LOGIC AS A TOOL

Series: STUDIES IN LOGIC, GRAMMAR AND RHETORIC 11(24)

Under the Auspices of the Polish Association
for Logic and Philosophy of Science

LOGIC AS A TOOL

Essays in Discourse and Information Sciences

edited by
Dariusz Surowik

University of Białystok
Białystok 2007

Refereed by Andrzej Trybulec

Series: STUDIES IN LOGIC, GRAMMAR AND RHETORIC 11(24)
http://logika.uwb.edu.pl/studies/

Cover design: Krzysztof Tur
Type-setting: Stanisław Żukowski

ISBN 978–83–7431–158–8
ISSN 0860–150X

WYDAWNICTWO UNIWERSYTETU W BIAŁYMSTOKU
15-097 Białystok, ul. Marii Skłodowskiej-Curie 14, tel. 0857457059
http://wydawnictwo.uwb.edu.pl, e-mail: ac-dw@uwb.edu.pl
Nakład 300 egz.

Druk i oprawa: Sowa – druk na życzenie
www.sowadruk.pl tel. 022 431-81-40

CONTENTS

STUDIES IN LOGIC, GRAMMAR AND RHETORIC 11 (24) 2007

Andrzej Malec
University of Białystok

THE IS-OUGHT PROBLEM AND LEGAL RATIONALITY

The aim of this paper is to discuss the following issue: what is rational behaviour of a lawyer (mainly in the process of the interpretation of legal texts)? We are especially interested if the 'is-ought problem' affects the issue.

Two Meanings of Rationality

Those from countries in the west generally have a firm belief in rationality. They consider being rational as one of the most important of virtues. Thinking and behaving in a rational manner can be seen as an almost ethical value. However, the notion of rationality, as taken from natural language, is rather vague. "Rational" naturally means something like "being in accordance with reason". But what does it mean to be in accordance with reason? Philosophers, logicians and others try to make it clearer[1].

When searching for a clear understanding of rationality we initially discover that the expression "rationality" is an abstract term. It refers to a certain feature: the feature of *being rational*. So we should determine which objects the term "being rational" does refer to. Presently it is said that there are two independent meanings of "being rational": (i) as a feature of thinking ("rational beliefs") and (ii) as a feature of behaviour ("rational behaviours")[2]. All other meanings of "being rational" can be defined in terms of the two above.

[1] A complex investigation of the idea and concepts of rationality can be found in: Ryszard Kleszcz, *O racjonalności. Studium epistemologiczno-metodologiczne* (in Polish; On Rationality. An Epistemological and Methodological Investigation), Wydawnictwo Uniwersytetu Łódzkiego, Łódź 1998.

[2] See: Ryszard Kleszcz, *O racjonalności...*, p. 39

ISBN 978–83–7431–158–8 ISSN 0860-150X

Rational Thinking

When philosophers or logicians are talking about the rationality of thinking, they are declaring that rationality mainly consists in preserving three principles. Firstly: the language we use should be as clear as possible. We should avoid vague notions in science as well as in the humanities. Secondly: logical rules should be preserved in all considerations. The best way to achieve this is to be aware of every step of our argumentation: what the premises are, what rules are applied. In particular we should be aware of the logical status, the logical "force", of the rules we are applying: are they deductive or merely giving us a probability of valid conclusion? Thirdly: being aware of the way science is developing over time, i.e., that old theories are continuously being replaced by new, we must be open to the arguments of others. We should be open to critique. And even more: self-criticism should be our motto.

Such understanding of being rational is quite clear. Undoubtedly anyone (including lawyers) should be rational in the above sense to be considered a rational being.

On the other hand, the meaning described above does not determine any exact way of thinking: what premises should be accepted, what rules of reasoning should be used. Certainly, we should accept facts, tautologies, logical rules, etc. But the above principles will not help us solve most of the real problems connected to a lawyer's practice. Lawyers defending opposite views during a court trial can be equally rational in the above sense. So, maybe it is possible to put forward more principles of rationality? Principles not restricted to thinking.

Rational Behaviour

When philosophers or logicians are talking about the rationality of behaviour it is often connected to praxiology – the theory that deals with the efficiency of activities of any kind. An action is considered to be efficient if it is both effective (an action is or can be successful) and economic (goals are achieved at reasonable costs).

Conveying rationality of behaviour is also possible in terms of theories of choice, game theories, etc. Then, rationality can be defined in terms of expected gains.

Such concepts of rational behaviour are much clearer than the vague notion of rationality taken from natural language. However, there are still many concepts of rationality, and no one among them is clearly a dominating concept.

We must also take into account that all such concepts are usually defined in natural language and so remain uncertain to a certain degree. If we want to be precise, we can restate the above in the following way: the term “rational behaviour” has several meanings (equivocality) and in every (or in most) of these meanings is also not completely clear (unclarity).

Therefore, we do not have a “ready to use” notion of rational behaviour for classifying the deeds of lawyers as rational or non-rational. So, maybe lawyers have such a notion elaborated within the legal sciences?

Lawyers and Rationality

The notion of rationality is very important to lawyers. At least lawyers talk about rationality a lot. They talk about rationality investigating theoretical issues, as well as solving practical problems. We dare to say that the idea of rationality is one of the most important ideas connected to law.

On the other hand, any person who has learnt even a tiny amount of the practical aspects of law knows perfectly well that lawyers earn their money defending just about anything we want them to. And, they win as frequently as they lose. Theoretically, their understanding of law should be determined by legal texts and the rules of legal interpretation. In practice however, what we find is a complete sense of chaos whenever legal texts are interpreted. The same is true in relation to a court of law practice[3]. It is generally known that the same text is often interpreted in completely different ways by different courts. And every judge is convinced about her/his rationality. But could they all be rational giving various interpretations of the same text? Probably not. Nevertheless they all claim they are right, they are rational. This is possible because there is no commonly accepted set of criteria of rationality of legal interpretation. This means there is no commonly accepted concept of legal rationality. Is the above fact accidental? Or, is there a reason for this fact?

[3] Anyway, it is true regarding the Polish system of courts of law.

The Lawmaker's Rationality

One of the most important issues regarding legal studies is the issue of choosing the best way for inferring legal norms from legal texts[4]. This issue is investigated within the theory of legal interpretation[5]. In their investigations, Polish theoreticians use concepts of rationality of the lawmaker.

The lawmaker is a fictitious person who is the supposed author of all legal texts. It is supposed that the lawmaker is rational in several aspects. Firstly, he is perfectly aware of the language he uses: he knows the precise meaning of every word, he understands all the grammar, etc. Having perfect knowledge of the language he is able to communicate all his ideas clearly and completely in accordance with his will. Such rationality is called "semiotic rationality".

The lawmaker is also aware of what justice is. He knows the desired goals that should be achieved by humankind, i.e., he knows the value of every event or situation and is able to compare such values. Such rationality is called "axiological rationality".

Among other "rationalities" there are suppositions that the lawmaker has a perfect understanding of factual situations and a perfect comprehension regarding the rules governing reality – including social reality. Finally, it is necessary to foresee all possible results of establishing any new regulation.

Two Models of Legal Interpretation

In legal theory, two key models of interpretation are considered. The first model is based on the idea that rules prescribed by the lawmaker, and somehow "concealed" by him in legal texts, should be derived from legal texts merely by semiotic procedures or at least – by procedures that prefer semiotic measures. This model is justified by assuming the semiotic rationality of the lawmaker.

The second model is based on the idea that rules derived from legal texts should be just. So, if a legal text implies something unjust, then we are supposed to abandon the direct meaning of the text and interpret the

---

[4] Legal norms are rules of behaviour prescribed by authorities. Legal texts are sets of inscriptions from which legal norms can be inferred. A text to be a legal text must be accepted in a due course by a legitimated do to so authority of a state.

[5] The theory of legal interpretation is the theory examining ways in which legal texts are understood by lawyers and formulating principles in accordance with which legal texts should be understood by lawyers.

text in such a way that the rules derived from the text are just. This model is founded on the supposition of the axiological rationality of the lawmaker.

So lawyers have two opposing models of interpretation regarding legal texts. Therefore, they are able to defend anything. If we want them to defend something that is in accordance with "the letter of the law", they will use the model of interpretation based on the semiotic rationality of the lawmaker. However, if we want them to defend something that is in conflict with "the letter of the law" they can then use the model of interpretation based on the axiological rationality of the lawmaker. In both cases they can be seen as somehow being rational. But are they equally rational by using opposing models?

The Is-Ought Problem

It is Hume's statement that moral distinctions cannot be derived from reason:

> *Reason is the discovery of truth or falsehood. Truth or falsehood consists in an agreement or disagreement either to the real relations of ideas, or to real existence and matter of fact. Whatever, therefore, is not susceptible of this agreement or disagreement, is incapable of being true or false, and can never be an object of our reason*[6].
>
> *Laudable or blameable, therefore, are not the same with reasonable or unreasonable. The merit and demerit of actions frequently contradict, and sometimes controul our natural propensities. But reason has no such influence. Moral distinctions, therefore, are not the offspring of reason*[7].

Presently, the above thesis of Hume can be restated in a more general way:

> *Deontic statements are logically separated from non-deontic statements, i.e., neither can deontic statements be derived from non-deontic statements (simple Hume's thesis) nor can non-deontic statements be derived from deontic statements (reverse Hume's thesis)*[8].

[6] David Hume, *A Treatise of Human Nature*, Penguin Classics 1985, p. 510

[7] *ibidem*

[8] See: Jan Woleński, *Uogólniona teza Hume'a* (in Polish; A Generalized Hume's Thesis), in: I. Bogucka, Z. Tobor (editors) "Prawo a wartości. Księga jubileuszowa Profesora Józefa Nowackiego" Zakamycze Kraków 2003, p. 293–303. Deontic statements are statements that describe norms, e.g., "John ought to open the window".

Pursuant to the above thesis, it is impossible to infer obligations from facts or to infer facts from obligations. Intuitively, the thesis holds. Some formal argumentation is also possible[9]. The true meaning of the thesis is that the so called "positive sciences" cannot help us with moral dilemmas.

What is important here is that a similar thesis can be put forward in relation to axiological modalities: it is impossible to infer values from facts or to infer facts from values. But if so, then how can we put forward binding arguments for any model of legal interpretation?

Facts and Values

Several arguments based on intuition can be put forward to support the thesis that axiological statements and non-axiological statements are logically separated.

The first argument is based on the idea of ***intractability of social phenomena***[10]. Since our ability to compute future states of society is limited, we are not aware of all the effects of our behaviour. Therefore, we are not able to value our behaviour. So, it is impossible to say if the final result of any action will be good or bad. This is reflected in a Polish proverb: "Nie ma tego złego, co by na dobre nie wyszło" (If an event seems to be bad, don't worry. It may turn out that the event is good).

The second argument is based on the idea of ***expected time horizon of events***[11]. Let us suppose that we are able to compute future states of society. So, we are able to compute all the consequences of all possible (alternatively attainable) ways of behaving. The problem is as follows: how far should computations go and be acted upon? Is it enough to compute the future effects of present deeds for one year? Or, should we compute our future to take into account the lifetime of our generation? Or perhaps several generations? Let us suppose that we can act in a way such as A or (alternatively) in a way such as B. It may turn out that the consequences of A are better than those of B if we are say looking at a time scale of 5 years, but are worse if we are looking at a time span of 10 years, but are

[9] *ibidem*

[10] See: Witold Marciszewski, *Undecidability and Intractability in Social Science*, in: W. Marciszewski (editor) "Issues of Decidability and Tractability" University of Bialystok 2006, p. 143–174.

[11] The idea of expected time horizon of events was used in literature by Stanislaw Lem in his famous novel *Powtórka* (in Polish; Repetition), Iskry 1979, p. 55–57. In the novel two scientists discuss how to re-create the world to make it perfect.

again better if we are looking at a time span of 20 years and so on. When should we stop our computation in order to have a ***real*** valuation of the effects of alternatively attainable ways of behaving?

The third argument is based on ***the Ant and the Grasshopper Paradox***[12]. The ant is working very hard all summer to survive during winter. The grasshopper in contrast is enjoying life as much as possible all summer long, aware that he will die during winter. In consequence, the ant survived the winter and the grasshopper did not. However, the grasshopper was happy during the summer, the ant was not happy either during the summer (since it worked very hard), nor during the winter (since the winter is a bad season to enjoy life). Who was right? Who was wrong? Are we able to answer the questions in a rational manner? Probably not.

Formal Rationality – Material Rationality

If we accept the thesis that axiological statements and non-axiological statements are logically separated, then we have to admit that it is impossible to establish a universal model of legal interpretation.

Of course, we can state some formal features of a good interpretation (like clarity of language, being in accordance with logical rules, etc.). If a legal interpretation has such features it is rational from the formal point of view.

On the other hand, we cannot put forward binding arguments concerning the choice between semiotic and axiological rationalities of interpretation. Therefore, if we try to define the material rules of legal interpretation, they have to have a hypothetical form: "if you choose semiotic rationality you should do A, but if you choose axiological rationality you should do B". And there is no universal argumentation to choose between A and B.

Therefore the material rationality of lawyers is merely hypothetical.

Andrzej Malec
Department of Logic, Informatics and Philosophy of Sciences
University of Białystok
e-mail: malets@uwb.edu.pl

[12] The paradox was put forward by Martin Hollis in his *The Cunning of Reason*, Cambridge University Press 1987, p. 95–96.

Kazimierz Trzęsicki
University of Białystok

POLISH LOGICIANS' CONTRIBUTION TO THE WORLD'S INFORMATICS*

The position of Polish informatics, both in research and teaching, in the world of informatics, has its roots in the achievements of the Polish mathematicians of the Warsaw School and the logicians of the Lvov-Warsaw School.
Jan Łukasiewicz is the most famous Polish logician in the world of computer science. Invented by him, the parenthesis-free notation is known as PN (Polish Notation) and RPN (Reverse Polish Notation). Łukasiewicz created multi-valued logic as a separate subject. The idea of multi-valuedness is applied to hardware design (many-valued or fuzzy switching, analogue computer). A many-valued approach to vague notions and commonsense reasoning is the method of expert systems, databases and knowledge-based systems, as well as data and knowledge mining.
Stanisław Jaśkowski's system of natural deduction is the basis of systems regarding automatic deduction and theorem proving. He created a system of paraconsistent logic. Such logics are used in AI.
Kazimierz Ajdukiewicz, with his categorial grammar, participated in the development of formal grammars, the field significant for programming languages.
Andrzej Grzegorczyk has made an important contribution to the development of the theory of recursiveness.
Alfred Tarski, and the significance of his work for informatics, is not under consideration in the paper. His achievements are the subject of S. Feferman's article "Tarski's Influence on Computer Science".

Keywords: parenthesis-free notation, many-valued logic, paraconsistent logic, categorial grammar, theory of recursiveness, Polish notation, fuzzy switching, analogue computer, AI

When we hear of the successes achieved by Polish students of informatics at the Students' World Championships in Programming, or the winners of competitions for young scientists in the European Union, or the fact that Warsaw University occupies a top position in the world rankings of informatics studies; we have to ask ourselves, why this is so and in which place we should look for the causes of this success. Undoubtedly, their achievements, and this includes those of students from other Polish universities as well,

* I would like to thank the anonymous reviewer for the comments which made this article better. The work was supported by KBN (The Commission for Scientific Research) Grant 3 T11F 01130.

ISBN 978–83–7431–158–8 ISSN 0860-150X

are their very own successes – the result of their talents, hard work and ambition. All this however, would matter little without good teachers, who being scientists themselves, contribute significantly to the development of informatics. We can be so bold as to say that informatics stemming from Warsaw University leads the world. While commenting on the fact that Warsaw University was listed in 2003 as one of the top universities (this according to a number of the most often quoted publications), professor Damian Niwiński (2003) points out that informatics has been developing systematically at Warsaw University since 1960. He sees the cause of this state of affairs in the attitudes of great Polish mathematicians,[1] the successors of the Polish school of mathematics: Kazimierz Kuratowski, Stanisław Mazur, Wacław Sierpiński, Hugo Steinhaus,[2] and Helena Rasiowa.[3] At the same time he emphasizes the role of logicians. It was mainly in this field that the critical mass of today's successes was created. It could have been a matter of chance in a sense, but when in 1948 the first Polish institution dealing with computers came into existence (the Mathematical Apparatus Group), professor Kuratowski appointed the logician and statistician Henryk Greniewski[4] (1930–1972) as its first director. It was Greniewski who initiated the establishment of the Polish Cybernetics Society[5] in 1962. The 23rd of December, 1948 however can be considered the launch date regarding the history of Polish informatics. Romuald W. Marczyński remembers that six people met on that very day in the mathematics seminar room in the Institute of Physics. Those present were Professor Kazimierz Kuratowski, Professor Andrzej Mostowski (logician), Doctor Henryk Greniewski and the three engineers Krystyn Bochenek, Leon Łukaszewicz and Marczyński himself. During the meeting they discussed the possibilities of

[1] There are two other important names worth mentioning in this context: professor Oskar Lange – economist, professor Janusz Groszkowski – director of the State Institute of Telecommunications, later deputy chairman of the State Council of the Polish People's Republic.

