JOURNAL OF COMPUTER AND SYSTEM SCIENCES **29**, 338–358 (1984)

GYO Reductions, Canonical Connections, Tree and Cyclic Schemas, and Tree Projections

NATHAN GOODMAN, ODED SHMUELI, AND Y. C. TAY

Aiken Computation Laboratory, Harvard University, Cambridge, Massachusetts 02138

Received November 3, 1983; revised June 27, 1984

Database schemas may be partitioned into two subclasses: three schemas and cyclic schemas. The analysis of tree vs . cyclic schemas introduced the concepts of GYO reductions, canonical connections, and tree projections. This paper investigates the intricate relationships among these concepts in the context of universal relation databases.

1. INTRODUCTION

A universal relation database (UR database) is a collection of relations that are projections of a single "universal" relation. This paper explores connections among four concepts of database theory applied to UR databases:

(1) the so-called *Graham reductions* of Graham [10], and Yu and Oszoyoglu [19];

(2) the *canonical connections* of Maier and Ullman [16];

(3) *tree* and *cyclic schemas* of Beeri *et al.* [3, 4, 8] and Bernstein, Chiu, Goodman, and Shmueli [5, 6, 11–14]; and

(4) the *tree projections* of Goodman and Shmueli [13].

This paper gives a coherent framework for viewing results in [3–5, 7, 8, 10–16, 18].

Sections 2 and 3 give terminology and define the above concepts. Sections 4–6 use these concepts to analyze three basic problems in UR database theory:

(1) Computing $\prod_X(\bowtie_{R\in D} R)$ over database D using joins followed by a single project.

(2) Determining whether a "sub-database" has a lossless join.

(3) Computing $\prod_X(\bowtie_{R\in D} R)$ using joins, semijoins, and projects in any combination.

We also treat some aspects of γ-acyclicity, a type of tree schema characterized by Fagin [7].

0022-0000/84 $3.00

2. Terminology

A *relation schema* is a set of *attributes* and a *database schema* (or simply *schema*) is a multiset of relation schemas. (All sets and multisets in this paper are finite.) A *relation state* R for relation schema $\mathbf{R}$ is a relation over the attributes of $\mathbf{R}$; a *database state* for schema $\mathbf{D}$ is an assignment of relation states to the relation schemas of $\mathbf{D}$. We use $\mathbf{D}=(\mathbf{R}_1,..., \mathbf{R}_n)$ to denote a database schema and $D=(R_1,..., R_n)$ for a corresponding state. If for all $\mathbf{R}' \in \mathbf{D}'$, there exists $\mathbf{R} \in \mathbf{D}$ such that $\mathbf{R}' \subseteq \mathbf{R}$, we denote this by $\mathbf{D}' \leqslant \mathbf{D}$. $\mathbf{D}$ is *reduced* if no relation schema in $\mathbf{D}$ is a subset of another relation schema in $\mathbf{D}$. The *reduction* of $\mathbf{D}$ is the schema obtained by eliminating relation schemas that are subsets of other. $\bigcup(\mathbf{D})=\bigcup_{i=1}^{n} \mathbf{R}_i$ denotes the attributes of $\mathbf{D}$.

As usual, we use $\bowtie$ for natural join, $\prod_X$ for projection onto attribute set X, and $\ltimes$ for natural semijoin. ($R \ltimes S \triangleq \prod_{\mathbf{R}}(R \bowtie S)$.) We use $Q=(\mathbf{D}, X)=\prod_X(\bowtie_{i=1}^{n} \mathbf{R}_i)$ to denote the natural join query with target X, i.e., Q applied to a state D for $\mathbf{D}$ is $Q(D)=\prod_X(\bowtie_{i=1}^{n} R_i)$.

We shall only consider *universal databases*, i.e., databases of the form $D=\{\prod_{\mathbf{R}} I \mid \mathbf{R} \in \mathbf{D}\}$, where I is a universal relation. Q is *weakly contained* in Q', $Q \subseteq Q'$, if for all universal databases $Q(D) \subseteq Q'(D)$. Q is *weakly equivalent* to Q', $Q \equiv Q'$, if for all such databases, $Q \subseteq Q'$ and $Q' \subseteq Q$.

3. Key Concepts

3.1. *Tree and Cyclic Schemas*

A *qual graph* for $\mathbf{D}$ is an undirected graph whose nodes are in one–one correspondence with the relation schemas of $\mathbf{D}$, such that for each $A \in \bigcup(\mathbf{D})$, the subgraph induced by the nodes whose corresponding relation schemas contain A is connected [6]. $\mathbf{D}$ is a *tree schema* if some qual graph for it is a tree; else $\mathbf{D}$ is a *cyclic schema*. See Fig. 1. (These are called acyclic and cyclic schemes and hypergraphs in much of the literature.)

The following fact is useful. Let T be a qual tree for $\mathbf{D}$. Let r and s be nodes in T and p a node along the path in T from r to s. Let $\mathbf{R}$, $\mathbf{S}$, and $\mathbf{P}$ be the relation schemas corresponding to r, s, and p, respectively. If $A \in \mathbf{R} \cap \mathbf{S}$, then $A \in \mathbf{P}$. We call this property *attribute connectivity*.

Let $U=\{A_1,..., A_n\}$, $n>2$. The schema $\mathbf{D}=(\{A_1, A_2\}, \{A_2, A_3\},..., \{A_{n-1}, A\}, \{A_n, A_1\})$ is called an *Aring* of *size n*. The schema $D=(U-\{A_1\}, U-\{A_2\},..., U-\{A_n\})$ is called an *Aclique* of *size n*. (Any schema isomorphic to an Aring or an Aclique is an Aring or Aclique simply by appropriately ordering the attributes.) Arings and Acliques are the "building blocks" of cyclic schemas in the following sense.

Lemma 3.1 [12]. *Schema* $\mathbf{D}$ *is cyclic iff there exists* $X \subseteq \bigcup(\mathbf{D})$ *such that*

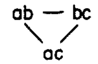
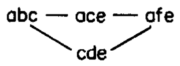
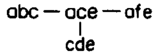
C	A qual graph representing C	Type
(ab,bc,cd)	ab — bc — cd	tree
(ab,bc,ac)		this is the only qual graph for C; so C is cyclic
(abc,cde,ace,afe)		C is a tree schema since  is also a qual graph for C.

FIG. 1. Tree and cyclic schemas: use a, b, c,..., for attributes, concatenate elements to denote sets, and identify nodes of a qual graph with the corresponding sets.

eliminating subset and duplicate relation schemas from $\mathbf{D}' = (\mathbf{R} - X \mid \mathbf{R} \in \mathbf{D})$, *results in an Aring or an Aclique.* (*See Fig.* 2.)

In particular, Arings and Acliques are cyclic (let $X = \varnothing$).

3.2. *Tree Projection*

Let $\mathbf{D} \leqslant \mathbf{D}'' \leqslant \mathbf{D}'$. $\mathbf{D}''$ is a *tree projection* of $\mathbf{D}'$ wrt $\mathbf{D}$, written $\mathbf{D}'' \in \mathrm{TP}(\mathbf{D}', \mathbf{D})$, if $\mathbf{D}''$ is a tree schema [13]. Let $Q = (\mathbf{D}, X)$. $\mathbf{D}''$ is a *tree projection* of $\mathbf{D}'$ wrt Q, denoted $\mathbf{D}'' \in \mathrm{TP}(\mathbf{D}', Q)$ if $\mathbf{D}'' \in \mathrm{TP}(\mathbf{D}', \mathbf{D} \bigcup (X))$.

EXAMPLE. $\mathbf{D} = (ab, bc, cd, de, ef, fg, gh, ha)$,

$\mathbf{D}'' = (ab, abch, cdgh, defg, ef)$,

$\mathbf{D}' = (abef, abch, cdgh, defg, e)$.

Clearly, $\mathbf{D} \leqslant \mathbf{D}'' \leqslant \mathbf{D}'$. Also, $\mathbf{D}''$ is a tree schema, viz., $ab - abch - cdgh - defg - ef$. Hence, $\mathbf{D}''$ is a tree projection of $\mathbf{D}'$ wrt $\mathbf{D}$. One can show that both $\mathbf{D}$ and $\mathbf{D}'$ are cyclic schemas.

3.3. *GYO Reductions*

Let $\mathbf{D}$ be a schema and $X \subseteq \bigcup(\mathbf{D})$. Consider the following operations.

(1) (Isolated attribute deletion.) Eliminate an attribute $A \notin X$ which belongs to exactly one relation schema of $\mathbf{D}$. (Hence, X consists of "sacred nodes" that are never eliminated.)

(2) (subset elimination.) Eliminate a relation schema in $\mathbf{D}$ contained in another relation schema.

$\mathbf{D}'$ is a *partial GYO reduction* of $\mathbf{D}$ wrt X, denoted $\mathbf{D}' \in \mathrm{pGR}(\mathbf{D}, X)$, if $\mathbf{D}'$ can be obtained from $\mathbf{D}$ by a sequence of zero or more of the above operations. It is easy to show [12] that (1) and (2) preserve schema *type*, i.e., $\mathbf{D}$ and $\mathbf{D}'$ are either both tree schemas or both cyclic schemas. $\mathbf{D}'$ is the *GYO reduction* of $\mathbf{D}$ wrt X, denoted

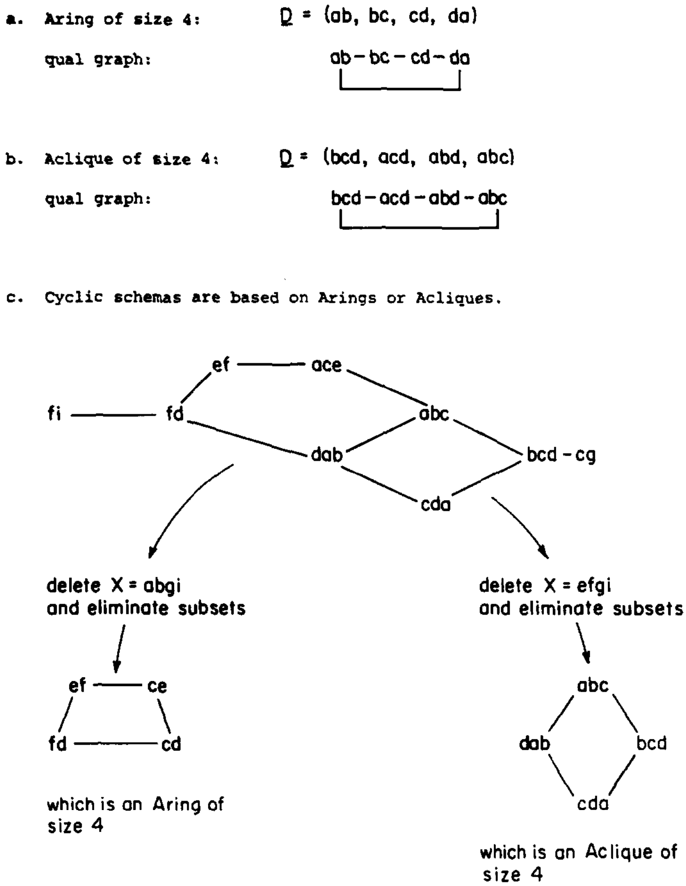


FIG. 2. Arings and Acliques (using the notation of Fig. 1).

