

Set-theoretic foundations of the modern relational databases: representations of table algebras operations

Author details:

As'ad Mahmoud As'ad Alnaser

**Department of Computer Science, Al-Balqa' Applied University, Ajlun
University College**

P.O. Box 158, Postal Code 21166, Irbid-Aidun, Jordan

E-mail address: asad1_99@yahoo.com

asad1_99@yahoo.com

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, 2, 3, 4, 5, 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, 8, 9, 10] and in SQL-like languages: the union, intersection, difference, selection, projection, join, active complement, renaming and grouping mechanisms on example of division operation.

2. TABLE ALGEBRAS: BASIC DEFINITIONS

Examining 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].

41 Let's make more precise the tables (the relations) in terms of nominal sets [11, 12, 13]. For
 42 this we fix two sets: $\mathbf{A}$, whose elements are called attributes, and $\mathbf{D}$ – the universal domain.
 43 The arbitrary finite set of attributes $R \subseteq \mathbf{A}$ is called scheme. We'll denote schemes by R ,
 44 $R_1, R_2, \dots$.

45 Row of the scheme R is a nominal set on a pair $R, \mathbf{D}$, in which projection on the first
 46 component equals to R . Thus, the row of the scheme R is a function of the form $s : R \rightarrow \mathbf{D}$.

47 The rows are denoted by $S, S_1, S_2, \dots$.

48 Table of the scheme R is called a finite set of rows of the scheme R . Thus, table of the
 49 scheme R is a finite set of the form $t = \{S_1, \dots, S_n\}$, where $n \geq 0$ and

50 $s_i : R \rightarrow \mathbf{D}$. The tables are defined as $t, t_1, t_2, \dots$. For $n = 0$ we have a case of an
 51 empty table, i.e. a table which does not contain rows. The set of all rows (tables) of the
 52 scheme R are defined as $S(R)$ ($T(R)$), respectively, the set of all rows (tables) are defined as

53 $S(T, \text{ respectively })$. Thus, we'll have $S \stackrel{\text{def}}{=} \bigcup_{R \subseteq \mathbf{A}} S(R)$, $T(R) \stackrel{\text{def}}{=} 2^{S(R)}$, $T \stackrel{\text{def}}{=} \bigcup_{R \subseteq \mathbf{A}} T(R)$.

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

55 The scheme can be empty, then in addition there exists a single empty scheme row which is
 56 defined by $\mathcal{E}$ (empty row is an undefined function).

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

60

61 Here $s \upharpoonright X$ is a restriction of the row S on set X which is defined standard as
 62 $s \upharpoonright X = s \cap (X \times \mathbf{D})$. The restriction of binary relation on a set is introduced completely

63 analogously: $U \upharpoonright X \stackrel{\text{def}}{=} U \cap (X \times \pi_2^2 U)$, where $\pi_i^2(U)$ is a projection of a binary relation on i -
 64 th component.

65

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

67

68 **Definition 1.** The unary parametric operation π_X is a projection on a set of attributes X .

69 In this case its values are the tables which consist of restrictions of all rows of initial tables
 70 on set of attributes X .

71 Thus, $\pi_X : T \rightarrow T$, $\pi_X(t) \stackrel{\text{def}}{=} \{s \upharpoonright X \mid s \in t\}$.

72

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

75 Thus, $\otimes : T \times T \rightarrow T$, $t_1 \otimes t_2 \stackrel{\text{def}}{=} \{s_1 \cup s_2 \mid s_1 \in t_1 \wedge s_2 \in t_2 \wedge s_1 \approx s_2\}$.

76 Here $\approx$ is a binary relation of binary relations compatibility (in particular, the compatibility of
 77 the functions and rows): $U_1 \approx U_2 \stackrel{\text{def}}{\Leftrightarrow} U_1 \upharpoonright X = U_2 \upharpoonright X$, where $X = \pi_1^2 U_1 \cap \pi_1^2 U_2$ is

78 an intersection of projections on first component of initial binary relations.