[2] At first he was deputy director of the Mathematical Apparatus Group dealing with applications. Later this post was held by Professor Stanisław Turski (1906–1986).

[3] Professor Rasiowa was very dedicated to publishing *Fundamenta Informaticae*. This journal started to appear in 1977 mainly thanks to her efforts. She was its Editor-in-Chief until her death. She never ceased to deal with it, even when she was ill. Let us add that she was an active member of the editorial board of *Studia Logica* (from 1974) and *Journal of Approximate Reasoning* (from 1986).

[4] He was expelled from the Planning Commission for political reasons, as a result of the growing class struggle.

[5] The name was ideologically conditioned. This is how informatics was referred to in the Soviet Union as well.

constructing mathematical apparatuses. It should be added that the first GAM-1 machine was built in 1950 by Zdzisław Pawlak, but was not used for calculations.

One of the achievements on a world scale was the language *KLIPA*, created in the 1960s by a team headed by Professor Władysław M. Turski: Marek Greniewski, Jadwiga Empacher, Jadwiga Zdanowska and Ryszard Solich. *KLIPA* was the external language for the *URAL* digital computer (Greniewski, Turski 1963). In the 1970s Andrzej Salwicki created the object-oriented programming language *LogLan.* A few years before dynamic logic was appreciated in the West, the team headed by Salwicki – Grażyna Mirkowska, Antoni Kreczmar and others – created algorithmic logic as a tool for examining and describing problems connected with the verification of programmes. Following Niwiński, one should also mention the works of Jerzy Tiuryn and his successors concerning the place of logic in informatics (type theory, the lambda calculation, functional programming, programming logic, the computing power of programming languages, complexity issues in logic and finite model theory). Tiuryn is presently in charge of a bio informatics team. Professor Jan Madey, head of the Section of Operating Systems in the Institute of Informatics, Warsaw University, director of the Centre for Open Multimedia Education (COME) at Warsaw University and the author of the first Polish handbooks of the languages *Algol 60* and *Pascal*, conducted innovative classes for students at various levels. He is the author of the system OS Kit designed to examine operating systems, problems of parallel computation and the efficiency of information systems. This world class specialist in the field of software engineering, who is probably best known for the methodology he developed in co-operation with David Lorge Parnas known in the literature as "Parnas-Madey Four Variable Model", finds a great deal of satisfaction in the achievements of his students. He rates their academic successes as his greatest success. His supervisees won the Academic World Championships in Team Programming and European Union Contests for Young Scientists (Szumiec-Presch 2004). He himself studied with professors Andrzej Kiełbasiński (doctor at the time), Karol Borsuk, Kazimierz Kuratowski, Stanisław Mazur and Andrzej Mostowski. He points out how important it was for him to enjoy the confidence of Professor Andrzej Turski, the rector of Warsaw University, and at the same time one of the people who made the greatest contributions to Polish informatics: "He threw me into the deep end of the pool", he said, "but was always there keeping an eye on me and supporting me from a distance". From 1964–70, Jan Madey was deputy head of the Section of Numerical Calculations at Warsaw University under Professor Stanisław Turski. Professor Stanisław Turski was a very important figure in the history of informatics at Warsaw University, who as its rector established the first computing centre (the Section of Numerical Calculations), and in 1975 also launched the Institute of Informatics within the department bearing a new name: *the Department of Mathematics, Informatics and Mechanics*. These institutional changes were connected with the launch of full informatics MA programmes (in place of studies offered within the section of numerical methods). Other areas of research in the Institute comprised automata theory (Stanisław Waligórski and others) and applied linguistics (Leonard Bolc, Janusz Bień and others), especially in connection with issues concerning artificial intelligence and programming in logic.

It is undoubtedly difficult to answer the question as to how important Polish logic was for achieving the aforementioned results. Many apparently irrelevant ideas may have significance in creating the right climate. Many enterprises, regardless of the intentions of their architects, may have an unexpected importance in other areas. As far as logic is concerned, however,

> ... the best known Pole in informatics is the logician Jan Łukasiewicz (1876–1956), who in 1917[6] introduced a way of expressing arithmetic expressions which avoids the use of parentheses, known as the Polish Notation. This notation is now commonly applied to automatic calculations of the value of expressions, used, among other things, in various calculators. (Madey, Sysło 2000)

This is the reason why solving the problem regarding the economisation of the notation, not in the least inspired by informatics issues, gave Łukasiewicz a lasting position in informatics.

Logic, alongside algorithmics, is a component of theoretical informatics.[7] In this sense all output by Polish logicians would be significant for informatics, and would render my initial question regarding their contribution to the world's informatics irrelevant. Therefore I will point out only those ideas which seem to be more directly connected with informatics, as was the case of the aforementioned Polish notation. Accordingly, I will not comment on the authors' intentions, nor will I ponder over the fact as to

[6] For more information (on this subject) see the next page.

[7] Electronics should also be added to contemporary informatics as a whole. Electronic solutions have turned out to be more effective than mechanical ones. Perhaps in the future electronics will be replaced by some biotechnological solutions.

whether those ideas were taken from their authors directly or indirectly or whether, as it was in the case of multi-valued logic, the same idea occurred independently to different scholars at the same time. Their importance for informatics will be discussed in as much as it is deemed necessary for their comprehension.

Polish Notation (Parenthesis-Free Notation)

The idea of the notation which avoids the use of parentheses appeared in connection with examining formal systems. Polish logicians, alongside other current issues, found the independence of the set of primitive terms and axioms equally important. Subsequently the problem of 'economisation' arose; in particular, a system with the smallest possible number of primitive terms and one shortest axiom was sought out.

From the point of view of semiotics (and informatics) – due to an economy of expression – it was interesting to find out whether a language without punctuation marks in general, and parentheses in particular, was possible. This kind of notation was invented by Jan Łukasiewicz. Łukasiewicz (1931, p. 165), who states that he laid down the principles of parenthesis-free symbolism in 1924, used it for the first time in his article *O znaczeniu i potrzebach logiki matematycznej* (On the importance and needs of mathematical logic) published in 1929, and although it was Chwistek who at the beginning of the 1920s talked about placing conjunctions before arguments, as Woleński (1985, p. 93) writes, there is more to parenthesis-free symbolism than writing conjunctions in front of arguments, hence there is no conflict in attributing the creation of parenthesis-free symbolism to Łukasiewicz and the idea of placing conjunctions in front of arguments to Chwistek.

It turned out that whenever all the conjunctions were *prefixes* (i.e. when they were written before their arguments) or when all of them were *suffixes* (i.e. written after their arguments), it was possible to eliminate the parentheses. Łukasiewicz's notation, apart from the economisation of means of expression, has an additional advantage in that the structure of an expression is defined by the position of symbols of which it is built. This very feature displays an advantage from the viewpoint of informatics (and not only informatics).

The importance of Łukasiewicz's notation for informatics was noted by Turing, who met Łukasiewicz in 1949. According to Turing, it is more advantageous for mechanical devices to have function symbols at the beginning of formulas. In informatics it is a suffixal notation that is particularly important. It was Hamblin who found a way of applying it. According to Pearcey (1994), Hamblin, who had gained some experience from radar services during World War II, was employed to run a third university computer in Australia in 1956. He became aware of the problems connected with (a) computing mathematical formulas containing parentheses and (b) loading memory with proper names of memory resources. As a formal logician he knew Łukasiewicz's work.[8] The solution to the first problem was supplied by Łukasiewicz's notation. Instead of writing, for example: $(a+b)\times c$, one can write: $\times, +abc$. The other problem, enabling the machine to access resources which do not require an address (a current operation would be always carried out on the results of the operations immediately preceding it, left and always remaining in the resources), was solved by applying Łukasiewicz's reverse notation (**R**everse **P**olish **N**otation – RPN). Instead of writing: $\times + abc$, one writes: $ab + c\times$). This is how the idea of organising resources into a stack was born – *last-in, first-out* (LIFO). Humblin presented his results at the First Australian Conference on Computing and Data Processing (1957). Representatives of the English Electric Company who were present at the conference carried his ideas to England and the company used Hamblin's architecture (and even his terminology) (Lavington 1980). Hamblin presented his conception in (1962) as well. One of the designers of the American computer B5000 (announced in 1961 and produced in 1963), in which RPN was used, R. S. Burton, wrote (1970) that the idea had occurred to him independently of Hamblin, when he was reading a handbook of symbolic logic. 10 years after Hamblin's first publication, the RPN idea was used by engineers from Hewlett-Packard in a calculator which appeared on the market in 1968 and then in HP-35 from 1972. In this way RPN became popular in scientific and engineering circles.[9] It is worth adding at this point that Hamblin was the precursor of many conceptions, among others, the application of temporal logic in informatics (Allen 1984, Allen 1985, Hamblin 1987, Williams 1985).

[8] Łukasiewicz's notation was used by A. N. Prior, a logician from New Zealand, among others in a handbook of logic (1955), which 'hindered' readability (see a comment on this in Woleński 1985, pp. 94–95).

[9] Edsger W. Dijkstra, converting infix notation into RPN, invented an algorithm which due to its similarity to the way a railroad shunting yard operates was called the 'shunting yard'.

Multi-Valued Logic

Jan Łukasiewicz (1878–1956) is best known for his concept of multi-valued logic.[10] Łukasiewicz was convinced not only that it was a discovery comparable to non-euclidian geometry, when he wrote (1930, p. 161):

> It is not easy to predict the impact of non-chrysippian[11] systems of logic upon philosophical speculation. It seems however that the philosophical importance of the systems presented here may be at least as great as the importance of non-euclidian systems of geometry.

Łukasiewicz designed his systems as the basis for mathematical research in arithmetic and multiplicity theory.[12] As far as practical application is concerned, having cybernetics in mind, he wrote in a letter to Lejewski in 1951 (Woleński 2005, p. 261):

> Multi-valued systems already today have important practical applications and may become a source of significant income.

One may agree with Woleński when he writes that:

> At present it is beyond any doubt that Łukasiewicz's expectations have not been fulfilled. Multi-valued logics have not revolutionised either logic or mathematics, or philosophy. (Woleński 1985, pp. 122–123)

It should be added however that the thesis concerning the practical benefits from multi-valued logics, and I do not mean those used in metatheoretical research but those in widely understood informatics, seems to have a chance to be confirmed. It is worth noting however that both multi-valued logics and the concept of the notation were not created for the sake of informatics. In the case of multi-valued logic it was philosophical motives. There is a monograph devoted to the use of multi-valued logics in informatics (Rine 1977). Interest in using multi-valued logics in informatics is reflected in conferences devoted to this issue. In 2006 the 36$^{\text{th}}$ annual symposium organised by *The Multi-Valued Logic Technical Committee of the IEEE Computer Society* will be held in Singapur.

---

[10] Independently of Łukasiewicz (1920*c*, 1920*b*, 1920*a*) multi-valued logics were created by E. Post (1921), born in Augustów, Poland.

[11] For more on the history of creating Łukasiewicz's multi-valued logics, see (Woleński 1985, pp. 115–122).

[12] This is how Łukasiewicz referred to multi-valued logics.

The Multi-Valued Logic – An International Journal (www.csi.uottawa.ca/~ivan/mvl.html), among issues in its sphere of interest, mentions the following:

> MVL[13] and Soft Computing: neural networks, evolutionary computation, fuzzy systems, computational intelligence cost-effectiveness;
>
> Engineering aspects of MVL: circuit design, programmable logic, hardware and software verification, testing, analog and digital VLSI and ULSI, new concept devices and architectures, carrier computing (biocomputing, optical computing, ...);
>
> MVL and Automated Reasoning: machine learning, reasoning, theorem proving, expert systems;
>
> Computer Science and MVL: databases, massively parallel systems, collision-based computing;
>
> Fuzzy Logic and MVL: theoretical and practical aspects;
>
> Philosophical aspects of MVL

Ewa Orłowska from the Institute of Telecommunications is a member of the journal's editorial board.

One can distinguish between the applications of multi-valued logic in designing informatics equipment and in the methods of artificial intelligence.

Engineering Applications

To put it simply, just as multi-valued logics are a generalisation of two-valued logics, so are electrical circuits where m states a generalisation of circuits with two states. This issue has been addressed for a long time. Henryk Greniewski, who, as already mentioned, was the first director of the Mathematical Apparatus Group, was interested in the technical application of multi-valued logics. Let me add in passing that his book *Elementy cybernetyki systemem niematematycznym wyłożone* (*Cybernetics without Mathematics*) (1959) was translated into German, English and French and is still available from Pergamon Press (1960). In the German Democratic Republic Greniewski was an authority, among other things, in the field of applying cybernetics (informatics) in planning economic development (Segal 1999). His views on this issue have been quoted to this day (Greniewski 1962). There is at least one major publication in Polish on the subject of the application

[13] Multi-Valued Logic

of Łukasiewicz's logics. It is a two-volume work by Moisil (1966, 1967).[14] Epstein (1993) is a good introduction to the problem of multi-valued (fuzzy) switches.[15]

Currently the binary standard is obligatory in informatics. It was Leibniz who already in his day opted for this solution, but – and one has to remember that it was the era of mechanics – based his 'computer' on the decimal system. The architecture for contemporary computers was postulated by Von Neumann in the report 'First Draft of a Report on the EDVAC' (1981) written in 1945 under the auspices of the University of Pennsylvania and the United States Army Ordinance Department. In his justification for the choice of the binary system, similarly to Leibniz, he points to the simplicity of the system. The report states (Von Newman 1981):

> 5.1 ... Since these tube arrangements are to handle numbers by means of their digits, it is natural to use a system of arithmetic in which the digits are also two valued. This suggests the use of the binary system.
>
> 5.2. A consistent use of the binary system is also likely to simplify the operations of multiplication and division considerably. Specifically it does away with the decimal multiplication table. ... In other words: Binary arithmetic has a simpler and more one-piece logical structure than any other, particularly than the decimal one.

The report emphasises that:

> An important part of the machine is not arithmetical, but logical in nature. Now logics, being a yes-no system, is fundamentally binary. Therefore, a binary arrangement of the arithmetical organs contributes very significantly towards a more homogenous machine, which can be better integrated and is more efficient.

If designing computers operating with the decimal system can be explained in terms of it being natural, for other systems it is theoretical and practical arguments that are important. Such arguments can be found not only for the binary system but also for the ternary system.

The first designer of the machine operating in the ternary system was Thomas Fowler. In May 1840 he demonstrated his wooden calculating machine. It was described by De Morgan (1840, (1837–1843)).[16]

---

[14] His earlier work (1959) was also published in English by Pergamon Press. See also (1972).

[15] Cf. (Gottwald Winter 2004).

[16] The bibliography concerning this subject can be found at: http://www.mortati.com/glusker/fowler/refslinks.htm.

In the Soviet Union 50 ternary computers were built – *Setun* and *Setun 70*. The designer of these computers, Brousentsov, writes as a co-author (Brousentsov, Maslov, Ramil, Zhogolev 2005):

> It is known that the ternary arithmetic has essential advantages as compared with the binary one that is used in present-day computers. In connection with this Donald Knuth assumed that the replacement of "flip-flop" for "flip-flap-flop" one a "good" day will nevertheless happen [1].[17] Now, when the binary computers predominate, it is hard to believe in a reality of such assumption, but if it would happen not only the computer arithmetic, but the informatics on the whole would become most simple and most perfect. The third value (Aristotle named it σνμβεβηκος – attendant) what is very actual but hidden in binary logic, will become obvious and direct manipulated. Ternary logic has better accordance with the Nature and human informal thinking [2]. Unfortunately the modern researches of the multi-valued (non-binary) logic are formal and not associated with practical requests.
>
> A remarkable exclusion is the experience of creating the ternary computers "Setun" and "Setun 70" at Moscow State University [...]. This experience convincingly confirms practical preferences of ternary digital technique.
>
> 1. Knuth D. E. The art of computer programming. Vol. 2. Seminumerical algorithms. – Addison-Wesley, 1969.
> 2. Brousentsov N. P. Origins of informatics. – Moscow, The New Millenium Foundation, 1994. (In Russian).

Brousentsov points out in an interview (Rumyantsev 2004) the technical sources of the idea. He noticed however the importance of three-valued logic, stating that these issues were not well thought out in his computers.

The Analogue Computer[18]

Nowadays digital computers predominate. It seems that the concept of the analogue computer has finally been abandoned. However, it would not be for the first time in informatics that predictions have turned out to be wrong. Suffice to recall the prognosis concerning the number of computers necessary for the United States. Professor Jonathan W. Mills from Indiana University Bloomington[19] believes in the success of such machi-

[17] Let me add that for Knuth this must be a balanced system, which the system based on $\{-1, 0, 1\}$ supposedly is. More on the advantages of such a system in (Hayes 2001).

[18] It was Professor Witold Marciszewski who pointed out this application of Łukasiewicz's logic to me.

