British Journal of Mathematics & Computer Science
4(23): 3286-3293, 2014
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Set-theoretic Foundations of the Modern Relational Databases: Representations of Table Algebras Operations

As'ad Mahmoud As'ad Alnaser[1*]

[1]Department of Computer Science, Al-Balqa' Applied University, Ajlun University College, P.O. Box 158, Postal Code 21166, Irbid-Aidun, Jordan.

Article Information

DOI: 10.9734/BJMCS/2014/13330
Editor(s):
(1) Dragoş - Pătru Covei, Department of Mathematics, University "Constantin Brâncuşi" of Târgu-Jiu, România.
Reviewers:
(1) Anonymous, Mathematics, Rajshahi University, Rajshahi, Bangladesh.
(2) Leonardo Simal Moreira, UniFoa – Centro Universitário de Volta Redonda, Brazil.
Peer review History: http://www.sciencedomain.org/review-history.php?iid=669&id=6&aid=6200

Original Research Article

Received: 13 August 2014
Accepted: 06 September 2014
Published: 24 September 2014

Abstract

The article is dedicated to the creation of the fragment of table algebras theory constructed on the basis of classical relational Codd's algebras. The distinctive peculiarity of the adapted technique is the use of set-theoretic properties of some constructions (full image of the set with respect to the function, function restriction with respect to the set, generalized direct (Cartesian) product, binary relation of functions compatibility) and their transference on a case of tables. The transference of these properties is possible in view of simplicity of signature operations representations in terms of indicated set-theoretic constructions.

Keywords: Relational Codd's algebras, table algebras, restriction, generalized direct product, compatibility relation.

1 Introduction

The table algebras theory forms a theoretic base of modern table databases query languages. The papers [1-6] are dedicated to the investigation of table algebras. The carrier elements of these algebras are the models of the relational data structures, the signature operations are constructed on the basis of basic table manipulations in relational Codd's algebras [7-10] and in SQL-like languages: the union, intersection, difference, selection, projection, join, active complement, renaming and grouping mechanisms on example of division operation.

Corresponding author: asad1_99@yahoo.com;

2 Table Algebras: Basic Definitions

Examinating data structures were abstracted from different attributes' domains (i.e. we consider one universal domain), from special value NULL as well as from duplicatoins of the rows in the table without a primary key (PRIMARY KEY) [6].

Let's make more precise the tables (the relations) in terms of nominal sets [11-13]. For this we fix two sets: $\boldsymbol{A}$, whose elements are called attributes, and $\boldsymbol{D}$ – the universal domain.

The arbitrary finite set of attributes $R \subseteq \boldsymbol{A}$ is called scheme. We'll denote schemes by R, R_1, R_2, ….

Row of the scheme R is a nominal set on a pair R, $\boldsymbol{D}$, in which projection on the first component equals to R. Thus, the row of the scheme R is a function of the form $s : R \to D$. The rows are denoted by $s, s_1, s_2, \ldots$.

Table of the scheme R is called a finite set of rows of the scheme R. Thus, table of the scheme R is a finite set of the form $t = \{ S_1, \ldots, S_n \}$, where $n \geq 0$ and $s_i : R \to D$. The tables are defined as t, t_1, t_2, …. For $n = 0$ we have a case of an empty table, i.e. a table which does not contain rows. The set of all rows (tables) of the scheme R are defined as $S(R)$ ($T(R)$, respectively), the set of all rows (tables) are defined as S (T, *respectively)*. Thus, we'll have $S \overset{def}{=} \bigcup_{R \subseteq A} S(R)$, $T(R) \overset{def}{=} 2^{S(R)}$, $T \overset{def}{=} \bigcup_{R \subseteq A} T(R)$.

Here, 2^X denotes a set of all finite subsets of set X.

The scheme can be empty, then in addition there exists a single empty scheme row which is defined by ε (empty row is an undefined function).

The set of all tables T is a carrier of a table algebra. We shall define signature's operations which have natural representations in terms of set-theoretic constructions like full image constructions, restriction and generalized direct product.

Here $s \mid X$ is a restriction of the row s on set X which is defined standard as $s \mid X = s \cap (X \times D)$. The restriction of binary relation on a set is introduced completely analogously: $U \mid X \overset{def}{=} U \bigcap (X \times \pi_2^2 U)$, where $\pi_i^2 (U)$ is a projection of a binary relation on *i-th* component.

Let $X \subseteq \boldsymbol{A}$ is a finite set of attributes.

