)))$ □

Before we proceed to the main result of this paper, we would like to summarize the above proof by providing the generalized version (we will call it version 2) of Paredaens' algorithm

The following algorithm constructs an algebraic expression for $CG(s)$

(1) Construct ALMOSTFLAT(s)

Suppose ALMOSTFLAT(s) has m tuples

Consider u = ALMOSTFLAT(s)^m defined over the scheme U

There is a tuple t in u that contains all the tuples of ALMOSTFLAT(s)

We obtain $CG(s)$ by the following two constructions

(2) **For every** $N, M \in \mathbf{C}(U)$ **do**

if $t(N) = t(M)$ replace u by $\sigma_{N=M}(u)$

else replace u by $\sigma_{N \neq M}(u)$

(3) t, which still belongs to u, contains all the elements $e_1, \ldots, e_k$ of $D(s)$, say on the attributes $E_1, \ldots, E_k \in \mathbf{C}(U)$

$v \leftarrow \Pi_{(E_1, \ldots, E_k)}(u)$

We showed in the proof of Lemma 11 that $v = CG(s)$

We can now state the main result

Theorem 2 Let s_1 and s_2 be structures Then $s_1 \in BI(s_2)$ if and only if $CG(s_2) \subset CG(s_1)$ and $D(s_1) \subset D(s_2)$

Proof A recursive application of Lemma 7 proves the implication from left to right

We shall now prove the implication from right to left The proof is divided into several steps

STEP 1

In this step, we show that if we know that for any structure s, $s \in BI(CG(s))$, the right to left implication follows Thus, assume we know that for any structure s, $s \in BI(CG(s))$ We then have the following implications

- If $CG(s_2) \subset CG(s_1)$ and $D(s_1) \subset D(s_2)$ (notice that $CG(s_1)$ and $CG(s_2)$ are flat relations), it follows from Theorem 1 and Remark 1 that $CG(s_1) \in BI(CG(s_2))$
- Since, by our assumption, $s_1 \in BI(CG(s_1))$, it follows that $s_1 \in BI(CG(s_2))$
- Since by Lemma 11, $CG(s_2) \in BI(s_2)$, it follows that $s_1 \in BI(s_2)$ as was to be shown

STEP 2

In this step, we show that the problem of showing that $s \in BI(CG(s))$ can be reduced to another problem We first review version 2 of Paredaens' algorithm Let u be the structure obtained after finishing step 2 of this algorithm It was shown in [Par] that ALMOSTFLAT(s) $\in BI(u)$ By Lemma 9, we know that $s \in BI$(ALMOSTFLAT(s)) Thus $s \in BI(u)$ If we can show that $u \in BI(CG(s))$, we will have shown that $s \in BI(CG(s))$ In the rest of this proof, we show that indeed $u \in BI(CG(s))$

STEP 3

We first need to establish some additional properties of the structure u

i Consider the tuple t selected at the end of step 1 of version 2 of Paredaens' algorithm We can show that for any pair of tuples $v_1, v_2 \in u$, there exists a $\psi \in CG(s)$ such that $\psi^e \circ v_1 = v_2$

ii We will use the notation of STEP 4 of the proof of Lemma 11 We showed there that for each tuple $v \in u$ and each $N \in \mathbf{C}(U)$ such that height$(N) > 1$

1 For every $x \in v(N)$, there exists a $N_x \in \{N_1, \ldots, N_m\}$ such that $v[\mathcal{M}_{N_x}] = x$ and $v(N) = v(N_x)$ We will denote the set $\{M \in \mathbf{C}(U) |$ there exists an $x \in v(N)$ such that $v[\mathcal{M}_M] = x$ and $v(N) = v(M)\}$ by $\mathcal{N}(v, N)$ and the set $\{\mathcal{M}_M | M \in \mathcal{N}(v, N)\}$ by $\mathcal{M}(v, N)$ Clearly, for every $x \in v(N)$, $N_x \in \mathcal{N}(v, N)$