[19] For his conception of the analogue computer see (Mills 1993) and (Mills, Walker, Himebaugh 2003).

nes. He is not alone in his beliefs. In 1995 Lee Rubel[20] wrote to him (Mills 2006):

> The future of analog computing is unlimited. As a visionary, I see it eventually displacing digital computing, especially, in the beginning, in partial differential equations and as a model in neurobiology. It will take some decades for this to be done. In the meantime, it is a very rich and challenging field of investigation, although (or maybe because) it is not in the current fashion.

Mills became interested in the idea of the analogue computer back in 1990 in connection with his studies of Łukasiewicz's multi-valued logic. He collaborated at the time with J. Michael Dunn, professor of philosophy, and Oscar R. Ewing, professor of informatics. Together with Ch. Daffinger and M. G. Beavers he started designing electric circuits based on infinitely valued Łukasiewicz's logic. Mills considered this logic suitable for describing analogue circuits. The construction of the machine was inspired by Kirchhoff's research on electricity. Mills writes the following about his Kirchhoff-Łukasiewicz machine:

> I'm thinking that within five to ten years, we will find a niche in which these processors are superior, efficient, and cost-effective. (Hedger 2006)

He predicted that:

> We may develop sensors that would detect chemicals in the environment or toxins within our bodies, such as life-threatening cholesterol levels. We might develop an implant that could predict heart attacks – sort of a biological beeper. (Hedger 2006)

Applications in AI

Applications in AI seem to be the most promising of all the possible applications of multi-valued logics.

Multi-valued logics form the basis for the description of vague concepts, which are characteristic of natural language and non-formal reasoning. This has an importance, among others, for expert systems.

The most famous conception is the theory of fuzzy sets developed in the 1960s by Lofti A. Zadeh (1965). He applied Łukasiewicz's logic to elements

[20] The author, among others, of the article *The Extended Analog Computer* (1993), which prompted the invention of Kirchhoff-Łukasiewicz Machine.

of a set, thereby creating an algebra of fuzzy sets. They were not put to use until the mid 1970s, when Ebrahim H. Mamdani of Queen Mary College in London designed a 'fuzzy' controller for a steam engine.

A similar solution in connection with the research on expert systems was worked out in Poland by Z. Pawlak. Rough set theory was developed in many publications, for example (Pawlak 1982, Pawlak 1991, Pawlak 1993) and (Komorowski, Pawlak, Polkowski, Skowron 1999).

The theories of fuzzy and rough sets are applied in artificial intelligence and expert systems. They are used for the automation of data and knowledge exploration. In connection with the applications of multi-valued logic in informatics the notion of fuzzy logic is used (see, for example, http://plato.stanford.edu/entries/logic-menyvalued/).

Natural Deduction

In contemporary informatics, natural logic is applied first of all in broadly understood issues relating to artificial intelligence. It is the starting point of the basic systems of theorem proving and/or proof verification. It was created independently by Genzten (1934) and Jaśkowski (1906–1965). Jaśkowski testifies that in 1926 Łukasiewicz formulated the problem of a logical system which would be in line with the practice of proving mathematical theorems. Gentzen approached the issue in a similar way (1934, p. 176):

> My initial point of view was as follows: The formalisation of logical reasoning, especially in the way it was developed by Frege, Russell and Hilbert, differs significantly from the way of reasoning practised in mathematical proofs. One gets significant formal advantages in return. I would like therefore to present a formalism, which is as close to real-life reasoning as it is possible.[21]

Jaśkowski published the solution to the problem in 1934, creating an assumption-based system. The first announcement concerning this issue appeared as early as 1929, in *The Commemorative Book of the Polish Mathematical Symposium*, 1927. In this announcement, Jaśkowski wrote about his results, which had been presented at Łukasiewicz's seminar in 1926.

[21] Mein erster Gesichtspunkt war folgender: Die Formalisierung des logischen Schließens, wie sie insbesondere durch Frege, Russell und Hilbert entwickelt worden ist, entfernt sich ziemlich weit von der Art des Schließens, wie sie in Wirklichkeit bei mathematischen Beweisen geübt wird. Dafür worden beträchtliche formale Vorteile erzielt. Ich wollte nun zunächst einmal einen Formalismus aufstellen, der dem wirklichen Schließen möglichst nahe kommt.

Andrzej Trybulec, the author of the MIZAR system used for verifying mathematical proofs, directly refers to Jaśkowski's system (and the non-Fregian logic of Roman Suszko). Witold Marciszewski (1994, 2005) holds the view that Jaśkowski's system is more useful in computer-assisted proof verification, while Gentzen's system is better in computer-assisted proving. As far as the problems of the mechanisation of reasoning are concerned, it is worth noting the work of (Marciszewski, Murawski 1995).

Temporal Logic

Undoubtedly the founder of temporal logic is Arthur Norman Prior. One should note however the influence of the Lvov-Warsaw School and especially that of Łukasiewicz upon the development of Prior as a logician and upon his early temporal considerations.[22] Among the works that are mentioned as important for its founding, one should mention the work of Jerzy Łoś (1948). Prior (1996, p. 46) was astonished by the usefulness of temporal logic, learning from, among others, Dov Gabbay and Dana Scott that:

> There are practical gains to be had from this study too, for example in the representation of time-delay in computer circuits.

Presently, temporal logic is a recognised and important subject from the point of view of informatics.

Paraconsistent Logics

Jaśkowski formulated and designed discursive logic. His work "Rachunek zdań dla systemów dedukcyjnych sprzecznych" (Propositional calculus for contradictory deductive systems) (1948, 1969) was written as a response to a political need. Marxists rejected the principle of inconsistency. Therefore there appeared a need in logicians' circles for a work which would show the rationality of such a stance. The concerns of logic for paraconsistent systems, however, are more deeply rooted. Contradiction on classical logic grounds leads to trivialisation and its rejection is one of the oldest postulates of logic, clearly formulated, for example, by Aristotle. We know, however, that in 'real-world' cognitive activity, despite hidden or revealed

[22] I write more on this subject in (2005).

inconsistencies, the complete rejection of the system does not have to take place. The importance of paraconsistent systems reveals itself in informatics in connection with the fact that data a computer programme has to deal with may be, in a sense, contradictory. Man can somehow deal with the inconsistency of his views. Therefore artificial intelligence should also have this ability. A programme may draw data from various resources and hence collect contradictory data. Since man can cope with this, a computer should too. Man, in his activity, can give up reason, while a computer has to run according to a formal programme. If a programme is to be able to deal with an inconsistency, it has to be based on systems of paraconsistent logic. It was Stanisław Jaśkowski who created an important system of this kind, which is commonly recognised and well known.

Categorial Grammar

The idea of categorial grammar was formulated by Kazimierz Ajdukiewicz (1890–1963) in his work (1935). Undoubtedly informatics used other grammars for its purposes. What is important, however, is the very fact that this is a formal grammar. It can be used in various applications of informatics, especially in linguistics, but not only (Park 2001). The theory of categorial grammars is developed in connection with the Lambek Calculus. The work is also carried out in Poland. Among major world-ranking publications one should mention (Buszkowski, Marciszewski, van Benthem 1988).

The Theory of Recursive Functions

Andrzej Grzegorczyk[23] joined in on the mathematical milieu at the time when "the political situation was conducive to staying in the safe circle of logical and mathematical speculations". In 1950 he received his doctoral degree and was promoted by Andrzej Mostowski. Three years later, on the basis of his work *Some Classes of Recursive Functions* (1953), he was promoted to the position of 'docent'. This very work is an important historical contribution to world informatics. It is his most frequently cited work in the area of broadly understood issues of theoretical informatics. He showed his interest in the problems of decidability in his works *Zagadnienia rozstrzy-*

[23] My text devoted to A. Grzegorczyk is almost entirely based on the information taken from Stanisław Krajewski's work *Andrzej Grzegorczyk*, (2005).

galności (issues of decidability) (Grzegorczyk 1957, Grzegorczyk 1961). The problems regarding the concept of decidability, the concept of computability and the concept of recursive function, which arose in connection to Hilbert's programme and eventually resulted in the creation of the theoretical foundations of informatics, were taken up by Gödl, Church, Turing and Kleene. The work of Grzegorczyk contributed in a significant way to a better understanding of them. As Krajewski writes (2005, p. 109): "Throughout the whole period of his scientific academic activity he was faithful to the problems of decidability and computable functions." Let us add that for him it was (Krajewski 2005, p. 109) "connected with an in-depth study of concrete, empirical, 'tangible' aspects of the world, which are approached mathematically."

The contribution of Polish logicians to the problems of decidability and computability is much greater if one considers the achievements of Alfred Tarski, one of the most remarkable logicians. The influence of Alfred Tarski and the Polish logicians collaborating with him (A. Mostowski, L. Szczerba and others) goes beyond the scope of the problems of decidability and computability. This issue calls for separate treatment. Finally, let us add that a Polish presence in the world as far as these issues are concerned has found its expression in publications like that by Roman Murawski (1999).

Bibliography

(1929) 'Księga pamiątkowa pierwszego polskiego zjazdu matematycznego, 1927', Kraków.

Ajdukiewicz K. (1935), 'Die syntaktische Konnexität', *Studia Philosophica, Leopoli* 1,

Ajdukiewicz K. (1960), *Język i poznanie. Wybór pism z lat 1920–1939*, t. 1, PWN, Warszawa.

Allen J. F. (1984), 'Towards a general theory of action and time', *Artificial Intelligence* 23 (2), 123–154.

Allen J. F. (1985), 'Charles Hamblin (1922–1985)', *The Australian Computer Journal* 17 (4), 194–195.

Barton R. S. (1970), Ideas for computer systems organization: a personal survey, *w* J. S. Jou, red., 'Proceedings of the Third Symposium on Computer and Information Sciences held in Miami Beach, Florida, December 1969', t. 1 *Software Engineering*, Academic Press, New York, 7–16.

Brousentsov, N. P., Maslov, S. P., Ramil, A. J., Zhogolev, E. (2005), 'Development of ternary computers at Moscow State University', Internet.

Buszkowski W., Marciszewski W., van Benthem J. (1988), 'Categorial Grammar', John Benjamins Publishing Company.

De Morgan, A. (1837–1843), 'Description of a calculating machine, invented by Mr. Thomas Fowler of Torrington in Devonshire', *Abstracts of the Papers Printed in the Philosophical Transactions of the Royal Society of London 4* 243–244. Abstract (De Morgan 1840).

De Morgan, A. (1840), 'Description of a calculating machine, invented by Mr Thomas Fowler of Torrington in Devonshire', AP.23.24., London: The Royal Society.

Epstein, G. (1993), *Multiple-Valued Logic Design*, Institute of Physics Publishing, Bristol.

Genzten, G. (1934), 'Untersuchungen über das logische Schliessen', *Mathematische Zeitschrift* 39, 176–210, 405–431. Translation into english: "Investigation into logical deduction", 68–131 of *The Collected Papers of Gerhard Gentzen*, red. M. E. Szabo. North-Holland, Amsterdam, 1969 Translation into polish: Trzęsicki, K. *Badania nad wnioskowaniem logicznym*, Białystok 1980.

Gottwald, S. (Winter 2004), Manyvalued logic, *w* E. N. Zalta, red., 'The Stanford Encyclopedia of Philosophy'.
http://plato.stanford.edu/archives/win2004/entries/logic-manyvalued

Greniewski, H. (1959), *Elementy cybernetyki systemem niematematycznym wyłożone*, Warszawa.

Greniewski, H. (1960), *Cybernetics without Mathematics*, Pergamon Press, New York.

Greniewski, H. (1962), 'Logique et cybernétique de la planification', *Cahiers du Séminaire d'Econométrie*, ss. 115–164.
http://www.globalsecurity.org/wmd/library/report/2005/art137-teheran2005.htm.

Greniewski, H. (1966), *Kybernetische Systemtheorie ohne Mathematik*, Dietz, Berlin.

Greniewski, M., Turski, W. (1963), 'The external language KLIPA for the URAL-2 digital computer', *Communications of the Association for Computing Machinery* 6 (6), 322–324.

Grzegorczyk, A. (1953), *Some Classes of Recursive Functions*, IV *w* 'Rozprawy Matematyczne', Instytut Matematyczny PAN, Warszawa.

Grzegorczyk, A. (1957), *Zagadnienia rozstrzygalności*, PWN, Warszawa.

Grzegorczyk, A. (1961), *Fonctions Récursives*, Gauthier-Villars, Paris.

Hamblin, C. L. (1957), An addressless coding scheme based on mathematical notation, *w* 'Proceedings of the First Australian Conference on Computing and Data Processing', Salisbury, South Australia: Weapons Research Establishment.

Hamblin, C. L. (1962), 'Translation to and from Polish notation', *Computing Journal* 5, 210–213.

Hamblin, C. L. (1987), *Imperatives*, Basil Blackwell, Oxford.

Hayes, B. (2001), 'Third base', *American Scientist* 89(6), 490-494. http://www.americanscientist.org/content/AMSCI/AMSCI/ArticleAltFormat/20035214317_146.pdf.

Hedger, L. (2006), 'Analog computation: Everything old is new again', http://www.indiana.edu/rcapub/v21n2/p01.html.

Jaśkowski, S. (1934), 'On the rules of supposition in formal logic', *Studia Logica* 1.

Jaśkowski, S. (1948), 'Rachunek zdań dla systemów dedukcyjnych sprzecznych', *Studia Societatis Scientiarun Torunesis* 1 (5), 55–77.

Jaśkowski, S. (1969), 'Propositional calculus for contradictory deductive systems', *Studia Logica* 24, 143–157. Translation into english (Jaśkowski 1948).

Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A. (1999), Rough sets: A tutorial, *w* 'Rough Fuzzy Hybridization. A new Trend in Decision-Making', Springer Verlag, Singapore.

Krajewski, S. (2005), Andrzej Grzegorczyk, *w* W. Mackiewicz, red., 'Polska filozofia powojenna', t. 3, Agencja Wydawnicza Witmark, Warszawa, pp. 99–118.

Lavington, S. H. (1980), *Early British Computers: The Story of Vintage Computers and the People who Built Them*, Manchester University Press, Manchester.

Łoś, J. (1948), 'Podstawy analizy metodologicznej kanonów Milla (the foundations of the methodological analysis of Mill's canons)', *Annales Universitatis Mariae Curie-Skłodowska, Section FII (5)*, 269–301.

Łukasiewicz, J. (1920a), 'Logika trójwartościowa', *Ruch Filozoficzny* V, 166–171.

Łukasiewicz, J. (1920b), 'O pojęciu możliwości', *Ruch Filozoficzny* V.

Łukasiewicz, J. (1920c), 'On three-valued logic', *Ruch Filozoficzny* 5, 170–171. Translation into english in: Borkowski, L. (red.) 1970. Jan Łukasiewicz: Selected Works. Amsterdam: North Holland.

Łukasiewicz, J. (1929), 'O znaczeniu i potrzebach logiki matematycznej', *Nauka Polska* 10, 604–20.

Łukasiewicz, J. (1930), 'Philosophische Bemerkungen zu mehrwertigen Systemem des Aussgankalkül', *Sprawozdania Towarzystwa Naukowego Warszawskiego Wydział III 23*, 51–77.

Łukasiewicz, J. (1931), Uwagi o aksjomacie Nicoda i «dedukcji uogólniającej», *w* 'Księga Pamiątkowa Polskiego Towarzystwa Filozoficznego we Lwowie', Lwów.

Łukasiewicz, J. (1961), *Z zagadnień logiki i filozofii. Pisma wybrane*, Państwowe Wydawnictwo Naukowe, Warszawa. Choice of papers done by J. Słupecki.

Madey, J., Sysło, M. M. (2000), 'Początki informatyki w Polsce', *Informatyka* 9/10.

Marciszewski, W. (1994), A Jaśkowski-style system of computer-assisted logic, in J. Woleński, red., 'Philosophical Logic in Poland', Synthese Library, Kluwer, Dordrecht. Changed version: *A System of Suppositional Logic as Embodied in the Proof Checker Mizar MSE* http://www.calculemus.org/MathUniversalis/3/marc-jas.html.

Marciszewski, W. (2005), 'A system of suppositional logic as embodied in the proof checker Mizar MSE', http://www.calculemus. org/MathUniversalis/3/marc-jas.html.

Marciszewski, W., Murawski, R. (1995), *Mechanization of Reasoning in a Historical Perspective*, Rodopi, Amsterdam/Atlanta.

McCall, S., red. (1967), *Polish Logic in 1920–1939*, Clarendon Press, Oxford.

Mills, J. (1993), Lukasiewicz' insect: The role of continuous-valued logic in a mobile robot's sensors, control, and locomotion, *w* 'Proc. 23rd Int. Symp. On Multiple-Valued Logic', IEEE Computer Society.

Mills, J. W. (2006), 'Kirchhoff-Lukasiewicz machines', http://www.cs.india-na.edu/∼jwmills/ANALOG.NOTEBOOK/klm/klm.html.