$\mathbf{D}' = \mathrm{GR}(\mathbf{D}, X)$, if neither operation has any effect on $\mathbf{D}'$. (This is usually called Graham reduction in the literature. We propose the name GYO reduction in recognition of the early work by Yu and Ozsoyoglu on this problem [10, 19].) If $X = \varnothing$, we write $\mathrm{GR}(\mathbf{D})$ for $\mathrm{GR}(\mathbf{D}, X)$. Maier and Ullman have proved that $\mathrm{GR}(\mathbf{D}, X)$ is unique and reduced [16].

3.4. *Canonical Connections*

Let $\mathrm{Tab}(\mathbf{D}, X)$ denote the "standard" tableau [2] for the query $(\mathbf{D}, X)$; i.e., $\mathrm{Tab}(\mathbf{D}, X)$ is a tableau with rows $r_1, \ldots, r_n$ such that

(i) $\prod_A (r_i) =$ the distinguished variable a iff $A \in \mathbf{R}_i \cap X$.

(ii) $\prod_A (r_i) =$ the nondistinguished variable a' iff $A \in \mathbf{R}_i - X$.

(iii) All other entries of r_i are unique nondistinguished variables.

(iv) Entry A of the summary is a if $A \in X$, and blank otherwise.

If T and T' are tableaux with the same distinguished variables, and the rows of T are rows of T', then T is a *subtableau* of T'. If T has less rows, it is a *proper* subtableau.

A containment mapping is a row-to-row mapping which is induced by a symbol-to-symbol mapping and preserves distinguished variables [2].

T and T' are *equivalent*, denoted $T \equiv T'$, if there is a containment mapping from T to T' and a containment mapping from T' to T. Since a composition of containment mapping is a containment mapping, tableau equivalence is transitive. T and T' are *isomorphic*, denoted $T \simeq T'$, if there is a one-to-one correspondence between their rows which is a containment mapping in both directions. If $T' \equiv \text{Tab}(\mathbf{D}, X)$ and T' is not equivalent to a tableau with fewer rows, T' is said to be a *minimal tableau* for $(\mathbf{D}, X)$.

LEMMA 3.2 [2]. $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$ *iff* $\text{Tab}(\mathbf{D}, X) \equiv \text{Tab}(\mathbf{D}', X)$.

Given any $\text{Tab}(\mathbf{D}, X)$, we can construct a *canonical schema*, $\text{CS}(\mathbf{D}, X)$, as follows. For each row $r_i \in \text{Tab}(\mathbf{D}, X)$, construct the relation schema $\mathbf{R}_i = \{A \mid \prod_A(r_i) = a$ or $\prod_A(r_i) = \prod_A(r_j)$ for some $r_j \neq r_i\}$. Then $\text{CS}(\mathbf{D}, X)$ is the reduction of the schema

$$(\mathbf{R}_i \text{ as constructed above} \mid r_i \in \text{Tab}(\mathbf{D}, X)).$$

The following lemma is an immediate consequence of the above definitions.

LEMMA 3.3. (i) *If* $\text{Tab}(\mathbf{D}, X) \simeq \text{Tab}(\mathbf{D}', X)$, *then* $\text{CS}(\mathbf{D}, X) = \text{CS}(\mathbf{D}', X)$.

(ii) *If* $\text{CS}(\mathbf{D}, X) = \text{CS}(\mathbf{D}', X)$, *then* $\text{Tab}(\mathbf{D}, X) \equiv \text{Tab}(\mathbf{D}', X)$.

LEMMA 3.4 [2]. *If T and T' are equivalent tableaux, and neither is equivalent to a tableau with fewer rows, than $T \simeq T'$.*

By Lemma 3.4, two minimal tableaux for $(\mathbf{D}, X)$ are isomorphic, and so by Lemma 3.3(i) the two have the same canonical schemas. This unique schema is called the canonical connection of $(\mathbf{D}, X)$, denoted $\text{CC}(\mathbf{D}, X)$ [16].

LEMMA 3.5. $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$ *iff* $\text{CC}(\mathbf{D}, X) = \text{CC}(\mathbf{D}', X)$.

Proof. ($\Rightarrow$) By Lemma 3.2, $\text{Tab}(\mathbf{D}, X) \equiv \text{Tab}(\mathbf{D}', X)$. Let T and T' be minimal tableaux for $(\mathbf{D}, X)$ and $(\mathbf{D}', X)$, respectively. Then $T \equiv T'$ and by Lemma 3.4, $T \cong T'$. By Lemma 3.3(i), $\text{CC}(\mathbf{D}, X) = \text{CC}(\mathbf{D}', X)$.

($\Leftarrow$) By Lemma 3.3(ii), $T \equiv T'$, where T is a minimal tableau for $(\mathbf{D}, X)$ and T' is a minimal tableau for $(\mathbf{D}', X)$. Hence, $\text{Tab}(\mathbf{D}, X) \equiv \text{Tab}(\mathbf{D}', X)$ and by Lemma 3.2, $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$. ∎

3.5. *Some Connections among These Concepts*

GYO reductions and tree schemas are closely related. Let $\mathbf{D}$ be a tree schema, $X \subseteq \bigcup(\mathbf{D})$ and $\mathbf{D}' \in \mathrm{pGR}(\mathbf{D}, X)$. Since GYO reduction preserves schema type, $\mathbf{D}'$ is a tree schema. Let T' be a qual tree for $\mathbf{D}'$.

LEMMA 3.6. *Define* $\mathbf{D}$, $\mathbf{D}'$, *and* T' *as above. There exists a qual tree* T *for* $\mathbf{D}$ *such that* T' *is a connected subgraph of* T.

Proof. By induction on k, the number of operations applied to $\mathbf{D}$ to obtain $\mathbf{D}'$.

Basis ($k = 0$). Then $\mathbf{D} = \mathbf{D}'$ and T and T' are identical.

Induction hypothesis. If $k \leqslant l$, then Lemma 3.6 holds.

Induction step. Consider $k = l + 1$ and let $\mathbf{D}''$ be the schema after applying l operations to $\mathbf{D}$. There are two cases to consider:

(i) $\mathbf{D}'$ is obtained from $\mathbf{D}''$ by attribute deletion. Then, T' is a qual tree for $\mathbf{D}''$ as well. By the induction hypothesis, the desired tree T exists.

(ii) $\mathbf{D}'$ is obtained from $\mathbf{D}''$ by deleting a relation schema, say $\mathbf{R}$, which is a subset of some relation schema $\mathbf{S}$ in $\mathbf{D}'$. We contruct a qual tree T'' for $\mathbf{D}''$ by "attaching" a leaf r corresponding to $\mathbf{R}$ via an edge to the node s corresponding to $\mathbf{S}$—as $\mathbf{R} \subseteq \mathbf{S}$ attribute connectivity is maintained. By the induction hypothesis there is a qual tree T for $\mathbf{D}$ such that T''—and hence T'—is a connected subgraph of T. ∎

The process described in Lemma 3.6 is called qual tree construction by *reversal* of the GYO reduction. Let $\mathbf{D}$ be a tree schema and $\mathbf{D}' \subseteq \mathbf{D}$. $\mathbf{D}'$ is a *subtree* of $\mathbf{D}$ if there exists a qual tree T for $\mathbf{D}$ such that the nodes in T corresponding to the schemas in $\mathbf{D}'$ induce a connected subgraph of T. $\mathbf{D}' = (\mathbf{R}'_1, ..., \mathbf{R}'_m)$ is a *weak subtree* of $\mathbf{D}$ if there exists $\mathbf{D}'' = (\mathbf{R}''_1, ..., \mathbf{R}''_m)$ such that for $1 \leqslant i \leqslant m$, $\mathbf{R}'_i \subseteq \mathbf{R}''_i$, and $\mathbf{D}''$ is a subtree of $\mathbf{D}$.

LEMMA 3.7 [12]. *Let* $\mathbf{D}$ *be a tree schema represented by tree* T. *Let* r *be a leaf in* T *adjacent to node* s, *and* $\mathbf{R}$, $\mathbf{S}$ *the relation schemas corresponding to* r *and* s, *respectively. Then* $\mathbf{R} = X \cup Y$, *where* $X = \mathbf{R} \cap \mathbf{S}$, $Y = \mathbf{R} - \mathbf{S}$, *and all attributes in* Y *appear in no relation schema other than* $\mathbf{R}$.

THEOREM 3.1. *Let* $\mathbf{D}$ *be a tree schema,* $X \subseteq \bigcup(\mathbf{D})$.

(i) *If* $\mathbf{D}' \in \mathrm{pGR}(\mathbf{D}, X)$ *then* $\mathbf{D}'$ *is a weak subtree of* $\mathbf{D}$.

(ii) *Let* $\mathbf{D}' \subseteq \mathbf{D}$. $\mathbf{D}'$ *is a subtree of* $\mathbf{D}$ *iff* $\mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$; *here,* $\mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}')) = \mathbf{D}'$ *iff* $\mathbf{D}'$ *is reduced.*

Proof. (i) Starting with a qual tree T' representing $\mathbf{D}'$, construct a qual tree T representing $\mathbf{D}$ by reversing the GYO reduction. The relation schemas corresponding to the nodes of the connected subgraph T' of T are the required $\mathbf{R}_1, \ldots, \mathbf{R}_m$.

