CATEGORICAL SYSTEM ASPECTS OF MARKOV TOWER

Tolokonnikov G.K.
Corresponding Member of Russian Academy of Engineering
admcit@mail.ru

Five Star Paper

Article

The theory of algorithms and the mathematical platform of artificial intelligence are based on the constructive logic constructed by A.A. Markov and presented in a number of his latest articles. Markov's constructive logic and theory of algorithms are widely used in mathematical works, research on AI, as well as in philosophical research on constructivism. The constructive platform laid by A.A. Markov launched research in the field of fundamental science and in the field of applications. This report is devoted to solving the following problems in the specified areas. The report proposes a complete construction of the constructivist methodology used in the Markov approach. As is known, this problem is posed in detail in the book by I.D. Zaslavsky, who outlined a number of positions of the methodology, but we propose its complete construction, including those concerning the philosophy of constructivism, in the report. The categorical theory of systems that we are developing, formalizing the general systemic principles put forward by P.K. Anokhin, revealed the systemic aspects of the constructive approach of A.A. Markov, which are discussed in the report.

The set-theoretical paradigm requires us and other researchers to represent any objects of the external and mental world in our consciousness as sets and subsets (a set of students in a class, functions as sets, and so on). In the categorical theory of systems that we are developing, we replace the set-theoretical paradigm with a systems paradigm, when the specified objects are not sets, but systems and their collections. We have constructed an axiomatics of categorical systems theory similar to the axiomatics of set theory based on the theory of convolutional polycategories and categorical splices that we introduced [1], which turned out to be useful, in particular, for modeling neural networks [2]. While in set theory we rely on classical logic, the categorical theory of systems does not fix the logic for the theory of categorical splices in advance. The main attribute of a system is a system-forming factor, which generalizes the system-forming factor introduced by P.K. Anokhin in physiology as a useful result for the organism. This requirement can be viewed as a postulate of the categorical theory of systems. An example of a system-forming factor that differs from that adopted in physiology in the theory of functional systems is the system-forming factor of a biomechanical system, which includes the principle of survival as a biological part and the principle of least action as a mechanical part. The second postulate of the categorical theory of systems also generalizes a similar position of the theory of functional systems and consists in the requirement that both the system itself and all its properties must be obtained from the system-forming factor. The third postulate of the categorical theory of systems consists in the explicit consideration in the theory of researchers who build this theory for some reason. Science, like everything else, according to the systems paradigm is a system, its system-forming factor is the task, the goal of building a theory of the subject of research. Thus, in relation to science, we have a subject of research, researchers who are interested in studying the subject of research. The result of study in science is a set of true statements in some sense in a suitable language about the subject of science. Usually, science has a toolkit for conducting experiments and measurements. This toolkit allows for some statements about an object to be directly observed as true. However, not all statements about an object can be verified as true using tools and measurements. An example of such statements is statements of generality, as well as statements of existence in many cases. For example, researchers cannot directly verify the truth of the statement "all electrons have the same charge", there is a need for reasoning to justify the truth of the statement, in other words, logic is needed. However, there are many logics and which one to take, researchers must decide in accordance with the second postulate, based only on the consideration of the subject of research of this science. Logicians usually look for logic for science in natural language, but the criteria for such a search are not clear. Categorical systems theory gives an unambiguous answer to where to get logic for the theory of an object, logic is determined by the system-forming factor. So, for science, we have an object of research or a subject of science, researchers, those who study the object of research, tools for study, the toolkit also includes logic for reasoning. As a result, a set of true statements about the subject arises, which allows us to talk about its properties and, ultimately, that the subject has been studied, science is built in the form of a theory of its subject. Thus, initially we have researchers who are concerned with studying the subject of science and the subject itself. Let us emphasize that there is nothing else initially. Researchers must offer an alphabet, a language in which they will formulate statements about the subject, a concept of the truth of a statement about the subject, build a logic for obtaining the truth of some statements about the subject.

Let us note right away that a brilliant example of such a systematic construction of science is the theory of words in alphabets with logic in the form of constructive logic, constructed by Andrey Andreyevich Markov in the book on the theory of algorithms [3] and other works. Indeed, the subject of science, which is called constructive mathematics according to Markov, is the following constructive objects, sheets of paper in a box, stamps with paint (or a pen with ink) for imprinting letters on sheets, a constructive operation of imprinting letters in boxes to obtain words in the alphabet, and researchers who know how to do this. As the language of researchers, Markov takes a fragment of the Russian language, as one of the tools Markov takes the property of researchers to understand the graphic equality of two letters of the alphabet and the graphic inequality of two letters of the alphabet, he imposes requirements on the consciousness of researchers, he initially refuses to use classical or any other logic, sets the task of constructing a logic adequate to the subject of research. However, Markov was not interested in systemic issues, he did not clearly describe any complete constructivist methodology for his science. An attempt to complete the constructivist methodology was made by I.D. Zaslavsky [4], citing the fact that such a methodology had not been constructed by constructivists. However, he stopped at the stage of construction sufficient for his symmetric logic. We offer a complete constructivist methodology for words in alphabets in this work. Due to the limited volume of the article, a detailed version of the presentation of the methodology will be published in a separate work.

The constructivist methodology stops after describing constructive objects and constructive operations, and answers questions of truth only for those statements for which this can be done using constructive operations.

From the point of view of the systems approach, the constructivist methodology corresponds to general systemic constructiveness; it does not use the concept of infinity (actual infinity according to Cantor, potential infinity according to Brouwer and Markov, practical infinity according to Yesenin-Volpin).

Clarification of general systemic constructiveness leads to the following types of constructivity.