79 Considering the definition of join it is ought to take into account that the functionality property
 80 is broken in general case of functions union. Therefore, to justify the correctness of the join

81 definition it is necessary to take into account the functionality of the union of functions, which
82 are compatible [6].

83 The definition implies that $t_1 \otimes t_2 = \{s \in S(R_1 \cup R_2) \mid s \mid R_1 \in t_1 \wedge s \mid R_2 \in t_2\}$, where
84 $t_1 \in T(R_1)$, $t_2 \in T(R_2)$. This is a definition of a *natural join* or *equijoin* [14].

85
86

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

90

91 **Definition 3.** Let A is an attribute, t is a table of the scheme R . Then

92 $D_{A,t} \stackrel{\text{def}}{=} \{d \mid \exists s(s \in t \wedge \langle A, d \rangle \in s)\}$ is an *active domain of attribute A* with respect to table t
93 according to the terminology in [14].

94

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

97 Where, $C(t)$ is called *saturation of table t* [16, 17] and N is an operator of constructing of a
98 set of all nominal sets with respect to parameter-indexing. We shall give a general definition.

99 Let's fix sets V and Σ . Let $\{\Sigma_v\}_{v \in U}$ (i.e. $v \mapsto \Sigma_v$) is a certain indexing of denotation

100 subsets $\Sigma_v \subseteq \Sigma$ by names of a (finite) set of names $U \subseteq V$.

101

102 **Definition 5.** Under the *nominal set corresponding* to the given *indexing* a nominal set on
103 a pair $U, \bigcup_{v \in U} \Sigma_v$ is understood which is the projection on first component coincides with set

104 U and every $v \in U$ value (denotat) belongs to the set Σ_v .

105 The set of all such nominal sets are denoted by $N \Sigma_v$. By definition, we assume that there
106 exists a single nominal set which corresponds everywhere undefined indexing. This empty

107 nominal set is denoted by $\mathcal{E}$. Thus, $N \Sigma_v \stackrel{\text{def}}{=} \{\mathcal{E}\}$.

108 Note that the operator for construction of set of all nominal sets corresponding the indexing
109 coincides, in fact, with a set-theoretic construction of a generalized direct (Cartesian) product

110 $N \Sigma_v = \prod_{v \in U} \Sigma_v$ [15].

111

112 **Definition 6.** Under *active complement* an unary operation is understood $\sim$ which
113 corresponds every table its complement up to the saturation.

114 Thus, $\sim: T \rightarrow T$, $\tilde{t} \stackrel{\text{def}}{=} C(t) \setminus t$.

115

116 Under *renaming* an unary, generally speaking partial, parametric operation of the form

117 $Rt_\xi: T_\xi \xrightarrow{\sim} T_\xi$ is understood, where the parameter is an injective, generally speaking

118 partial, map defined on an attributes set $\xi : \mathbf{A} \xrightarrow{\sim} \mathbf{A}$. The operation only renames tables-
 119 arguments attributes according to parameter.
 120 The arrow of the form $\xrightarrow{\sim}$ is used to denote partial operations .
 121

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

127 Let $\eta : \mathbf{A} \rightarrow \mathbf{A}$ is a certain total function of attributes renaming, generally speaking, not
 128 an injective function, we denote the set of all finite binary relations on a pair of sets $\mathbf{A}, \mathbf{D}$ by
 129 S' .
 130

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

$$133 \quad Rs_\eta : S \rightarrow S', Rs_\eta(s) = \overset{def}{\{ \langle \eta(A), s(A) \rangle \mid A \in \pi_1^2(s) \}}, Rs_\eta(\varepsilon) = \varepsilon.$$

134
 135 Evidently the functionality rows property can be broken in case of such renaming, we require
 136 the attributes renaming function η to have the following structure: we fix an injective partial

137 function of the form $\xi : \mathbf{A} \xrightarrow{\sim} \mathbf{A}$ and assume $\eta \overset{def}{=} \xi \cup \text{id}_{\mathbf{A} \setminus \text{dom} \xi}$.