(ii) ($\Rightarrow$) Let T be a qual tree for $\mathbf{D}$ such that $\mathbf{D}'$ induces a connected subgraph T'. We now describe a sequence of GYO operations which transforms T to T'. The basic idea is to eliminate leaves of T until T becomes T'. The reason we can repeatedly eliminate leaves follows from Lemma 3.7. Once those attributes in a relation schema corresponding to a leaf, which appear only in this relation schema are eliminated, the relation schema becomes a subset of the relation schema corresponding to its parent in the tree and hence can be eliminated. Therefore,

$$\mathbf{D}' \in \mathrm{pGR}\left(\mathbf{D}, \bigcup (\mathbf{D}')\right) \quad \text{and} \quad \mathrm{GR}\left(\mathbf{D}, \bigcup (\mathbf{D}')\right) \subseteq \mathbf{D}'.$$

($\Leftarrow$) Since attribute deletion is not applicable to the attributes of $\bigcup(\mathbf{D}')$, the only operation permitted on $\mathbf{D}'$ in the GYO reduction of $\mathbf{D}$ wrt $\bigcup(\mathbf{D}')$ is subset deletion. But these subset deletions commute with "later" operations in the reduction; hence, given the sequence of operations that led to $\mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}'))$, we can construct another sequence of operations that has the subset deletions on $\mathbf{D}'$ coming after all other operations. Thus, $\mathbf{D}' \in \mathrm{pGR}(\mathbf{D}, \bigcup(\mathbf{D}'))$ and, furthermore, no operations had been performed on relation schemas in $\mathbf{D}'$ in this partial GYO reduction. Let T' be a qual tree representing $\mathbf{D}'$. By reversing the GYO reduction, we can construct a qual tree T representing $\mathbf{D}$ such that T' is a connected subgraph of T. Since no operations had been performed on relation schemas in $\mathbf{D}'$ in the partial GYO reduction, the relation schemas corresponding to the subgraph T' of T are precisely those in $\mathbf{D}'$. Hence $\mathbf{D}'$ is a subtree of $\mathbf{D}$.

Finally, suppose $\mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$. If $\mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}')) = \mathbf{D}'$, then since $\mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}'))$ is reduced, so is $\mathbf{D}'$. Conversely, if $\mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}')) \subsetneq \mathbf{D}'$, then some relation schema in $\mathbf{D}'$ was deleted in the GYO reduction, and so $\mathbf{D}'$ is not reduced. ∎

Theorem 3.1 establishes a link between the GYO reduction and qual trees. Theorem 3.1(ii) provides a characterization of subtrees of a tree schema that will prove useful when we consider joins in tree databases. The next theorem examines further the relationship between the GYO reduction and tree schemas.

THEOREM 3.2. *Let* $\mathbf{D}$ *be an arbitrary schema*:

(i) $\mathbf{D} \cup (\mathbf{R})$ *is a tree schema implies* $\mathrm{GR}(\mathbf{D}) \cup (\mathbf{R})$ *is a tree schema*;

(ii) $\mathbf{D} \cup (\bigcup(\mathrm{GR}(\mathbf{D})))$ *is a tree schema*;

(iii) *if* $\mathbf{D} \cup (\mathbf{S})$ *is a tree schema then* $\mathbf{S} \supseteq \bigcup(\mathrm{GR}(\mathbf{D}))$;

(iv) *if* $\mathrm{GR}(\mathbf{D}) \cup (\mathbf{S})$ *is a tree schema then* $\mathbf{S} \supseteq \bigcup(\mathrm{GR}(\mathbf{D}))$.

Proof. (i) Let T be a qual tree for $\mathbf{D}_1 = \mathbf{D} \cup (\mathbf{R})$. Construct a new database schema by uniformly deleting from $\mathbf{D}_1$ attributes not in $\bigcup(\mathrm{GR}(\mathbf{D}))$ except that $\mathbf{R}$ is left unchanged. Clearly, T is a qual tree for the new database schema as attribute connectivity is maintained. Now, eliminate subsets from the new schema but do not use $\mathbf{R}$ as a subset or superset. The result is $\mathrm{GR}(\mathbf{D}) \cup (\mathbf{R})$. It is a tree schema as subset elimination preserves schema type.

(ii) Let $\mathbf{R} = \bigcup(\mathrm{GR}(\mathbf{D}))$. Clearly, $(\mathbf{R}) \cup \mathrm{GR}(\mathbf{D}) \in \mathrm{pGR}(\mathbf{D} \cup (\mathbf{R}))$. But $\mathbf{R} = \bigcup(\mathrm{GR}(\mathbf{D}))$ implies that $(\mathbf{R}) \cup \mathrm{GR}(\mathbf{D})$ is a tree schema. Since GYO preserves schema type, $\mathbf{D} \cup (\mathbf{R})$ is a tree schema.

(iii) By (i), $\mathrm{GR}(\mathbf{D}) \cup (\mathbf{S})$ is a tree schema. We claim that if $\mathrm{GR}(\mathbf{D}) \cup (\mathbf{S})$ is represented by qual tree T, then T is "star-shaped" around S; i.e., the node corresponding to S is the root, and all nodes corresponding to other relation schemas are leaves. If T is not star-shaped, then T contains a leaf l and a node p adjacent to l with the associated relations $\mathbf{R}_l$, $\mathbf{R}_p$ in $\mathrm{GR}(\mathbf{D})$. By Lemma 3.7 and the fact that no attribute in $\mathrm{GR}(\mathbf{D})$ appears only in one schema, $\mathbf{R}_l \subseteq \mathbf{R}_p$, this is impossible as $\mathrm{GR}(\mathbf{D})$ is reduced. The contradiction implies that T is star-shaped. Since every attribute in $\mathrm{GR}(\mathbf{D})$ appears in at least two relation schemas, and S is on the path between them, by attribute connectivity arguments every attribute in $\mathrm{GR}(\mathbf{D})$ must appear in the root S, i.e., $\bigcup(\mathrm{GR}(\mathbf{D})) \subseteq \mathbf{S}$.

(iv) By (iii), if $\mathrm{GR}(\mathbf{D}) \cup (\mathbf{S})$ is a tree schema, then $\mathbf{S} \supseteq \bigcup(\mathrm{GR}(\mathrm{GR}(\mathbf{D})))$, i.e., $\mathbf{S} \supseteq \bigcup(\mathrm{GR}(\mathbf{D}))$. ∎

The following result of [3, 19] can be proved nicely from the preceding theorems.

COROLLARY 3.1. *$\mathbf{D}$ is a tree schema iff* $\mathrm{GR}(\mathbf{D}) = \varnothing$.

Proof. ($\Rightarrow$) Consider any $\mathbf{R} \in \mathbf{D}$. Since $(\mathbf{R})$ is a subtree of $\mathbf{D}$, by Theorem 3.1(ii), $\mathrm{GR}(\mathbf{D}, \bigcup((\mathbf{R}))) = (\mathbf{R})$. It follows that $\mathrm{GR}(\mathbf{D}) = \varnothing$.

($\Leftarrow$) By Theorem 3.2(ii), $\mathbf{D} \cup (\varnothing)$ is a tree schema, so $\mathbf{D}$ must be a tree schema. ∎

COROLLARY 3.2. *The relation schema of least cardinality whose addition to* $\mathbf{D}$ *makes it a tree schema is* $\bigcup(\mathrm{GR}(\mathbf{D}))$.

GYO reductions are also related to canonical connections.

THEOREM 3.3. *Let* $\mathbf{D}$ *be an arbitrary schema and* $X \subseteq \bigcup(\mathbf{D})$:

(i) $\mathrm{CC}(\mathbf{D}, X) \leqslant \mathrm{GR}(\mathbf{D}, X)$.

(ii) *If* $\mathbf{D}$ *is a tree schema, then* $\mathrm{CC}(\mathbf{D}, X) = \mathrm{GR}(\mathbf{D}, X)$ [16].

(iii) *If* $\bigcup(\mathrm{GR}(\mathbf{D}, X)) \subseteq X$, *then* $\mathrm{CC}(\mathbf{D}, X) = \mathrm{GR}(\mathbf{D}, X)$.

Proof. (i) It is proved in [16] that, if $\mathbf{D}'' = \mathrm{GR}(\mathbf{D}, X)$, then $\mathrm{Tab}(\mathbf{D}, X) \equiv \mathrm{Tab}(\mathbf{D}'', X)$. By Lemma 3.2, $(\mathbf{D}, X) \equiv (\mathbf{D}'', X)$, and so $\mathrm{CC}(\mathbf{D}, X) = \mathrm{CC}(\mathbf{D}'', X)$ by Lemma 3.5. But $\mathrm{CC}(\mathbf{D}'', X) \leqslant \mathbf{D}'' = \mathrm{GR}(\mathbf{D}, X)$. Hence the claim.

(ii) This is proved in [16].

(iii) Let $\mathbf{D}'' = \mathrm{GR}(\mathbf{D}, X)$. As in (i), $\mathrm{Tab}(\mathbf{D}, X) \equiv \mathrm{Tab}(\mathbf{D}'', X)$. We shall argue that $\mathrm{Tab}(\mathbf{D}'', X)$ is, in fact, a minimal tableau for $(\mathbf{D}, X)$. Now, $\bigcup(\mathbf{D}'') \subseteq X$ implies $\bigcup(\mathbf{D}'') = X$, so $\mathrm{Tab}(\mathbf{D}'', X) = \mathrm{Tab}(\mathbf{D}'', \bigcup(\mathbf{D}''))$. Thus all variables in $\mathrm{Tab}(\mathbf{D}'', X)$ for attributes are distinguished, and any containment mapping from $\mathrm{Tab}(\mathbf{D}'', X)$ to a tableau with fewer rows would correspond to relation schema elimination in a GYO reduction of $\mathbf{D}''$ w.r.t. $\bigcup(\mathbf{D}'')$, which is impossible. Therefore, $\mathrm{Tab}(\mathbf{D}'', X)$ is a minimal tableau for $(\mathbf{D}, X)$ and $\mathbf{D}'' = \mathrm{CC}(\mathbf{D}, X)$. ▌

4. Solving Queries Using Joins

Let $X \subseteq \bigcup(\mathbf{D})$. The problem is to characterize those $\mathbf{D}' \leqslant \mathbf{D}$ for which $(\mathbf{D}', X) \equiv (\mathbf{D}, X)$. This problem is addressed in [15, 18]. We use the concepts of Section 3 to unify their results.

