Using a Tableau Method for Checking the Database Logical Structure Correctness

Gennady V Svetlov, Aleksey I. Baranchikov, Natalya N. Grinchenko, Nataliy S. Fokina
Joint stock company "Ryazan Production and Tehnological enterprise "Granit"
Ryazan, Russia
alexib@inbox.ru

Abstract– **The paper has considered the issue of checking the database logical structure correctness. At the present, there are several methods for solving this issue in the world, but to test existing databases with different complexity in the structures, it is used a method based on the new algorithm using a tableau. This algorithm uses the chase method to check the join- and functional dependencies of the databases. The assessment of algorithm time complexity and convergence are presented in the next part. The last part of this paper, we try to illustrate an example by using this algorithm.**

Keywords- database, algorithm, logical structure, tableau, the chase method.

I. INTRODUCTION

The purpose of the algorithm is to achieve a successful checking of the right representability of relations out of constraint set ***C*** with its projections on the relation schemes of some database ***R***.

At the present time, within the limits of relational approaches in the theory of relational databases, there is a pressing issue of the correctness checking of schemes' logical structure. A properly structured database has the following advantages:

- High request processing speed;
- Smaller amount of taken memory;
- Proper functioning;
- Lack of redundancy;
- Comprehensibility and certainty.

The task of the correctness checking of the logical structure is still open.

II. THEORETICAL RESEARCH

At the present time, there are several options for tracking the correctness of constructing the databases[1,2]. For example, when designing the base from scratch different systems of database management allow monitoring the structures online. Their disadvantage is that it is impossible to analyze finished databases[3]. There are some automated systems which allow getting the information about the structure of such bases. Usually, as the result of these software solutions the user gets the information in the ER-chart form and after that, he needs to analytically check the correctness of the structure. In cases of difficult databases, this task becomes intractable.

The checking of databases without using a tableau is also possible through its decomposition and search for losses. For the decomposition, it is necessary to have the full information about the scheme of the base, in particular, all the attributes and connections between them, and about existing dependencies. The processing is very labor-intensive due to the absence of structuring in data storage. It is particularly hard in the cases of dealing with big and difficult databases[4,5,6].

The tableau allows ignoring the content of the database. It gives the possibility to keep all the information in a visualized and convenient to handle format without using any extra data or parameters.

In order to check the database, it is also possible to use the method of representative samples. A representative sample is a sample out of the sampled population with the distribution *F(x)* representing the main features of the sampled population. The sample (empirical) distribution function $\hat{F}(x)$ subject to the large range of samples, provides a good enough indication of the sample distribution function *F(x)* of the original sampled population. However, this method is also rather labor-intensive and, just like the other statistical method, is relatively low reliable.

The developed algorithm of checking the logical structure of schemes is based on the usage of scheme decomposition and the application of tableau to detect date losses.

As a result of the performed algorithm, the answer to the question "Is the decomposition without the losses of relation out of constraint set ***C*** into ***R*** possible" will be given.

A tableau is a tabular procedure of representing *PJ*-reflections (reflections "projection-connection") [1,7,8,9].

A tableau reduction is a tableau that consists out of the set of all lines of the original tableau, and none of them is absorbed by the other lines [2,10,11,12].

The equivalence of two tableaux with the restrictions gives us the opportunity to check the cases when *PJ*-reflection does not have any losses in the constraint set.

ISSN 2337-0343

Two tableaux are equivalent when their reductions are the same to within the one-to-one renaming of not distinguished symbols [2].

The algorithm uses the method of the chase.

The method of chase is a computational method with the help of which for a given tableau *T* and the dependency set **C** a new tableau *T** is constructed, the kind that $T \equiv T^*$, and *T** as a relation belongs to SAT(**C**) which is a subset of database scheme set, satisfying ***C*** [3,14,15,16].

With the help of the chase, the tableau is tested for equivalence on ***C***.

In the terms of equivalency of a tableau, for the successful testing of discovering the dependencies in the connection the fulfillment of the condition $TR \equiv cTI$ is needed, where *TI* is the tableau consisting out of one line of distinguished variables. The letter c means the existence of a set with all kinds of constraints applicable to the tableau. The equivalency $T1 \equiv cT2$ is true if and only if

$chase\ c(T1) \equiv chase\ c(T2)$,

i.e. if the final tableau *T1*, according to the method of the chase, is equivalent to the final tableau *T2*.

This means that it is enough for us when this condition is fulfilled

$chase\ c(TR) \equiv chase\ c(TI)$.

But as far as $chase\ c(TI) = TI$, then

$chase\ c(TR) \equiv TI$

Therefore, the necessary and sufficient condition of the checking is the availability of the line of distinguished elements in *chase c(TR)* [17].