Mills, J., Walker, T., Himebaugh, B. (2003), 'Lukasiewicz' insect: Continuous-valued robotic control after ten years', *Int. Jour. Multiple-Valued Logic* 9 (2).

Moisil, G. C. (1959), *The Algebraic Theory of Switching Circuits (in Romanian)*, Bucharest. English translation: Pergamon Press 1969.

Moisil, G. C. (1966), *Zastosowanie algebr Łukasiewicza do teorii układów przekaźnikowo-stykowych*, t. 1, Wrocław – Warszawa – Kraków.

Moisil, G. C. (1967), *Zastosowanie algebr Łukasiewicza do teorii układów przekaźnikowo-stykowych*, t. 2, Wrocław – Warszawa – Kraków.

Moisil, G. C. (1972), *Essais sur les logiques non Chrysippiennes*, Editions de l'Acad. de la Rép. Soc. de Roumania, Bucharest.

Murawski, R. (1999), *Recursive Functions and Metamathematics. Problems of Completeness and Decidability, Gödel's Theorems*, Kluwer Academic Publishers, Dordrecht – Boston – London.

von Neuman, J. (1981), First draft of a report on the EDVAC, *w* N. Stern, red., 'From ENIAC to UNIVAC: An Appraisal of the Eckert-Mauchly Computers', Digital Press, Bedford, Massachusetts, ss. 177–246.

Niwiński, D. (2003), 'University of Warsaw', http://www.in-cites.com/institutions/UniversityofWarsaw.html.

Park, J. (2001), 'Using combinatory categorial grammar to extract biomedical information', *IEEE Intelligent Systems and Their Applications* 16, 62–67.

Pawlak, Z. (1982), 'Rough sets', *International Journal of Computer and Information Sciences* 11, 341–356.

Pawlak, Z. (1991), *Rough Sets – Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publisher, Dordrecht.

Pawlak, Z. (1993), 'Rough sets. Present state and the future', *Foundations of Computing and Decision Sciences* 18, 157–166.

Pearcey, T. (1994), Australian computing: The second generation, *w* J. M. Bennett, R. Broomham, P. M. Murton, T. Pearcey, R. W. Rutledge, red., 'Computing in Australia: The Development of a Profession', Australian Computer Society.

Post, E. (1921), 'Introduction to a general theory of elementary propositions', *American Journal of Mathematics* XLIII.

Prior, A. N. (1955), *Formal Logic*, Clarendon Press, Oxford. II wyd. 1962.

Prior, A. N. (1996), A statement of temporal realism, *w* B. J. Copeland, red., 'Logic and Reality: Essays on the Legacy of Arthur Prior', Oxford University Press.

Rescher, N. (1969), *Many-valued Logic*, New York.

Rine, D. C., red. (1977), *Computer Science and Multiple Valued Logic*, North-Holland Publ. Comp., Amsterdam. The second edition. 1984.

Rubel, L. A. (1993), 'The extended analog computer', *Adv. In Appl. Math.* 14, 39–50.

Rumyantsev, D. (2004), 'Interview s konstruktorom troicznoj ewm (interviews with the designer of the ternary computer)', *Upgrade* 33 (175).

Segal, J. (1999), L'introduction de la cybernétique en R.D.A. Rencontres avec l'idéologie marxiste, in 'Proceedings of the XXth International Congress of History of Science (Liege, 20–26 July 1997)', t. 1, pp. 67–80. Die Einführung der Kybernetik in der DDR. Begegnung mit der marxistischen Ideologie http://jerome-segal.de/Publis/Kyb-DDR.htm.

Szumiec-Presch, B. (2004), 'Madey: Najbardziej cieszą mnie sukcesy moich studentów i wychowanków', http://www.naukawpolsce.pap.pl/nauka/index.jsp?place=Lead20&news_cat_id=62&news_id=319&layout=1&page=text.

Trzęsicki, K. (2005), Arthura Normana Priora związki ze szkołą lwowsko-warszawską, in K. Trzęsicki, editor., 'Ratione et studio', Uniwersytet w Białymstoku, Białystok, pp. 269–288.

Williams, G. (1985), 'A shy blend of logic, maths and languages (obituary of Charles Hamblin)', Morning Herald, Sydney.

Woleński, J. (1985), *Filozoficzna szkoła lwowsko-warszawska*, PWN, Warszawa.

Woleński, J. (2005), Dzieje pewnego przypisu, *w* K. Trzęsicki, editor., 'Ratione et studio', Wydawnictwo Uniwersytetu w Białymstoku, Białystok, pp. 249–268.

Zadeh, L. (1965), 'Fuzzy sets', *Journal of Information and Control* 8, 338–353.

Andrzej Malec
University of Białystok

A FORMAL APPROACH TO NATURAL LAW

The idea of the paper is to use a symbolism taken from logic to explicate some notions of civil law, such as: subjective right, relative right, right, liability, obligation and claim. By having a formal explication of the above notions we will be able to find some logical consequences of these notions. In particular, we should be able to establish relations between the explicated notions. Legal statements that are logical consequences of generally accepted legal notions truly deserve the name of "natural law": if we accept the notions, logic itself will force us to accept the statements.

Two Meanings of the Expression "Legal Rationality"

When using the expression "legal rationality" we usually have in mind either (1) some attributes of lawyers ("rational lawyer") or (2) some attributes of law itself ("rational law").

Rational Lawyer

To explain legal rationality as the rationality of lawyers it is necessary to define the expected rules of behaviour of rational lawyers (i.e., to define what the expression "rational behaviour" means in relation to lawyers). Generally speaking, we can define rational behaviour in the following way: a behaviour is rational if and only if:
(a) it is based on a certain schema of action (an "algorithm"),
(b) it is efficient (in some sense[1]),
(c) it leads us to a good in a moral sense.

[1] There are several notions of efficiency. They are defined in praxiology, rational choice theory, expected utility theory, game theory, etc.

ISBN 978–83–7431–158–8 ISSN 0860-150X

Assuming the above general idea of rational behaviour to be true, we need to have an accepted (rational) hierarchy of values to be able to determine if a behaviour is rational or not. So, a choice of hierarchy of values determines a set of rational behaviours. Let us set forth an example related to lawyers[2]. There is a conflict of legal formalism and legal activism in the theory of law. According to legal formalism, legal norms applied by lawyers should be strictly connected to legal texts[3]. This means that a judge should be no more than a "logical device" for inferring norms that are coded by a lawmaker in legal texts (for example: coded by parliament in a bill). On the contrary, according to legal activism, lawyers should avoid blind subordination to legal texts: if a norm inferred from a legal text is "unjust", then a lawyer should (is supposed to) ignore it. The above opposite conceptions of rational behaviour of lawyers are based on different hierarchies of values. If we accept that the predictability of law is more important than the justice of the law, then we make a choice in favour of legal formalism. Otherwise – we make a choice in favour of legal activism.

The Is-Ought Problem

However, since Hume we have become aware that it is impossible to infer any statement about values from any statement about facts (Hume's "is-ought problem"). Today, we say that factual statements and deontic statements are logically separated[4]. Therefore, it is impossible to justify any hierarchy of values by methods of the so called "positive sciences". Respectively, it is difficult (if at all possible) to find a universal (absolute) hierarchy of values.

But if so, then we are not able to make a rational choice between legal formalism and legal activism. In other words, we are not able to determine by reason who is rational: a judge that subordinates himself to unjust norms coded in legal texts, or a judge that ignores such unjust norms. Therefore, the rationality of lawyers can be understood merely in terms of the so called

[2] The paper is connected with continental (especially Polish) tradition in theory of law. Nevertheless, methods and ideas presented in the paper apply to common law as well.

[3] Legal norms are rules of behaviour prescribed by the authorities. Legal texts are sets of inscriptions from which legal norms can be inferred. A text to be a legal text must be accepted in due course by a legitimate authority of the state.

[4] See: Jan Woleński, *Uogólniona teza Hume'a*, in: I. Bogucka, Z. Tobor (editors) "Prawo a wartości. Księga jubileuszowa Profesora Józefa Nowackiego" Zakamycze Kraków 2003, p. 293–303.

"instrumental rationality" ("hypothetical rationality"): any action taken by a lawyer can be classified as rational or not rational only from a viewpoint of a certain accepted hierarchy of values. Therefore, we cannot say "behavior x is rational"; all we can say is: "if our aim is y, then behavior x is rational".

Rational Law

To explain legal rationality as the rationality of law it is necessary to say what the attributes of a rational system of legal norms are. We can indicate some formal attributes such as consistency or completeness, but the question remains: can we indicate any material attributes? In other words: can we indicate any universal (absolute) legal norms? Or: is there a natural law?

At first sight the answer to the above questions is "no". The thesis that factual statements and deontic statements are logically separated still holds. So, if it is impossible to justify any hierarchy of values by using the methods supplied by the positive sciences, then it is probably also impossible to indicate any universal (absolute) norms[5].

So, it is difficult (if possible) to set forth any material attributes of a rational system of law without a prior acceptance of a hierarchy of values.

There is a way however. A way that enables us to set forth some material attributes of a rational system of law that avoids simultaneously all the possible discussions about which system of values is better.

The Way

The way consists in:

(a) obtaining a symbolic explication of certain legal notions and
(b) having such an explication – inferring logical consequences of these notions.

The notions that shall be examined are those which belong to civil law, evident since the Roman Empire. Such notions are present in our language. They are a part of our language. Therefore we need not accept any system of values to accept such notions: we learnt these notions when we were learning the language we use. We can say that the system of values connected with these notions is an intrinsic part of our language.

---

[5] Deontic statements are not norms but are closely related to norms. "John is obliged to close the window" is a deontic statement. "John, close the window, please!" is a norm.

Obviously, the logical consequences of these notions constitute a set of sentences that are analytical in relation to these notions (analytical statements)[6]. So, since the notions in question (and the system of values connected with them) are a part of our language, the logical consequences in question constitute a set of analytical statements from the viewpoint of our language. Since these notions are legal in nature and the related statements are about legal relations, the logical consequences in question constitute a set of analytical statements related to the matter of law. It is quite in accordance with common intuition to name such a set "natural law".

Some Notions of Civil Law

Some of the most general (and also/therefore most important) notions of civil law are the following notions:
(a) subjective right,
(b) relative right,
(c) right,
(d) liability,
(e) obligation,
(f) claim[7].

The above notions arose in Roman civil law or from the inspiration derived from Roman civil law. In the Polish Civil Code of 1964 (the code still remains in force) they are not defined[8]. Nevertheless, the correct understanding of the notions in question has a key role in understanding regulations of civil law: any person that interprets a regulation of civil law has to take into account not only the regulation itself, but also the notions in question. It is necessary for finding a rule of behaviour prescribed by the regulation, i.e., for finding a legal norm.

---

[6] A statement is analytical if and only if the issue of whether it is true or false can be determined exclusively by analyzing the meaning of words that constitute the statement.

[7] There are no exact English equivalents of the Polish legal terms "prawo podmiotowe" (subjective right) and "wierzytelność" (relative right). I have proposed the above translation having in mind the meanings of the terms: a subjective right is a right attributed to a subject of law (e.g., to a physical person) and is valid in relation to all other subjects of law; on the contrary – a relative right is valid only between parts of a legal relation (e.g., a legal relation that occurred as a consequence of a contract or a tort).

[8] With the exception of the notion of liability that is defined in article 353. There are also some consequences of the notions in the code.

On the other hand, it can be observed that the notions in question do not have a clear meaning for many lawyers[9]. It is a consequence of their abstract character. For example, what is the difference between the meanings of "right" and "claim" in the sentences: "I have a right" and "I have a claim"?

Some Definitions

Let us look at some definitions:

(a) a definition of liability given in article 353 of the Polish Civil Code of 1964:

> Liability consists in that a creditor may demand from a debtor to fulfil the debtor's debt and the debtor ought to fulfil the debt.

(b) definitions and relations given by the theory of civil law[10]:

- (i) Subjective right – a sphere of the ability to act in a way defined by a legal norm (i.e., to act according to the matter/essence of subjective right) that is granted by the norm to a subject of legal relation.
- (ii) A subjective right brings rights. The rights are correlated to liabilities of uncertain (undefined) subjects or liabilities of certain (defined) subjects. If a right is correlated to a liability of a certain (defined) subject, the right is a claim.
- (iii) A claim consists in an ability to demand from a certain (defined) subject to behave in a certain (defined) way (to act, to give up, to bear).
- (iv) Relative right – a sphere of the ability to act in a way defined by a legal norm in relation to a defined (other) party of legal relation.

Symbolism

Let us construct a theory based on the first order predicate logic.

To the axioms of the first order predicate logic we add some new axioms that are supposed to be explanations of the legal notions in question. We

[9] The author of the present paper has made several observations of this kind.

[10] Reconstructions based on "System prawa cywilnego" (in Polish: "The System of Civil Law") – a system of fundamental inquiries concerning Polish civil law, prepared when the Polish Civil Code was issued.

presume that the additional axioms define the meanings of the notions in question as they are in Roman law. We also admit that it is a partial explanation: i.e., that the explanation indicates only a number, but not all of the relations between the notions.

For every subjective right, relative right, right, liability or claim we should have a separate axiom (axioms). But all axioms that define subjective rights are of the same schema (schemas). Respectively, all axioms that define relative rights are of the same schema (schemas), all axioms that define claims are of the same schema (schemas), *et cetera*. Therefore, we will analyze schemas of axioms.

Subjective Right versus Right

According to the definitions stated above, a subjective right is a sphere of the ability to act in a way which is defined by a legal norm that is granted by the norm to a subject of a legal relation. We also have that a subjective right brings rights. We can express the above in the following way:

$$\forall x\{SR(x) \equiv \forall y[R_1(x, y) \wedge R_2(x, y) \wedge \ldots \wedge R_n(x, y)]\}$$

where:

the domain is the set of subjects of law,
SR stands for a subjective right,
$R_1, R_2, \ldots, R_n$ stand for rights.

Right versus Claim

According to the definitions stated above, a subjective right brings rights. The rights are correlated to liabilities of uncertain (undefined) subjects or liabilities of certain (defined) subjects. If a right is correlated to a liability of a certain (defined) subject, the right is a claim. We can express the above in the following way:

$$\forall x \forall y\{R_i(x, y) \equiv [S_i(y) \rightarrow C_i(x, y)]\}$$

where:

S_i stands for the status of y (we read the expression $S_i(y)$ as "y is a subject in the situation S_i (y is a defined subject"),
C_i stands for a claim.

Claim versus Actions of a Creditor and a Debtor

According to the definitions stated above, a claim consists in the ability to demand from a certain (defined) subject to behave in a certain (defined) way. We can express the above in the following way:

$$\forall x \forall y \{C_i(x, y) \equiv [D_i(x) \rightarrow B_i(y)]\}$$

where:

D_i stands for the status of x (we read the expression $D_i(x)$ as "x demands a behaviour"),

B_i stands for the status of y (we read the expression $B_i(y)$ as "y ought to behave").

Liability versus Obligation

According to the definitions stated above, a liability consists in that a creditor may demand from a debtor to fulfil the debtor's debt and the debtor ought to fulfil the debt. We can express the above in the following way:

$$\forall x \forall y \{L_i(y, x) \equiv [S_i(y) \rightarrow O_i(y, x)]\}$$

$$\forall x \forall y \{O_i(y, x) \equiv [D_i(x) \rightarrow B_i(y)]\}$$

where:

L_i stands for a liability,

O_i stands for an obligation.

Therefore a debtor is obliged to fulfil a debt if and only if a creditor has demanded to fulfil the debt.

Some Consequences

The following relations can also be established.

$$\forall x \forall y \{C_i(x, y) \equiv O_i(y, x)\}$$

(Every claim is correlated to an obligation.)

$$\forall x \forall y \{R_i(x, y) \equiv L_i(y, x)\}$$

(Every right is correlated to a liability.)

$$\exists x\ SR(x) \to \exists x \forall y\ R_i(x, y)$$

(Every subjective right brings rights.)

$$\neg \forall x \forall y \{R_i(x, y) \to C_i(x, y)]\}$$

(Some rights do not bring claims.)

Subjective Liability

So, we found that the term "right" forms a pair with the term "liability" and the term "claim" pairs with the term "obligation". However, there is no term that forms a pair with the term "subjective right". Do we have a notion that pairs with the notion of subjective right? We should imagine such a notion (which we can call "subjective liability") in the following way:

$$\forall y \{SL(y) \equiv \forall x [L_1(y, x) \wedge L_2(y, x) \wedge \ldots \wedge L_n(y, x)]\}$$

where:

SL stands for subjective liability.

Such a notion can be found in the Polish Civil Code of 1964 – namely in article 919:

> *Who publicly declared a prize for an action is obliged to fulfil the declaration.*

So, there is no such term as "subjective liability" in legal language. However, using formal tools, we came to the idea of a relevant notion. And after that we found such a notion in a legal text. It shows that a formal approach can enlarge the theoretical apparatus of the theory of law.