Theorem 4.1. *Let* $\mathbf{D}' \leqslant \mathbf{D}$. *The following are equivalent*: (i) $\mathrm{CC}(\mathbf{D}, X) \leqslant \mathbf{D}'$; (ii) $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$; (iii) $\mathrm{CC}(\mathbf{D}, X) = \mathrm{CC}(\mathbf{D}', X)$.

Proof. (i) ⇒ (ii) Let $\mathbf{D}'' = \mathrm{CC}(\mathbf{D}, X)$. Since $\mathbf{D}'' \leqslant \mathbf{D}' \leqslant \mathbf{D}$, we have $(\mathbf{D}, X) \subseteq (\mathbf{D}', X) \subseteq (\mathbf{D}'', X)$. But by definition of $\mathbf{D}''$, $\mathrm{Tab}(\mathbf{D}, X) \equiv \mathrm{Tab}(\mathbf{D}'', X)$. By Lemma 3.2, $(\mathbf{D}, X) \equiv (\mathbf{D}'', X)$, so $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$.

(ii) ⇒ (iii) Lemma 3.5.

(iii) ⇒ (i) Simply observe that $\mathrm{CC}(\mathbf{D}', X) \leqslant \mathbf{D}'$. ▌

Corollary 4.1. *To solve* $(\mathbf{D}, X)$ *by joining the relations in* $\mathbf{D}' \subseteq \mathbf{D}$ *and then projecting the result onto* X, *it is necessary and sufficient that* $\mathrm{CC}(\mathbf{D}, X) \leqslant \mathbf{D}'$.

Corollary 4.2. *Suppose* $\mathbf{D}' \leqslant \mathbf{D}$. *Checking* $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$ *can be done by minimizing* $\mathrm{Tab}(\mathbf{D}, X)$ [2] *to obtain* $\mathrm{CC}(\mathbf{D}, X)$ *and then verifying that* $\mathrm{CC}(\mathbf{D}, X) \leqslant \mathbf{D}'$.

If $\mathbf{D}$ is a tree schema then, by Theorems 3.3 and 4.1, $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$ iff $\mathrm{GR}(\mathbf{D}, X) \leqslant \mathbf{D}'$. This was proved by Hull [15] and Yanakakis [18]. They also proved that for cyclic schemas, $\mathrm{GR}(\mathbf{D}, X) \leqslant \mathbf{D}'$ is a sufficient condition for $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$. Theorem 4.1 strengthens this result by stating that for cyclic schemas $\mathrm{CC}(\mathbf{D}, X) \leqslant \mathbf{D}'$ is a necessary and sufficient condition.

We now discuss the case of non-UR databases briefly. One strategy for solving $(\mathbf{D}, X)$ is to transform the database into a UR database and identify $\mathbf{D}'$ as above. If $\mathbf{D}$ is a tree schema, the non-UR transformation can be done efficiently using semijoins [5]. If $\mathbf{D}$ is cyclic, a possible strategy is (i) transform $\mathbf{D}$ into a tree schema by adding one or more relation schemas; (ii) use joins and projects to build relation states for those relation schemas—this reduces the cyclic case to the previous tree case; (iii) use semijoins as in the previous case to obtain a UR database. If step (i) is done by adding a single relation schema, Corollary 3.2 tells us the best choice for that schema, namely $\bigcup(\mathrm{GR}(\mathbf{D}))$. If, however, step (i) is done by adding multiple relations, we run into the following NP-complete problem:

Fixed Treefication. Given a schema $\mathbf{D}$ and integers K, B, are there $\mathbf{R}'_1,..., \mathbf{R}'_K$ such that $\mathbf{D} \cup (\mathbf{R}'_1,..., \mathbf{R}'_K)$ is a tree schema, and for $1 \leqslant i \leqslant k$, $|\mathbf{R}'_i| \leqslant B$?

Fixed treefication is straightforwardly in NP; to prove completeness, we use the following NP-complete problem [9]:

Bin Packing. Given a set I of items, a size $s(i) \in Z^+$ for each $i \in I$, a positive bin capacity B and a positive integer K. Is there a partition of I into sets $I_1,..., I_K$ such that the sum of the sizes of the items in each I_l is B or less? (W.l.o.g. we may assume that each $s(i)$ and B are divisible by 3.)

THEOREM 4.2. *Fixed treefication is NP-complete.*

Proof. We reduce bin packing to fixed treefication by constructing a database schema $\mathbf{D} = \bigcup_{i \in I} \mathbf{R}_i$. Each $\mathbf{R}_i$ is an Aclique of size $s(i)$ over a unique set of attributes. (This means that an integer $s(i)$, represented by using $\log_2(s(i))$ bits in the bin packing instance, is transformed into $\log_2(s(i)) \cdot s(i)^2$ bits in the fixed treefication instance. However, bin packing is NP-complete in the *strong sense* [9], which renders the reduction proper.)

Claim. There is a fixed treefication for $\mathbf{D}$, K, B iff there is a bin packing assignment for I, K, B:

($\Rightarrow$) Let $\mathbf{D}' = \mathbf{D} \cup (\mathbf{R}'_1,..., \mathbf{R}'_K)$. Since $\mathbf{D}'$ is a tree schema, $\mathrm{GR}(\mathbf{D}') = (\varnothing)$. This implies that each Aclique in $\mathbf{D}$ is eliminated in the reduction process; as no Aclique attribute appears only in one relation, the attributes in each Aclique in $\mathbf{D}$ must appear together in some $\mathbf{R}'_j$. Assign $i \in I$ to a bin j such that $\bigcup(\mathbf{R}_i) \subseteq \mathbf{R}'_j$. Observe that each i is assigned to a bin. Also, the sum of sizes of items in some bin j cannot be more than the number of attributes in $\mathbf{R}'_j$, which cannot exceed B. Hence we have a bin packing assignment for I, K, B.

($\Leftarrow$) Let $I_1,..., I_K$ be the partition. Construct, for $1 \leqslant l \leqslant k$, $\mathbf{R}'_l = \bigcup \{\bigcup(\mathbf{R}_i) \mid i \in I_l\}$, i.e., $\mathbf{R}'_l$ contains all the attributes associated with Acliques corresponding to items in I_l. Let $\mathbf{D}' = \mathbf{D} \cup (\mathbf{R}'_l \mid 1 \leqslant l \leqslant K)$. Since the sum of sizes of items in no bin exceeds B, and item i of size $s(i)$ generates an Aclique of size $s(i)$, $|\mathbf{R}'_l| \leqslant B$. Also, as the attributes of each Aclique appear together and the set of attributes of Acliques are disjoint, $\mathrm{GR}(\mathbf{D}') = (\varnothing)$ and so $\mathbf{D}'$ is a tree schema. ∎

5. LOSSLESS JOINS

5.1. *Canonical Connections and Lossless Joins*

A *join dependency* (jd) is a statement of the form $\bowtie \mathbf{D}$; this jd *holds* in a universal relation I, written $I \models \bowtie \mathbf{D}$ if $\prod_{\bigcup(\mathbf{D})} I = \bowtie_{\mathbf{R} \in \mathbf{D}} (\prod_{\mathbf{R}} I)$ [17]. (If $\bigcup(\mathbf{D}) \subsetneq U$, this is usually called an *embedded* join dependency.) We use $\bowtie \mathbf{D} \models \bowtie \mathbf{D}'$ to mean

$I \models \bowtie \mathbf{D}$ implies $I \models \bowtie \mathbf{D}'$. In this case we also say that $\bowtie \mathbf{D}$ implies that $\mathbf{D}'$ has a *lossless join* [1].

Let D be a UR database for schema $\mathbf{D}$, and let $I = \bowtie_{R \in D} R$. Trivially, $I \models \bowtie \mathbf{D}$. This and Theorem 4.1 give

THEOREM 5.1. *Let* $\mathbf{D}' \leqslant \mathbf{D}$. *The following are equivalent*: (i) $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$, (ii) $\bowtie \mathbf{D} \models \bowtie \mathbf{D}'$, *and* (iii) $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) = \mathrm{CC}(\mathbf{D}', \bigcup(\mathbf{D}'))$. (*There is equality in* (i) *iff* $\mathbf{D}'$ *is reduced.*)

Proof. By Theorem 4.1, the following are equivalent: (i) $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \leqslant \mathbf{D}'$; (ii) $(\mathbf{D}, \bigcup(\mathbf{D}')) \equiv (\mathbf{D}', \bigcup(\mathbf{D}'))$; (iii) $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) = \mathrm{CC}(\mathbf{D}', \bigcup(\mathbf{D}'))$. Suppose $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \leqslant \mathbf{D}'$, $\mathbf{R} \in \mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}'))$, $\mathbf{S} \in \mathbf{D}'$, $\mathbf{R} \subseteq \mathbf{S}$. Since $\mathbf{S} \in \mathbf{D}'$ and $\mathbf{D}' \leqslant \mathbf{D}$, there exists $\mathbf{S}' \in \mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}'))$ such that $\mathbf{S} \subseteq \mathbf{S}'$, and so $\mathbf{R} \subseteq \mathbf{S}'$. But $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}'))$ is reduced, so $\mathbf{R} = \mathbf{S} = \mathbf{S}'$ and $\mathbf{R} \in \mathbf{D}'$. Hence $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \leqslant \mathbf{D}'$ implies $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$. Conversely, it is obvious that $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$ implies $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \leqslant \mathbf{D}'$. $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \leqslant \mathbf{D}'$ and $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$ are thus equivalent.