Similarly, for the successful testing of discovering the functional dependencies of $X \rightarrow Y$ kind the necessary and sufficient condition of the checking is the availability of only distinguished elements in the column, corresponding to the attribute *Y*, the final tableau according to the method of *chase c(TR)*.

We outline the method of checking.

Input data:

- database scheme ***R***;

- constraint set ***C***.

The general scheme of the algorithm can be seen in Figure 1.

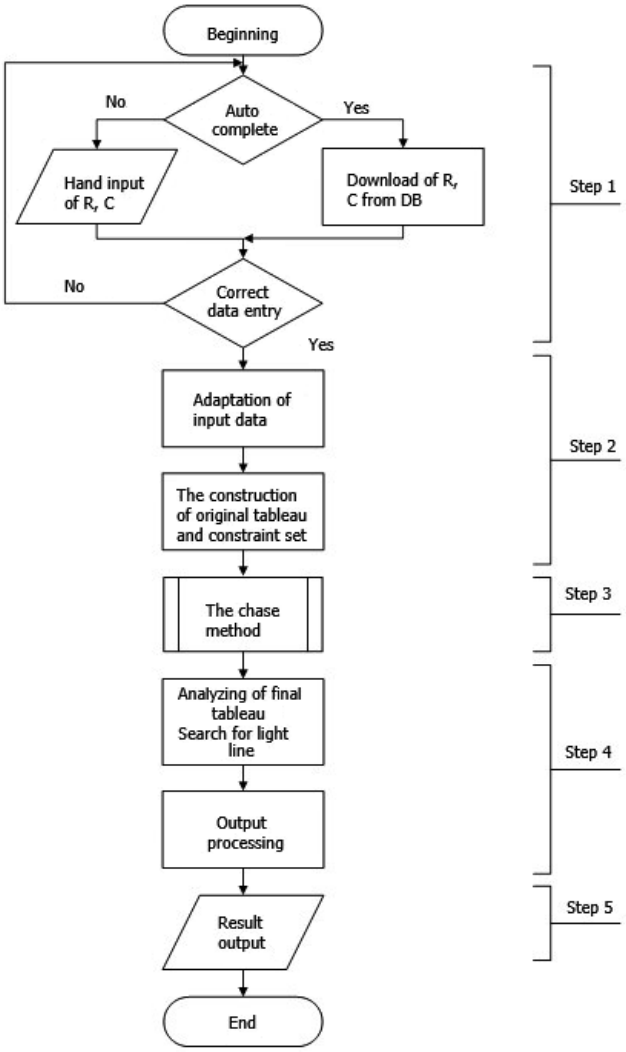

Figure 1 – The general scheme of the algorithm

Let us take a closer look at the performance of the algorithm step by step.

At the first step, there is a correct data entry: the scheme ***R*** and the constraint set ***C***, which is a population of *F*- and *J*-regulations (functional and join), there is the checking of the correctness of data entered.

At the second step of the algorithm the adaptation of input data is in progress, i.e. the data are transformed in a way it can be best processed, and the original tableau *TR* in the scheme ***R*** is constructed.

The third step is the performance of the chase method.

The method is as follows: *for the specified T and **C** F- and J-regulations are applied, corresponding to F- and J-regulations out of **C** until they cause changes.*

At the fourth step, there is an assessment of equivalence. For the dependencies of the connection, the final tableau *T** is checked for the equivalence to *TI* (the tableau that consists out

of the only line of distinguished elements). *T** is equivalent to *TI* if there is a line of distinguished elements in *T**. For the functional dependencies, there is a check only for distinguished elements in the corresponding column.

At the fifth step, the result output happens.

The chase method and the assessment are presented schematically in Figure 2.

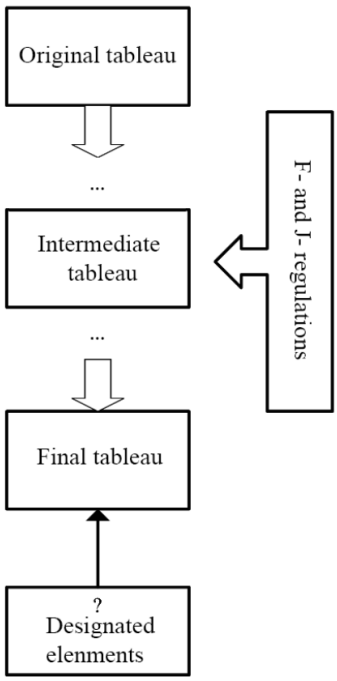

Figure 2 – The chase method and correctness checking

III. THE ASSESSMENT OF ALGORITHM TIME COMPLEXITY AND CONVERGENCE

The tableau is a set of lines, and none of the *F*- or *J*-regulations does not enter any new variables, there is only a finite number of tableaux which can appear in a generating sequence *T* relatively to ***C*** [18,19]. That is why the algorithm always converges.