Relative Right

Relative rights constitute a kind of subjective rights. Namely, a subjective right is a relative right if and only if it constitutes a sphere of ability to act in a way defined by a legal norm in relation to a defined (other) party of a legal relation. We can express the above in the following way:

$$\forall x \{RR(x) \equiv \forall y [R_1(x, y) \wedge R_2(x, y) \wedge \ldots \wedge R_n(x, y)] \wedge \\ \wedge\ \exists y [S_1(y) \wedge S_i(y) \wedge \ldots \wedge S_n(y)]\}$$

where:

RR – stands for relative right.

And we can infer some consequences:

$$\forall x\{R_i(x,b) \equiv C_i(x,b)]\}$$

(Rights derived from any relative right are claims.)

$$\forall x\{RR(x) \to \exists y[C_1(x,y) \wedge C_2(x,y) \wedge \ldots \wedge C_n(x,y)]\}$$

(Relative rights bring claims.)

$$\forall x\{L_i(b,x) \equiv O_i(b,x)\}$$

(Liabilities corresponding to relative rights are obligations.)

$$L_i(b,a) \equiv C_i(a,b)$$

(Any debtor's liability corresponds to a claim of a creditor.)

$$L_i(b,a) \equiv [D_i(a) \to B_i(b)]$$

(A liability consists in that a creditor may demand from a debtor to fulfil the debtor's debt and the debtor ought to fulfil the debt – i.e., the norm expressed in article 353 of the Polish Civil Code of 1964)

Beyond First Order Logic?

The aim of the paper is to examine whether a logical symbolism can be effectively used in the theory of law for explication of legal notions. In previous paragraphs we were concerned with means taken from first order logic and some basic notions of civil law: subjective right, relative right, right, liability, obligation and claim. However more results can be obtained if we enrich our formal apparatus with some means of the second order logic and temporal logic. Then we are able to explain, e.g., how we should understand a Roman definition of ownership as *ius possidendi, disponendi, utendi – fruendi et abutendi*[11]:

$$\forall x\{OS(x,a) \equiv \forall R[A(R,a) \to \forall y\ R(x,y)]\}$$

where:

a is a constant denoting a property,

OS stands for ownership,

A stands for a kind of connection between R and a (we read "R is a right relevant to a").

[11] "Ownership is a right to possess, to dispose of, to use and to abuse."

Or, how we should define the transition of a subjective right[12]:

$$TR(x, y, SR, t_k) \equiv \{SR(x, t_{k-1}) \wedge SR(y, t_{k+1})\}$$

where:

TR stands for transition,

t_k, t_{k-1}, t_{k+1} stand for moments of time,

having as an important consequence of the above:

$$\neg SR(x, t_{k-1}) \rightarrow \neg TR(x, y, SR, t_k)$$

that constitutes a Roman rule: *Nemo plus iuris in alium transferre potest quam ipse habet*[13].

Having temporal notions of right and claim we can express a legal rule that any claim will expire whereas no right can expire:

$$\forall x \forall y \{R_i(x, y) \equiv \forall t[S_i(y, t) \rightarrow C_i(x, y, t)]\}$$

$$\forall x \forall y \{C_i(x, y, t) \equiv [D_i(x, t) \rightarrow B_i(y, t)]\}$$

$$\forall y \forall t \{S_i(y, t) \rightarrow \exists t_j < t\ [\neg S_i(y, t_j)] \wedge \exists t_k > t\ [\neg S_i(y, t_k)]\}$$

Natural Law?

So, we have outlined some explications of legal notions by means of logic. As a result, several legal relations were established as logical consequences of the explications:

(a) Any claim corresponds to an obligation,
(b) Any right corresponds to a liability,
(c) Subjective rights bring rights,
(d) Some rights do not bring claims,
(e) Rights derived from any relative right are claims,
(f) Liabilities corresponding to relative rights are obligations,
(g) Any debtor's liability corresponds to a claim of a creditor,
(h) A liability consists in that a creditor may demand from a debtor to fulfil the debtor's debt and the debtor ought to fulfil the debt,
(i) Any relative right brings claims,
(j) Ownership is a right to possess, to dispose of, to use and to abuse (i.e., contains all rights connected to a property),

[12] However, for this purpose we need a temporal explication of subjective right: $\forall x \forall t \{SR(x, t) \equiv \forall y[R_1(x, y, t) \wedge R_2(x, y, t) \wedge \ldots \wedge \forall R_n(x, y, t)]\}$ where t is a variable for time moments.

[13] "No one can transfer more than she/he has."

(k) *Nemo plus iuris in alium transferre potest quam ipse habet*,
(l) Any claim will expire, whereas no right can expire.

As stated at the beginning of the paper, the logical consequences in question constitute a set of analytical statements related to the matter of law. It is quite in accordance with common intuition to name such a set "natural law".

Conclusions

There are two conclusions from the above examination.

Firstly and fundamentally, it is possible to indicate material properties, which any rational system of law should possess, without choosing any hierarchy of values. As a consequence, it is possible to develop an objective theory of natural law which is not limited to formal considerations (as consistency or completeness of a system of norms or so on).

Secondly, logical formal tools are really useful for legal reasoning (at any rate – in the theory of law) and therefore such tools should be propagated among students of law. These tools, when used correctly, can significantly improve a student's understanding of fundamental legal notions.

Andrzej Malec
Department of Logic, Informatics and Philosophy of Sciences
University of Białystok
e-mail: malets@uwb.edu.pl

STUDIES IN LOGIC, GRAMMAR AND RHETORIC 11 (24) 2007

Andrew Schumann
Belarusian State University (Minsk, Belarus)

NON-ARCHIMEDEAN FOUNDATIONS OF MATHEMATICS

Finite foundations of mathematics developed by D. Hilbert are presently considered in computer science as an original mathematical canon. Nevertheless, transfinite foundations of mathematics proposed by G. Cantor can also be urgent for soft computing. In this paper I consider some perspectives of transfinite foundations, namely I propose non-Archimedean foundations of mathematics and non-Archimedean multiple-validity. Further, I construct a logical language with non-Archimedean valued semantics.

1. Finite and Transfinite Foundations of Mathematics

In the foundational views of late-19th century mathematicians we can observe two approaches to foundations of mathematics, which are presently called finite and transfinite foundations. According to the first approach developed, e.g. by Kronecker and Brouwer, only objects that are "*intuitively present as immediate experience prior to all thought*" [4] are interpreted as initial objects of algebra and analysis. These objects are considered to be natural numbers. This means that all the mathematical operations must be defined on finite sets or on sets of potential attainable objects, i.e. they must have a reduction to operations on natural numbers. As a result, completed infinite totalities must be rejected in mathematical research. What comes to mind here the Kronecker's famous aphorism: "*Die ganze Zahl schuf der liebe Gott, alles Übrige ist Menschenwerk*" ("*God created natural numbers, all others are fashioned by human beings*").

For instance, David Hilbert considered natural numbers to be the finitary numerals, those which have no meaning, i.e., they do not stand for abstract objects, but can be operated on and compared. According to Hilbert, knowledge of their properties and relations is intuitive and unmediated by logical inference.

ISBN 978–83–7431–158–8 ISSN 0860-150X

In the transfinite approach to foundations of mathematics created by Georg Cantor, operations on completed infinite totalities are possible if we have no contradiction regarding their restrictions to finite sets. "*Mathematics is in its development entirely free and is only bound in the self-evident respect that its concepts must both be consistent with each other, and also stand in exact relationships, ordered by definitions, to those concepts which have previously been introduced and are already at hand and established. In particular, in the introduction of new numbers, it is only obligated to give definitions of them which will bestow such a determinacy and, in certain circumstances, such a relationship to the other numbers that they can in any given instance be precisely distinguished. As soon as a number satisfies all these conditions, it can and must be regarded in mathematics as existent and real*" [1].

Cantor distinguished between the improper infinite ("*Uneigentlich-Unendliches*") and the proper infinite ("*Eigentlich-Unendliches*"). He called the first kind of infinity the variable finite (*veränderliches Endliches*) and syncategorematic infinity ($\alpha\pi\varepsilon\iota\rho o\nu$, *synkategorematice infinitum*). The second kind of infinity is regarded by him as the actual infinite (*transfinitum*) and categorematic infinity ($\alpha\phi\omega\rho\iota\sigma\mu\varepsilon\nu o\nu$, *kategorematice infinitum*). According to Cantor, the set of natural numbers is improper infinite. He believed that there exists the proper infinite, which includes unattainable objects.

Modern mathematics and computer science are based, as a rule, on discrete objects, therefore they are constructed in the framework of finite foundations of mathematics. These foundations were formulated by Hilbert and they came to be known as *Hilbert's Program*, in which a formalization of all mathematics in axiomatic form, together with proof that this axiomatization of mathematics is consistent, is supposed. The consistency proof itself was to be carried out using only what Hilbert called "finitary" methods. Hilbert's Program was optimistic – it was assumed that the subject of mathematics consists only of the problems that can have a positive or negative solution by means of finitary methods. This optimistic standpoint allows us to set up 23 mathematical problems, which Hilbert addressed to the International Congress of Mathematicians in 1900 (see [5]). Hilbert thought that every well defined mathematical problem can have a solution: "*Take any definite unsolved problem, such as the question as to the irrationality of the Euler-Mascheroni constant C, or the existence of an infinite number of prime numbers of the form $2^n + 1$. However unapproachable these problems may seem to us and however helpless we stand before them, we have, nevertheless, the firm conviction that their solution must follow by a finite number of purely logical processes*" [5].

However there exist mathematical problems that are effectively insolvable, i.e. it is possible to set an effective proof of their insolvability. For example, Gödel and Cohen showed that the first of Hilbert's 23 problems does not have a solution (see [3], [2]). On the other hand, Matiyasevich proved the same result concerning the tenth of Hilbert's 23 problems (see [7]). These results show the limits of application of regarding the finite approach.

The tenth of Hilbert's problems was the following. "*Given a Diophantus equation with any number of unknowns and with rational integer coefficients: devise a process, which could determine by a finite number of operations whether the equation is solvable in rational integers*". Yuri Matiyasevich proved the insolvability of this problem using Cantor's diagonal construction (see [7]).

The first problem was the following: *The set of real numbers is well ordered*, or equivalently there is no a transfinite number between that of a denumerable set and the numbers of the continuum, i.e. $\mathfrak{c} = \aleph_1 = 2^{\aleph_0}$. This statement was proposed by Cantor and it is called *continuum hypothesis*. It is also connected to the axiom of choice. Since the continuum hypothesis is true, every object must be attainable – *for any two upper bounded subsets of an ordered set there exists an upper bound*.

Also, if we accept continuum hypothesis, then we set infinite objects as an infinite union of finite objects. For example, we can define the set ω of natural numbers as follows: $\omega = \sup\{n : n \text{ is a natural number and } n < \omega\}$. In Hilbert's opinion, we cannot avoid infinite constructions, therefore we must obtain infinite unions of finite objects using continuum hypothesis and the axiom of choice. For instance, a set of real numbers should be obtained by the union of rational number sets (denumerable sets): "*It should also be remarked that the process just described for obtaining an upper bound amounts to forming a* union set. *In fact every real number is defined by a partition of the rational numbers into larger and smaller ones or by the set of the smaller rational numbers. The given set of real numbers is therefore represented as a set* $\mathfrak{M}$ *of sets of rational numbers. And the upper bound of the set* $\mathfrak{M}$ *is formed from the set of those rational numbers which belong to at least one of the sets in* $\mathfrak{M}$*. The totality of these rational numbers is, however, exactly the union set of* $\mathfrak{M}$" [4]. On the other hand, "*it follows from the property of an upper bound that for every integer n there is a number* c_n *in the set such that* $a - \frac{1}{n} < c_n \leq a$ *and so* $|a - c_n| < \frac{1}{n}$*. The numbers* c_n *constitute therefore a sequence which converges toward a, and they all belong to the set under consideration.*

When we argue in this way our manner of expression hides a fundamental point in the proof. For when we use the notation c_n *we presuppose that* *for each number n among those real numbers c belonging to the set under consideration and satisfying the inequality* $a - \frac{1}{n} < c \leq a$*, a certain one is distinguished.*" [4].

Also, Hilbert's first problem is to constraint or formalize set theory and thereby prove continuum hypothesis and axiom of choice. Let us take Zermelo-Fraenkel's set theory **ZF**. Suppose M is a model for **ZF** and α is an ordinal. Define M_α by setting (1) $M_0 = \emptyset$, (2) if $\alpha \neq 0$, then $M_\alpha = \bigcup_{\beta<\alpha} M_\beta$. A set x is constructible if there exists an ordinal α such that $x \in M_\alpha$. Gödel showed that continuum hypothesis and the axiom of choice are provable in **ZF** if we take his universe of constructible sets.

At the same time, Cohen proved that we can refute continuum hypothesis and the axiom of choice in other models for **ZF**. Define a new model N that is a denumerable and transitive extension of M. Then we can prove in N that there is no universe of constructible sets. Let $\mathrm{Fn}(A, B) := \{p \subset A \times B$: p is finite function$\}$. Then continuum hypothesis is refuted on the base of $\mathrm{Fn}(\omega \times \lambda, 2)$, where λ is an ordinal, and on the basis of other assumptions, because already $\aleph_1 \neq 2^{\aleph_0}$. Cohen's idea is that there exist infinite unions of finite objects that are not attainable.

Thus, Gödel and Cohen proved that the solution to the first Hilbert's problem depends on the particular version of the set theory assumed, i.e. the acceptance or rejection of the continuum hypothesis and the axiom of choice is subjective and it is not connected to the axiomatic formalization of the set concept. Consequently, the setting of infinite sets as attainable infinite union (as upper bound) is no less subjective than its setting as unattainable infinite union (as actual infinity). The finite approach to foundations of mathematics is no less subjective than a transfinite one!

Consider some examples of sets that cannot be regarded as attainable infinite union (as upper bound).

1. **The set of incomputable functions**. Let $d(n) = U(n, n)$, where U is a two-place computable function that is universal for the class of one-place computable functions. Define the new function as follows

$$d''(x) = \begin{cases} 1 & \text{if } d(x) = 0, \\ 0 & \text{if } d(x) > 0. \end{cases}$$

Each completely defined extension of the function $d''(x)$ will be incomputable. This implies that the set of all completely defined extensions of $d''(x)$ is actual infinity, i.e. we cannot get this set as a minimal one in the given class (it has no an upper bound).

2. **The set of non-constructible real numbers**. Assume that any real numbers of $(0, 1)$ are contained in a sequence $x_1, x_2, \ldots, x_n, \ldots$ and

every number x_n can be represented as an infinite decimal fraction $x_n = 0, a_1^{(n)} a_2^{(n)} \ldots a_n^{(n)} \ldots$ that is not repeating with a repeater of 9. Take a number b_n for any $a_n^{(n)}$ such that $b_n \neq a_n^{(n)}$. Let us consider the infinite decimal fraction $0, b_1 b_2 \ldots b_n \ldots$ This number is called *non-constructible real number*. The set of such numbers is not attainable infinite union – it is the maximal set in the given class.

3. **The set of transcendental numbers**. According to Liouville's theorem, for any algebraic number α with degree $n > 1$, there exists positive λ such that $\left|\alpha - \frac{p}{q}\right| \geq \frac{\lambda}{q^n}$ for any rational number $\frac{p}{q}$. From this it follows that each transcendental number α satisfies the following inequality $\left|\alpha - \frac{p}{q}\right| < \frac{\lambda}{q^n}$, i.e., it has a stronger convergence than an algebraic number. This means that if we set the union of two transcendental numbers, then we obtain their maximum such that the law of strong convergence absorbs the law of weak convergence. Conversely, the law of weak convergence absorbs the law of strong convergence for the union of two algebraic numbers. In fact, if $\alpha_1 \geq \alpha_2$, the law of convergence for an algebraic number α_1 is weaker.

Also, in some cases we cannot use Hilbert's idea of an upper bound as an attainable infinite union for the transfinite setting of mathematical objects and we must postulate actual totalities in advance. We can propose the following understanding of actual infinity. *A set of mathematical objects of the same nature is called an actual infinite set if we are always able to obtain their infinite intersection, but cannot obtain their infinite union (as upper bound) in a general sense.* For instance, if A is the set of incomputable functions (resp. the set of non-constructible real numbers or the set of transcendental numbers) and $A \subset B$, then B is also the set of incomputable functions (resp. the set of non-constructible real numbers or the set of transcendental numbers). Therefore the union for these sets is not upper bound. In the same way, we can postulate the following principle.

Principle 1 (Principle of non-linearity) *An actual infinite totality cannot be represented as linear sequence of its objects.*

Suppose that we have built *a transfinite metalanguage of mathematics*, in which we define well formed formulas by setting some actual infinities. Suppose also that our formulas and proofs can be coded by some numbers (evidently, it is possible that they are actual infinite numbers). Then, our formulas can be seen as a number of transfinite programs to set operations on actual infinite numbers. In consequence we obtain the following new principle.