It therefore suffices to show that

$$(\mathbf{D}, \textstyle\bigcup(\mathbf{D}')) \equiv (\mathbf{D}', \bigcup(\mathbf{D}')) \qquad \text{iff} \quad \bowtie \mathbf{D} \models \bowtie \mathbf{D}'.$$

($\Rightarrow$) Given $(\forall I)[\prod_{\bigcup(\mathbf{D}')}(\bowtie_{\mathbf{R} \in \mathbf{D}}(\prod_{\mathbf{R}} I)) = \bowtie_{\mathbf{R}' \in \mathbf{D}'}(\prod_{\mathbf{R}'} U)]$. Suppose $I \models \bowtie \mathbf{D}$. Then

$$\begin{aligned} \mathop{\bowtie}_{\mathbf{R}' \in \mathbf{D}'} \Big(\prod_{\mathbf{R}'} I\Big) &= \prod_{\bigcup(\mathbf{D}')} \Big(\mathop{\bowtie}_{\mathbf{R} \in \mathbf{D}} \Big(\prod_{\mathbf{R}} I\Big)\Big) \\ &= \prod_{\bigcup(\mathbf{D}')} \prod_{\bigcup(\mathbf{D})} I \\ &= \prod_{\bigcup(\mathbf{D}')} I \qquad \text{since} \quad \textstyle\bigcup(\mathbf{D}') \subseteq \bigcup(\mathbf{D}) \end{aligned}$$

and hence $I \models \bowtie \mathbf{D}'$.

($\Leftarrow$) For any I, let $J = \bowtie_{\mathbf{R} \in \mathbf{D}}(\prod_{\mathbf{R}} I)$. Then $J \models \bowtie \mathbf{D}$, and so $J \models \bowtie \mathbf{D}'$, i.e., $\prod_{\bigcup(\mathbf{D}')} J = \bowtie_{\mathbf{R}' \in \mathbf{D}'}(\prod_{\mathbf{R}'} J)$. Hence $\prod_{\bigcup(\mathbf{D}')}(\bowtie_{\mathbf{R} \in \mathbf{D}}(\prod_{\mathbf{R}} I)) = \bowtie_{\mathbf{R}' \in \mathbf{D}'}(\prod_{\mathbf{R}'} J) = \bowtie_{\mathbf{R}' \in \mathbf{D}'}(\prod_{\mathbf{R}'} I)$ (since $\mathbf{D}' \leqslant \mathbf{D}$), and so $(\mathbf{D}, \bigcup(\mathbf{D}')) \equiv (\mathbf{D}', \bigcup(\mathbf{D}'))$.

Suppose now that $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$. If $\mathbf{D}'$ is reduced, then every relation schema in $\mathbf{D}'$ must be in $\mathrm{CC}(\mathbf{D}, \bigcup \mathbf{D}'))$; hence $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) = \mathbf{D}'$. Conversely, if $\mathbf{D}'$ is not reduced, then the subsets in $\mathbf{D}'$ are surely not in $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}'))$, and so $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \subsetneq \mathbf{D}'$. ∎

COROLLARY 5.1. *Suppose* $\mathbf{D}' \leqslant \mathbf{D}$. *Checking* $\bowtie \mathbf{D} \models \bowtie \mathbf{D}'$ *can be done by minimizing* $\mathrm{Tab}(\mathbf{D}, \bigcup(\mathbf{D}'))$ *to obtain* $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}'))$ *and then verifying that* $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$.

For tree schemas, the lossless join question has a very appealing answer.

COROLLARY 5.2. *Let* $\mathbf{D}$ *be a tree schema and* $\mathbf{D}' \subseteq \mathbf{D}$. *Then* $\bowtie \mathbf{D} \models \bowtie \mathbf{D}'$ *iff* $\mathbf{D}'$ *is a subtree of* $\mathbf{D}$.

Proof. $\bowtie \mathbf{D} \models \bowtie \mathbf{D}'$ iff $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$ (Theorem 5.1) iff $\mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$ (Theorem 3.3) iff $\mathbf{D}'$ is a subtree of $\mathbf{D}$ (Theorem 3.1) ∎

As an example, consider $\mathbf{D} = \{abc, ab, bc\}$ and $\mathbf{D}' = \{ab, bc\}$. It is easy to see that $\bowtie \mathbf{D} \not\models \bowtie \mathbf{D}'$ and $\mathbf{D}'$ is not a subtree of $\mathbf{D}$.

THEOREM 5.2. *Let* $\mathbf{D}' \leqslant \mathbf{D}$ *be a minimum cardinality schema such that* $\mathrm{CC}(\mathbf{D}', X) = \mathrm{CC}(\mathbf{D}, X)$. *Then* $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) = \mathbf{D}'$.

Proof. By Theorem 4.1, $\mathrm{CC}(\mathbf{D}', X) = \mathrm{CC}(\mathbf{D}, X)$ iff $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$, and by Lemma 3.2, $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$ iff $\mathrm{Tab}(\mathbf{D}, X) \equiv \mathrm{Tab}(\mathbf{D}', X)$, so there is containment mapping h from $\mathrm{Tab}(\mathbf{D}, X)$ to $\mathrm{Tab}(\mathbf{D}', X)$. We shall argue that h is also a containment mapping from $\mathrm{Tab}(\mathbf{D}, \bigcup(\mathbf{D}'))$ to $\mathrm{Tab}(\mathbf{D}', \bigcup(\mathbf{D}'))$:

Since $\mathbf{D}' \leqslant \mathbf{D}$, for each $\mathbf{R} \in \mathbf{D}'$, select some $\mathbf{S} \in \mathbf{D}$ such that $\mathbf{R} \subseteq \mathbf{S}$, and denote $\mathbf{S} = \mathrm{parent}(\mathbf{R})$. Let $\mathrm{parent}(\mathbf{D}') = \{\mathrm{parent}(\mathbf{R}) \mid \mathbf{R} \in \mathbf{D}'\}$. Clearly, $\mathrm{parent}(\mathbf{D}') \subseteq \mathbf{D}$ and $|\mathrm{parent}(\mathbf{D}')| \leqslant |\mathbf{D}'|$. There is an obvious containment mapping h_p from $\mathrm{Tab}(\mathbf{D}', X)$ to $\mathrm{Tab}(\mathrm{parent}(\mathbf{D}'), X)$ that maps the row for $\mathbf{R}$ to that for $\mathrm{parent}(\mathbf{R})$.

We may consider $\mathrm{Tab}(\mathrm{parent}(\mathbf{D}'), X)$ to be a subtableau of $\mathrm{Tab}(\mathbf{D}, X)$. Suppose that h maps the rows of $\mathrm{Tab}(\mathrm{parent}(\mathbf{D}'), X)$ to a proper subset of the rows of $\mathrm{Tab}(\mathbf{D}', X)$. Then by composing h, h_p, and h (Fig. 3), there is a containment mapping from $\mathrm{Tab}(\mathbf{D}, X)$ to a proper subtableau of $\mathrm{Tab}(\mathbf{D}', X)$, thus contradicting the minimality of $\mathbf{D}'$.

Hence h maps the rows of $\mathrm{Tab}(\mathrm{parent}(\mathbf{D}'), X)$ onto those of $\mathrm{Tab}(\mathbf{D}', X)$, and $|\mathrm{parent}(\mathbf{D}')| = |\mathbf{D}'|$. Since the mapping is one–one onto, if a is a repeated variable in $\mathrm{Tab}(\mathrm{parent}(\mathbf{D}'), X)$ (i.e., a appears in two rows), so is $h(a)$. Therefore $h(a) = a$ for every distinguished or repeated variable a in $\mathrm{Tab}(\mathrm{parent}(\mathbf{D}'), X)$. By the one–one subset correspondence between $\mathbf{D}'$ and $\mathrm{parent}(\mathbf{D}')$, the distinguished and repeated variables in $\mathrm{Tab}(\mathbf{D}', X)$ are also distinguished and repeated variables in $\mathrm{Tab}(\mathrm{parent}(\mathbf{D}'), X)$, hence $h(a) = a$ for every distinguished or repeated variable in $\mathrm{Tab}(\mathbf{D}', X)$.

Consider now a' in $\mathrm{Tab}(\mathbf{D}', X)$ that is, neither distinguished nor repeated. If $h(a') \neq a'$, and a' appears in row s for an attribute in some $\mathbf{S} \in \mathbf{D}'$, then h must map

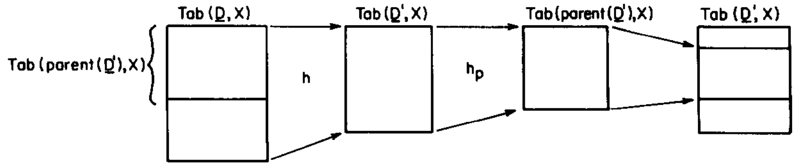


FIG. 3. Composition of h, h_p, and h.

the row for parent(**S**) to some row r for $\mathbf{R} \in \mathbf{D}'$, $\mathbf{R} \neq \mathbf{S}$. Then the mapping h_s from Tab($\mathbf{D}'$, X) to subtableau Tab($\mathbf{D}' - (\mathbf{S})$, X) defined by

$$\begin{aligned} h_s(t) &= t \quad \text{if} \quad t \neq s, \\ &= r \quad \text{if} \quad t = s, \end{aligned}$$

is a containment mapping since all distinguished and repeated variables map onto themselves. Again, by composing h and h_s, we get a containment mapping from Tab($\mathbf{D}$, X) to a proper subtableau of Tab($\mathbf{D}'$, X), which contradicts the minimality of $\mathbf{D}'$.

It must therefore be the case that h maps all variables for attributes in $\mathbf{D}'$ to themselves. Thus, if all variables for attributes in $\mathbf{D}'$ were made distinguished, h remains a valid containment mapping from Tab($\mathbf{D}$, $\bigcup(\mathbf{D}')$) to Tab($\mathbf{D}'$, $\bigcup(\mathbf{D}')$). But there is an obvious containment mapping from Tab($\mathbf{D}'$, $\bigcup(\mathbf{D}')$) to Tab($\mathbf{D}$, $\bigcup(\mathbf{D}')$) because $\mathbf{D}' \leqslant \mathbf{D}$, so Tab($\mathbf{D}$, $\bigcup(\mathbf{D}')$) $\equiv$ Tab($\mathbf{D}'$, $\bigcup(\mathbf{D}')$). This implies, by Lemma 3.2, that $(\mathbf{D}' \bigcup(\mathbf{D}')) \equiv (\mathbf{D}', \bigcup(\mathbf{D}'))$, and therefore $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) = \mathrm{CC}(\mathbf{D}', \bigcup(\mathbf{D}'))$ (Theorem 4.1). Since $\mathbf{D}'$ minimal, $\mathrm{CC}(\mathbf{D}', \bigcup(\mathbf{D}')) = \mathbf{D}'$; hence the theorem. ▮

COROLLARY 5.3 [18]. *Let* $\mathbf{D}' \leqslant \mathbf{D}$ *be a minimal cardinality schema such that* $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$. *Then* $\bowtie \mathbf{D}'$ *implies* $\mathbf{D}'$ *has a lossless join.*

Proof. By Theorem 4.1, $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$ implies $\mathrm{CC}(\mathbf{D}, X) = \mathrm{CC}(\mathbf{D}', X)$. Hence $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) = \mathbf{D}'$ (Theorem 5.2), and so $\bowtie \mathbf{D} \models \bowtie \mathbf{D}'$ (Theorem 5.1). ▮

These results are related to problems studied by Yannakakis [18], Goodman and Shmueli [11], and Fagin [7].

