

Original Research Article

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

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 (whole 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, 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].

Let's make more precise the tables (the relations) in terms of nominal sets [11, 12, 13]. For this we fix two sets: $\mathbf{A}$, whose elements are called attributes, and $\mathbf{D}$ – the universal domain.

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

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

The rows are denoted by $s, s_1, s_2, \dots$

41 Table of the scheme R is called a (finite) set of rows of the scheme R . Thus, table of the
 42 scheme R is a finite set of the form $t = \{S_1, \dots, S_n\}$, where $n \geq 0$ and
 43 $s_i : R \rightarrow D$. The tables are defined as $t, t_1, t_2, \dots$. For $n = 0$ we have a case of an
 44 empty table, i.e. a table which does not contain rows. The set of all rows (tables) of the
 45 scheme R are defined as $S(R)$ ($T(R)$, respectively), the set of all rows (tables) are defined as

46 $S(T, \text{ respectively})$. Thus, we'll have $S \stackrel{\text{def}}{=} \bigcup_{R \subseteq A} S(R)$, $T(R) \stackrel{\text{def}}{=} 2^{S(R)}$, $T \stackrel{\text{def}}{=} \bigcup_{R \subseteq A} T(R)$.

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

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

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

53

54 Here $s|X$ is a restriction of the row s on set X which is defined standard as
 55 $s|X = s \cap (X \times D)$. The restriction of binary relation on a set is introduced completely

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

58

59 Let $X \subseteq A$ is a (finite) set of attributes.

60

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

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

65

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

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

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

70 an intersection of projections on first component of initial binary relations.
 71

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

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

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

81

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

83 $D_{A,t} \stackrel{def}{=} \{d \mid \exists s(s \in t \& \langle A, d \rangle \in s)\}$ is an *active domain of attribute A* with respect to
84 table t according to the terminology in [14].
85

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

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

90 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
91 subsets $\Sigma_v \subseteq \Sigma$ by names of a (finite) set of names $U \subseteq V$.
92

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

95 U and every name's $v \in U$ value (denotat) belongs to the set Σ_v .

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

98 nominal set is denoted by $\mathcal{E}$. Thus, $N \Sigma_v \stackrel{def}{=} \{\mathcal{E}\}$.
99

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

$$102 \quad N \Sigma_v = \prod_{v \in U} \Sigma_v \quad [15].$$

102

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

$$105 \quad \text{Thus, } \sim: T \rightarrow T, \tilde{t} \stackrel{def}{=} C(t) \setminus t.$$

106

107 Under *renaming* an unary, generally speaking partial, parametric operation of the form
108 $Rt_\xi: T_\xi \xrightarrow{\sim} T_\xi$ is understood, where the parameter is an injective , generally speaking

109 partial, map defined on an attributes set $\xi: A \xrightarrow{\sim} A$. The operation only renames tables-
110 arguments attributes according to parameter.

111 The arrow of the form $\xrightarrow{\sim}$ is used to denote partial operations .
112

113 The formal definition of renaming requires additional notions. The table renaming is a
114 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 whole image construction. In turn, the row renaming is renaming of first pair components, e.g. the row elements. Let's go down to precise definitions.

Let $\eta : A \rightarrow A$ 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 A, D by S' .

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

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

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 $\xi : A \rightarrowtail A$ and assume $\eta \stackrel{def}{=} \xi \cup id_{A \setminus dom \xi}$.

Here $id_X : X \rightarrow X$ is the identical function (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 $A \setminus dom \xi$ 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 $\xi[R] \cap (R \setminus dom \xi) = \emptyset$, where

$\xi[R] \stackrel{def}{=} \{ A \mid \exists B (B \in R \ \& \ \xi(B) \simeq A) \}$ is a whole 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 -admissible 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 -admissible schemes [6].

Definition 9. Renaming corresponding to injective partial renaming attributes function

$\xi : A \rightarrowtail A$ 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,

$\eta = \xi \cup id_{A \setminus dom \xi}$ and the right part of the last equality is a whole image of table-argument.

Assume by definition

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

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

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

153

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

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

156

157 For Examining tables and their schemes we need to take into account some
158 peculiarities. By definition the empty set or rows in the table is called *empty table* and
159 denoted by $t_\emptyset$. As far as exists single nominal set on pair $\emptyset, D$, namely, empty nominal set
160 ε , then there exists a single row ε of a scheme $\emptyset$; i.e. $S(\emptyset) = \{\varepsilon\}$. Concerning empty

161 scheme $\emptyset$ tables: there are two of them - $t_\emptyset$ and $t_\varepsilon \stackrel{def}{=} \{\varepsilon\}$. Really,

$$162 \quad T(\emptyset) = 2^{S(\emptyset)} = \{\emptyset, \{\varepsilon\}\} = \{t_\emptyset, t_\varepsilon\}.$$

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

$$166 \quad \forall R_1 R_2 (R_1 \neq R_2 \Rightarrow T(R_1) \cap T(R_2) = \{t_\emptyset\}).$$

167

168

169 3.REPRESENTATIONS OF SIGNATURE OPERATIONS OF TABULAR 170 ALGEBRA

171

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

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

177 particular, $\bar{\cup}: S \times S \xrightarrow{\sim} S$, $dom \bar{\cup} = \{ \langle s_1, s_2 \rangle \mid s_1 \approx s_2 \}$, $s_1 \bar{\cup} s_2 = s_1 \cup s_2$ for all

178 $\langle s_1, s_2 \rangle \in dom \bar{\cup}$.