Principle 2 (Transcendental principle) *Transfinite logical language is infinite-order.*

According to this, transfinite logical language has no a metatheory in the sense that the truth concept is formalized in such metalanguage.

It is known that actual infinite numbers does not satisfy Archimedes' axiom. Let us remember that this axiom is the formula of infinite length which has one of the following notations:

- For any ε that belongs to the interval $[0,1]$, we have

$$(\varepsilon > 0) \supset [(\varepsilon \geq 1) \vee (\varepsilon + \varepsilon \geq 1) \vee (\varepsilon + \varepsilon + \varepsilon \geq 1) \vee \ldots], \tag{1}$$

- For any positive integer ε, we have

$$[(1 \geq \varepsilon) \vee (1 + 1 \geq \varepsilon) \vee (1 + 1 + 1 \geq \varepsilon) \vee \ldots]. \tag{2}$$

Formulas (1) and (2) are valid in the field $\mathbf{Q}$ of rational numbers as well as in field $\mathbf{R}$ of real numbers. In the ring $\mathbf{Z}$ of integers, only formula (2) has a nontrivial sense, because $\mathbf{Z}$ does not contain numbers of the open interval $(0,1)$. Also, Archimedes' axiom affirms the existence of an integer multiple of the smaller of the two numbers which exceeds the greater: for any positive real or rational number ε, there exists a positive integer n such that $\varepsilon \geq \frac{1}{n}$ or $n \cdot \varepsilon \geq 1$. The negation of Archimedes' axiom has one of the following forms:

- There exists ε which belongs to the interval $[0,1]$ such that

$$(\varepsilon > 0) \wedge [(\varepsilon < 1) \wedge (\varepsilon + \varepsilon < 1) \wedge (\varepsilon + \varepsilon + \varepsilon < 1) \wedge \ldots], \tag{3}$$

- There exists a positive integer ε such that

$$[(1 < \varepsilon) \wedge (1 + 1 < \varepsilon) \wedge (1 + 1 + 1 < \varepsilon) \wedge \ldots]. \tag{4}$$

Notice that (3) is the negation of (1). It is obvious that formula (3) reveals there exist *infinitely small numbers* (or *infinitesimals*), i. e., numbers that are smaller than all real or rational numbers of the open interval $(0,1)$. In other words, ε is said to be an infinitesimal if and only if, for all positive integers n, we have $|\varepsilon| < \frac{1}{n}$. Further, formula (4) reveals there exist *infinitely large integers* that are greater than all positive integers. Infinitesimals and infinitely large integers are called *nonstandard numbers* or *actual infinities.*

The field that satisfies all the properties of $\mathbf{R}$ without Archimedes' axiom is called the field of *hyperreal numbers* and it is denoted by ${}^*\mathbf{R}$. The field that satisfies all the properties of $\mathbf{Q}$ without Archimedes' axiom is called the field of hyperrational numbers and it is denoted by ${}^*\mathbf{Q}$. By definition of field, if $\varepsilon \in \mathbf{R}$ (respectively $\varepsilon \in \mathbf{Q}$), then $1/\varepsilon \in \mathbf{R}$ (respectively $1/\varepsilon \in \mathbf{Q}$). Therefore ${}^*\mathbf{R}$ and ${}^*\mathbf{Q}$ simultaneously contain infinitesimals and infinitely large integers: for an infinitesimal ε, we have $N = \frac{1}{\varepsilon}$, where N is an infinitely large integer.

The ring that satisfies all the properties of $\mathbf{Z}$ without Archimedes' axiom is called the ring of hyperintegers and it is denoted by ${}^*\mathbf{Z}$. This ring includes infinitely large integers. Notice that there exists a version of ${}^*\mathbf{Z}$ that is called the ring of p-adic integers and is denoted by $\mathbf{Z}_p$.

The main originality of non-Archimedean number systems consists in that the set of *hypernumbers* cannot be well ordered – e.g., there is no effective ordering relation on the set of infinitesimals. Therefore hypernumbers satisfy the principle of non-linearity.

Set up a problem to construct a metalanguage of mathematics in that well formed formulas have their truth values in the set of hypernumbers (namely, in ${}^*\mathbf{R}$ and ${}^*\mathbf{Q}$). I will show that this metalanguage is infinite-order, i.e., it satisfies the transcendental principle and it truly is a transfinite metalanguage.

2. Non-Archimedean Valued Matrix

Consider a set Θ. Let I be any infinite index set. Then, we can construct an indexed family Θ^I, i.e., we can obtain the set of all functions: $f\colon I \to \Theta$ such that $f(\alpha) \in \Theta$ for any $\alpha \in I$. The set of all complements for finite subsets of I is a filter and it is called a *Frechet filter*. A maximal filter (ultrafilter) that contains a Frechet filter is called a *Frechet ultrafilter* and is denoted by $\mathcal{U}$. Let $\mathcal{U}$ be a Frechet ultrafilter on I. Define a new relation $\approx$ on the set Θ^I by $f \approx g \equiv \{\alpha \in I\colon f(\alpha) = g(\alpha)\} \in \mathcal{U}$. It is easily proven that the relation $\approx$ is an equivalence. Notice that the aforementioned formula means that f and g are equivalent iff f and g are equal on an infinite index subset. For each $f \in \Theta^I$ let $[f]$ denote the equivalence class of f under $\approx$. The *ultrapower* $\Theta^I/\mathcal{U}$ is then defined to be the set of all equivalence classes $[f]$ as f ranges over Θ^I: $\Theta^I/\mathcal{U} := \{[f]\colon f \in \Theta^I\}$.

Also, we can say that each nonempty set Θ has an ultrapower with respect to a Frechet filter/ultrafilter $\mathcal{U}$. (Notice that if $\mathcal{U}$ is a Frechet filter, we have no well-ordering relation. On the other hand, suppose that $\mathcal{U}$ is a Frechet ultrafilter, in this case we obtain an ineffective well-ordering relation. In the sequel we propose that $\mathcal{U}$ is a Frechet filter.) This ultrapower $\Theta^I/\mathcal{U}$ is said to be a *proper nonstandard extension* of Θ and it is denoted by ${}^*\Theta$. There exist two groups of members of ${}^*\Theta$: (1) functions that are constant, e.g., $f(\alpha) = m \in \Theta$ for an infinite index subset $\{\alpha \in I\}$ (a constant function $[f = m]$ is denoted by ${}^*m$), (2) functions that are not constant. The set of all constant functions of ${}^*\Theta$ is called a *standard set* and is denoted by ${}^\sigma\Theta$.

The members of ${}^{\sigma}\Theta$ are called *standard*. It is readily seen that ${}^{\sigma}\Theta$ and Θ are isomorphic: ${}^{\sigma}\Theta \simeq \Theta$. If Θ was a number system, then members of ${}^{*}\Theta$ are called *hypernumbers*.

Assume that ${}^{*}\mathbf{Q}_{[0,1]} = \mathbf{Q}_{[0,1]}^{\mathbf{N}}/\mathcal{U}$ is a nonstandard extension of the subset $\mathbf{Q}_{[0,1]} = \mathbf{Q} \cap [0,1]$ of rational numbers and ${}^{\sigma}\mathbf{Q}_{[0,1]} \subset {}^{*}\mathbf{Q}_{[0,1]}$ is the subset of standard members. We can extend the usual order structure on $\mathbf{Q}_{[0,1]}$ to a partial order structure on ${}^{*}\mathbf{Q}_{[0,1]}$: (1) for rational numbers x, $y \in \mathbf{Q}_{[0,1]}$ we have $x \leq y$ in $\mathbf{Q}_{[0,1]}$ iff $[f] \leq [g]$ in ${}^{*}\mathbf{Q}_{[0,1]}$, where $\{\alpha \in \mathbf{N}\colon f(\alpha) = x\} \in \mathcal{U}$ and $\{\alpha \in \mathbf{N}\colon g(\alpha) = y\} \in \mathcal{U}$, i.e., f and g are constant functions such that $[f] = {}^{*}x$ and $[g] = {}^{*}y$, (2) each positive rational number ${}^{*}x \in {}^{\sigma}\mathbf{Q}_{[0,1]}$ is greater than any number $[f] \in {}^{*}\mathbf{Q}_{[0,1]} \backslash {}^{\sigma}\mathbf{Q}_{[0,1]}$, i.e., ${}^{*}x > [f]$ for any positive $x \in \mathbf{Q}_{[0,1]}$ and $[f] \in {}^{*}\mathbf{Q}_{[0,1]}$, where $[f]$ is not constant function.

These conditions have the following informal sense: (1) The sets ${}^{\sigma}\mathbf{Q}_{[0,1]}$ and $\mathbf{Q}_{[0,1]}$ have an isomorphic order structure. (2) The set ${}^{*}\mathbf{Q}_{[0,1]}$ contains actual infinities that are less than any positive rational number of ${}^{\sigma}\mathbf{Q}_{[0,1]}$.

Define this partial order structure on ${}^{*}\mathbf{Q}_{[0,1]}$ as follows:

$\mathcal{O}_{{}^{*}\mathbf{Q}}$ For any hyperrational numbers $[f], [g] \in {}^{*}\mathbf{Q}_{[0,1]}$, we set $[f] \leq [g]$ if $\{\alpha \in \mathbf{N}\colon f(\alpha) \leq g(\alpha)\} \in \mathcal{U}$. For any hyperrational numbers $[f], [g] \in {}^{*}\mathbf{Q}_{[0,1]}$, we set $[f] < [g]$ if $\{\alpha \in \mathbf{N}\colon f(\alpha) \leq g(\alpha)\} \in \mathcal{U}$ and $[f] \neq [g]$, i.e., $\{\alpha \in \mathbf{N}\colon f(\alpha) \neq g(\alpha)\} \in \mathcal{U}$. For any hyperrational numbers $[f], [g] \in {}^{*}\mathbf{Q}_{[0,1]}$, we set $[f] = [g]$ if $f \in [g]$.

Introduce two operations max, min in the partial order structure $\mathcal{O}_{{}^{*}\mathbf{Q}}$: (1) $\min([f],[g]) = [h]$ iff there exists $[h] \in {}^{*}\mathbf{Q}_{[0,1]}$ such that $\{\alpha \in \mathbf{N}\colon \min(f(\alpha),\ g(\alpha)) = h(\alpha)\} \in \mathcal{U}$; (2) $\max([f],[g]) = [h]$ iff there exists $[h] \in {}^{*}\mathbf{Q}_{[0,1]}$ such that $\{\alpha \in \mathbf{N}\colon \max(f(\alpha), g(\alpha)) = h(\alpha)\} \in \mathcal{U}$.

Note there exist the maximal number ${}^{*}1 \in {}^{*}\mathbf{Q}_{[0,1]}$ and the minimal number ${}^{*}0 \in {}^{*}\mathbf{Q}_{[0,1]}$ under condition $\mathcal{O}_{{}^{*}\mathbf{Q}}$.

Now define *hyperrational valued matrix logic* $\mathbf{M}_{{}^{*}\mathbf{Q}}$ as the ordered system $\langle V_{{}^{*}\mathbf{Q}}, \neg, \supset, \vee, \wedge, \widetilde{\exists}, \widetilde{\forall}, \{{}^{*}1\}\rangle$, where (1) $V_{{}^{*}\mathbf{Q}} = {}^{*}\mathbf{Q}_{[0,1]}$ is the subset of hyperrational numbers, (2) for all $x \in V_{{}^{*}\mathbf{Q}}$, $\neg x = {}^{*}1 - x$, (3) for all $x, y \in V_{{}^{*}\mathbf{Q}}$, $x \supset y = {}^{*}1 - \max(x,y) + y$, (4) for all $x, y \in V_{{}^{*}\mathbf{Q}}$, $x \vee y = (x \supset y) \supset y = \max(x,y)$, (5) for all $x, y \in V_{{}^{*}\mathbf{Q}}$, $x \wedge y = \neg(\neg x \vee \neg y) = \min(x,y)$, (6) for a subset $M \subseteq V_{{}^{*}\mathbf{Q}}$, $\widetilde{\exists}(M) = \max(M)$, where $\max(M)$ is a maximal element of M, (7) for a subset $M \subseteq V_{{}^{*}\mathbf{Q}}$, $\widetilde{\forall}(M) = \min(M)$, where $\min(M)$ is a minimal element of M, (8) $\{{}^{*}1\}$ is the set of designated truth values.

The truth value ${}^{*}0 \in V_{{}^{*}\mathbf{Q}}$ is false, the truth value ${}^{*}1 \in V_{{}^{*}\mathbf{Q}}$ is true, and other truth values $x \in V_{{}^{*}\mathbf{Q}}$ are neutral.

If we replace the set $\mathbf{Q}_{[0,1]}$ by $\mathbf{R}_{[0,1]}$ and the set ${}^{*}\mathbf{Q}_{[0,1]}$ by ${}^{*}\mathbf{R}_{[0,1]}$ in all above definitions, then we obtain *hyperreal valued matrix logic* $\mathbf{M}_{{}^{*}\mathbf{R}}$.

3. Non-Archimedean Valued Propositional Logical Language

An infinite-order propositional logical language $\mathcal{L}_V^\infty$ consists of the following symbols: (1) first-order propositional formulas: φ, ϕ, ψ, ... of **n**-valued Łukasiewicz's logic Ł_V, where $|V| = \mathbf{n}$; (2) logical symbols: (i) various order propositional connectives of arity n_j: $F_0^{n_0}$, $F_1^{n_1}$, ..., $F_r^{n_r}$, which are built by a superposition of negation $\neg$ and implication $\supset$, (ii) vertical quantifiers of various order Q_1, Q_2,...,Q_{i-1},... such that an i-order quantifier has the lower index $i-1$; (3) auxiliary symbols: (,), and , (comma).

If $V = \mathbf{Q}_{[0,1]}$, then we have $\aleph_0$ universal vertical quantifiers at the level i and $\aleph_0$ existential vertical quantifiers at the level i: $\forall_i^{y\in\mathbf{Q}_{[0,1]}}$, ..., $\forall_i^{y'\in\mathbf{Q}_{[0,1]}}$, ..., $\exists_i^{y\in\mathbf{Q}_{[0,1]}}$, ..., $\exists_i^{y'\in\mathbf{Q}_{[0,1]}}$, ... If $V = \mathbf{R}_{[0,1]}$, then we have $2^{\aleph_0}$ universal vertical quantifiers at the level i and $2^{\aleph_0}$ existential vertical quantifiers at the level i: $\forall_i^{y\in\mathbf{R}_{[0,1]}}$, ..., $\forall_i^{y'\in\mathbf{R}_{[0,1]}}$, ..., $\exists_i^{y\in\mathbf{R}_{[0,1]}}$, ..., $\exists_i^{y'\in\mathbf{R}_{[0,1]}}$, ...

Well-formed formulas of $\mathcal{L}_V^\infty$ are inductively defined as follows: (1) If φ is a first-order propositional formula of **n**-valued Łukasiewicz's logic Ł_V, where $|V| = \mathbf{n}$, and $\mathrm{Q}_1, \mathrm{Q}_2, \ldots, \mathrm{Q}_{i-1}$ are a finite sequence of vertical quantifiers, then

$$\mathrm{Q}_{i-1}(\ldots(\mathrm{Q}_1\varphi)\ldots)\ldots\mathrm{Q}_1\varphi(\varphi)$$

is an i-order formula denoted sometimes by φ_i or by $\mathrm{Q}_{i-1}\varphi(\varphi_{i-1})$ to emphasize what is the least quantifier in φ_i. It is called atomic or an atom. Its outermost logical symbols are $\mathrm{Q}_1, \mathrm{Q}_2, \ldots, \mathrm{Q}_{i-1}$. If the first of these quantifiers is $\forall_1^{y_1\in V}$, then the other are also universal with the upper indices that are equal to y_1. If the first of these quantifiers is $\exists_1^{y_1\in V}$, then the other are also existential with the upper indices that are not possibly equal to y_1. (2) If $\varphi_i, \ldots, \psi_i$ are formulas of i-order and F^n is a propositional connective of arity n, then $F^n(\varphi_i, \ldots, \psi_i)$ is an i-order formula with outermost logical symbol F^n. (3) If φ is a first-order propositional formula and $\mathrm{Q}_1, \mathrm{Q}_2, \ldots, \mathrm{Q}_{i-1}, \ldots$ are an infinite sequence of vertical quantifiers, then

$$\ldots\mathrm{Q}_{i-1}(\ldots(\mathrm{Q}_1\varphi)\ldots)\ldots\mathrm{Q}_1\varphi(\varphi)$$

is an infinite-order formula denoted sometimes by φ_∞. It is called atomic or an atom. Its outermost logical symbols are $\mathrm{Q}_1, \mathrm{Q}_2, \ldots, \mathrm{Q}_{i-1}, \ldots$ If the first of these quantifiers is $\forall_1^{y_1\in V}$, then the other are also universal with the upper indices that are equal to y_1. If the first of these quantifiers is $\exists_1^{y_1\in V}$, then the other are also existential with the upper indices that are not possibly equal to y_1. (4) If $\varphi_\infty, \ldots, \psi_\infty$ are formulas of infinite order and F^n is a propositional connective of arity n, then $F^n(\varphi_\infty, \ldots, \psi_\infty)$ is a formula with outermost logical symbol F^n and this formula is an infinite-order formula.