Yannakakis considered the following problem. Given $\mathbf{D}' \subseteq \mathbf{D}$ and $X \subseteq \bigcup(\mathbf{D}')$, when does $I \models \bowtie \mathbf{D}$ imply $\prod_X (\bowtie_{\mathbf{R}' \in \mathbf{D}'}(\prod_{\mathbf{R}'} I)) = \prod_X I$? In our terms, this amounts to asking whether $(\mathbf{D}, X) \equiv (\mathbf{D}', X)$. This can be answered using simple tableau techniques [2], or by applying Corollary 4.2. Taking $X = \bigcup(\mathbf{D}')$, Yannakakis's problem is precisely the lossless join problem, i.e., deciding whether $\bowtie \mathbf{D} \models \bowtie \mathbf{D}'$. Once again, tableau equivalence can be used to answer this question. Alternatively, Corollary 5.1 can be invoked. We note that Corollary 5.3 appears in [18] for the case $\mathbf{D}' \subseteq \mathbf{D}$. Corollary 5.2 also appears there, although the proof using our techniques is more direct.

In [11], the problem of *ultra join reduction* is considered. Let D be a database for $\mathbf{D}$. D is UJR if for all minimum size (minimum number of edges) qual graphs G for $\mathbf{D}$, and for all connected subgraphs of G, say consisting of nodes $r_1, \ldots, r_k$ corresponding to $\mathbf{R}_{i_1}, \ldots, \mathbf{R}_{i_k}$, the following holds:

$$\prod_X \left(\bowtie_{i=1}^{n} R_i \right) = \bowtie_{j=1}^{k} R_{i_j}, \qquad \text{where} \quad X = \bigcup_{j=1}^{k} \mathbf{R}_{i_j}.$$

Every UJR database is trivially UR. In [11], it is proved that for tree schemas the converse holds—i.e., for all tree schemas **D**, every UR database for **D** is also UJR; while for cyclic schemas the converse fails—for all cyclic schemas **D**, there exists a UR database for **D** that is not UJR.

Let us interpret these results in light of the current treatment. For tree schemas, a minimum size qual graph is simply a tree, and so the question addressed is this: Does $\bowtie \mathbf{D} \models \bowtie \mathbf{D}'$ for all subtrees $\mathbf{D}'$ of $\mathbf{D}$? Corollary 5.2 gives the answer, "yes." Moreover, Corollary 5.2 strengthens [11] by answering the converse to this question, too. It is also easy to see why UR does not imply UJR for cyclic schemas: If the nodes corresponding to $\mathbf{D}'$ induce a connected subgraph in some minimal qual graph representing $\mathbf{D}$, it is not necessary that $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$, Thus, by Theorem 5.1, $\bowtie \mathbf{D} \not\models \bowtie \mathbf{D}'$.

5.2. *γ-Acyclic Databases*

Fagin [7] characterized those schemas $\mathbf{D}$ such that $\bowtie \mathbf{D} \models \bowtie \mathbf{D}'$ for all "connected" $\mathbf{D}' \subseteq \mathbf{D}$ [7]. ($\mathbf{D}'$ is *connected* if every pair or relation schemas $\mathbf{R}$ and $\mathbf{S}$ is connected by a *path* $\mathbf{R}_{i_1}, \mathbf{R}_{i_2}, ..., \mathbf{R}_{i_l}$, where $\mathbf{R} = \mathbf{R}_{i_1}$, $\mathbf{S} = \mathbf{R}_{i_l}$, and adjacent relation schemas share at least one attribute.)

Fagin's characterization is based on γ-cycles. A weak *γ-cycle* in $\mathbf{D}$ is a sequence $(\mathbf{R}_1, A_1, \mathbf{R}_2, ..., \mathbf{R}_m, A_m, \mathbf{R}_1)$ such that $m \geqslant 3$, the A_is are distinct, $A_i \in \mathbf{R}_i \cap \mathbf{R}_{i+1}$ $(m+1 \equiv 1)$, A_1 is only in $\mathbf{R}_1$ and $\mathbf{R}_2$, and A_2 is only in $\mathbf{R}_2$ and $\mathbf{R}_3$ [7]. $\mathbf{D}$ is *γ-acyclic* if it contains no weak γ-cycles. Fagin proves

$$\bowtie \mathbf{D} \models \bowtie \mathbf{D}' \qquad \text{for all connected} \quad \mathbf{D}' \subseteq \mathbf{D} \text{ iff } \mathbf{D} \text{ is } \gamma\text{-acyclic.} \qquad (*)$$

We offer the following alternative characterizations of γ-acyclic schemas.

THEOREM 5.3. *The following are equivalent*:

(i) $\mathbf{D}$ *is γ-acyclic.*

(ii) *For all* $\mathbf{R}_1$, $\mathbf{R}_2$ *in* $\mathbf{D}$ *such that* $\mathbf{R}_1 \cap \mathbf{R}_2 \neq \varnothing$, *deleting attributes* $\mathbf{R}_1 \cap \mathbf{R}_2$ *from* $\mathbf{D}$ *results in a schema in which* $\mathbf{R}_1 - (\mathbf{R}_1 \cap \mathbf{R}_2)$ *and* $\mathbf{R}_2 - (\mathbf{R}_1 \cap \mathbf{R}_2)$ *are not connected.*

(iii) $\mathbf{D}$ *is a tree schema and every connected* $\mathbf{D}' \subseteq \mathbf{D}$ *is a subtree of* $\mathbf{D}$.

Proof. (i) ⇒ (ii) Suppose, for the sake of deriving a contradiction, that there exists $\mathbf{R}, \mathbf{S} \in \mathbf{D}$ violating (ii). Let $X = \mathbf{R} \cap \mathbf{S}$ and delete X from all relation schemas in $\mathbf{D}$. Let $\mathbf{R}_1 - X, \mathbf{R}_2 - X, ..., \mathbf{R}_n - X$ be a path connecting $\mathbf{R}_1 - X$ and $\mathbf{R}_n - X$, where $\mathbf{R}_1 = \mathbf{R}$ and $\mathbf{R}_n = \mathbf{S}$. Since we deleted $\mathbf{R} \cap \mathbf{S}$, $n \geqslant 3$.

If $(\mathbf{R}_i - X) \cap (\mathbf{R}_j - X) \neq \varnothing$ for some $1 \leqslant i < j \leqslant n$ and $i+1 < j$, then the path can be shortened to

$$\mathbf{R}_1 - X, \mathbf{R}_2 - X, ..., \mathbf{R}_{i-1} - X, \mathbf{R}_i - X, \mathbf{R}_j - X, \mathbf{R}_{j+1} - X, ..., \mathbf{R}_n - X$$

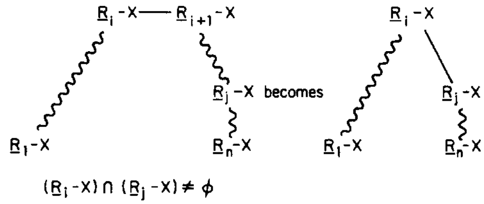


FIG. 4. Shortening a path connecting $\mathbf{R}_1-X$ and $\mathbf{R}_n-X$.

(see Fig. 4). Note that this shorter path must still have at least 3 relation schemas. We may therefore assume that $(\mathbf{R}_i-X)\cap(\mathbf{R}_j-X)=\varnothing$ for $2\leqslant i+1<j\leqslant n$, where $n\geqslant 3$.

Let $A_i\in(\mathbf{R}_i-X)\cap(\mathbf{R}_{i+1}-X)$ for $1\leqslant i\leqslant n-1$ and $A_n\in X=\mathbf{R}_1\cap\mathbf{R}_n$ $(\neq\varnothing)$; by the preceding remark, the A_is are distinct. Furthermore, $A_1\notin\mathbf{R}_i$ for $i\neq 1, 2$, and $A_2\notin\mathbf{R}_i$ for $i\neq 2, 3$. We thus have a γ-cycle $(\mathbf{R}_1, A_1, \mathbf{R}_2,..., \mathbf{R}_n, A_n, \mathbf{R}_1)$.

(ii) $\Rightarrow$ (i). Suppose, for the sake of deriving a contradiction, that $\mathbf{D}$ has the γ-cycle $C=(\mathbf{R}_1, A_1, \mathbf{R}_2, A_2,..., \mathbf{R}_n, A_n, \mathbf{R}_1)$. If, for some i, j such that $1<i<j<n$, $\mathbf{R}_i\cap\mathbf{R}_{i+1}\subseteq\mathbf{R}_j\cap\mathbf{R}_{j+1}$, then $C_1=(\mathbf{R}_1, A_1, \mathbf{R}_2, A_2,..., A_{i-1}, \mathbf{R}_i, A_i, \mathbf{R}_{j+1}, A_{j+1},..., A_n, \mathbf{R}_1)$ is a smaller-γ-cycle (see Fig. 5).

