N-Tuple Algebra as a Unifying System to Process Data and Knowledge

D

Boris Alexandrovich Kulik
Institute of Problems in Mechanical Engineering RAS, Russia

Alexander Yakovlevich Fridman
Institute for Informatics and Mathematical Modelling, Kola Science Centre of RAS, Russia

BACKGROUND

Theory of relations is mostly used in databases modeled by means of relational algebra (Codd, 1971). However, many diverse and (at first sight) different mathematical objects, namely graphs, semantic networks, expert rules, predicates, logical formulas, etc., can be expressed as relations. Relational algebra does not provide sufficient techniques to deal with such models and structures.

Modern intelligence systems comprise two types of dissimilar objects, namely databases (DBs) and knowledge bases (KBs). Their structures are quite distinct since they exploit fundamentally different theoretical approaches. Nowadays, data structures (numbers, vectors, graphs, networks, texts, etc.) correspond to algebraic methods (Wirth, 1976). As for KBs, their basic models (predicates, frames, semantic networks, rules and so on) are designed on the basis of declarative approach (Russel & Norvig, 2003). This incompatibility results in significantly different programming systems and structures for DBs and KBs, and consequently, in big difficulties to integrate a DB and a KB within one software system.

Conventional attempts to do so relate to the declarative approach mostly. The model of deductive databases (Ullman, 1989; Ceri, Gotlob, & Tanca, 1990) gives an example of such attempts. It is grounded on a modification of Prolog language designed within the declarative approach.

Conversely, some shortcomings inhere in this approach. One of the major problems is in the necessity to reduce many tasks of logical analysis to satisfiability checks for a certain logical formula, this check being able to return only two possible answers ("yes" or "no"). Such a reduction is not simple. Moreover, it is unrealizable in cases when one needs to not only receive a binary answer but also to estimate the value of some parameters in the formal system or to assess the structure and/or number of objects that satisfy the given conditions.

To model and analyze modified reasoning (that is reasoning with hypotheses and abductive conclusions), many authors propose usage of non-classical logics (Russel & Norvig, 2003). However, such logics often have no interpretation or their interpretations do not correspond to semantics of the objects to be modeled. In the authors' opinion, there are some reasons to suppose that problems in modeling modified reasoning within classical logic result from features of the declarative approach rather than from drawbacks of classical logic.

Algebraic approach to logical modeling can be originated from Aristotle's syllogistics and Boole's logical matrices, in more recent times only Arkady Zakrevskij (for instance, see (Zakrevskij & Gavrilov, 1969)) investigated set-theoretical approach to logic and proposed a programming language LYaPAS (Logical Language for Representation of Synthesis Algorithms) oriented toward programming of synthesis algorithms for finite-state and discrete devices. However, this language does not provide any means for logical inference.

DOI: 10.4018/978-1-5225-2255-3.ch173

To overcome the mentioned and other problems, the mathematical model of relational algebra was generalized by the authors to a more universal system for processing n-ary relations called n-tuple algebra (Kulik, 1995, 2007; Kulik, Fridman, & Zuenko, 2013, 2015).

N-TUPLE ALGEBRA: BASICS AND FEATURES

N-tuple algebra is a mathematical system to deal with arbitrary n-ary relations. In NTA, such relations can be expressed as four types of structures called *NTA objects*. Every NTA object is immersed into a certain space of *attributes*. Names of NTA objects contain an identifier followed by a sequence of attributes names in square brackets; these attributes determine the *relation diagram* in which the NTA object is defined. For example, $R[XYZ]$ denotes an NTA object defined within the space of attributes

$X \times Y \times Z$ that is $R[XYZ] \subseteq X \times Y \times Z$.

From the mathematical point of view, NTA is an *algebraic system* $A = <A, O, R>$, where *carrier* A is an arbitrary totality of relations expressed as NTA objects; O is a set of *operations* comprising *algebra of sets' operations* (union, intersection, complement) and *operations on attributes* (see below); R is a set of verified *relations* between NTA objects, namely equality and inclusion. This algebraic system is proved to be complete (i.e. all operations and checks of relations are feasible for any totalities of NTA objects).

NTA objects defined on the same relation diagram are called *homotypic* ones. NTA objects contain n-tuples of sets and provide a condensed representation of n-ary relations. In NTA, *elementary n-tuples* correspond to n-tuples of elements in ordinary relations. For instance, the record $T[XYZ] = (a, b, c)$ means that T is an elementary n-tuple where $a \in X$, $b \in Y$, $c \in Z$.

NTA objects look like matrices having subsets of corresponding attributes (*components*) rather than elements as their cells. The totality of NTA components includes two *dummy components*. They are: the *complete component* * that equals to the whole domain (i.e. the range of values) of the corresponding attribute in the relation diagram and the *empty component* $\varnothing$ that equals to the empty set and models the complement of the component *.

NTA admits 4 types of NTA objects, namely *C*-*n*-tuples and *C*-systems, *D*-*n*-tuples and *D*-systems.

A *C-n-tuple* is an n-tuple of components defined in a certain relation diagram, denoted within square brackets and interpreted as the set of elementary n-tuple contained in the Cartesian product of its components. For example, the right part of the formula $R[XYZ] = [A * C]$ expresses a *C*-*n*-tuple that can be transformed into an ordinary n-ary relation by calculating the Cartesian product $A \times * \times C$ where $A \subseteq X$, $* = Y$ is a complete component placed to the position of the attribute Y, and $C \subseteq Z$.

A *C-system* is a set of homotypic *C*-*n*-tuples that are denoted as a matrix in square brackets. As an example, the *C*-system

$$Q[XYZ] = \begin{bmatrix} A_1 & B_1 & * \\ * & B_2 & C_2 \end{bmatrix}$$

can be reduced to a set of elementary n-tuples calculated by formula

$$Q[XYZ] = (A_1 \times B_1 \times Z) \cup (X \times B_2 \times C_2).$$

A *D-n-tuple* expresses a structure that can be interpreted, in particular, as a complement of a certain *C*-*n*-tuple. Suppose, the *C*-*n*-tuple

$$R[XYZ] = [A\ B\ C]$$

is given. Its complement equals to the difference of *C*-*n*-tuples