General systemic constructiveness: a finite alphabet is specified, a finite set of rules for constructing words (it is possible to construct individual elements of algebra from generators, use relations) and statements, the tools used by researchers for direct establishment (graphic equality of letters or graphic difference of letters, etc.) of the truth of statements by them, other truth values, except for "true", are not introduced, the concept of the set of all words in the alphabet, logic is not considered;

Constructiveness according to A.A. Markov: constructive logic, constructed in [3, 5-13], is added to the general system constructiveness; potential infinity is used;

Constructiveness according to A.I. Maltsev: elements of the theory of recursive functions, classical logic and set theory, necessary for the constructive numbered algebras introduced by A.I. Maltsev (which are constructed using generators and relations), are added to the general system constructiveness, based on the concept of actual infinity (the approach was significantly developed in the works of the Siberian mathematical school of academicians Yu.L. Ershov and S.S. Goncharov, see Goncharov, S.S., Ershov, Yu.L. Constructive Models, Novosibirsk: Nauchnaya Kniga, 1999, 345 p.).

We can talk about constructiveness according to A.S. Yesenin-Volpin, whose approach is based on the concept of practical infinity, as well as on constructiveness according to L. Brouwer, partially reflected in intuitionistic logic and using potential infinity.

The proposed constructivist methodology is based on the above postulates of the categorical theory of systems.

So, we have the subject of science "constructive mathematics" in the form of words in alphabets, as well as a group of researchers who, for some reason, seek to build a theory describing the properties of this subject. The constructivist methodology is nothing more than a clarification of the formulated position on the existence of a subject of science and researchers. Let's move on to its description.

Researchers are ourselves and a number of selected specific colleagues. To build a theory of words in alphabets, we must determine the requirements for researchers that are sufficient for this. It has long been known that a theory can depend on the level of knowledge of researchers; a formalized version of this phenomenon is provided by the well-known Kripke semantics in intuitionistic logic. So, we will require a certain level of education from researchers, for which we can take the least educated of the researchers we have selected; knowledge beyond his level, for example, is not used by other researchers when building a theory. This knowledge primarily includes pronunciation and understanding of some fragment of natural language, understanding and reproducibility by researchers of some commands corresponding to constructions in the theory of words. Researchers must be conscious, able to perform the following actions: Take sheets of paper lined in a cage from a warehouse (as sheets can be used, as in Turing machines, tapes with one row of cells, for example, from 100 cells in each tape), glue them together, increasing their area in a way that is obvious to us and them, take stamps for a selected number of letters from a warehouse, be able to, using paint, imprint one imprint per cell, thus obtaining words in the alphabet from the specified letters. Additional requirements for researchers are formulated as the constructivist methodology is described.

So, we have sheets of paper in a cage, a fixed set of stamps for letters, paint for making imprints in cages on sheets of paper. We refer to the above as a toolkit that will be supplemented further. Researchers must understand the statements "two prints located in adjacent cells are graphically equal", "two prints located in adjacent cells are graphically different from each other". In addition to insight, which consists in the act of understanding statements, researchers have the opportunity for insight, meaning the truth of each of these statements when examining letters on sheets of squared paper. We have introduced the truth value "true", we emphasize that we do not introduce other (for example, "false") truth values for statements. What we are talking about, G. Froege called "comprehension of thought" and "comprehension of the truth of thought", but we will not need his deep research in the search for classical logic in natural language. We recorded statements from which all other statements of an adequate language for the theory of words in alphabets will be built. We are distracted from the questions of how researchers have insight, what feelings they experience, what physiological processes underlie insights. Let us emphasize that we do not need to know how researchers came to this or that knowledge, and so on.

It is impossible to list everything that we are distracted from, but it is enough for us to record what we require from researchers.

We consider a whole series of concepts and actions that we will attribute to the use of scientific tools to be feasible and not defined in theory. Namely, how researchers understand each other and act when taking sheets of paper, gluing them together, when transmitting information to each other about the facts of performing constructive operations, in what language they communicate (this may include, for example, sign language), and so on. The question of what a sign is (this is one of the most difficult questions of philosophy) does not require an answer in our case, since we know perfectly well what a letter is in our specific case and can imprint letter prints on sheets of paper in a box.

According to system requirements, researchers cannot set the logic for their theory, for example, classical logic, simply by taking it from their knowledge. It is necessary to derive and construct the logic based on the object of study. However, the question of how researchers guess what associations they use lies, like similar physiological and psychological questions, outside the theory of words in alphabets.

Nevertheless, leaving behind the scenes the question of how this or that step in the theory appeared, we, like other researchers, can make incorrect steps. The discovery of incorrect steps leads to a revision and a search for other correct steps for the theory's construction. The question of the correctness or incorrectness of the theory, like the very concept of correctness, also goes beyond its scope.

There are at least five versions of constructive logic, in the book [3] ("Markov tower"), in the version [4] in French ("Markov tower 1971"), in his articles in the journal Reports of the Academy of Sciences [6-13] ("Markov tower 1974"), in the version developed under the supervision of N.M. Nagorny at the Computing Center of the Academy of Sciences [14] ("Markov-Nagorny tower") and in the version given by G.E. Mintz [15] ("Markov-Mintz tower"). Note that the development of a unified presentation of Markov's constructive logic (announced in [3]) has not yet been completed. For the sake of certainty in constructing Markov's constructivist methodology, we will consider a fragment of the Markov-Nagorny tower version. The letters of the alphabet A are the following |, ◊, *, =, ≠, ∃, ∧, ∨, ∀, < .

Using the constructive operation of adding a letter to a word on the right, we construct words in the alphabet ◊, * in the form of magma with the generator ◊ and with the sign * of the binary operation. We will call the resulting words constant terms.