Definition 1. The unary parametric operation π_X is a *projection on a set of attributes* X. In this case its values are the tables which consist of restrictions of all rows of initial tables on set of attributes X.

$$\text{Thus, } \pi_X : T \to T,\ \pi_X(t) \overset{def}{=} \{s \mid X \mid s \in t\}.$$

Definition 2. The binary operation ⊗ is a *join.* Under join we understood binary operation, whose values are tables consisting of all unions of compatible rows of the tables-arguments.

$$\text{Thus, } \otimes : T \times T \to T,\ t_1 \otimes t_2 \overset{def}{=} \{s_1 \bigcup s_2 \mid s_1 \in t_1 \wedge s_2 \in t_2 \wedge s_1 \approx s_2\}.$$

Here $\approx$ is a binary relation of binary relations compatibility (in particular, the compatibility of the functions and rows): $U_1 \approx U_2 \overset{def}{\Leftrightarrow} U_1 \mid X = U_2 \mid X$, where $X = \pi_1^2 U_1 \bigcap \pi_1^2 U_2$ is an intersection of projections on first component of initial binary relations.

Considering the definition of join it is ought to take into account that the functionality property is broken in general case of functions union. Therefore, to justify the correctness of the join definition it is necessary to take into account the functionality of the union of functions, which are compatible [6].

The definition implies that $t_1 \otimes t_2 = \{s \in S(R_1 \bigcup R_2) \mid s \mid R_1 \in t_1 \wedge s \mid R_2 \in t_2\}$, where $t_1 \in T(R_1)$, $t_2 \in T(R_2)$. This is a definition of a *natural join* or *equijoin* [14].

Reference [6,14] introduced the operation of active addition , which approximates a set-theoretic addition in a certain sence. Let's consider several subsidiary concepts necessary for an active addition introduction.

Definition 3. Let *A* is an attribute, *t* is a table of the scheme *R*. Then $D_{A,t} \overset{def}{=} \{d \mid \exists s (s \in t \wedge < A, d > \in s)\}$ is an *active domain of attribute A* with respect to table *t* according to the terminology in [14].

Definition 4. Assume $C(t) \overset{def}{=} \underset{A \in R}{N} D_{A,t}$. For an empty table we need to choose a nonempty scheme, where, as previously, *R* is the scheme of table *t*.

Where, *C*(*t*) is called *saturation of table t* [15,16] and *N* is an operator of constructing of a set of all nominal sets with respect to parameter-indexing. We shall give a general definition.

Let's fix sets *V* and Σ. Let $\{\Sigma_v\}_{v \in U}$ (i.e. $v \mapsto \sum_v$) is a certain indexing of denotation subsets $\Sigma_v \subseteq \Sigma$ by names of a (finite) set of names $U \subseteq V$.

Definition 5. Under the *nominal set corresponding* to the given *indexing* a nominal set on a pair $U, \bigcup_{v \in U} \Sigma_v$ is understood which is the projection on first component coincides with set *U* and every $v \in U$ value (denotat) belongs to the set Σ_v.

The set of all such nominal sets are denoted by $\underset{v \in U}{N} \Sigma_v$. By definition, we assume that there exists a single nominal set which corresponds everywhere undefined indexing. This empty nominal set is denoted by ε. Thus, $\underset{v \in \varnothing}{N} \Sigma_v \overset{def}{=} \{\varepsilon\}$.

Note that the operator for construction of set of all nominal sets corresponding the indexing coincides, in fact, with a set-theoretic construction of a generalized direct (Cartesian) product $\underset{v \in U}{N} \Sigma_v = \prod_{v \in U} \Sigma_v$ [17].

Definition 6. Under *active complement* an unary operation is understood ~ which corresponds every table its complement up to the saturation.

$$\text{Thus, } \sim: T \to T,\ \tilde{t} \overset{def}{=} C(t) \setminus t.$$

Under *renaming* an unary, generally speaking partial, parametric operation of the form $Rt_\xi : T_\xi \tilde{\to} T_\xi$ is understood, where the parameter is an injective , generally speaking partial, map defined on an attributes set $\xi : \boldsymbol{A} \tilde{\to} \boldsymbol{A}$. The operation only renames tables-arguments attributes according to parameter.

The arrow of the form $\tilde{\to}$ is used to denote partial operations .

The formal definion of renaming requires additional notions. The table renaming is a renaming of all attributes of its scheme, therefore table renaming is reduced to renaming its rows, where two such renaming are connected among themselves by a full image construction. In turn, the row renaming is renaming of first pair components, e.g. the row elements. Let's go down to precise definions.

Let $\eta : \boldsymbol{A} \to \boldsymbol{A}$ is a certain total function of attributes renaming, generally speaking, not an injective function, we denote the set of all finite binary relations on a pair of sets $\boldsymbol{A}, \boldsymbol{D}$ by S'.