Generally, the process of the chase method has an exponential time complexity [20]. If the tableau *T* has *k* columns and *m* lines, *chase c(T)* can have *m^k* lines. In the case of using the process of the chase method in order to check the nonexistence of information losses in the connection, the full procedure is not always needed. As soon as the line consisting only out of distinguished variables is gotten, there is no need for continuing the checking. If this line is a part of any tableau of generating the sequence, it will appear in the final tableau. However, the problem of identification of line of distinguished elements to *chase c(T)* probably does not have a polynomial-time decision because it is known that the problem of the checking of $C \models * [S]$ is NP-hard. For the checking of $C \models c$ other methods exist which, contrary to the chase method, have in case of *F*- or *J*-dependencies a polynomial time complexity.

By virtue of the fact that *F*-regulations does not induce any new lines, the process of chase method *chaseF (T)* for the set of *F*-dependencies *F* never has more lines than *T*. It is no wonder then that *chaseF (T)* can be calculated in a polynomial time. Let us suppose that the task inputs is the tableau *T* and the set *F*. For simplicity in what follows, we assume that every attribute or tableau variable takes one storage unit.

Suppose that $k = |U|$ = a number of *T* columns, *m* = a number of *T* lines, *p* = an amount of storage for the recording of *F*.

The amount of input is $n = O(k\text{-}t_n + p)$.

Let us show how to calculate *chase c(T)* in a time $O(n^3)$. We begin to make up rerunning on the set of *F*-dependencies. For every *F*-dependency $X \rightarrow A$ let us group together the lines with equal definitions of *X*-component. If $|X| = q$, the sorting takes $O(q\text{-}t_n)$ of time. After the sorting for $O(q\text{-}t_n)$ of time, we find the lines with equal definitions of *X*-component and identify their *A*-columns. The sum of the left parts amounts, according to all the *F*-dependencies out of *F*, does not exceed p. Therefore, one running through all the *F*-dependencies takes $O(p\text{-}m)$ of time.

We continue to make rerunning through *F* until the changes stop appearing in *T*. Let us stop at this. At the beginning, *T* can have no more than *k-m* different variables. Every rerunning, except for the last one, decreases the number of variables by one, thus, we have no more than $O(k\text{-}m)$ rerunning. The total time of the rerunning procedure is $O(k*p* m^2)f$ which does not exceed $O(n^3)$.

If the tableau complies with the database scheme, and there are only schemes put by as an input, the procedure described above takes $O(n^4)$ of time where *n* is the amount.

With a view to simplifying *F*-regulations, we have still assumed that all of our *F*-dependencies have one attribute on the right. *F*-regulation can be summarized in case of many attributes on the right side of *F*-dependency. If *w1* and *w2* are the lines in the tableau, such as $w1(X) = w2(X)$ and $X \rightarrow Y$, there is an *F*-dependency within the limitations, than it is possible for every *A* attribute, which is also a part of *Y*, to be identified to $w1(A)$ and $w2(A)$.

There is also a generalization of *J*-regulation which allows us to generate more than one line at a time. If $* [S]$ is a *J*-dependency out of constraint set, then it is possible to apply *PJ*-reflection of *ms* to the tableau and the result can be used for the making of the next tableau in a generating sequence.

IV. AN EXAMPLE OF ALGORITHM'S WORK

Let us consider the database scheme $\boldsymbol{R}=\{AB, BC, AD\}$ as an example. Let the constraint set be $\boldsymbol{C}=(A \rightarrow D, *[AB, BCD])$. We apply the worked out the algorithm in order to get the final tableau. Original T1 and final T1* tableaux are presented in Figure 3.

T1			
A	B	C	D
a1	a2	b1	b2
b3	a2	a3	b4
a1	b5	b6	a4

T1*			
A	B	C	D
a1	a2	b1	a4
b3	a2	a3	a4
a1	b5	b6	a4
a1	a2	a3	a4
b3	a2	b1	a4

Figure 3 – The original and the final tableaux for R, C

Given that *chase c(T1)* contains the line of distinguished elements, then any relation from *SAT(C)* without any losses decomposes into ***R***.

For the database scheme *S*={*AB, DC, CD*} *chase c(T2)*, found by the worked-out algorithm, does not contain the line of distinguished elements. In *SAT(C)* such relations exist which have losses while decomposing into *S*. The original T2 and the final T2* tableaux are presented in Figure 4.