Continue until no further contraction is possible. Let the resulting γ-cycle be $C'=(\mathbf{R}'_1, A'_1, \mathbf{R}'_2, A'_2,..., A'_{m-1}, \mathbf{R}'_m, A'_m, \mathbf{R}'_1)$, where $\mathbf{R}'_1=\mathbf{R}_1$, $\mathbf{R}'_2=\mathbf{R}_2$, $\mathbf{R}'_m=\mathbf{R}_n$, $A'_1=A_1$, $A'_m=A_n$. Let $X=\mathbf{R}'_1\cap\mathbf{R}'_m$ $(\neq\varnothing)$. By the definition of a γ-cycle, $A'_1\notin X$ and $A'_2\notin X$. Furthermore, by definition of C', there is no i such that $1<i<m-1$ and $\mathbf{R}'_i\cap\mathbf{R}'_{i+1}\subseteq\mathbf{R}'_1\cap\mathbf{R}'_m$. Hence if we remove X from all relation schemas in $\mathbf{D}$, $\mathbf{R}'_1-X$, and $\mathbf{R}'_m-X$ remain connected by the path $\mathbf{R}'_1-X$, $\mathbf{R}'_2-X$, $\mathbf{R}'_3-X,..., \mathbf{R}'_m-X$ (see Fig. 6); a contradiction.

(ii) $\Rightarrow$ (iii) By Lemma 3.1, if $\mathbf{D}$ is cyclic, then by appropriately deleting some attributes of $\mathbf{D}$ and omitting subsets, we can get either an Aring or an Aclique. In either case, if $\mathbf{R}$ and $\mathbf{S}$ are two relation schemas in this Aring or Aclique such that $\mathbf{R}\cap\mathbf{S}\neq\varnothing$, then deleting the intersection of the supersets in $\mathbf{D}$ of $\mathbf{R}$ and $\mathbf{S}$ does not disconnect those supersets (see Fig. 7). Thus (ii) implies $\mathbf{D}$ is a tree schema.

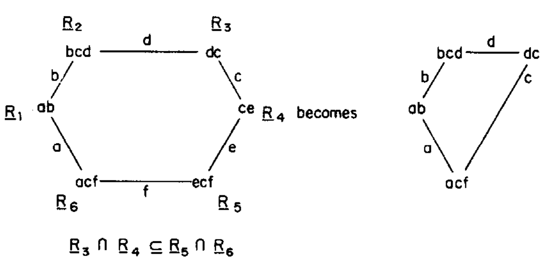


FIG. 5. Contracting a γ-cycle.

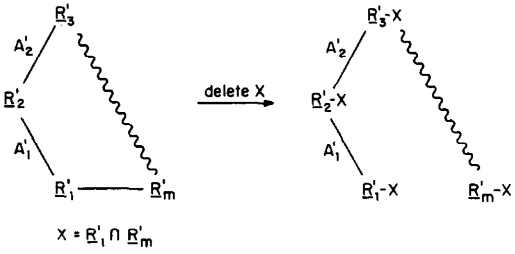


FIG. 6. Deleting $\mathbf{R}'_m \cap \mathbf{R}'_{m-1}$ does not disconnect $\mathbf{R}'_m$ and $\mathbf{R}'_{m-1}$.

We shall prove by induction on the size of $\mathbf{D}'$ that if it is connected, then it is a subtree of $\mathbf{D}$. This trivially holds for $|\mathbf{D}'| = 1$.

Suppose $\mathbf{D}''$ is a subtree of $\mathbf{D}$, and it induces connected subgraph T'' in qual tree T for $\mathbf{D}$. We must show that if $\mathbf{D}' = \mathbf{D}'' \cup (\mathbf{R})$ is connected, then $\mathbf{D}'$ is a subtree of $\mathbf{D}$. Consider any $\mathbf{R}$ in $\mathbf{D} - \mathbf{D}''$ connected to $\mathbf{D}''$, i.e., $\mathbf{R}$ has nonempty intersection with some relation schema in $\mathbf{D}''$. Let $\mathbf{S}$ be the node in T'' such that the path in T from $\mathbf{R}$ to $\mathbf{S}$ contains no edges in T''. (See Fig. 8. Here, we identify the nodes and their corresponding relation schemes.) If $\mathbf{R}$ is adjacent in T to $\mathbf{S}$, then T'' plus the edge $\{\mathbf{R}, \mathbf{S}\}$ is the required T' (Fig. 8a).

In $\mathbf{R}$ is not adjacent in T to $\mathbf{S}$, let $X = \mathbf{R} \cap \mathbf{S}$. Since $\mathbf{R}$ is connected to $\mathbf{D}''$ and T is a qual tree, $X \neq \varnothing$ (attribute connectivity). Furthermore, every node along the path

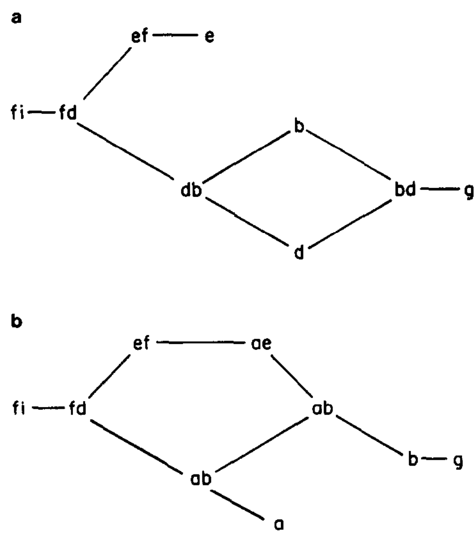


FIG. 7. Deleting the intersection of the supersets in $\mathbf{D}$ of $\mathbf{R}$ and $\mathbf{S}$ does not disconnect those supersets: Consider the Aring and Aclique in Fig. 2. (a) Let $\mathbf{R} = cd$ and $\mathbf{S} = ce$ in the Aring. They have supersets cda and ace, respectively. Deleting ac does not disconnect d and e; (b) Let $\mathbf{R} = bcd$ and $\mathbf{S} = cda$ in the Aclique. They have supersets bcd and cda, respectively. Deleting cd does not disconnect b and a.

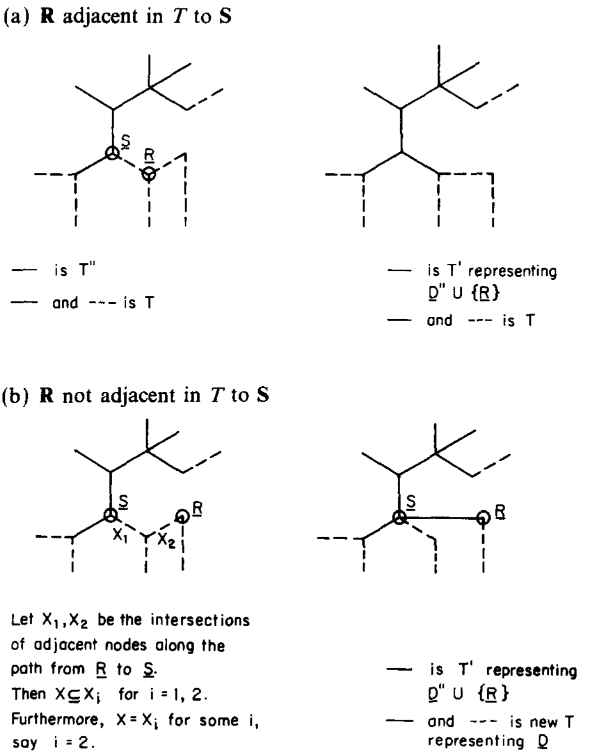


FIG. 8. Constructing T' to represent $\mathbf{D}' = \mathbf{D}'' \cup \{\mathbf{R}\}$.

from **R** to **S** contains X, so any pair of adjacent nodes along the path in T between **R** and **S** must have intersection containing X. One of these intersections must be equal to X; otherwise, deleting $\mathbf{R} \cap \mathbf{S}$ does not disconnect **R** and **S**. By joining **R** and **S** via an edge and deleting that edge along the path with intersection exactly X, we have a new qual tree for **D** and a connected subgraph T' representing $\mathbf{D}'' \cup \{\mathbf{R}\}$ (see 8b). This completes the induction.

(iii) $\Rightarrow$ (ii) Let $\mathbf{R}, \mathbf{S} \in \mathbf{D}$, $\mathbf{R} \cap \mathbf{S} \neq \varnothing$, comprise a connected schema $\mathbf{D}'$. Then there is a subtree T representing **D** such that **R** and **S** are adjacent in T. Deleting $\mathbf{R} \cap \mathbf{S}$ obviously disconnects $\mathbf{R} - (\mathbf{R} \cap \mathbf{S})$ and $\mathbf{S} - (\mathbf{R} \cap \mathbf{S})$. ∎

In the above theorem, (i) $\Leftrightarrow$ (iii) can be proved easily using Corollary 5.2 and Fagin's result (*). The proof above, however, is by qual graph techniques and independent of (*). Fagin's result can now be obtained as part of the following corollary:

COROLLARY 5.3. *The following are equivalent*:

(i) $\mathbf{D}$ *is* γ*-acyclic.*

(ii) *For all connected* $\mathbf{D}' \subseteq \mathbf{D}$, $\mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$.

(iii) *For all connected* $\mathbf{D}' \subseteq \mathbf{D}$, $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$.

(iv) *For all connected* $\mathbf{D}' \subseteq \mathbf{D}$, $\bowtie \mathbf{D} \models \bowtie \mathbf{D}'$.

Proof. (i) $\Rightarrow$ (ii) By Theorem 5.3 ((i) $\Rightarrow$ (iii)), $\mathbf{D}$ is a tree schema and every connected $\mathbf{D}' \subseteq \mathbf{D}$ is a subtree of $\mathbf{D}$. Hence, $\mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$ (Theorem 3.1(ii)).

(ii) $\Rightarrow$ (i) Let $\mathbf{R} \in \mathbf{D}$. Since $\mathrm{GR}(\mathbf{D}, \mathbf{R}) \subseteq (\mathbf{R})$, we have $\mathrm{GR}(\mathbf{D}) = \varnothing$. Hence $\mathbf{D}$ is a tree schema (Corollary 3.1). By Theorem 3.1(ii), every connected $\mathbf{D}' \subseteq \mathbf{D}$ is a subtree of $\mathbf{D}$, so $\mathbf{D}$ is γ-acyclic (Theorem 5.3(iii) $\Rightarrow$ (i)).

(ii) $\Rightarrow$ (iii) By Theorem 3.3(iii), since $\mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$, we have $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) = \mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}'))$. Hence, $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) \subseteq \mathbf{D}'$ for all connected $\mathbf{D}' \subseteq \mathbf{D}$.