We will depict constant terms on white sheets of paper in a cage. We will make yellow sheets of paper in a cage (you can paint existing white sheets yellow) on them we will depict words in the alphabet A, the seals used for white sheets we also use for yellow sheets. Words of the form |...|◊ will be called variables. We will add them as additional generators to the magma indicated above, obtaining terms. We use the symbols =, ≠ as binary predicate letters on terms, we will call their application to terms elementary formulas, from elementary formulas, as generators, we will construct new formulas using the symbols ∧, ∨, considered as binary operations on formulas, the symbol ∃ is used in the usual way to form formulas, the symbols ∀, < are used to form formulas of the form ∀|...|◊ F (F -formula). We have obtained a version of the language $Я_1$ of the Markov tower. It is easy to construct a constructive operation of replacing variables with constant terms in formulas from the existing operations. Formulas without variables, for example, obtained by the indicated replacement, are called closed. Similarly, it is possible to construct a constructive operation of replacing one word with another, and also to obtain a construction of the normal Markov algorithm as a constructive operation. For closed formulas, it is possible to define a constructive operation of establishing truth, extending this operation from elementary formulas. =pq is a true formula if constant terms p, q are graphically equal, ≠pq is a true formula if constant terms p, q are graphically different from each other.

Based on these true statements, we can determine the truth of statements with connectives $\wedge$, $\vee$, namely, the statement $\wedge PQ$ is true when both closed formulas are true, $\vee PQ$ is true when it is shown that P is a true closed formula, the second case of truth is when it is shown that Q is a true closed formula.

To establish the truth of formulas with quantifier icons, it is necessary to use the constructive operation of replacing variables with constant terms, taking into account which we define: $\exists xF$ is true (x is a variable, F has only one parameter in the form of a variable x), if we can present a constructive operation of constructing a constant term, which, when substituted into all occurrences of the variable in F, turns it into a true formula.

Unlike the previous case, in this case there is no guarantee of establishing the truth, since no recipe is given for finding the required constant term.

The truth for a closed formula $\forall x<tP$ is determined through the truth of P for a number of constant terms using some constructive operation.

Now we will take into account another property of researchers - the ability to draw imprints of letters in the imagination. We, like other researchers, can reproduce in the imagination sheets in a cage and the imprinting of imprints of letters. We will call the imaginary table on which the sheets in a cage are laid out, a screen of inner vision. We, like other researchers, can verify the truth of statements about the graphic equality or graphic difference of imprints on imaginary sheets in a cage. For imaginary sheets in a cage, we will expand the alphabet, including letters of the Russian alphabet, as well as letters, for example, of the Latin alphabet, which we will need for statements on the screen of inner vision.

Since the imprints on the white sheets and on the yellow sheets are imprinted with the same seals, both statements about the graphic equality and graphic difference of the imprints are determined to be true on the basis of, in fact, the same insights that we discussed only for the white sheets. A significantly different kind of insight about graphic equality or graphic difference is necessary for researchers if the statement includes one or both mental imprints. We accept such a requirement for researchers.

Now we will collect postulates for the concept of potential infinity. The main one is that if a certain word is printed, then an imprint of another letter can be added to it on the right. The second postulate is that we and other researchers have an unlimited number of sheets of paper, paint, and time to write out words in alphabets. Since this is impossible, we are deliberately constructing an approximate theory. It is possible to construct a more precise theory of words in alphabets (for example, using the approach of A.S. Yesenin-Volpin), but here we will not deviate from Markov's constructivism.

With the help of one single constructive physical operation of assigning a letter to a word on the right and the mental operation of establishing the truth of equality and graphic inequality of letters, it is possible to construct a whole series of other constructive operations, for example, the operation of assigning a letter to a word on the left, the operation of erasing a letter on the right or left, the operation of establishing graphic equality and graphic inequality not only of letters, but also of words.

Now we will move on to comparing the statement on yellow sheets of paper with the statement of the Russian language, which we and researchers can form on the screen of inner vision using the Russian alphabet.

Note. What the reader sees on the monitor screen or printed on paper can be considered a copy of the text displayed on the screen of inner vision. Thus, on the screen of inner vision we and other researchers can easily depict the statements “the imprint of some letter is graphically equal to the imprint of some letter” or “the imprint of some letter is graphically different from the imprint of some letter”. They express a certain thought according to G. Frege, which we and other researchers comprehend, realizing the appropriate insight.

Truth is objective (the question of the objectivity of truth is the most difficult in philosophy) in the precise sense that we and other researchers comprehend the same statements in the same way and comprehend their truth in the same way.

Now we introduce another requirement for us and researchers, consisting in taking into account in the theory another type of insight, which is also familiar to us. This refers to the act of comparing two statements of graphic comparison written on the screen of inner vision with two statements of formal language on yellow sheets of paper. We can, by matching the symbols of the connections with their usual names ("and", "or", etc.) unambiguously match any statement of the formal language on the yellow sheets of paper with the corresponding statement in Russian on the screen of the inner vision.

Since a thought can be expressed on the screen of the inner vision with the help of different statements of the Russian language, the translation of these statements onto the yellow sheets of paper is not always unambiguous.

We could use a computer monitor instead of sheets of paper, then we would get the transfer of statements and the transfer (with the necessary programming) of the procedure for determining the truth of statements to the computer. At the same time, we do not transfer the act of comprehending the thought contained in the statement, the act of comprehending the truth of the statement, which are insights, to the computer. This remark is important for thinking about modeling consciousness in a computer.

The next important insight that we introduce into consideration and require us and researchers to take it into account for the theory consists of matching (fixing any matching, we will talk about numbering) words on yellow sheets of paper to words (constant terms) on white sheets of paper. This is an analogue of the Gödel numbering of language.