2 $v[\mathcal{M}_N] \in v(N)$

We can show that for each pair of tuples $v_1, v_2 \in u$ and each $N \in \mathbf{C}(U)$ such that height$(N) > 1$, $\mathcal{N}(v_1, N) = \mathcal{N}(v_2, N)$ and $\mathcal{M}(v_1, N) = \mathcal{M}(v_2, N)$ The proof of this fact relies heavily upon i

iii Define $P(N) = \bigcup_{M \in \mathcal{N}(N)} \mathcal{M}_M$ It follows from ii that for any tuple $v \in u$, $v(N) = \{v[\mathcal{M}_M] | M \in \mathcal{N}(N)\}$ As a simple consequence it follows that u satisfies the FD $P(N) \rightarrow N$

iv Suppose height$(U) = n+1$ We define for each i such that $1 \leq i \leq n$ the set LEVEL$_i = \{M \in \mathbf{C}(U) |$height$(M) \leq i\}$ Now let i be such that $1 \leq i < n$ and let $N \in \mathbf{C}(U)$ be such that height$(N) = i + 1$ and let $v_1, v_2 \in u$ We can show that if $v_1[\text{LEVEL}_i - P(N)] = v_2[\text{LEVEL}_i - P(N)]$ then $v_1(N) = v_2(N)$ The proof of this fact relies on iii and the properties of version 2 of Paredaens' algorithm

STEP 4

We define for all i, $1 \leq i \leq n$, $u_i = \Pi_{\text{LEVEL}_i}(u)$ Clearly,

1 $u_i = \Pi_{\text{LEVEL}_i}(u_{i+1})$ for $1 \leq i < n$

2 $u_n = u$

Remember that our goal is to show that $u \in BI(CG(s))$ We can decompose this problem into showing that

1 $u_1 \in BI(CG(s))$ and

2 for any i, $1 \leq i < n$, $u_{i+1} \in BI(u_i)$

If we can show these two items, it follows from the fact that $u_n = u$, that $u \in BI(CG(s))$

We first show that $u_1 \in BI(CG(s))$ Since u_1 is a flat relation and $D(u_1) = D(CG(s))$ it follows from Theorem 1 that we have to show that $CG(CG(s)) \subset CG(u_1)$ Since $CG(CG(s)) = CG(s)$, we have to show that $CG(s) \subset CG(u_1)$ Since $u \in BI(s)$ and $u_1 \in BI(u)$, we have that $u_1 \in BI(s)$ It follows from the left to right implication of this theorem that $CG(s) \subset CG(u_1)$ as was to be shown

We now show that for any i, $1 \leq i < n$, $u_{i+1} \in BI(u_i)$ Let $N_{i+1} = \{N \in C(U) | \text{height}(N) = i+1\}$ Clearly, $N_{i+1} = \text{LEVEL}_{i+1} - \text{LEVEL}_i$

Induction Hypothesis For each set $X \subset N_{i+1}$, $0 \leq |X| < k$, $\Pi_{\text{LEVEL}_i \cup X}(u_{i+1}) \in BI(u_i)$

Basis If $|X| = 0$, $X = \emptyset$, hence $\Pi_{\text{LEVEL}_i \cup X}(u_{i+1}) = \Pi_{\text{LEVEL}_i}(u_{i+1}) = u_i$ which is obviously an element of $BI(u_i)$

Induction Step Let $X \subset N_{i+1}$ such that $|X| = k$ Let $N \in X$ and let $X' = X - \{N\}$ Since $X' \subset N_{i+1}$ and $|X'| = k - 1$, we have by the induction hypothesis that $\Pi_{\text{LEVEL}_i \cup X'}(u_{i+1}) \in BI(u_i)$ We now show that $\Pi_{\text{LEVEL}_i \cup X}(u_{i+1}) \in BI(\Pi_{\text{LEVEL}_i \cup X'}(u_{i+1}))$

The results obtained in STEP 3, in particular item iv, imply that the following algebraic expressions obtain $\Pi_{\text{LEVEL}_i \cup X}(u_{i+1})$ from the structure $\Pi_{\text{LEVEL}_i \cup X'}(u_{i+1})$

