**Buletin Teknik Elektro dan Informatika**
(Bulletin of Electrical Engineering and Informatics)
Vol. 3, No. 3, September 2014, pp. 223~228
ISSN: 2089-3191



# Inductive Generalization in Logical Inference and Techniques to Estimate It

**B.A. Kulik[1], A.Ya. Fridman[2]**
[1] Institute of Problems in Mechanical Engineering, Russian Academy of Sciences (RAS),
61 Bol'shoi pr., 199178 St. Petersburg, Russia
[2] Institute for Informatics and Mathematical Modelling, Kola Science Centre of RAS,
24A Fersman str., 184209 Apatity, Russia
e-mail: [1]ba-kulik@yandex.ru; [2]fridman@iimm.ru

***Abstract***

*The paper presents a novel approach to problems of deductive reasoning in frames of n-tuple algebra (NTA) earlier developed by the authors. Investigations of such problems let us determine the minimal consequence in logical inference and develop techniques to find it. Besides, we have proved that many formally correct consequences are inductive generalizations of this minimal consequence. An NTA-based method is proposed to obtain a numerical estimation for the degree of such an inductive generalization. In particular, it becomes possible to predict the number of consequences for a given system of premises and the share of a minimal consequence in a universe.*

***Keywords**: deductive logical inference, n-tuple algebra, minimal consequence, inductive generalization, numerical measure*

## 1. Introduction

Logical inference in classical logic stipulates sequential application of inference rules to some statements chosen as axioms. This method produces a lot of diverse consequences while many of them are not interesting for the researcher. Usually, he\she formulates the supposed consequence together with the premises and sets some desired properties of this consequence, for instance, a number or a list of its variables. Within such an approach, it is not easy to forecast a sequence of steps needed to check correctness of the supposed consequences or to derive consequences with certain properties. This kind of problems requires for exponentially complex search. In this paper, we propose to use algebraic techniques, NTA [1],[2] in particular, in order to estimate consequences quantitatively.

## 2. Main Terms and Structures of NTA

NTA is mathematically designed as an algebraic system, so it needs a support, a totality of operations and relations as well as their properties to be defined. Sometimes, these properties can be uniquely fixed by proving an isomorphism between a given algebraic system and a known one. Particularly, we have proved that NTA is isomorphic to algebra of sets and belongs to the class of Boolean algebras [2].

NTA *support* is an arbitrary set of *n*-ary relations expressed by specific structures called *NTA objects*. We will introduce them some later. Every NTA object is immersed into a certain space of *attributes*. *Domain* is a set of values of an attribute. Names of NTA objects contain an identifier followed by a sequence of attributes names in square brackets; this sequence determines the *relation diagram* in which the NTA object is defined. For example, $R[XYZ]$ denotes an NTA object given within the space of attributes $X$, $Y$ and $Z$.

NTA *basic operations* include the *algebra of sets operations*, namely intersection, union, and complement as well as *attributes operations* (renaming and transposition of attributes,

*Received October 9, 2013; Revised December 23, 2013; Accepted January 16, 2014*

elimination and addition of a dummy attribute). Combinations of the listed operations allows defining *auxiliary operations* upon relations, they are join, composition, transitive closure, etc. To compare NTA objects, we use two *basic relations*, namely *inclusion* and *equality*. By its analytical capabilities, we can compare NTA with predicate calculus. NTA objects model truth areas of predicates and logic formulas.

NTA objects provide a condensed representation of *n*-ary relations. When necessary, some specific algorithms can transform these objects into ordinary *n*-ary relations, which contain sets of *n*-tuples called *elementary n-tuples* in NTA. Cartesian product of domains for a given sequence of attributes determines a certain *partial universe*.

NTA objects are called *homotypic* ones if defined on the same relation diagram. In NTA, it is possible to implement operations of algebra of sets upon NTA objects with different relation diagrams as well.

NTA objects, namely *C*-n-tuples, *C*-systems, *D*-n-tuples, and *D*-systems are formed as matrices of subsets of attributes domains called *components*. The components include two types of *dummy components*. One of them is called the *complete component*; it is used in *C*-n-tuples and denoted by "*". A dummy component "$*$" added in the *i*-th place of a *C*-n-tuple or a *C*-system equals to the whole domain of the *i*-th attribute in the relation diagram. Another dummy component ($\varnothing$) called an *empty set* is used in *D*-n-tuples.

Let us now proceed with description of NTA major structures; they are *C*-systems and *D*-systems.