We do not introduce the concept of a set, but we can introduce a constructive concept of a property. Formulas with one parameter are called generated formulas, they generate sets (are synonyms for properties) of constant terms, which, when substituted into this formula, make it true. We can say that a term has a property, but we do not introduce sets of terms that have a given property.

Now we can talk about the property of closed formulas to be true. We can construct a generated formula that corresponds to the property for a closed formula "to be true". Using the numbering, we obtain the property of "being true" for a closed formula if its number (the constant term associated with it) when substituted into the generated formula yields a true closed formula. As we can see, if we recall Tarski's theorem on the non-representability of truth in ordinary formal arithmetic, the language of $Я_1$ is very strong. To illustrate this, we will give another formula for the statement "there exists an odd perfect natural number" (see the article [16] for explanations and notations)

$$\exists v \exists v_1 \ldots \exists v_s \Sigma_s(v_1, v_2, \ldots, v_s; v) \wedge [\wedge_{l,j}{}^{s}(v_l \neq v_j) \wedge_{k=1}{}^{s} D(v, v_k)] \wedge H(v).$$

References

1. Tolokonnikov G.K. Categorical splices, categorical systems and their applications in algebraic biology. Biomachsystems, v.5, no.1, 2021, pp. 148-235
2. Tolokonnikov G.K. Convolution Polycategories and Categorical Splices for Modeling Neural Networks. Advances in Intelligent Systems and Computing, Volume 938, 2020, pp. 259-267, https://doi.org/10.1007/978-3-030-16621-2,
3. Markov A.A., Nagorny N.M. Theory of Algorithms, Moscow, Nauka, 1984, 432 p.
4. Zaslavsky I.D. Symmetric Constructive Logic, Yerevan, 1978, 281 p.
5. Markov A.A. Essay on the Construction of a Mathematical Constructive Logic, Revue Intern. de Philosophie, 1971, no. 98, Fasc. 4.
6. Markov A.A. (1) On the Language $Я_0$, Reports of the USSR Academy of Sciences, 1974, 214, no.1, pp. 40–43.
7. Markov A.A. (2) On the Language $Я_1$, Reports of the USSR Academy of Sciences, 1974,214,no.2,pp.279–282.
8. Markov A. A. (3) On the language $Я_2$, Dokl. Academy of Sciences of the USSR, 1974, 214, No. 3, p. 513–516.
9. Markov A. A. (4) On the language $Я_3$, Dokl. Academy of Sciences of the USSR, 1974, 214, No. 4, p. 765–768.
10. Markov A. A. (5) On languages $Я_4$, $Я_5$,.., Dokl. Academy of Sciences of the USSR, 1974, 214, No. 5, p. 1031–1034.
11. Markov A. A. (6) On the language $Я_\omega$, Dokl. Academy of Sciences of the USSR,1974,214,No.6, p.1262–1264.
12. Markov A. A. (7) On the language $Я_{\omega|}$, Dokl. Akad. Sci. USSR, 1974, 215, No. 1, pp. 57–60.
13. Markov A. A. (8) On the completeness of the classical predicate calculus in constructive mathematical logic, Dokl. Akad. Sci. USSR, 1974, 215, No. 2, pp. 266–269.
14. Vladimirov A. A., Dombrovsky-Kabanchenko M. N. Stepped semantic system, Moscow, Computing Center of the Russian Academy of Sciences, 2009, 305 pp.
15. Mintz G. E. Stepped semantics of A. A. Markov, pp. 348–357, in the book. Barwise J. Handbook of Mathematical Logic, v.4, Moscow, Nauka, 1983, 392 p.
16. Tolokonnikov G.K. Logic for Categorical Systems, Biomachsystems, 2019, v.3, no.1, pp.77-109.

Abut the Author

Georgy Konstantinovich Tolokonnikov Candidate of Physical and Mathematical Sciences, employee of FNATS VIM RAS. He is Co-Head of the Section "Mathematical Biology and Systems Theory" of the Scientific Council under the Presidium of the Russian Academy of Sciences on Artificial Intelligence (NSMII RAS). He is the author of 3 monographs and about 100 articles in journals He is a specialist in the field of mathematical physics, algebra, category theory, theory of functional and categorical systems and artificial intelligence. The main directions open and developed by him: the method of functional equations in the theory of algebras; categorical systems theory.

End of Article

1974 version of the Markov tower

This material has been included here just for reference purposes.

E. G. Rajan

The Languages { $Я_\alpha$}

Language $Я_0$

The starting point of Markov's logic is the language $Я_0$. A *Literoid* is a nonempty word form a one-letter alphabet. *Variables* are nonempty words from a one-letter alphabet. Literoids and variables are also known as *atoms*. *Verboids* are formed by concatenating literoids to verboids. The null string Λ is a verboid. By a *term* we mean either a verboid or an atom. Let us assume that there occurs a variable X in a term U. Substitution of the variable X by another term T is a permissible operation. By an *operation,* we mean the action of a normal algorithm $\mathcal{N}$ on a term. The result of the operation of substitution of X by T in U by means of a normal algorithm $\mathcal{N}$ is also a term and is denoted by $\Im\lfloor XU\square T\rfloor$ where $\square$ is an auxiliary symbol and the symbols $\lfloor$ and $\rfloor$ play the role of the left and the right parentheses respectively. For example, for any term T, $\Im\lfloor X\Lambda\square T\rfloor \underset{=}{\simeq} \Lambda$ where the symbol $\underset{=}{\simeq}$ stands for *equal by definition*. The notion of *result of the operation of substitution* could be extended to all the remaining languages.

