N-Tuple Algebra as a Generalized Theory of Relations

5

Boris A. Kulik
Institute for Problems in Mechanical Engineering of RAS, Russia

Alexander Y. Fridman
https://orcid.org/0000-0003-2408-6892
Institute for Informatics and Mathematical Modeling of RAS, Russia

INTRODUCTION

In information technologies, analysis of heterogeneous information often necessitates unification of presentation forms and processing procedures for such data. One of the promising ways to solve this problem is the search for a universal structure, which allows reducing various data and knowledge models to a single mathematical model with unified analysis methods. One of such universal structures is the relation, which is mainly associated with databases modeled by means of relational algebra (Codd, 1971). However, relations are used not only in databases. They can model as different, at first glance, mathematical objects as graphs, networks, artificial intelligence structures, predicates, logical formulas, etc. Representation and analysis of such structures and models requires for more expressive means and methods than relational algebra provides.

So, with a view to developing a general theory of relations, the authors propose *n*-tuple algebra (NTA) developed as a theoretical generalization of structures and methods applicable in intelligence systems. NTA allows for formalizing a wide set of logical problems (deductive, abductive and modified reasoning, modeling uncertainties and so on). Besides, this paper considers matters of metrization and clustering for NTA objects with ordered domains of attributes.

BACKGROUND

Nowadays, intelligence systems include two types of dissimilar objects, namely databases (DBs) and knowledge bases (KBs). Their structures are fundamentally different, since their construction is based on different theoretical approaches. Modern data structures (numbers, vector graphs, networks, texts, etc.) exploit algebraic techniques (Wirth, 1976). As for KBs, their basic models (predicates, frames, semantic networks, rules) are built on the basis of declarative approaches (Russel & Norvig, 2003). This discrepancy results in significant differences of programming systems for DBs and KBs and, accordingly, in large expenditures of time and money to couple DBs and KBs in one software system. Conventional cases to attempt this integration are mostly accomplished within declarative approaches. An example is the deductive database model (Ullman, 1989; Ceri, Gotlob, & Tanca, 1990), which is based on a modified Prolog programming language intended for computer implementation of the declarative approach.

Conversely, some shortcomings inhere in this approach. One of the main problems is that when using a declarative approach, many tasks of logical analysis need to be reduced to satisfiability checks for a

DOI: 10.4018/978-1-7998-3479-3.ch048

certain logical formula where only two possible answers ("yes" or "no") are possible. 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.

When modeling and analyzing modifiable reasoning (i.e., reasoning with hypotheses and abductive conclusions), many researchers recommend using non-classical logic (default logic, non-monotonic logic, etc.) (Russel & Norvig, 2003), since it is believed that classical logic is ill-suited for solving such problems. However, in many cases, interpretations of non-classical logics are either missing or not corresponding to the semantics of the objects being modeled. There are reasons to assume that the difficulties of modeling modifiable reasoning within the framework of classical logic are not due to the shortcomings of this logic, but to the specifics of the declarative approach.

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.

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, 2017). Its main features and provided opportunities are introduced below.

N-TUPLE ALGEBRA: BASICS AND FEATURES

The algebraic approach to modeling systems stipulates that some homogeneous objects (numbers, vectors, matrices, etc.) are used as basic structures. In n-tuple algebra (NTA), such objects are relations, which can be expressed as four types of structures called ***NTA objects***. Every NTA object is immersed into a certain space of variables called ***attributes***. Each attribute is represented by a set of possible values called ***domains***, and the spaces themselves are defined as Cartesian products of these domains.

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 $R[XYZ] \in X \times Y \times Z$.

From the mathematical point of view, NTA is an ***algebraic system*** $\mathbf{A} = <A, O, R>$, where the ***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 concise 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 $a \in X$, $b \in Y$, $c \in Z$.