Consider the set $^*V$ of all equivalence classes $[f]$ under a Frechet ultrafilter $\mathcal{U}$ such that $f\colon \mathbf{N} \to V$. Recall that for each $i \in V$, $^*i = [f = i]$, i.e., it is a constant function. Every element of $^*V$ has the form of an infinite tuple $[f] = \langle y_0, y_1, \ldots \rangle$.

Let 1 be the designated truth value of **n**-valued Łukasiewicz's logic Ł$_V$, where $|V| = \mathbf{n}$. An i-order truth assignment is a function $v_i(\cdot)$ whose domain is the set of all i-order formulas of $\mathcal{L}_V^\infty$ and whose range is the set $^*V$ of truth values such that:

1. For any first-order propositional formula φ_1, $v_1(\varphi_1)$ is a truth assignment of **n**-valued Łukasiewicz's logic.
2. For any first-order propositional formula φ_1, $v_i(\varphi_1) = \langle \underbrace{y_1, y_1, \ldots y_1}_{i} \rangle$ iff $v_1(\varphi_1) = y_1$.
3. For any i-order atomic propositional formula $\forall_{i-1}^{y_{i-1} \in V} \varphi(\varphi_{i-1})$, (1) if $v_{i-1}(\forall_{i-2}^{y_{i-2} \in V} \varphi(\varphi_{i-2})) = \langle \underbrace{y_{i-1}, \ldots, y_{i-1}}_{i-1} \rangle$ for all valuations, then
$$v_i(\forall_{i-1}^{y_{i-1} \in V} \varphi(\varphi_{i-1})) = \langle \underbrace{y_{i-1}, \ldots, y_{i-1}}_{i} \rangle;$$
(2) if (i) $v_{i-1}(\forall_{i-2}^{y_{i-2} \in V} \varphi(\varphi_{i-2})) \neq \langle \underbrace{y_{i-1}, \ldots, y_{i-1}}_{i-1} \rangle$ for some valuations, (ii) $v_{i-1}(\forall_{i-2}^{y_{i-2} \in V} \varphi(\varphi_{i-2})) = \langle y'_1, \ldots, y'_{i-1} \rangle$ and $v_1(\varphi) = y'_0$ for some valuations, then $v_i(\forall_{i-1}^{y_{i-1} \in V} \varphi(\varphi_{i-1})) = \langle y'_0, y'_1 \ldots, y'_{i-1} \rangle$.
4. For any i-order atomic propositional formula
$$\varphi_i = \exists_{i-1}^{y_{i-1} \in V} (\ldots (\exists_1^{y_1 \in V} \varphi) \ldots) \ldots \exists_1^{y_1 \in V} \varphi(\varphi),$$
(1) if (i) $v_{i-1}(\varphi_{i-1}) = \langle y_1, \ldots, y_{i-1} \rangle$ for some valuations and (ii) $v_1(\varphi_1) = y_0$ for some valuations, then $v_i(\varphi_i) = \langle y_0, y_1, \ldots, y_{i-1} \rangle$; (2) if (i) $v_{i-1}(\varphi_{i-1}) \neq \langle y_1, \ldots, y_{i-1} \rangle$ for all valuations, (ii) $v_{i-1}(\varphi_{i-1}) = \langle y'_1, \ldots, y'_{i-1} \rangle$ and $v_1(\varphi) = y'_0$ for some valuations, then $v_i(\varphi_i) = \langle y'_0, y'_1 \ldots, y'_{i-1} \rangle$.
5. For any formula φ_i, $v_i(\neg\varphi_i) = \langle 1 - y_0, 1 - y_1, \ldots, 1 - y_{i-1} \rangle$, where $v_i(\varphi_i) = \langle y_0, y_1, \ldots, y_{i-1} \rangle$.
6. For any formulas φ_i and ψ_i, $v_i(\varphi_i \supset \psi_i) = \langle (1 - \max(x_0, y_0) + y_0), (1 - \max(x_1, y_1) + y_1), \ldots, (1 - \max(x_{i-1}, y_{i-1}) + y_{i-1}) \rangle$, where $v_i(\varphi_i) = \langle x_0, x_1, \ldots, x_{i-1} \rangle$ and $v_i(\psi_i) = \langle y_0, y_1, \ldots, y_{i-1} \rangle$.

7. For any formulas φ_i and ψ_i, $v_i(\varphi_i \vee \psi_i) = \langle \max(x_0, y_0), \max(x_1, y_1), \ldots, \max(x_{i-1}, y_{i-1})\rangle$, where $v_i(\varphi_i) = \langle x_0, x_1, \ldots, x_{i-1}\rangle$ and $v_i(\psi_i) = \langle y_0, y_1, \ldots, y_{i-1}\rangle$.
8. For any formulas φ_i and ψ_i, $v_i(\varphi_i \wedge \psi_i) = \langle \min(x_0, y_0), \min(x_1, y_1), \ldots, \min(x_{i-1}, y_{i-1})\rangle$, where $v_i(\varphi_i) = \langle x_0, x_1, \ldots, x_{i-1}\rangle$ and $v_i(\psi_i) = \langle y_0, y_1, \ldots, y_{i-1}\rangle$.

An infinite-order truth assignment is a function $v_\infty[\cdot]$ whose domain is the set of all infinite-order formulas of $\mathcal{L}_V^\infty$ and whose range is the set ${}^*V$ of truth values such that:

1. For any first-order propositional formula φ_1, $v_1(\varphi_1)$ is a truth assignment of **n**-valued Łukasiewicz's logic.
2. For any first-order propositional formula φ_1, $v_\infty[\varphi_1] = {}^*y_1 = \langle y_1, y_1, \ldots\rangle$ iff $v_1(\varphi_1) = y_1$.
3. For any infinite-order atomic propositional formula

$$\varphi_\infty = \ldots \forall_{i-1}^{y_{i-1}\in V}(\ldots(\forall_1^{y_1\in V}\varphi)\ldots)\ldots\forall_1^{y_1\in V}\varphi(\varphi),$$

(1) if $v_\infty[\varphi_\infty] = {}^*y_1 = \langle y_1, \ldots, y_1, \ldots\rangle$ for all valuations, then $v_\infty[\varphi_\infty] = {}^*y_1$; (2) if (i) $v_\infty[\varphi_\infty] \neq {}^*y_1$ for some valuations and (ii) $v_\infty[\varphi_\infty] = [f]$ for some valuations, then $v_\infty[\varphi_\infty] = [f]$.

4. For any infinite-order atomic propositional formula

$$\varphi_\infty = \ldots \exists_{i-1}^{y_{i-1}\in V}(\ldots(\exists_1^{y_1\in V}\varphi)\ldots)\ldots\exists_1^{y_1\in V}\varphi(\varphi),$$

(1) if $v_\infty[\varphi_\infty] = [f] = \langle y_0, y_1, \ldots, y_{i-1}, \ldots\rangle$ for some valuations, then $v_\infty[\varphi_\infty] = [f]$; (2) if $v_\infty[\varphi_\infty] \neq [f]$ for all valuations and $v_\infty[\varphi_\infty] = [f']$ for some valuations, then $v_\infty[\varphi_\infty] = [f']$.

5. For any formula φ_∞, $v_\infty[\neg\varphi_\infty] = {}^*1 - v_\infty[\varphi_\infty]$.
6. For any formulas φ_∞ and ψ_∞, $v_\infty[\varphi_\infty \supset \psi_\infty] = {}^*1 - \max(v_\infty[\varphi_\infty], v_\infty[\psi_\infty]) + v_\infty[\psi_\infty]$.
7. For any formulas φ_∞ and ψ_∞, $v_\infty[\varphi_\infty \vee \psi_\infty] = \max(v_\infty[\varphi_\infty], v_\infty[\psi_\infty])$.
8. For any formulas φ_∞ and ψ_∞, $v_\infty[\varphi_\infty \wedge \psi_\infty] = \min(v_\infty[\varphi_\infty], v_\infty[\psi_\infty])$.

Note that the function $v_\infty[\cdot]$ is an infinite sequence of functions $v_i(\cdot)$.

Suppose that **n**-valued Łukasiewicz's logic Ł_V, where $|V| = \mathbf{n}$, is truth-functionally complete thanks to Słupecki's operators $\mathrm{T}_k(\varphi) : v(\varphi) \mapsto k \in V/\{0, 1\}$, where $v(\varphi)$ is any truth valuation of $\varphi \in \text{Ł}_V$. In this case some formulas of $\mathcal{L}_V^\infty$ have truth values of the form ${}^*k$. Without Słupecki's operators, only ^{*0}, ^{*1} are constant functions that can be truth values for formulas of $\mathcal{L}_V^\infty$.

4. Non-Archimedean Valued Logic

A hyperrational valued logic denoted by $\text{Ł}_{{}^*\mathbf{Q}_{[0,1]}}$ (resp. a hyperreal valued logic denoted by $\text{Ł}_{{}^*\mathbf{R}_{[0,1]}}$) is built on the basis of language $\mathcal{L}^\infty_{\mathbf{Q}_{[0,1]}}$ (resp. $\mathcal{L}^\infty_{\mathbf{R}_{[0,1]}}$).

The following properties of higher-order formulas are evident without proofs:

- $v_i(\forall_{i-1}^{y_{i-1}\in V}\varphi(\varphi_{i-1})) = \langle y_{i-1},\ldots,y_{i-1}\rangle$:= "a formula φ has the truth value y_{i-1} for all truth valuations, a formula φ_1 has the truth value $\langle y_{i-1}, y_{i-1}\rangle$ for all truth valuations, etc.";
- $v_i(\exists_{i-1}^{y_{i-1}\in V}(\ldots(\exists_1^{y_1\in V}\varphi)\ldots)\ldots\exists_1^{y_1\in V}\varphi(\varphi)) = \langle y_0,\ldots,y_{i-1}\rangle$:= "a formula φ has the truth value y_0 for some truth valuations, a formula φ_1 has the truth value $\langle y_0, y_1\rangle$ for some truth valuations, etc.";
- a formula $\forall_{i-1}^{y_{i-1}\in V}\varphi(\varphi_{i-1})$ or $\exists_{i-1}^{y_{i-1}\in V}(\ldots(\exists_1^{y_1\in V}\varphi)\ldots)\ldots\exists_1^{y_1\in V}\varphi(\varphi)$ is a tautology iff a formula φ is a tautology;
- a formula $\forall_{i-1}^{y_{i-1}\in V}\varphi(\varphi_{i-1})$ or $\exists_{i-1}^{y_{i-1}\in V}(\ldots(\exists_1^{y_1\in V}\varphi)\ldots)\ldots\exists_1^{y_1\in V}\varphi(\varphi)$ is satisfiable iff a formula φ is satisfiable.

These properties allow setting a non-Archimedean calculus. Consequently, we can extend the Hilbert's type calculus of infinite-valued Łukasiewicz's logic Ł_∞ for the non-Archimedean case. *The axioms of the Hilbert's type calculus for* $\text{Ł}_{{}^*\mathbf{Q}_{[0,1]}}$ (resp. for $\text{Ł}_{{}^*\mathbf{R}_{[0,1]}}$) are as follows.

$$(\varphi_i \supset \phi_i) \supset ((\phi_i \supset \psi_i) \supset (\varphi_i \supset \psi_i)), \tag{5}$$

$$\varphi_i \supset (\phi_i \supset \varphi_i), \tag{6}$$

$$((\varphi_i \supset \phi_i) \supset \phi_i) \supset ((\phi_i \supset \varphi_i) \supset \varphi_i), \tag{7}$$

$$(\neg\varphi_i \supset \neg\phi_i) \supset (\phi_i \supset \varphi_i), \tag{8}$$

$$(\varphi_\infty \supset \phi_\infty) \supset ((\phi_\infty \supset \psi_\infty) \supset (\varphi_\infty \supset \psi_\infty)), \tag{9}$$

$$\varphi_\infty \supset (\phi_\infty \supset \varphi_\infty), \tag{10}$$

$$((\varphi_\infty \supset \phi_\infty) \supset \phi_\infty) \supset ((\phi_\infty \supset \varphi_\infty) \supset \varphi_\infty), \tag{11}$$

$$(\neg\varphi_\infty \supset \neg\phi_\infty) \supset (\phi_\infty \supset \varphi_\infty), \tag{12}$$

$$(\neg(\psi_1 \equiv \psi_\infty) \wedge \neg(\phi_1 \equiv \bot)) \supset (\psi_\infty \supset \phi_1), \tag{13}$$

where $\bot$ is a contradiction.

Axioms (5)–(8) are called *horizontal*. Axiom (13) is called *vertical*. Axioms (9)–(12) are called *axioms of infinite length*. Notice that the upper indices of vertical quantifiers belong to the countable set $\mathbf{Q}_{[0,1]}$ in $Ł_{^*\mathbf{Q}_{[0,1]}}$ and belong to the uncountable set $\mathbf{R}_{[0,1]}$ in $Ł_{^*\mathbf{R}_{[0,1]}}$.

Inference rules are as follows: (1) *modus ponens* (if two formulas φ_i (resp. φ_∞) and $\varphi_i \supset \psi_i$ (resp. $\varphi_\infty \supset \psi_\infty$) hold, then we deduce a formula ψ_i (resp. ψ_∞)); (2) *substitution rule*: we can substitute a formula of the same order for an atomic formula.

The non-Archimedean valued Hilbert's type calculus has all the deductive and semantic properties of the Hilbert's type calculus of infinite-valued Łukasiewicz's logic $Ł_\infty$. At the same time the truth concept of $Ł_\infty$ can be syntactically expressed by means of non-Archimedean valued logic. Thus, non-Archimedean valued logic is more formally expressive than $Ł_\infty$.

5. Conclusions

In this paper I considered some perspectives of transfinite foundations of mathematics, namely I built the non-Archimedean valued propositional logic. This new metalanguage also has a lot of practical applications, in particular it can be applied in non-Kolmogorovian probability theory and in soft computing (see [6] and [8]).

References

[1] Cantor, G. Gesammelte Abhandlungen mathematischen und philosophischen Inhalts (hrsg. Ernst Zermelo). Berlin Heidelberg New York 1980.

[2] Cohen, P. J. Set Theory and the Continuum Hypothesis, New York, 1966.

[3] Gödel, K. The consistency of the Axiom of Choice and of the Generalized Continuum-Hypothesis with the Axioms of the Set Theory, [in:] Annals of Math. Studies, 3 (1940).

[4] D. Hilbert and P. Bernays, Grundlagen der Mathematik, Vol. 1. Zweite Auflage. Springer-Verlag, 1968.

[5] Hilbert, D. Mathematical Problems, [in:] Bulletin of the American Mathematical Society 8 (1902), pp. 437–479.

[6] A. Khrennikov and A. Schumann, Logical Approach to p-adic Probabilities, [in:] Bulletin of the Section of Logic Volume 35/1 (2006), pp. 49-57.

[7] Matiyasevich, Yu. Hilbert's Tenth Problem. MIT Press, Cambridge, Massachusetts, 1993.

[8] Schumann, A. DSm Models and non-Archimedean Reasoning, [in:] F. Smarandache, J. Dezert (Editors), Advances and Applications of DSmT (Collected works), Vol. 2, American Research Press, Rehoboth. 2006, pp. 183–204.

[9] Schumann, A. Non-Archimedean Fuzzy Reasoning, *Fuzzy Systems and Knowledge Discovery (FSKD'07)*, IEEE Press (2007)

[10] Schumann, A. Non-Archimedean Valued Predicate Logic, *Bulletin of the Section of Logic*, 36/1 (2007) pp. 1–12.

[11] Schumann, A. Non-Archimedean Valued Sequent Logic, *Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'06)*, IEEE Press (2006) pp. 89–92.

[12] Schumann, A. p-Adic Multiple-Validity and p-Adic Valued Logical Calculi, *Journal of Multiple-Valued Logic and Soft Computing*, 13, 1–2 (2007) pp. 29–60.

Katarzyna Zbieć
University of Białystok

A LINEAR APPROXIMATION METHOD IN PREDICTION OF CHAOTIC TIME SERIES

This paper presents a method to make predictions regarding the chaotic time series, known as a linear approximation method. After embedding a time series in a phase space, it is necessary to replace nonlinear mapping using a local approximation. This allows making a short-term prediction of future behaviour, using information based only on past values. The effectiveness of this method is demonstrated by applying it to the prediction of share prices.

Introduction

The basic problem of scientific investigation is forecasting – How can we predict the future, given the past? The behaviour of periodical structure we can predict in infinity, but chaotic structure is predictable in the short term only. It is connected with a basic property – the sensitivity on initial conditions.