We record a *C-system* as a matrix of component sets framed with square brackets.
For example:

$R[XYZ] = \begin{bmatrix} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \end{bmatrix}$ is a *C*-system that can be transformed into an ordinary relation (i.e. a set of elementary *n*-tuples) calculated by the formula $R[XYZ] = (A_1 \times A_2 \times A_3) \cup (B_1 \times B_2 \times B_3)$ where $A_1, B_1 \subseteq X$; $A_2, B_2 \subseteq Y$; $A_3, B_3 \subseteq Z$.

*C*-systems are convenient for representing disjunctive normal forms of unary predicates. A one-line *C*-system is called a *C-n-tuple*; it is similar to a row vector in matrix algebra. In logic, a *C*-n-tuple corresponds to a separate conjunct.

*D-systems* model conjunctive normal forms of unary predicates. We denote a *D*-system as a matrix of component sets framed with reversed square brackets. *D*-systems provide easy calculating complements of *C*-systems.

For instance, the *D*-system $\overline{T}[XYZ] = \left] \begin{matrix} \overline{A} & \varnothing & \overline{C} \\ \overline{D} & \overline{E} & \varnothing \end{matrix} \right[$ is the complement of the *C*-system $T[XYZ] = \begin{bmatrix} A & * & C \\ D & E & * \end{bmatrix}$. Alike a *C*-system, a one-line *D*-system is called a *D-n-tuple*. In logic, a *D*-n-tuple models a separate disjunct.

Calculations of unions and intersections for *C*- and *D*-structures are specific; you can find their detailed description in [1],[2]. Please note that NTA provides implementing all operations of algebra of sets and all checks of relations among NTA objects (equality, inclusion, etc.) in matrix form, without having to represent these objects as sets of elementary *n*-tuples.

To process NTA objects defined on different diagrams, we have developed some *attributes operations*, *addition of a dummy attribute* (+*Attr*) and *elimination of an attribute* (-*Attr*) in particular. The operation +*Attr* corresponds to the rule of generalization in predicate calculus, so it does not change the semantics of any relations. For any NTA object, this operation simultaneously adds the name of a new attribute into the relation diagram and a new column with dummy components into the respective place of the NTA matrix representation.

Given the relation $R_k[XZ] = \begin{bmatrix} A_1 & A_3 \\ B_1 & B_3 \end{bmatrix}$ models the predicate $R_k(x, z)$, adding a dummy attribute $Y$ into $R_k[XZ]$ results in the formula $\forall y(R_k(x, z))$. This operation is done as $+Y(R_k[XZ]) = \begin{bmatrix} A_1 & * & A_3 \\ B_1 & * & B_3 \end{bmatrix}$.

The operation +*Attr* is often used to reduce some different-type NTA objects to the same relation diagram. Then we can perform all necessary operations and checks by means of standard NTA algorithms. Considering this, we have introduced *generalized operations* ($\cap_G$, $\cup_G$). They are possible after reducing NTA objects to the same relation diagram and semantically correspond to logical connectives: conjunction and disjunction. Our algebra of relations with these generalized operations is proved to be isomorphic to the ordinary algebra of sets. This way we have eliminated the restriction existed in the theory of relations and stated that algebra-of-sets' laws are only applicable to the relations defined upon the same Cartesian product.

For corresponding logical formulas, elimination of an attribute (let it be $X$ for example) from $C$-structures results in quantification $\exists x$, and from $C$-structures – in quantification $\forall x$ [1]. In NTA, this operation leads to deleting an attribute from the relation diagram and the respective column from the matrix representation of an NTA object. For instance, calculating $-Y(R[XYZ])$ for the $D$-system $R[XYZ] = \left]\begin{matrix} A & \varnothing & B \\ C & D & \varnothing \end{matrix}\right[$ gives us $Q[XZ] = \left]\begin{matrix} A & B \\ C & \varnothing \end{matrix}\right[$.

## 3. Logical Inference in NTA

Suppose that we have a system of premises (or axioms) $A_1$ , $A_2$ ,..., $A_n$ represented as NTA objects. Then NTA-based solving the two main problems of deductive inference looks as follows.

1. *Problem of correctness check for a consequence*. An alleged consequence $B$ follows from the premises, if the following generalized inclusion is true [1],[2]:

$$A = A_1 \cap_G \ldots \cap_G A_n \subseteq B. \quad (1)$$

This relation allows for correctness checks not only for the inference rules of classical logic, but also for rules specific to a certain knowledge system.

2. *Problem of derivation of consequences*. In order to solve this problem, we calculate an NTA object $A = A_1 \cap_G \ldots \cap_G A_n$ first and then choose a $B_i$ for which $A \subseteq B_i$ is true. We have developed special techniques that allow to find all possible corollaries of a known $A$ using the relation (1) [1],[2].

   Here we define NTA object $A$ as the *minimal consequence*. It is easy to prove that any reduction of the minimal consequence gives a formula underivable from the axioms.