T2			
A	B	C	D
a1	a2	b1	b2
b3	a2	a3	b4
b5	b6	a3	a4

T2*			
A	B	C	D
a1	a2	b1	b2
b3	a2	a3	b2
b5	b6	a3	a4
a1	a2	a3	b2
b3	a2	b1	b2

Figure 4 – The original and the final tableaux for S, C

V. CONCLUSION

The worked-out algorithm which uses the tableau gives the opportunity to analyze the correctness of logical structure of the databases made from scratch or already existing. In case of further improving the algorithm, there is an aim to add the checking given multivalued dependencies and the possibility to check database structure with the attributes of varying degrees of protection.

REFERENCES

[1] Date C. J., Hugh Darwen Foundation for Future Database Systems: The Third Manifesto, Addison-Wesley, ISBN-10: 0201709287 (ISBN-13: 978-0201709285), 2000, 608 p.

[2] Connolly T., Begg C. Database Solutions: A step by step guide to building databases, ISBN-10: 0321173503 (ISBN-13: 978-0321173508), Addison-Wesley, 2003, 552 p.

[3] Maier D. Theory of Relational Database, Computer Science Press; 1st edition (March 1, 1983) ISBN-10: 0914894420 (ISBN-13: 978-0914894421) 656 p.

[4] Aleksey I. Baranchikov, Aleksey Yu. Gromov, Viktor S. Gurov, Natalya N. Grinchenko and Sergey I. Babaev "The technique of dynamic data masking in information systems" in Proceedings of the 5th Mediterranean Conference on Embedded Computing MECO 2016, Montenrgro, Bar, pp. 473-476.

[5] Xiao-Bai Li, Luvai Motiwalla BY “Protecting Patient Privacy with Data Masking” WISP 2009 Ravikumar G K et al. / International Journal of Engineering Science and Technology (IJEST)ISSN : 0975-5462Vol. 3 No. 6 June 20115158

[6] Oracle White Paper—Data Masking Best Practices JULY 2010[14]

[7] Ravikumar G K, Manjunath T N, Ravindra S Hegadi,Umesh I M:A Survey on Recent Trends ,Process and Development in Datamasking for testing, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011, p-535-544

[8] Baranchikov A.I., Baranchikov P.A. Methods of data masking concerning DB for different access models // Management systems and information technologies: scientific journal № 2(52). Moscow - Voronezh, 2013. Pp. 58 – 61.

[9] Baranchikov A.I., Gromov A.Yu., Kostrov B.V. Method and algorithm of data domain description based on the scheme of relation database // Modern problems of science and education. – 2015. – № 1; URL: www.science-education.ru/121-18676 (last accessed date: 02.10.2015) 7 p. 0,12 МБ

[10] Godin, R. and Missaoui, R.An Incremental Concept Formation Approach for Learning from Databases, Theoretical Computer Science, Special Issue on Formal Methods in Databases and Software Engineering, 133, 387-419.

[11] Carpineto C., Romano G. Inferring Minimal Rule Covers from Relations, Computational Intelligence, Volume 15, numer 4, 1999, pages 415-441.

[12] J. Hammer, M. Schmalz, W. O’Brien, S. Shekar and N. Haldavnekar Knowledge Extraction in the SEEK Project Part I: Data Reverse Engineering, Dept. of CISE, University of Florida, Gainesville, FL 32611-6120, Technical Report TR-02-008, July 2002.

[13] Rakesh Agrawal, Ramakrishnan Srikant Fast Algorithms for Mining Association Rules, Proc. 20th Int. Conf. Very Large Data Bases, VLDB, 1994

[14] Manilla H., Raiha K.-J. Algorithms for Inreffing Functional Dependencies // Data & Knowledge Engineering, 1994, 12. P.83-99.

[15] Yao H., Hamilton H.J., Butz C.J. FD_MINE: Discovering Functional Dependencies in a Database Using Equivalemces // University of Regina. Computer Science Department. Technocal Report CS-02-04. August, 2002. ISBN 0-7731-0441-0.

[16] Novelli N., Cicchetti R. Functional and Embedded Dependency Inference: A Data Mining Point of View // Information System, 2001, 26 (7). P.477-506.

[17] Date C.J. An Introduction to Database Systems, Pearson Education, ISBN 8177585568, 2006, 968 p.

[18] Date C.J. Databases Design and Relational Theory: Normals Forms and All that Jazz, O'Reilly Media, Inc, USA, ISBN10 1449328016, 2013, 278 p.

[19] Strategic Database Technology: Management for the Year 2000 by Alan R. Simon , ISBN 155860264X (ISBN13: 9781558602649) Paperback, 446 pages, Published April 28th 1995 by Morgan Kaufmann Publishers

[20] Domingo-Ferrer J., and Mateo-Sanz, J. M. 2002.“Practical Data-Oriented Microaggregation for Statistical Disclosure Control,” IEEETransactions on Knowledge and Data Engineering (14:1), pp. 189-201.[6]