138 Here $\text{id}_X : X \rightarrow X$ is the identical function (diagonal) defined on set X .

139 Informally speaking, attributes of the set $\text{dom} \xi$ (i.e. the attributes of domain of map ξ) are
 140 renamed, attributes of the differences $\mathbf{A} \setminus \text{dom} \xi$ are remained the same. This corresponds
 141 to situation when only part of attributes of database tables is renamed globally, the rest
 142 attributes remain unchanged.
 143

144 **Definition 8.** Scheme R is called ξ -admissible, if $\xi[R] \cap (R \setminus \text{dom} \xi) = \emptyset$, where

$$145 \quad \xi[R] \overset{def}{=} \{A \mid \exists B (B \in R \wedge \xi(B) \simeq A)\} \text{ is a full image of scheme } R \text{ with respect to map } \xi.$$

146 Here $\simeq$ – is a generalized equality (strong Kleene's equality) [18].

147 The set of all tables of ξ -admissible schemes are defined by T_ξ . It is easily to check that
 148 the functionality property is not broken in case of renaming the rows of ξ -admissible
 149 schemes [6].
 150

151 **Definition 9.** Renaming corresponding to injective partial renaming attributes function
 152 $\xi : \mathbf{A} \xrightarrow{\sim} \mathbf{A}$ is understood as unary parametric operation Rt_ξ which domain is T_ξ and the
 153 values are set by equality $Rt_\xi(t) = Rs_\eta[t]$, $t \in T_\xi$, where, as earlier,
 154 $\eta = \xi \cup \text{id}_{\mathbf{A} \setminus \text{dom} \xi}$ and the right part of the last equality is a full image of table-argument.
 155 Assume by definition

$$\Omega_{P,\Xi} \stackrel{def}{=} \left\{ \bigcup_R, \bigcap_R, \setminus_R, \sigma_p, \pi_X, \otimes, \div_{R_1}^{R_2}, Rt_\xi, \sim \right\}_{\substack{p \in P, \xi \in \Xi \\ X, R, R_1, R_2 \subseteq A}}$$

where P, Ξ are parameter's sets (for selection and renaming 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 Examining 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 $t_\emptyset$. As far as exists single nominal set on pair $\emptyset, D$, namely, empty nominal set ε , then there exists a single row ε of a scheme $\emptyset$; i.e. $S(\emptyset) = \{\varepsilon\}$. Concerning empty

scheme $\emptyset$ tables: there are two of them - $t_\emptyset$ and $t_\varepsilon \stackrel{def}{=} \{\varepsilon\}$. Really, $T(\emptyset) = 2^{S(\emptyset)} = \{\emptyset, \{\varepsilon\}\} = \{t_\emptyset, t_\varepsilon\}$.

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. $t_\emptyset \in T(R)$ for any scheme R . The last assertion can be written in form

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

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 statements all designations are used in sense of the monograph [6], in

particular, $\bar{\bigcup} : S \times S \rightarrow S$, $dom \bar{\bigcup} = \{ \langle s_1, s_2 \rangle \mid s_1 \approx s_2 \}$, $s_1 \bar{\bigcup} s_2 = s_1 \cup s_2$ for all $\langle s_1, s_2 \rangle \in dom \bar{\bigcup}$.

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

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

The proof implies from definitions of projection and restriction.

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

$$t_1 \otimes t_2 = \bar{\bigcup}[t_1 \times t_2]. \quad (2)$$

194
195 Proof. We show that the left part of equality (2) is included the right one, the right part is
196 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

197 $s_1 \approx s_2$ and $s = s_1 \cup s_2$. It remains to note accessories $\langle s_1, s_2 \rangle \in \text{dom} \bar{\cup}$ and full
198 image definition.

199 Now let $s \in \bar{\cup}[t_1 \times t_2]$, then there are rows $s_1 \in t_1$ and $s_2 \in t_2$, such that
200 $\langle s_1, s_2 \rangle \in \text{dom} \bar{\cup}$ and $s = s_1 \cup s_2$. The definition of operation $\bar{\cup}: S \times S \rightarrow S$
201 implies that $s_1 \approx s_2$. From here follows that $s \in t_1 \otimes t_2$.