179

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

$$182 \quad \pi_X(t) = \uparrow X[t], \text{ where } \uparrow X: S \rightarrow S, \uparrow X(s) = s \mid X. \quad (1)$$

183 The proof implies from definitions of projection and restriction.

184

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

$$186 \quad t_1 \otimes t_2 = \bar{\cup}[t_1 \times t_2]. \quad (2)$$

187

188 Proof. We show that the left part of equality (2) is included the right one, the right part is
189 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

190 $s_1 \approx s_2$ and $s = s_1 \cup s_2$. It remains to note accessories $\langle s_1, s_2 \rangle \in \text{dom} \bar{\cup}$ and whole
191 image definition.

192 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

193 $\langle s_1, s_2 \rangle \in \text{dom} \bar{\cup}$ and $s = s_1 \cup s_2$. The definition of operation $\bar{\cup}: S \times S \rightarrow S$
194 implies that $s_1 \approx s_2$. From here follows that $s \in t_1 \otimes t_2$.

195

196

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

199

$$200 \quad C(t) = \prod_{A \in R} D_{A,t}, \quad (3)$$

201

202 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
203 respect to table t , R is the scheme a table t .

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

205

206

207 **Lemma 4 (about renaming representation).** The renaming has the following
208 representation:

$$209 \quad R_{t_\xi}(t) = R_{s_\eta}[t], \text{ where } \eta \stackrel{\text{def}}{=} \xi \cup \text{id}_{A \setminus \text{dom} \xi}, t \in T_\xi. \quad (4)$$

210

211 The proof implies from definitions of tables and rows renaming.

212

213

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

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

217

218 Thus, the representations of basic operations of table algebras through set-theoretic
219 constructions of whole image, restriction and direct product have been established. It is
220 worth to note one more time that the presence of these representations allows to get
221 properties of table operations in the capacity of corollaries of properties of indicated set-
222 theoretic constructions. This will be done in the next part of the paper.

223

224 4. CONCLUSION

225

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

232

233

234 REFERENCES

235

236 [1] Julia Brana. Key relationships in table algebras. Bulletin of Kiev University. Series:
237 Physics & Mathematics. 1997; Issue 3: 143-148.

238

239 [2] Bui Dmitry,. Ms. Brana, Set-theoretic structures in relational database theory .Bulletin of
240 Kiev University. Series: Physics & Mathematics. 1996; Issue 1:216-224.

241

242 [3] Bui Dmitry. The theory of composition type of program algebras and its applications:
243 thesis Doctor of physical and mathematical sciences: 01.05.03 – Mathematical and
244 software of computers and systems. Kyiv. 2002; 365 p.

245 [4] V.N Red'ko, Y.I Brana, D.B. Bui. Information aspects of Case-technologies: basic
246 relations in table algebras. Problems of programming.1997; 1: 5-11.

247 [5] V.N. Red'ko, Y.I. Brana, D.B. Bui. Relational algebra: operations of division and
248 renaming. Cybernetics and Systems Analysis. 1997; 5: 3-15.

249 [6] V.N. Red'ko, Y.I. Brana, D.B. Bui, S.A. Polyakov. Relational databases: table algebras
250 and SQL-like language. Kyiv: Publishing House "Akademperiodika". 2001;198p.

251 [7] Codd E.F. A Relational Model of Data for Large Shared Data Banks. Communications of
252 the ACM. 1970; Vol. 13, № 6: 377-387.
253 <http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>

254 [8] Codd E.F. Further Normalization of Data Base Relational Model. Data Base
255 Systems.1972; 33-64 P.

256 [9] Codd E.F. Relational Completeness of Data Base Sublanguages. Data Base Systems.
257 197;65-93P.
258 <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.9277&rep=rep1&type=pdf>

259 [10] Codd E.F. Relational Database: A Practical Foundation for Productivity.
260 Communications of the ACM.1982; Vol. 25, № 2: P. 109-117.

261 [11] Red'ko Vladimir. Compositions of programs and compositional programming.
262 Programming. 1978; № 5: P. 3-24.

263 [12] Red'ko Vladimir. The foundations of compositional programming. Programming. 1979;
264 № 3: P. 3-13.

265 [13] Red'ko Vladimir. Semantic structure of programs. Programming. 1981; № 1: P. 3-19.

266 [14] Maier D. The Theory of Relational Databases. Computer Science Press,1983; 608 p.

267 [15] Kuratovski K. Topology. Academic Press, 1966; Vol. 1: 594 p.

268 [16] Tsalenko Mihail. Semantic and mathematical models of databases.Moscow: VINITI,
269 1985; 207 p.

270 [17] Tsalenko Mihail. Modeling of semantics in databases. Moscow: Nauka, 1989; 288 p.

271 [18] Cutland N. Computability. An introduction to recursive function theory. – Cambridge
272 University Press. 1980; 256 p.