An *elementary formula* is a string of the form T σ U where T and U are terms and σ is a *comparer.* By a *comparer,* we refer to any one of the symbols = or ≠ or $\doteq$ or $\not\doteq$ where the symbols = and ≠ compare two terms and the symbols $\doteq$ and ≠ compare two words or verboids.

The only logical connectives that are permitted in $Я_0$ are (i) & (conjunction) and (ii) V (disjunction). The quantifier symbols ∀ and ∃ are allowed in a restricted manner. Quantifiers and connectives are called *logical symbols*. Formulas of $Я_0$ are constructed using the following rules

(i) If A is an elementary formula then it is a formula $Я_0$. We shall designate a formula of $Я_0$ as Fm0.
(ii) If A and B are Fm0's and λ is a connective, then λAB is an Fm0.
(iii) If A is an Fm0, Q is a quantifier, X is a variable, U is a term and β is a limiter (either the symbol < or the symbol >), then the string QUβXA is an Fm0.

The variable X is allowed to be substituted only by certain words or verboids. For example, the formula ∀U<XA where U is a *constant term,* express that every formula of the type $\Im_0\lfloor XAQ\rfloor$ is true where Q is the verboidal prefix of the *meaning of* U. Three items are to be explained now: (i) Constant Term: It is either the null string Λ or of the form PQ where P is a constant term and Q is a literoid. (ii) The meaning of a constant term is always a verboid determined by a normal algorithm (iii) $\Im_0\lfloor XAQ\rfloor$ is the result of substituting the variable X by a term Q in the formula A which is also a formula. To be more specific, the quantification in an Fm0 is to be understood as that which allows the bounded variable to range in a predetermined, bounded set of words or verboids only.

The notion of a *parameter* is important in constructive logic. By *parameters* we mean the variables that occur either in an elementary formula or in the formulas A and B contained in the formula λAB or in U and A of the formula QUβXA excluding X. The term parameter conveys the same meaning as the term *free variable* conveys in classical logic. As a result, we have the notion of a *closed formula* in constructive logic similar to that of a *sentence* in the classical logic. In general, a closed formula of a language $Я_\alpha$ is a formula without parameters as defined in $Я_\alpha$. Closed formulas of a language $Я_\alpha$ are designated as CFα's. The set of all formulas of a language $Я_\alpha$ is denoted by {FMα} and the set of closed formulas by {CFα}. {CFα} is a subset of {FMα}. So, {CF0} is the set of all closed formulas in the languages $Я_0$ and it is a subset of {FM0}.

The language $Я_0$ does not allow the direct use of a negation symbol in an FM0. However, it is possible to obtain an FM0 which is semantically a negative equivalent of another by means of a normal algorithm $\mathcal{N}^{\neg}{}_{Я0}$ whose scheme is given below:

$\mathcal{N}^{\neg}{}_{Я0}$:	Substitution Formula				Formula Number
	$\alpha\xi$	→	$\xi\alpha$	(ξ denotes every symbol of $Я_0$ other than =, ≠, $\doteq$ and $\not\doteq$)	(00)
	α=	→	=α		(01)
	α≠	→	≠α		(02)
	$\alpha\doteq$	→	$\doteq\alpha$		(03)
	$\alpha\not\doteq$	→	$\not\doteq\alpha$		(04)
	α&	→	&α		(05)

$\alpha\vee \rightarrow \vee\alpha$ (06)

$\alpha\forall \rightarrow \forall\alpha$ (07)

$\alpha\exists \rightarrow \exists\alpha$ (08)

$\alpha \rightarrow \cdot$ (09)

$\rightarrow \alpha$ (10)

Language Я₁

The languages $Я_1$ allows the construction of the formula FM1's by means of the following rules:

(i) If A is an Fm0, then it is also an FM1.

(ii) It λ is a connective (& or $\vee$), A and B are Fm1's then the string λAB is an Fm1 provided either A or B is certainly not an Fm0.

(iii) If X is a variable and A is an Fm1 then $\exists$XA is an Fm1.

Parameters and closed formulas are defined in the same manner as they are defined in $Я_0$.
$Я_1$ describes the operation of normal algorithms in some alphabet $\mathcal{A}$ in the following manner. Let x, y and z be variables. Let us denote the translation of a normal algorithm $\mathcal{N}$ by $[\mathcal{N}]^{\mathcal{T}}$ and its transcription by $\{\mathcal{N}\}$. In the same manner, the translation of a word P is denoted by $[P]^{\mathcal{T}}$. Now, the following types of constructions are permitted:

(i) $(x\neg)_{\mathcal{N}}$ is an Fm0 without parameters different from x such that $\Im_0\lfloor x(x\neg)_{\mathcal{N}}[P]^{\mathcal{T}}\rfloor$ is a valid CF0 provided the word P is not accepted by $\mathcal{N}$.

(ii) $(x\vdash y)_{\mathcal{N}}$ is an Fm0 without parameters different from x and y such that $\Im_0\lfloor y\Im_0\lfloor x\ (x\vdash y)_{\mathcal{N}}[P]^{\mathcal{T}}[Q]^{\mathcal{T}}\rfloor$ is a valid CF0 provided the word P is simple rewritten as Q by $\mathcal{N}$.

(iii) $(x\vdash y)_{\mathcal{N}}$ is an Fm0 without parameters different from x and y such that $\Im_0\lfloor y\Im_0\lfloor x(x\vdash .\ y)_{\mathcal{N}}[P]^{\mathcal{T}}[Q]^{\mathcal{T}}\rfloor$ is a valid CF0 provided the word P is concludingly rewritten as Q by $\mathcal{N}$.