   NTA provides several techniques that allow to generate possible corollaries. If we want to fix the number and/or the structure of attributes in the consequence, the search of $B_j$ reduces to choosing suitable projections of $A$. We calculate such projections by eliminating "redundant" attributes from the minimal consequence $A$ expressed as a $C$-system.

   As an example, let us consider the following system of premises: $A_1 = x_4 \supset (x_2 \vee x_3)$; $A_2 = x_1 \supset x_4$; $A_3 = x_2 \supset (x_1 \vee x_4)$. Suppose we want to find such consequences of this system as they contain one or two variables only. To solve this problem in NTA, we use the algorithm described below.

   1. Express logical formulas as NTA objects.
   2. Calculate the minimal consequence $A = A_1 \cap_G A_2 \cap_G A_3$. If $A$ is a $D$-system, we transform it into the equal $C$-system.

3. Choose uni- and bidimensional projections and check their informativity. A projection is not informative, if it contains the complete set of attributes values, because the corresponding consequence will be a valid formula.

For our example, let us define binary attributes $X_1$, $X_2$, $X_3$ and $X_4$ for all variables. After transformation premises into disjuncts, we obtain the following *D*-n-tuples:

$A_1[X_2X_3X_4]$ = ]{1} {1} {0}[; $A_2[X_1X_4]$ = ]{0} {1}[; $A_3[X_1X_2X_4]$ = ]{1} {0} {1}[.

Calculation of the minimal consequence *A* gives the *D*-system $A[X_1X_2X_3X_4] = \left]\begin{matrix} \varnothing & \{1\} & \{1\} & \{0\} \\ \{0\} & \varnothing & \varnothing & \{1\} \\ \{1\} & \{0\} & \varnothing & \{1\} \end{matrix}\right[$ that can be expressed by NTA standard algorithm as the *C*-system $A[X_1X_2X_3X_4] = \begin{bmatrix} \{0\} & \{0\} & * & \{0\} \\ * & \{1\} & * & \{1\} \\ * & \{0\} & \{1\} & \{1\} \end{bmatrix}$.

Let us check its projections. It is easy to see that all one-dimensional projections contain complete set of values of the corresponding attributes. So, there are no derivable consequences with one variable. As for bidimensional projections, the only projection $[X_2X_4]$ contains an incomplete set of values. After uniting the *C*-n-tuples of this projection, we obtain $B_j[X_2X_4] = \begin{bmatrix} \{0\} & * \\ \{1\} & \{1\} \end{bmatrix}$ that corresponds to the logical formula $x_2 \supset x_4$. This formula is one of the consequences from the given system of premises. If we express this structure in the relational diagram of the minimal consequence *A*, we will get $B_j[X_1X_2X_3X_4] = \begin{bmatrix} * & \{0\} & * & * \\ * & \{1\} & * & \{1\} \end{bmatrix}$.

**4. Inductive Generalizations in Consequences and Their Properties**

Let $A_1$, $A_2$, …, $A_n$ be a system of premises transformed into NTA objects and ***U*** is the universe of this system. ***U*** equals to the Cartesian product of all attributes domains. Using (1), we can conclude, in particular, that consequences of premises $A_1$, $A_2$, …, $A_n$ equal either the minimal consequence or the set $A \cup S$ where *S* is any non-empty subset of the set $\boldsymbol{U} \setminus A$. Sometimes, this added formula *S* increases uncertainty of the consequence. For instance, if the minimal consequence of certain premises states that “The weather will be sunny tomorrow” and $\boldsymbol{U} \setminus A$ contains another statement “It will rain tomorrow”, then the disjunction of these statements i.e. the statement “The weather will be sunny or rainy tomorrow” is a correct consequence too. However, often a certain superset of the minimal consequence can be a suitable consequence. As an example, we can take the famous rule of dilemma in the natural calculus where premises $A \supset C$, $B \supset C$ and $A \vee B$ infer *C*.

NTA structures are simple to represent in a metric space. We can assign them a probabilistic measure [3],[4] as well as, for instance, a power measure defined as the number of elementary *n*-tuples contained it an NTA object. Then the power of a *C*-n-tuple equals to the product of powers of its components. For example, if $R = [A\ B\ C]$, then $|R| = |A| \times |B| \times |C|$ where $|X|$ denotes the power of the set *X*. To count measures for other types of NTA objects, we have developed *orthogonalization techniques* [4] transforming any NTA objects into *C*-systems with disjoint *C*-n-tuples. Then the power of any NTA object immersed into a space with finite domains can be calculated as the sum of powers of such *C*-n-tuples.

The possibility to count powers of NTA objects and the relation (1) provide for estimating the number of all possible consequences for a certain system of premises [5]. Given $A_1$, $A_2$, …, $A_n$ is an NTA-expressed system of premises and ***U*** is its universe, the following theorem is true.