Let

a $temp_1 = \Pi_{\text{LEVEL}_i \cup X'}(u_{i+1})$

b $temp_2 = \text{NEST}_{P(N)}(temp_1)$

c $temp_3 = \text{TAG}_{P(N)^*}(temp_2)$

d $temp_4 = \text{UNNEST}_{P_1(N)^*}(temp_3)$ ($P_1(N)^*$ correspond to $P(N)^*$ after tagging $temp_3$)

e $temp_5 = \bigcup_{M \in M(N)} \Pi_M(temp_4)$

f $temp_6 = \text{UNNEST}_{P(N)^*}(\text{NEST}_M(temp_5))$

It can be seen that $temp_6 = \Pi_{\text{LEVEL}_i \cup X}(u_{i+1})$ This shows that $\Pi_{\text{LEVEL}_i \cup X}(u_{i+1}) \in BI(\Pi_{\text{LEVEL}_i \cup X'}(u_{i+1}))$ If we choose $X = N_{i+1}$, we have that $u_{i+1} \in BI(u_i)$ as was to be shown This completes the proof of Theorem 2 □

4 Conclusion

In this paper, we showed that the extended relational algebra is BP-complete This result generalizes a well-known result concerning the BP-completeness of the relational algebra It suggests that the extended relational algebra is to the unnormalized relational model what the relational algebra is to the relational model Whereas it was previously shown that the URDM is a natural practical extension of the relational model, this result shows that it is also a natural theoretical extension

On the other hand, however, we realize that BP-completeness is not the strongest notion of the completeness of a query languages for a database model Chandra and Harel [CH] have introduced a stronger notion of completeness and constructed a language for the relational model which is complete in that sense We conjecture that this result can also be extended to unnormalized relational models, as well as to other data models, and plan to study this problem

5 References

[AB] S Abiteboul, N Bidoit, "Non First Normal Form Relations to Represent Hierarchically Organized Data", Proc Third ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, 1984, 191-200

[AH] A V Aho, J D Ullman, "Universality of Data Retrieval Languages", Proc 6th ACM Symposium on Principles of Programming Languages, San-Antonio, Texas, Jan 1979, 110-117

[Ban] F Bancilhon, "On the Completeness of Query Languages for relational Data Bases", Proceedings 7th Symposium on Mathematical Foundations of Computer Science, Zakopane, Poland, sept 1978, Lecture Notes in Computer Science, Springer-Verlag, Berlin/New York/Heidelberg

[BRS] F Bancilhon, P Richard, M Scholl, "On Line Processing of Compacted Relations", Proc 8th Int'l Conf on Very Large Data Bases, 1982, 263-269

[CH] A Chandra, D Harel, "Computable Queries for Relational Data Bases", *Journal of Computer and System Sciences 21*, 1980, pp 156-178

[CT] J Clifford, A Tansel, "On an Algebra For Historical Relational Databases Two Views", Proc ACM SIGMOD Int'l Conf on Management of Data, 1985, 247-

265

[Cod] E F Codd, "A Relational Model for Large Shared Data Banks", *Comm ACM 13*,6 (June 1970), 377-387

[FSTV] P C Fischer, L V Saxton, S J Thomas, D Van Gucht, "Interactions Between Dependencies and Nested Relational Structures", *J Computer System Sciences 31* (1985), 343-354

[FT] P C Fischer, S J Thomas, "Operators for Non-First-Normal Form Relations", Proc IEEE Computer Software and Applications Conference, 1983, 464-475

[FV1] P C Fischer, D Van Gucht, "Weak Multivalued Dependencies", Proc Third ACM SIGACT-SIGMOD Symposium on Principles of Database Systems", 1984, 266-274

[FV2] P C Fischer, D Van Gucht, "Structure of Relations Satisfying Certain Families of Dependencies", *Proc 2nd Symposium on Theoretical Aspects of Computer Science*, K Mehlhorn, Ed , Springer-Verlag, Berlin, 1985, 132-142

[FV3] P C Fischer, D Van Gucht, "Determining When a Structure is a Nested Relation", Proc 11th Int'l Conf on Very Large Data Bases, 1985, 171-180

[FVT] P C Fischer, D Van Gucht, S J Thomas, "Some Principles and Uses of Nested Relational Structures", Technical Report CS-84-20, Vanderbilt University, 1984

[FA] J Freitag, H J Appelrath, "Modelling IR by S-NF^2 Relations", *Computing 85 A Broad Perspective of Current Developments*, G Bucci and G Valle (eds), Elsevier Science Publishers, B V (North Holland) 1985