(iv) $(x!)_{\mathcal{N}}$ is an Fm1 without parameters different from x such that $\Im_1\lfloor x(x!)_{\mathcal{N}}[P]^{\mathcal{T}}\rfloor$ is a valid CF1 provided the word P from A is accepted by $\mathcal{N}$.

(v) $(x\Rightarrow y)_{\mathcal{N}}$is an Fm1 without parameters different from x and y such that $\Im_1\lfloor y\Im_1\lfloor x(x\Rightarrow y)_{\mathcal{N}}[P]^{\mathcal{T}}[Q]^{\mathcal{T}}\rfloor$ is a valid CF1 provided P is simply transformed to Q by $\mathcal{N}$.

(vi) $(x\Rightarrow\cdot y)_{\mathcal{N}}$ is an Fm1 without parameters different from x and y such that $\Im_1\lfloor y\Im_1\lfloor x(x\Rightarrow\cdot y)_{\mathcal{N}}[P]^{\mathcal{T}}\ [Q]^{\mathcal{T}}\rfloor$ is a valid CF1 provided P is concludingly transformed to Q by $\mathcal{N}$.

(vii) $(x\!:\ y\Rightarrow z)_{\mathcal{N}}$ is a valid Fm1 without parameters different from x, y and z so that $\Im_1\lfloor z\Im_1\lfloor x(x\!:\ y\Rightarrow z)\{\mathcal{N}\}\rfloor\,[Q]^{\mathcal{T}}$ $[R]^{\mathcal{T}}\rfloor$ is a valid CF1 provided the verboid $[Q]^{\mathcal{T}}$ which is the translated version of the word Q from $\mathcal{A}_0 = \{O\,|\}$, is transformed to verboid $[R]^{\mathcal{T}}$ when the transcription of $\mathcal{N}$ is applied to it.

An important notion that $Я_1$ introduces the calculus $\mathcal{C}$, which is based on the notion of *deducibility* of a CF1 from another. Let S be a finite series of CF1's (that is, a finite number of CF1's are written one after the other in the form of a string). Now we shall call a CF1: C as an immediate consequence of S only in the following cases: (i) if there is a valid CF0, (ii) if A and B are in S such that $C \doteqdot \&AB$, (iii) if either A or B is in S such that $C \doteqdot \vee AB$ and (iv) if $\Im_1\lfloor XDQ\rfloor$ is in S such that $C \doteqdot \exists XD$. Now the notion of deducibility is to be understood in the following manner. If S is a deduction and a series of CF1's and C is an immediate consequence from S then SC is a deduction. Obviously the null string Λ is a deduction. In general, the notion of deducibility is extended to all the remaining languages, in the sense that, for every language $Я_\alpha$ there is a system of rules of deduction S_α using with a CFα could be deduced from another.

Language Я₂

The language $Я_2$ is richer than $Я_1$, in the sense that, it provides rules for the use of *implication* and *negation* of 0^{th} order and *universal quantifier* in constructing Fm2's. Implication is denoted by the symbol $\supset$

An Fm2 is constructed using the following rules:

(i) If A is an Fm1 then it is also an Fm2.

(ii) If A and B are Fm1's then $\supset$AB is an Fm2.

(iii) If A and B are Fm2's and one of them is certainly not an Fm1 then & AB is an Fm2.

(iv) If X is a variable and A is an Fm2 then $\forall$XA is an Fm2.

It is important to note that CF2's cannot be combined using the logical connective of disjunction. Also existential quantifiers cannot be used in the construction of Fm2's. Fm2's of the form $\supset$AB where A and B are Fm1's are called implications of the 0^{th} order. The implication of the 0^{th} order is to be understood in the following manner. Let S be a series of Fm1's. Let A and B be two Fm1's. Then the Fm2: $\supset$AB is interpreted as *for an arbitrary S,* ((S is a deduction of A) or B). The negation of the 0^{th} order is defined as $\neg A \underset{=}{\simeq} A\ (\neq)$ where A is an Fm1. The following are the basic deductive rules of the language $Я_2$:

(i) $\dfrac{A \supset AB}{B}$ (ii) $\dfrac{\supset AB \supset BC}{\supset AC}$ (iii) $\dfrac{B}{\supset AB}$

(iv) $\dfrac{\supset AB \supset AC}{\supset A\&BC}$ (v) $\dfrac{\supset AC \supset BC}{\supset \vee ABC}$ (vi) $\dfrac{DE}{\&DE}$

(vii) $\dfrac{\&DE}{E}$ (viii) $\dfrac{\&DE}{D}$

(ix) $\dfrac{\Im_2 \lfloor XHQ \rfloor \text{ for every verboid } Q(x)}{\forall XH}$ $\dfrac{\forall XH}{\Im_2 \lfloor XHQ \rfloor}$

(xi) $\dfrac{\forall X \supset GA}{\supset \exists XHA}$

In addition to these rules, Я2 provides a *semiformal system* S_2 consisting of thirteen rules of deduction, which decide the deducibility of a CF2 from another.

Let K be a CF2. Let Y be a condition, which could be meaningfully imposed on a CF2. Then Y is called *K-inductive* if the following thirteen conditions hold:

(i) K satisfies Y.
(ii) Every valid CF2 satisfies Y.
(iii) Whenever the CF2's A and ⊃AB satisfy Y, B satisfy Y.
(iv) Whenever the CF2's ⊃AB and ⊃BC satisfy Y, then ⊃AC satisfies Y.
(v) Whenever the CF2 B satisfies Y, then ⊃AB satisfies Y.
(vi) Whenever the CF2's ⊃AB and ⊃AC satisfy Y, then ⊃A&BC satisfies Y.
(vii) Whenever the CF2's ⊃Ac and ⊃BC satisfy Y, then ⊃∨ABC satisfies Y.
(viii) Whenever D and E satisfy Y, then the CF2 & Dc satisfies Y.
(ix) Whenever the CF2 & DE satisfies Y, then the CF2 D satisfies Y.
(x) Whenever the CF2 & DE satisfies Y, then the CF2 E satisfies Y.
(xi) Whenever we have a general method enabling us to establish for fixed X and H and for any verboid Q that the CF2: $\Im_2 \lfloor XHQ \rfloor$ satisfy Y, then CF2 ∀XH satisfies Y.
(xii) Whenever the CF2 ∀XH satisfies Y, then CF2$\Im_2 \lfloor XHQ \rfloor$ satisfies Y.
(xiii) Whenever the CF2 ∀X⊃GA satisfies Y, then the CF2 ⊃∃XGA satisfies Y.