*Theorem*. The number of possible consequences from $n$ premises $A_i$ equals $2^N$ where $N = |\boldsymbol{U}| - |A|$ and $A = A_1 \cap_G \ldots \cap_G A_n$.

If $A \cup_G S$ is a consequence from the system of premises $A_1, A_2, \ldots, A_n$ and $S \neq \varnothing$, we can suppose that $S$ increases uncertainty of this consequence. However, sometimes the new consequence contains less variables than $A$. Then we can conjecture that the problem of finding consequences with certain properties is similar to the problem of seeking out regularities. In the case when the notation of the obtained consequence is simpler than the expression for the minimal consequence, the inductive generalization occurs. It is possible to find an NTA solution of this problem by investigating projections of the minimal consequence $A$.

By calculating powers of NTA objects, we estimate the degree of the inductive generalization. It seems reasonable to define this measure as a ratio between powers of the obtained consequence and the minimal one.

*Definition*. The measure of inductive generalization for a consequence $B_j$ inferred from the premises $A_1, A_2, \ldots, A_n$ equals to $\mu(B_j) = \dfrac{|B_j|}{|A|}$ where $A = A_1 \cap_G \ldots \cap_G A_n$.

Let us analyze two sets of attributes, namely $Attr(B_j)$ contained in the consequence $B_j$ and $Attr(A)$ present in the minimal consequence. If we consider $B_j$ as a separate formula in the same relation diagram with $A$, then $B_j$ contains only dummy components in the attributes belonging to $Attr(A) \setminus Attr(B_j)$, i.e. it contains all possible combinations of values for these attributes. Conversely, while the minimal consequence $A$ and $B_j$ have the same regularity in attributes from $Attr(B_j)$, $A$ does not contain all possible combinations of values for attributes from $Attr(A) \setminus Attr(B_j)$.

Evidently, $\mu(B_j) \geq 1$ and the more is the measure of generalization for an inferred consequence, the bigger is $\mu(B_j)$. Let us calculate $\mu(B_j)$ for the problem of deriving consequences considered above. As $A[X_1X_2X_3X_4]$ is an orthogonal $C$-system, it is easy to calculate its power. Since powers of dummy components "$*$" equal 2, we get $|A[X_1X_2X_3X_4]| = 8$. It is not also difficult to calculate the power of the consequence $B_j$ expressed as an orthogonal $C$-system $B_j[X_1X_2X_3X_4] = \begin{bmatrix} * & \{0\} & * & * \\ * & \{1\} & * & \{1\} \end{bmatrix}$ (see section 3). The power of the first $C$-n-tuple equals 8; the power of the second $C$-n-tuple equals 4. As a result, $|B_j| = 12$.

Now we compute the measure of generalization for the obtained consequence:

$$\mu(B_j) = 1.5.$$

Thus, the number of all possible elementary $n$-tuples satisfying the relation $x_2 \supset x_4$ in the projection $[X_2X_4]$ is 1.5 times bigger than the number of elementary $n$-tuples contained in the minimal consequence $A$.

## 5. Conclusion

The calculation technique described in this paper allows to estimate the degree of inductive generalization for consequences and obtain quantitative estimates for other results related to logical analysis of reasoning and premises. In particular, it is possible to predict the number of consequences for a given system of premises and the share of a minimal consequence in a universe.

**Acknowledgment**

The authors would like to thank the Russian Foundation for Basic Researches (grants 12-07-00302, 12-07-00550, 12-07-00689, 13-07-00318, 14-07-00256, 14-07-00257, 14-07-00205) and the Chair of the Russian Academy of Sciences (project 4.3 "Intelligent Databases" of the Programme # 16 of Basic Scientific Researches) for their aid in partial funding of this research.

## References


[1] Kulik B, Fridman A, Zuenko A. Algebraic Approach to Logical Inference Implementation. *Computing and Informatics (CAI).* 2012; 31(6): 1295-1328.

[2] Kulik B, Fridman A, Zuenko A. Logical Inference and Defeasible Reasoning in N-tuple Algebra. In: Naidenova X, Ignatov D. *Editors*. Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems. IGI Global, Hershey PA. 2013: 102-128.

[3] Ryabinin IA. Reliability of Engineering Systems: Principles and Analysis. Moscow: Mir Publishers. 1976.

[4] Kulik BA. N-Tuple Algebra-Based Probabilistic Logic. *Computer and Systems Sciences International*. 2007; 46(1): 111-120.

[5] Kulik BA, Zuenko AA, Fridman AYa. Deductive and Revised Reasoning on the Basis of Uniform Algebraic Approach. *Artificial Intelligence and Decision Making.* 2013; 4: 95-105.