Definition 7. Under *rows renaming* corresponding to function of attributes renaming η is understood the map of the form

$$Rs_\eta : S \to S', Rs_\eta(s) \overset{def}{=} \{ < \eta(A), s(A) > \mid A \in \pi_1^2(s)\},\ Rs_\eta(\varepsilon) = \varepsilon.$$

Evidently the functionality rows property can be broken in case of such renaming, we require the attributes renaming function η to have the following structure: we fix an injective partial function of the form $\xi : \boldsymbol{A} \tilde{\to} \boldsymbol{A}$ and assume $\eta \overset{def}{=} \xi \cup \mathrm{id}_{\boldsymbol{A} \setminus dom \xi}$.

Here $\mathrm{id}_X : X \to X$ is the identical fuction (diagonal) defined on set X.

Informally speaking, attributes of the set $dom\xi$ (i.e. the attributes of domain of map ξ) are renamed, attributes of the differences $\boldsymbol{A} \backslash dom\xi$ are remained the same. This corresponds to situation when only part of attributes of database tables is renamed globally, the rest attributes remain unchanged.

Definition 8. Scheme R is called ξ-admissible, if $\xi[R] \cap (R \setminus dom\xi) = \varnothing$, where $\xi[R] \overset{def}{=} \{A \mid \exists B(B \in R \wedge \xi(B) \simeq A\}$ is a full image of scheme R with respect to map ξ. Here $\simeq$ – is a generalized equality (strong Kleene's equality) [18].

The set of all tables of ξ-admissable schemes are defined by T_ξ. It is easily to check that the functionality property is not broken in case of renaming the rows of ξ-admissable schemes [6].

Definition 9. *Renaming corresponding to injective partial renaming attributes function* $\xi : \boldsymbol{A} \tilde{\to} \boldsymbol{A}$ is understood as unary parametric operation Rt_ξ which domain is T_ξ and the values are set by equality $Rt_\xi(t) = Rs_\eta[t]$, $t \in T_\xi$, where, as earlier, $\eta = \xi \cup \mathrm{id}_{\boldsymbol{A} \backslash \mathrm{dom}\xi}$ and the right part of the last equality is a full image of table-argument.

Assume by definition

$$\Omega_{P,\Xi} \overset{def}{=} \left\{ \cup_R, \cap_R, \backslash_R, \sigma_p, \pi_X, \otimes, \div_{R_2}^{R_1}, Rt_\xi, \sim \right\}_{X,R,R_1,R_2 \subseteq \boldsymbol{A}}^{p \in P, \xi \in \Xi},$$

where P, Ξ are parameter's sets (for selection and renaiming operations respectively).

The operations of union, intersection, difference, selection and division, which definitions are omitted, are understood in standard way [6].

Definition 10. The partial (parametric) algebra $\langle T; \Omega_{P,\Xi} \rangle$ will be called a table algebra.

Note, that division and intersection are derived with respect to other signature operations [3].

For Examinating tables and their schemes we need to take into account some peculiarities. By definition the empty set or rows in the table is called *empty table* and denoted by $t_\varnothing$. As far as exists single nominal set on pair $\varnothing$, D, namely, empty nominal set ε, then there exists a single row ε of a scheme $\varnothing$; i.e. $S(\varnothing) = \{\varepsilon\}$. Concerning empty scheme $\varnothing$ tables: there are two of them – $t_\varnothing$ and $t_\varepsilon \overset{def}{=} \{\varepsilon\}$. Really, $T(\varnothing) = 2^{S(\varnothing)} = \{\varnothing, \{\varepsilon\}\} = \{t_\varnothing, t_\varepsilon\}$.

At last, evidently, the scheme of non-empty table can be uniquely recover by the table itself, and we can ascribe any scheme to empty table, e.g. $t_{\varnothing} \in T(R)$ for any scheme *R*. The last assertion can be written in form

$$\forall R_1, R_2 (R_1 \neq R_2 \Rightarrow T(R_1) \bigcap T(R_2) = \{t_{\phi}\}.$$

3 Representations of Signature Operations of Tabular Algebra

The introduced signature operations of table algebras have natural simple representations in terms of set-theoretic constructions; namely these representations allow relatively easy to transfer properties of full image, restriction, generalized direct product and other constructions on a table case.

In the following statesments all designations are used in sense of the monograph [6], in particular, $\bar{\bigcup} : S \times S \tilde{\rightarrow} S$, $dom \bar{\bigcup} = \{< s_1, s_2 > | s_1 \approx s_2\}$, $s_1 \bar{\bigcup} s_2 = s_1 \bigcup s_2$ for all $< s_1, s_2 > \in dom \bar{\bigcup}$.

Lemma 1 (about projection representation). The projection has the following representation:

$$\pi_X(t) = \uparrow X[t], \text{ where } \uparrow X : S \rightarrow S, \uparrow X(s) = s \mid X . \tag{1}$$

The proof implys from definitions of projection and restriction.