Theorem
If a condition Y is K-inductive, then every CF2 which is deducible from K satisfies the condition Y.

Language Я3

This language is just an extension of Я2 in the sense that it provides rules for the use of implication and negation of the first order. They are denoted by the symbols ⊐and ¬| respectively. An Fm3 is constructed using the following rules:
(i) If A is an Fm2 then it is also an Fm3.
(ii) If A and B are Fm2's then ⊐|AB is an Fm3.
(iii) If C and D are Fm3's such that one of them is certainly not an Fm2, then & AB is an Fm3.
(iv) If X is a variable are E is an Fm3 but not an Fm2 then ∀XE is an Fm3.

The negation of the first order is defined as ¬|B $\simeq$ ⊐|B(≠) where B is an Fm2.

Theorem

$\dfrac{\neg|\neg A}{A}$ where A is arbitrary Fm1.

¬A is a quasi elementary formula since ¬A is nothing but ⊃A(≠). The above theorem is known as *Markov's Principle of Constructive Choice.*

Let K be a CF3 and Y be a condition, which could be meaningfully imposed on K. Then Y is called 3K-inductive if thirteen conditions similar to the ones given in Я2 hold. Now, if Y is 3K-inductive, then every CF3 which is 3-deducible [Refer to deducibility of CF2's] from K satisfies the condition Y. The notion of K-induction could be extended to all the remaining languages in a similar manner.

Languages Я$_4$, Я$_5$, ……

The languages Я$_4$, Я$_5$ and so on, are the generalizations of the successively extended languages Я$_0$, Я$_1$, Я$_2$ and Я$_3$. Any language Я$_3$ is identified by Я$_N$: N = 4, 5, 6, …. With the corresponding system of deductive rules S_N : N = 4, 5, 6, … For every transition from a language Я$_N$ to its successor Я$_{N|}$, a new kind of implication emerges, an implication of order N-1. Thus a languages Я$_N$ accumulates implications of orders 0 to N-2. However such an abundance of implications does no harm since they agree among themselves. An FmN| is constructed in the strength of the following rules:
(i) If A is an FmN then it is also an FmN|.
(ii) If A and B are FmN's then ⊃(N-1)AB is an FmN|.
(iii) If C and FmN|'s such that one of them is certainly not an FmN , then &CD is an FmN|.
(iv) If X is a variable and E is an FmN| but not an FmN then ∀XE is an FmN|.

N|-deducibility and N|K-induction are defined in exactly the same manner as they are defined in the previous languages. In general, the negation of order N in language Я$_{N||}$ is defined as ¬NA $\cong$ ⊃NA (≠) where A is an FmN|.

Theorem

$$\frac{\neg M \neg NA}{A}$$ (Proof is omitted here)

Language Я$_\omega$

The notion of *abstraction of potential realizability* [45] allows one to unite all the languages so far seen, to form what is known as Я$_\omega$. An Fmω is constructed as per the following rules:
(i) If A is an Fm1 then it is also an Fmω.
(ii) If C and D are Fmω's and one of them is certainly not an Fm1 then & CD is an FMω
(iii) If E and F are Fmω's then ⊃EF is an Fmω.
(iv) If X is a variable and E is an Fmω then ∀XE is an Fmω.

Fmω's are known as *normal formulas.* Two Fmω's cannot be combined disjunctively nor they be existentially quantified. However, Я$_\omega$ allows the use of quasi disjunction and quasi existential quantifier in the construction of Fmω's. We shall denote quasi disjunction by the symbol $\underline{\vee}$ and the quasi existential quantifier by the symbol $\underline{\exists}$ and agree to their definitions, so that the following hold:

(i) $\underline{\vee}$AB $\cong$ ¬&¬A¬B where A and B are Fmω's
(ii) $\underline{\exists}$XA $\cong$ ¬∀X¬A where X is a variable and A is an Fmω.

Language Я$_{\omega|}$

This language is an extension of Я$_\omega$ and is closed under all traditional logical connectives. Fmω|'s are constructed by virtue of the following rules:
(i) If A is an Fmω then it is also an Fmω|.
(ii) If A and B are Fmω's but one of them is certainly not an Fmω and λ is a logical connective either ⊃ or & then λAB is an Fmω|.
(iii) If A is an Fmω| but not an Fmω and X is a variable then ∀XA is an Fmω|.
(iv) If A is an Fmω| but not an Fm1 and X is a variable then the string ∃XA is an Fmω|.