202
203

204 **Lemma 3 (about saturation representation).** The saturation has the following
205 representation:
206

$$207 \quad C(t) = \prod_{A \in R} D_{A,t}, \quad (3)$$

208

209 Where $D_{A,t} \stackrel{\text{def}}{=} \{d \mid \exists s(s \in t \wedge \langle A, d \rangle \in s)\}$ is an active domain of attribute A with
210 respect to table t , R is the scheme of the table t .
211 The proof implies from definitions of saturation and generalized direct product.
212
213

214 **Lemma 4 (about renaming representation).** The renaming has the following
215 representation:

$$216 \quad R t_{\xi}(t) = R s_{\eta}[t], \text{ where } \eta \stackrel{\text{def}}{=} \xi \cup \text{id}_{A \setminus \text{dom} \xi}, t \in T_{\xi}. \quad (4)$$

217
218
219
220

221 **Theorem 1** (about representations of signature operations of table algebras). The
222 projection, join, saturation and renaming have the representations (1)-(4) respectively.
223 The proof implies from lemmas (1)-(4).
224

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

230
231

232 4. CONCLUSION

233 Informative fragment of table algebras theory has been constructed in this paper. The main
234 peculiarity of adapted technique is contained in establishment of natural representations of
235 basic signature operations of these algebras (projection, join, saturation, renaming) in terms

236 of set-theoretic constructions of full image, restriction, (generalized) direct product,
237 compatibility relation. In the second part of the paper the properties of considered set-
238 theoretic constructions will be transferred on table algebra.

239

240 REFERENCES

241

242 1. Julia Brana. Key relationships in table algebras. Bulletin of Kiev University. Series:
243 Physics & Mathematics. 1997;3:143-148.

244

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

247

248 3. Bui Dmitry. The theory of composition type of program algebras and its applications:
249 thesis Doctor of physical and mathematical sciences: 01.05.03 – Mathematical and
250 software of computers and systems. Kyiv. 2002; 365 p.

251 4. V.N Red'ko, Y.I Brana, D.B. Bui. Information aspects of Case-technologies: basic
252 relations in table algebras. Problems of programming.1997;1:5-11.

253 5. V.N. Red'ko, Y.I. Brana, D.B. Bui. Relational algebra: operations of division and
254 renaming. Cybernetics and Systems Analysis. 1997; 5: 3-15.

255 6. V.N. Red'ko, Y.I. Brana, D.B. Bui, S.A. Polyakov. Relational databases: table algebras
256 and SQL-like language. Kyiv: Publishing House "Akadempriodika". 2001;198p.

257 7. Codd E.F. A Relational Model of Data for Large Shared Data Banks. Communications of
258 the ACM. 1970; 13,(6): 377-387. <http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>

259 8. Codd E.F. Further Normalization of Data Base Relational Model. Data Base
260 Systems.1972; 33-64 .

261 9. Codd E.F. Relational Completeness of Data Base Sublanguages. Data Base Systems.
262 1972;65-93.
263 <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.9277&rep=rep1&type=pdf>

264 10. Codd E.F. Relational Database: A Practical Foundation for Productivity. Communications
265 of the ACM.1982;25 (2): 109-117.

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

268 12. Red'ko Vladimir. The foundations of compositional programming. Programming. 1979; 3:
269 3-13.

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

271 14. Maier D. The Theory of Relational Databases. Computer Science Press,1983; 608.

272 15. Kuratovski K. Topology. Academic Press, 1966;1: 594.

273 16.Tsalenko Mihail. Semantic and mathematical models of databases.Moscow: VINITI,
274 1985; 207p.

- 275 17. Tsalenko Mihail. Modeling of semantics in databases. Moscow: Nauka, 1989; 288p.
- 276 18. Cutland N. Computability. An introduction to recursive function theory. – Cambridge
- 277 University Press. 1980; 256p.