(iii) $\Rightarrow$ (ii) Suppose $\mathbf{D}$ is an Aring (or Aclique). Let $\mathbf{D}' = \mathbf{D} - \{\mathbf{R}\}$ for any $\mathbf{R} \in \mathbf{D}$. Then (iii) would not be true since $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) = \mathbf{D}$. It is straightforward to prove that a general cyclic schema cannot satisfy (iii). Hence, (iii) implies that $\mathbf{D}$ is a tree schema. By Theorem 3.3(ii), we have $\mathrm{CC}(\mathbf{D}, \bigcup(\mathbf{D}')) = \mathrm{GR}(\mathbf{D}, \bigcup(\mathbf{D}'))$, so (ii) follows from (iii).

(iii) $\Leftrightarrow$ (iv) Theorem 5.1 ((i) $\Leftrightarrow$ (ii)). ∎

6. Solving Queries Using Joins, Semijoins, and Projects

We are interested in solving $(\mathbf{D}, X)$ by translating it into a *program* that computes the result when applied to any UR database for $\mathbf{D}$. A program is a finite sequence of statements. A statement is one of:

(Join statement) $R_k := R_i \bowtie R_j$—a newly created relation R_k is assigned the values of $R_i \bowtie R_j$.

(Project statement) $R_k := \prod_X (R_i)$— a newly created relation R_k is assigned the value $\prod_X (R_i)$.

(Semijoin statement) $R_k := R_i \ltimes R_j$— a newly created relation R_k is assigned the value of $R_i \ltimes R_j$.

P solves $(\mathbf{D}, X)$ if for all UR databases for $\mathbf{D}$, the value produced by the last statement of P is the answer to $(\mathbf{D}, X)$.

We can think of P as mapping the original databases *schema and state* into a new schema and state. The new schema is the original schema plus the relation schemas for the relations created by the statements of P. The new database state assigns to each original relation schema its original relation state, and to each new relation schema the relation produced by the appropriate statement. We use $P(\mathbf{D})$ to denote the schema part of this mapping and $P(D)$ for the state part.

Some relation states may be considered irrelevant in solving $Q = (\mathbf{D}, X)$ over a universal database. Consider the following example: Let $\mathbf{D} = (\mathbf{R}_1 = abg, \mathbf{R}_2 = bcg, \mathbf{R}_3 = acf, \mathbf{R}_4 = ad, \mathbf{R}_5 = de, \mathbf{R}_6 = ea)$ and let $Q = (\mathbf{D}, abc)$. Clearly, to solve Q, $\mathbf{R}_4$, $\mathbf{R}_5$, and $\mathbf{R}_6$ are irrelevant, as is the f column in $\mathbf{R}_3$. Hence given $D = (R_1, ..., R_6)$ over $\mathbf{D}$, we can solve $(\mathbf{D}', abc)$, where $D' = (R_1, R_2, \prod_{ac} R_3)$.

Formally, the relations that remain correspond to elements of $\mathrm{CC}(\mathbf{D}, X)$. In addition, columns corresponding to attributes that appear in only one such relation, and are not in X, are "projected out." Thus, we basically perform tableau minimization and then eliminate useless columns.

For general databases (i.e., not necessarily UR), Goodman and Shmueli [13] proved the following theorem (recall the definition of tree projection (TP) from Section 3.2).

THEOREM 6.1 [13] (Tree Projection Sufficiency). *If there exists* $\mathbf{D}'' \in \mathrm{TP}(P(\mathbf{D}), \mathbf{D} \bigcup (X))$, *then P augmented by at most* $2 \cdot |\mathbf{D}|$ *semijoins solves* $(\mathbf{D}, X)$.

In light of the discussion above, we immediately obtain a specialization of Theorem 6.1 to UR databases.

THEOREM 6. [13] *If there exists* $\mathbf{D}'' \in \mathrm{TP}(P(\mathbf{D}), \mathrm{CC}(\mathbf{D}, X) \cup (\mathbf{X}))$, *then P augmented by at most* $2 \cdot |\mathrm{CC}(\mathbf{D}, X)|$ *semijoins solves* $(\mathbf{D}, X)$ *over all* UR *databases.*

For general databases, the following theorem characterizes the necessary actions P has to perform in order to solve $(\mathbf{D}, X)$.

THEOREM 6.3 [13] (Tree Projection Necessity). *If P solves* $(\mathbf{D}, X)$ *then there exists* $\mathbf{D}'' \in \mathrm{TP}(P(\mathbf{D}), \mathbf{D} \cup (X))$.

One may suspect that the UR property may weaken the above theorem when restricted to UR databases. Indeed, in light of Theorem 6.2, we know that $\mathbf{D}'' \in \mathrm{TP}(P(\mathbf{D}), \mathrm{CC}(\mathbf{D}, X) \cup (X))$ is sufficient. However, no further weakening of Theorem 6.3 is possible. By tracing the proof of Theorem 6.3 with $\mathrm{CC}(\mathbf{D}, X)$ replacing $\mathbf{D}$, we obtained the following.

THEOREM 6.4 [13] *If P solves* $(\mathbf{D}, X)$ *on all* UR *databases, then there exists* $\mathbf{D}'' \in \mathrm{TP}(P(\mathbf{D}), \mathrm{CC}(\mathbf{D}, X) \cup (X))$.

Theorems 6.1–6.4 are important for two reasons. First, they show that forming a tree projection is the crux of the query processing problem for both UR and non-UR databases. Second, the justaposition of the UR and non-UR results gives a clear indication of the value of the UR property for query processing. The UR property is helpful to the extent that $\mathrm{CC}(\mathbf{D}, X)$ is smaller than $\mathbf{D}$; but once $\mathrm{CC}(\mathbf{D}, X)$ has been taken, the benefit of UR is "used up."

7. Conclusions

Recent work has tied together relational database theory and graph theory (especially hypergraphs). Tree and cyclic schemas, tree projections and the GYO reduction have proved useful in analyzing problems in query processing, schema design, and dependency theory. The canonical connection introduced an essential link between tableaux (i.e., expressions) and schemas.

This paper presents a coherent organization of results concerning the canonical connection and lossless joins. Our results strengthen existing ones, and as corollaries we get previously known results. We exhibited a relationship between tree projections and canonical connections associated with query processing. In analyzing lossless joins, we provided a new characterization for γ-acyclic databases, and showed how to prove some of their properties using graph techniques.

Some new results concerning the GYO reduction were presented. In particular, we proved that if the addition of relation $\mathbf{R}$ to schema $\mathbf{D}$ transforms it into a tree schema then $\mathbf{R} \supseteq \mathrm{GR}(\mathbf{D})$. Such a simple characterization was not found for the case where more than one relation is added; in this case we run into *NP*-completeness.

Acknowledgment

We thank the referees for their careful reading of the manuscript and their many helpful suggestions.

References

1. A. V. Aho, C. Beeri, and J. D. Ullman, The theory of joins in relational databases, *ACM Trans. Database Systems* **4**, No. 3 (1979), 297–314.
2. A. V. Aho, Y. Sagiv, and J. D. Ullman, Equivalence among relational expressions, *SIAM J. Comput.* **8**, No. 2 (1979), 218–246.
3. C. Beeri, R. Fagin, D. Maier, A. Mendelzon, J. D. Ullman, and M. Yannakakis, Properties of acyclic database schemes, *in* "Proc. ACM Symposium on Theory of Computation, NewYork, 1981." pp. 353–362.
4. C. Beeri, R. Fagin, D. Maier, and M. Yannakakis, On the desirability of acyclic database schemes, *J. Assoc. Comput. Mach.* **30**, No. 3 (1983), 479–513.
5. P. A. Bernstein and D. W. Chiu, Using semijoins to solve relational queries, *J. Assoc. Comput. Mach.* **28**, No. 1 (1981), 25–40.
6. P. A. Bernstein and N. Goodman, The power of natural semijoins, *SIAM J. Comput.* **10**, No. 4 (1981), 751–771.
7. R. Fagin, Types of acyclicity for hypergraphs and relational database systems, *J. Assoc. Comput. Mach.* **30**, No. 3 (1983), 514–550.
8. R. Fagin, A. O. Mendelzon, and J. D. Ullman, A simplified universal relation assumption and its properties, *ACM Trans. Database Systems* **7**, No. 3 (1982), 343–360.
9. M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of *NP*-Completeness," Freeman, San Francisco, 1979.
10. M. H. Graham, "On the Universal Relation," Technical Report, Univ. of Toronto, 1979.

11. N. GOODMAN AND O. SHMUELI, Tree queries: A simple class of relational queries, *ACM Trans. Database Systems* **7**, No. 4 (1982), 653–677.
12. N. GOODMAN AND O. SHMUELI, Syntactic characterizations of database schemas, *J. Assoc. Comput. Mach.* **30**, No. 4 (1983), 767–786.
13. N. GOODMAN AND O. SHMUELI, The tree projection theorem and relational query processing, *J. Comput. System Sci.* **28**, No. 1 (1984), 60–79.
14. N. GOODMAN AND O. SHMUELI, Transforming cyclic schemas into trees, *in* "Proc. ACM SIGACT-SIGMOD Conference on Principles of Database Systems, Los Angeles, (1982)," pp. 49–54.
15. R. HULL, Acyclic join dependencies and database projections, *J. Comput. System Sci.* **27**, No. 3 (1983), 331–349.
16. D. MAIER AND J. D. ULLMAN, Connections in acyclic hypergraphs, *in* "Proc. ACM SIGACT-SIGMOD Conference on Principles of Database Systems, Los Angeles, 1981," pp. 34–39.
17. J. RISSANEN, Theory of joins for relational databases—a tutorial survey, *in* "Proc. Symposium on Mathematical Foundations of Computer Science No. 64," pp. 537–551, Springer Verlag Berlin/New York, 1978.
18. M. YANNAKAKIS, Algorithms for acyclic database schemes, *in* "Proc VLDB, Cannes France, 1981," pp. 82–94.
19. C. T. YU AND M. Z. OZSOYOGLU, An algorithm for tree-query membership of a distributed query, *in* "Proc. IEEE COMPSAC, 1979."