The languages Я$_1$ is a sublanguage of Я$_{\omega|}$. For any two Fmω|'s A and B in which one is certainly not an Fm1 the disjunctive formula ∨AB is defined as ∨AB $\cong$ ∃z&⊃(z=)A⊃(z≠)B where z is a variable other than the parameters of A and B. The negation is defined as ¬A $\cong$ ⊃A(≠) where A is an Fmω|. The rule Modus Ponens holds here: *Whenever the CFω|'s* A and B are such that both A and ⊃AB are true in Я$_{\omega|}$ *then* B *is also true in* Я$_{\omega|}$. The principle of constructive choice is stated in Я$_{\omega|}$ as

$$\supset\forall X\vee D\neg D\supset\neg\neg\exists XD\exists XD$$

The above principle plays the fundamental role in the formulation of various constructive theories. For example, let {CFα} be the set of closed formulas of the language Я$_\alpha$. Following standard terminology, any subset of {CFα} is known as a *constructive theory,* Φ_α, of that language, and a structure M is a model of that constructive theory Φ_α if every closed formula of the theory holds in M also. Consider now a constructive theory, Th($\Re$), defined by the following five closed formulas that are Я$_{\omega|}$-provable.

Th($\Re$):

Fmω|.1 $\supset\neg\neg\, !\mathcal{N}\lfloor P\rfloor\, !\mathcal{N}\lfloor P\rfloor$
Fmω|.2 $\supset\mathcal{N}\lfloor P\rfloor \doteq Q!\mathcal{N}\lfloor P\rfloor$

Fmω|.3 $\supset \forall xyz \mathfrak{I}_1 \lfloor z \mathfrak{I}_1 \lfloor y \mathfrak{I}_1 \lfloor x(x{:}\ y \Rightarrow z)\{\mathcal{N}_i\}\rfloor [P]^{\mathcal{T}}\rfloor [Q]^{\mathcal{T}}\rfloor \doteq \mathfrak{I}_1 \lfloor z \mathfrak{I}_1 \lfloor y \mathfrak{I}_1 \lfloor x(x{:}\ y \Rightarrow z)\{\mathcal{N}_j\}\rfloor [P]^{\mathcal{T}}\rfloor [Q]^{\mathcal{T}}\rfloor \{\mathcal{N}_i\} \simeq \{\mathcal{N}_j\}$
Fmω|.4 $\supset \forall XYZ\ \&\&\ \mathcal{N}_i \lfloor X \rfloor \doteq Y \mathcal{N}_j \lfloor Y \rfloor \doteq Z\, \mathcal{N}_k \lfloor X \rfloor \doteq Z \mathcal{N}_j \lfloor \mathcal{N}_i \lfloor X \rfloor \rfloor \doteq \mathcal{N}_k \lfloor X \rfloor$
Fmω|.5 $\supset \forall XYZ\ \&\&\ \mathcal{N}_i \lfloor X \rfloor \doteq Y\, \mathcal{N}_j \lfloor X \rfloor \doteq Z \mathcal{N}_k \lfloor X \rfloor \doteq YZ \mathcal{N}_i \lfloor X \rfloor\, \mathcal{N}_j \lfloor X \rfloor \doteq \mathcal{N}_k \lfloor X \rfloor$

Now, we denote as $C_{\mathfrak{R}}$, the class of signal processing normal algorithms and $C_{\mathfrak{R}}$ in a model of Th($\mathfrak{R}$). Fmω|.1 is a different version of Markov's principle of constructive choice. According to this principle, if the assertion of the inapplicability of a normal algorithm $\mathcal{N}$ to some specific word P is refuted then $\mathcal{N}$ is applicable to P. By *applicability definiteness* of a normal algorithm $\mathcal{N}$ to string P, we mean the effective use of at least one of the substitution formulas of the scheme of $\mathcal{N}$ in rewriting as (P) . The *applicability definiteness* (a-definiteness) of $\mathcal{N}$ to P is generally expressed as $!\mathcal{N}(P)$. The basic supporting argument underlying Fmω|.1 is the notion of abstraction of potential realizability. By virtue of this notion, if the antecedent $\neg\neg!\mathcal{N}\lfloor P \rfloor$ is accepted then the process of applying $\mathcal{N}$ to P by actually carrying out the operation of this algorithm step by step waiting for the conclusion of this operation. Since computation time and storage space are not considered to be the limiting factors in carrying out normal algorithmic signal processing operations, formula Fmω|.1 holds for $C_{\mathfrak{R}}$. Fmω|.2 is interpreted in the following manner.

A normal algorithm $\mathcal{N}$ is said to be a-definite for a word P only when $\mathcal{N}$ is applicable to P and the process of applying it to P terminates either naturally or by a terminal substitution formula and the output Q is a word other than P. An important question that arises here is of immediate concern to us. Does Q indicate the *desired output* in $\mathcal{N}\lfloor P \rfloor \doteq Q$ of the formula Fmω.2? The answer is negative, because the a-definiteness of a normal algorithm for a word does not guarantee the transformed word to be the desired output due to the intended operation for which the very scheme has been constructed. For example, let us consider a normal algorithm $\mathcal{N}$ over an alphabet $\mathcal{A}$, whose scheme is constructed with the purpose of carrying out a specific operation on words from $\mathcal{A}$. Let us assume that this scheme contains the simple substitution formula $\rightarrow \alpha$. Then $\mathcal{N}$ is a-definite for every word in the free monoid $\mathcal{A}^*$ But a-definiteness of $\mathcal{N}$ to the free monoid does not imply that every word of the free monoid is transformed to the relevant output due to the intended operation. In order to overcome this difficulty, we shall introduce here the notion of *successful applicability* of a normal algorithm. A normal algorithm $\mathcal{N}$ is said to be *successful applicability definite* (sa-definite) for a word P only when the result of applying $\mathcal{N}$ to P is the *desired output* that satisfies the purpose for which the scheme of $\mathcal{N}$ has been constructed. sa-definiteness of a normal algorithm implies its a-definiteness. But the converse in not true always.

Courtesy:
Theory and Applications of Symbolic Computing with Artificial Intelligence Perspective
London Journals Publications, United Kingdom, 2023