Lemma 2 (about join representation). The join has the following representation:

$$t_1 \otimes t_2 = \bar{\bigcup}[t_1 \times t_2]. \tag{2}$$

Proof. We show that the left part of equality (2) is included the right one, the right part is included left one. Thus, let $s \in t_1 \otimes t_2$, then there are rows $s_1 \in t_1$ and $s_2 \in t_2$ such, that $s_1 \approx s_2$ and $s = s_1 \cup s_2$. It remains to note accessories $< s_1, s_2 > \in dom \bar{\bigcup}$ and full image definition. Now let $s \in \overline{\cup}[t_1 \times t_2]$, then there are rows $s_1 \in t_1$ and $s_2 \in t_2$, such that $< s_1, s_2 > \in dom \bar{\bigcup}$ and $s = s_1 \cup s_2$. The definition of operation $\bar{\bigcup} : S \times S \tilde{\rightarrow} S$ implies that $s_1 \approx s_2$. From here follows that $s \in t_1 \otimes t_2$.

Lemma 3 (about saturation representation). The saturation has the following representation:

$$C(t) = \prod_{A \in R} D_{A,t}, \tag{3}$$

Where $D_{A,t} \overset{def}{=} \{d \mid \exists s(s \in t \wedge < A, d > \in s)\}$ is an active domain of attribute A with respect to table t, R is the scheme of the table t.

The proof implies from definitions of saturation and generalized direct product.

Lemma 4 (about renaiming representation). The renaiming has the following representation:

$$Rt_{\xi}(t) = Rs_{\eta}[t], \text{ where } \eta \overset{def}{=} \xi \cup \mathrm{id}_{A \setminus dom\xi}, t \in T_{\xi} . \quad (4)$$

The proof implies from definitions of tables and rows renamining.

Theorem 1 (about representations of signature operations of table algebras). The projection, join, saturation and renaming have the representations (1)-(4) respectively.

The proof implies from lemmas (1)-(4).

Thus, the representations of basic operations of table algebras through set-theoretic constructions of full image, restriction and direct product have been established. It is worth to note that the presence of these representations allows to get properties of table operations in consequences of properties of indicated set-theoretic constructions. This will be done in the next part of the paper.

4 Conclusion

Informative fragment of table algebras theory has been constructed in this paper. The main peculiarity of adapted technique is contained in establishment of natural representations of basic signature operations of these algebras (projection, join, saturation, renaming) in terms of set-theoretic constructions of full image, restriction, (generalized) direct product, compatibility relation. In the second part of the paper the properties of considered set-theoretic constructions will be transferred on table algebra.

Competing Interests

Author has declared that no competing interests exist.

References

[1] Julia Brona. Key relationships in tablle algebras. Bulletin of Kiev University. Series: Physics & Mathematics. 1997;3:143-148.

[2] Bui Dmitry. Ms. Brona, Set-theoretic structures in relational database theory. Bulletin of Kiev University. Series: Physics & Mathematics. 1996;1:216-224.

[3] Red'ko VN, Bui DB. Foundations of the theory of relational database models. Cybernetics and Systems Analysis. 1996;4:3-12.

[4] Red'ko VN, Brona YI, Bui DB. Information aspects of Case-technologies: Basic relations in table algebras. Problems of programming. 1997;1:5-11.

[5] Red'ko VN, Brona YI, Bui DB. Relational algebra: Operations of division and renaming. Cybernetics and Systems Analysis. 1997;5:3-15.

[6] Red'ko VN, Brona YI, Bui DB, Polyakov SA. Relational databases: Table algebras and SQL-like language. Kyiv: Publishing House "Akademperiodika". 2001;198.

[7] Codd EF. A relational model of data for large shared data banks. Communications of the ACM. 1970;13(6):377-387.
Available: http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf

[8] Codd EF. Further normalization of data base relational model. Data Base Systems. 1972;33-64.

[9] Codd EF. Relational completeness of data base sublanguages. Data Base Systems. 1972;65-93. Available:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.9277&rep=rep1&type=pdf

[10] Codd EF. Relational Database: A Practical Foundation for Productivity. Communications of the ACM. 1982;25(2):109-117.

[11] Red'ko Vladimir. Compositions of programs and compositional programming. Programming. 1978;5:3-24.

[12] Red'ko Vladimir. The foundations of compositional programming. Programming. 1979;3:3-13.

[13] Red'ko Vladimir. Semantic structure of programs. Programming. 1981;1:3-19.

[14] Maier D. The theory of relational databases. Computer Science Press, Rockville, MD. 1983;608.

[15] Tsalenko Mihail. Semantic and mathematical models of databases. Moscow: VINITI. 1985;9:207.

[16] Tsalenko Mihail. Modeling of semantics for databases. Moscow: Nauka. 1989;288.

[17] Kuratovski K. Topology. Academic Press. 1966;1:594.

[18] Cutland N. Computability. An introduction to recursive function theory. Cambridge University Press. 1980;256.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=669&id=6&aid=6200