[Gir] M Girkar, "Unnormalized Relational Structures with Nulls", Technical Report CS-86-07, Vanderbilt University, 1986

[Gon] G Gonnet, "Unstructured Data Bases", Proc Second ACM SIGACT SIGMOD Symposium on Principles of Database Systems, 1983, 117-124

[HY] R Hull, C K Yap, "The Format Model", *Journal of the ACM 31*, 3 (July 1984)

[Kor] H F Korth, "Extending the Scope of Relational Languages" *IEEE Software* (January 1986)

[Jac] B E Jacobs, "On Database Logic", *Journal of the ACM 29*, 2 (April 1982), 333-362

[Imi] Private communications

[JS] G Jaeschke, H -J Schek, "Remarks on the the Algebra on Non First Normal Form Relations", Proc ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, 1982, 124-138

[KTT] Y Kambayashi, K Tanaka, K Takeda, "Synthesis of Unnormalized Relations Incorporating More Meaning", *Information Sciences 29* (1983), 201-247

[KV] G M Kuper, M Y Vardi, "A New Approach to Database Logic", Proc Third SCM SIGACT-SIGMOD Symposium on the Principles of Database Systems, 1984, 86-96

[Mac] I A Macleod, "A Model for Integrated Information Systems", Proc Ninth Int'l Conf on Very Large Data Bases, 1983, 280-289

[Mak] A Makinouchi, "A Consideration of Normal Form of Not-Necessarily-Normalized Relations in the Relational Data Model", Proc 5th Int'l Conf on Very Large Data Bases, 1977, 447-453

[OMO] G Ozsoyoglu, V Matos, Z M Ozsoyoglu, "Extending Relational Algebra and Relational Calculus for Set-Valued Attributes and Aggregate Functions", Tech Report, Case Western Reserve University, 1983

[OY] Z M Ozsoyoglu, L Y Yuan, "A Normal form for Nested Relations", Proc Fourth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, 1985, 251-260

[Par] J Paredaens, "On the Expressive Power of the Relational Algebra", *Information Processing Letters 7*, 2 (Feb 1978), 107-111

[RKB] M A Roth, H F Roth, D S Batory, "SQL/NF A Query Language for ¬1NF Relational Databases", Tech Report TR-84-36, University of Texas at Austin, 1984

[RKS1] M A Roth, H F Korth, A Silberschatz, "Theory of Non-First-Normal-Form Relational Databases", Tech Report TR-84-36 (Revised January 1986), University of Texas at Austin, 1984

[RKS2] M A Roth, H F Korth, A Silberschatz, "Null Values in ¬1NF Relational Databases", Tech Report TR-84-32, University of Texas at Austin, 1985

[Sche] H J Schek, "Towards a Basic Relational NF^2 Algebra Processor", Proc of the International Conference of Data Organization, Kyoto, Japan, 1985, 173 182

[Scho] M H Scholl, "Theoretical Foundation of Algebraic Optimization Utilizing Unnormalized Relations", Proceedings of the International Conference on Database Theory, Rome, Italy, (Sept 1986)

[SP] H -J Schek, P Pistor, "Data Structures for an Integrated Data Base Management and Information Retrieval System", Proc 8th Int'l Conf on Very Large Data Bases, Mexico, 1982, 197-207

[SS] H -J Schek, M H Scholl, "An Algebra for the Relational Model with Relation-Valued Attributes", TR DVSI-1984-T1, Technical University of Darmstadt, West Germany, 1984

[Tho] S J Thomas, "A Non-First-Normal Form Relational Database Model", Ph D Dissertation, Vanderbilt University, 1983

[TF] S J Thomas, P C Fischer, "Nested Relational Structures", *Theory of Databases*, P C Kanellakis, Ed , JAI Press, 1985, to appear

[V] D Van Gucht, "Theory of Unnormalized Relational Structures", Ph D Dissertation, Vanderbilt University, 1985

[VF1] D Van Gucht, P C Fischer, "MVDs, Weak MVDs and Nested Relational Structures", Technical Report CS-84-19, Vanderbilt University, 1984

[VF2] D Van Gucht, P C Fischer, "Some Classes of Multilevel Relational Structures", Proc of the Fifth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, 1986, 60-69, submitted for publication