It consists in that very similar initial conditions sometimes give very different structure's behaviour. The reason for this is that we can establish initial conditions with finished exactitude, but miscalculations grow exponentially. (Therefore forecasting such structures is sensible only in short intervals). This means, that when we want to predict the behaviour of such a structure in any moment, we would dispose of the data entrance passed with infinite exactitude as well as execute all calculation with finite accuracy. Otherwise, small mistakes in setting initial values as well as miscalculations (e.g. the mistakes of roundings) grow in an exponential way. This means that the evolution of such systems is very complex and virtually unpredictable in the long-term.

There exist different methods of forecasting a chaotic time series. This article presents a method of the linear approximation applied to a short-term prediction regarding the share prices on the Warsaw Stock Exchange. The

ISBN 978–83–7431–158–8 ISSN 0860-150X

time series of share prices is a type of deterministic series and can behave chaotically [6, 1997].

A Linear Approximation Method

If we want to model nonlinear systems:

$$x_{i+1} = f(x_i), \quad i = 0, 1, ... \tag{1}$$

we might fit the data to combinations of nonlinear functions. This however is a very complicated method. We can therefore apply a method based on an approximation of behaviour in the midst of any point on an attractor[1] by a unique local function. Then, the evolution on an attractor is represented by the set of such functions. Functions are linear at each point. This means:

$$x_{i+1} = \mathbf{a} + \mathbf{b}x_i, \tag{2}$$

where matrix **b** and vector **a** are defined for every point. A class of the local map creates a global nonlinear map [1, 1993].

Suppose the time series folded from T observations: $x_1, x_2, \ldots, x_T$. We can establish a dimension of embedding m and make the reconstruction of the phase space[2]. In such a space we get the following set of vectors:

$$x_i^m = (x_i, x_{i-1}, \ldots, x_{i-m+1}), \quad i = m, m+1, \ldots, T. \tag{3}$$

We should predict the value of the time series with number $P - x_P$, which is the first component of a point: $x_P^m = (x_P, x_{P-1}, \ldots, x_{P-m+1})$. In the neighbourhood of this point we can estimate the following equation parameters:

$$x_P^m = \mathbf{a} + \mathbf{b}x_{P-1}^m + \varepsilon_P^m, \tag{4}$$

a is mx1 dimensional vector of parameters,
b os mxm dimensional matrix of parameters,
ε_P^m is mx1 dimensional vector of errors.

We assume that the function is linear.

[1] Attractor is a set to which the system evolves after a long enough time. For a set to be an attractor, trajectories which get close enough to the attractor must remain close even if slightly disturbed.

[2] Phase space is the space in which all possible states of a system are represented, with each possible state of the system corresponding to one unique point in the phase space. Dimension of the phase space depends on quantity of variables necessary to the description of the system. The reconstruction of phase space consists in reproducing the multidimensional attractor based on the one-dimensional time series.

Illustrated bellow is the matrix form:

$$\begin{bmatrix} x_P \\ x_{P-1} \\ \vdots \\ x_{P-m+1} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mm} \end{bmatrix} \cdot \begin{bmatrix} x_{P-1} \\ x_{P-2} \\ \vdots \\ x_{P-m} \end{bmatrix} + \begin{bmatrix} \varepsilon_P \\ \varepsilon_{P-1} \\ \vdots \\ \varepsilon_{P-m+1} \end{bmatrix}. \tag{5}$$

Therefore we should estimate only the first component a_1 of vector **a** and the first row of matrix **b**. We need to solve the following equation:

$$x_P = a_1 + \sum_{j=1}^{m} b_{1j} x_{P-j} + \varepsilon_P. \tag{6}$$

To calculate the parameters we use an approximation by k nearest neighbour x_{P-1}^m point.

The estimation of the parameters relating to equation 6 proceeds as follows [3, 1989]:

- We define components of k points in the m-dimensional reconstructed phase space: $x_{n_{1m}}, x_{n_{2m}}, \ldots, x_{n_{km}}$, where $k > m$. They are the nearest, in the sense the of the Euclidean metric, neighbours of the point $(x_{P-1}, x_{P-2}, \ldots, x_{P-m})$. We should consider $k \geq 2(m+1)$ the closest neighbours.
- We mark the first components of the nearest neighbours: $x_{n_1}, x_{n_2}, \ldots, x_{n_k}$, and then corresponding to them, the following points in time series: $x_{n_1+1}, x_{n_2+1}, \ldots, x_{n_k+1}$.
- As a result we form a system of k equations with $m+1$ unknowns:

$$x_{n_i+1} = a_1 + \sum_{j=1}^{m} b_{1i} x_{n_i+1-j} + \varepsilon_{n_i+1}, \quad i = 1, 2, \ldots, k. \tag{7}$$

- Parameters a_1, b_{1j} $(j = 1, ..., m)$ are estimated by the least squares method.
- Using equation 6 we predict the value of element x_P:

$$\hat{x_P} = \hat{a_1} + \sum_{j=1}^{m} \hat{b_{1j}} x_{P-j}. \tag{8}$$

Thus for any x_t there is a marked predicted value $\hat{x_t}$. It can precisely determine an absolute error $\varepsilon_t = x_t - \hat{x_t}$. Relative error ψ_t it is the percentage deviation of the obtained value from the real value.

The Share Prices on the Warsaw Stock Exchange Prediction

The method of linear approximation can be used for forecasting share prices. It has been proven was that the time series of share prices are generated by a deterministic system, showing a tendency for chaotic behaviours.

The time series, we studied, consisted from around 2500 observations. The method will be applied to an example time series of share prices for the Żywiec company. We take $t = 2001, \ldots, 2300$.

Firstly, we assume the dimension of embedding $m = 2$, and its nearest neighbours' number $k = 8$ (Fig. 1). For an exact analysis of the method we specified the fragment of time series for $t = 2110, \ldots, 2120$. The last point of this fragment is x_{2120}. The nearest eight neighbours of point x^2_{2119} in reconstructed phase space for $m = 2$ are points: x^2_{2111}, x^2_{2112}, x^2_{2113}, x^2_{2114}, x^2_{2115}, x^2_{2116}, x^2_{2117}, x^2_{2118}. The consequents are: x^2_{2112}, x^2_{2113}, x^2_{2114}, x^2_{2115}, x^2_{2116}, x^2_{2117}, x^2_{2118}, x^2_{2119}. We ascertained the following matrix equation:

$$\begin{bmatrix} x_{2112} \\ x_{2113} \\ x_{2114} \\ x_{2115} \\ x_{2116} \\ x_{2117} \\ x_{2118} \\ x_{2119} \end{bmatrix} = \begin{bmatrix} 1 & x_{2111} & x_{2110} \\ 1 & x_{2112} & x_{2111} \\ 1 & x_{2113} & x_{2112} \\ 1 & x_{2114} & x_{2113} \\ 1 & x_{2115} & x_{2114} \\ 1 & x_{2116} & x_{2115} \\ 1 & x_{2117} & x_{2116} \\ 1 & x_{2118} & x_{2117} \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ b_{11} \\ b_{12} \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \\ \varepsilon_8 \end{bmatrix}. \tag{9}$$

As a result of the estimation parameters we get the equation:

$$x_{2\hat{1}20} = 150{,}3669467 + 0{,}18326264 \cdot x_{2119} - 0{,}040650558 \cdot x_{2118}. \tag{10}$$

The predicted value comes to 175,23, with an error $\varepsilon_{2120} = -0,73$. The error amounts to $\psi_{2120} = 0,42\%$ of the real value. In most cases the mistakes did not exceed 2%, so this is quite an effective method (Fig. 2).

As a rule, experimental results show that an increase to the 12 the nearest neighbours numbers improves the effectiveness of a prediction (Fig. 3).

As we can see from above, the method of linear approximation gives enough first-rate results in predicting of share prices. However, we must bear in mind, that it relates only to short-term forecasting.

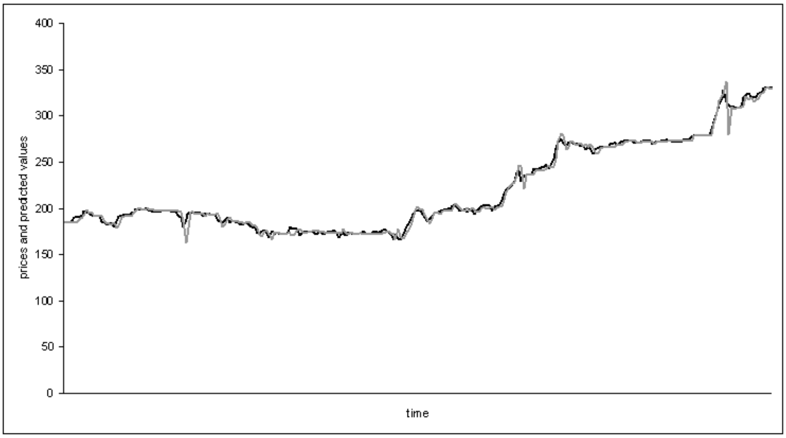


Figure 1. Comparison of the Żywiec's share prices with predicted values for $t = 2001 - 2300, m = 2, k = 8$

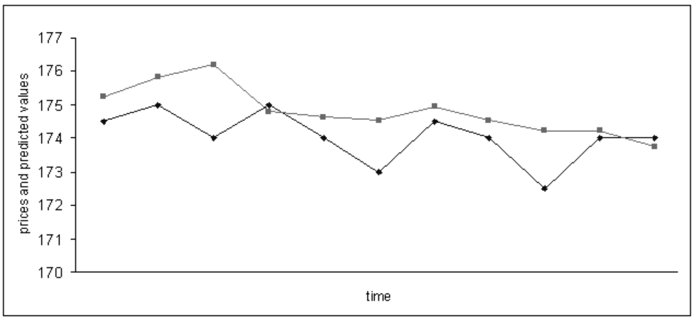


Figure 2. Comparison of the Żywiec share prices with predicted values for $t = 2110 - 2120, m = 2, k = 8$

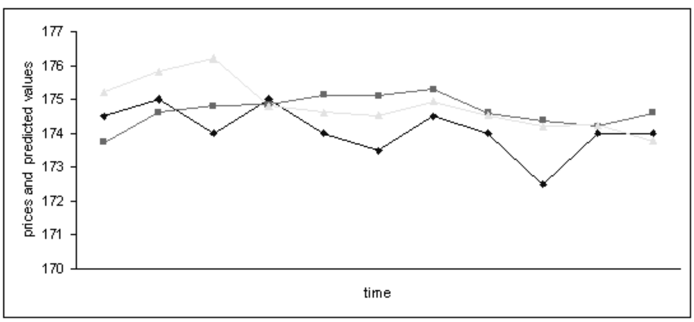


Figure 3. Comparison of the Żywiec share prices with predicted values for $t = 2110 - 2120, m = 2, k = 8$ and $k = 12$

References

[1] Abarbanel H. D. I. and Brown R. and Sidorowich J. and Tsimiring T. S. The analysis of observed chaotic data in physical systems. *Reviews of Modern Physics*, 65:1331–92, 1993.

[2] Baker G. L. and Gollub J. P. *Wstęp do dynamiki układów chaotycznych.* Wydawnictwo Naukowe PWN, Warszawa, 1998.

[3] Casdagli M. Nonlinear prediction of chaotic time series. *Physica D*, 35:335–56, 1989.

[4] Farmer J. D. and Sidorowich J. J. *Predicting Chaotic Time Series*, volume 59. 1987.

[5] Kacprzak D. O metodzie przewidywania stanów chaotycznych na przykładzie cen akcji z giełdy nowojorskiej NYSE. In *Energia w nauce i technice*, Białystok–Suwałki, 2005. Wydawnictwo Politechniki Białostockiej.

[6] Peters E. E. *Teoria chaosu a rynki kapitałowe.* Wig Press, Warszawa, 1997.

STUDIES IN LOGIC, GRAMMAR AND RHETORIC 11 (24) 2007

Vitaly. I. Levin
Penza State Technological Academy, Penza, Russia

BASIC CONCEPTS OF CONTINUOUS LOGIC

In this paper, a general description of a continuous (-valued) logic is given and some problems and particulars of their solutions are discussed. Firstly, we define algebra of continuous logic and enumerate its basic unary, binary and ternary functions. All laws of continuous logic are compared to laws of discrete binary logic. We discuss how to enumerate all the functions of continuous logic with a specified number of variables and how to represent such functions in a standard form. Procedures of minimization regarding continuous logical functions and their decomposition into functions with less clarity are exploited. The procedures are compared to their counterparts from binary logic. We also tackle problems of the analysis and synthesis of continuous logical functions, and show that the problem of synthesis may not have a solution. Basics of differential and integral calculus are applied to continuous-valued logic. We demonstrate that any continuous logical function has the points where a derivative does not exist. To conclude, we briefly discuss a problem of incompleteness regarding continuous logic, application of continuous logic in mathematics, engineering and economy, give examples, draw a perspective of further development and supply an extensive bibliography of Russian works in the field.

1. Introduction

A continuous logic (CL) is a natural generalization of a discrete logic (DL). A few laws of DL take place in CL as well. A general structure of CL differs from that of DL; thus, for example an operation of negation regarding CL cannot be defined in terms of addition (as it is done in binary logic) in a consistent way. CL now forms an independent scientific discipline where theoretical and applied key points lie in such diverse fields as:

- mathematics (approximation of functions, geometry, theory of sets, theory of numbers, interval analysis);
- engineering (design of electrical circuits, synthesis of functional generators and analogue-discrete transformers, design and analysis of analogue and digital devices, diagnostics and maintenance service);
- systems (theory of service systems, pattern recognition and analysis of scenes, decision making, information processing, synchronization);

ISBN 978–83–7431–158–8 ISSN 0860-150X

- economy (discrete optimization, schedule theory, simulation of economic systems),
- biology (simulation of neuron systems at all levels of representation),
- sociology (simulation of the dynamics of collective behaviour);
- politology (simulations of dynamics of a society),
- history (simulation of the streams of historical events).

The new results in CL can be obtained with the help of several straightforward techniques:

- calculation of the values of logical expressions;
- equivalent conversion of logical expressions;
- unification of individual logical expressions;
- partition of common logical expressions onto individual ones.

We can also embed the algebra of CL in a more general distributive structure; in this situation all methods of the theory of structures are useful.

A range of the main problems of CL includes the following categories:

1. enumeration of all CL functions for the given number of arguments,
2. representation of CL functions in a standard form (including unambiguous representation),
3. selection of elementary CL functions;
4. minimization and decomposition of CL functions,
5. analysis and synthesis of CL functions;
6. solution of the equations and inequalities of CL,
7. differentiation and integration of CL functions,
8. verification of completeness regarding the system of CL functions.

Problems 1–4 are similar in formulation and, partially in the methods of solution, to the appropriate DL tasks. Problems 5–8 belong exclusively to CL.

2. General Description of Continuous Logic

Let $C = [A, B]$ be a closed interval such that $M = (A + B)/2$. Basic operations of CL are defined on as follows:

$$\begin{aligned} a \vee b &= \max(a, b) && \text{(Disjunction)}, \\ a \wedge b &= \min(a, b) && \text{(Conjunction)}, \\ \bar{a} &= 2M - a && \text{(Negation)}. \end{aligned} \tag{1}$$

The sign $\wedge$ is usually omitted.

Sometimes the following operations are used as basic ones:

- inclusion $a \supset b = (\bar{a} + b) \wedge B$,
- implication $a \to b = \bar{a} \vee b$,
- equivalence $(a \equiv b) = (a \vee \bar{b})(\bar{a} \vee b)$,
- non-equivalence $(a \not\equiv b) = (a\bar{b} \vee \bar{a}b$,
- Sheffer $a|b = \overline{ab}$,
- Webb $a \downarrow b = \overline{a \vee b}$,
- contradiction $(a \neq a) = a\bar{a}$,
- tautology $(a \equiv a) = a \vee \bar{a}$,
- prohibition $a \overline{\to} b) = a\bar{b}$.

An algebraic system of the supporting set C and the basic operations is called an algebra of CL. A CL function is a function $C^n \to C$, which is represented by the superposition of a finite number of basic operations of the CL algebra with the arguments $x_1, \ldots, x_n \in C$. The number of CL functions is finite, though the set of all functions of the sort $C^n \to C$ is infinite.

A quasi-Boolean algebra:

$$\Delta = (C; \vee, \wedge, ^{-}) \tag{2}$$

is one of the most developed and investigated algebras of CL. The functions of algebra (2) are usually tabulated. One can easily transform tabular representation into an analytical one using a method of union. A reverse transition from analytical to tabular representation is carried out by a method of partition.

The number $P(n)$ of n-ary CL functions in quasi-Boolean algebra grows quite fast when n increases: $P(0) = 2$, $P(1) = 6$, $P(2) = 84$, $P(3) = 43918$. For $n = 0$ the functions are constant:

$$y_0 = A, \quad y_1 = B. \tag{3}$$

For $n = 1$ there are constants y_0, y_1 and 4 functions, essentially depending on argument x:

$$y_2 = x, \quad y_3 = \bar{x}, \quad y_4 = x \vee \bar{x}, \quad y_5 = x\bar{x}. \tag{4}$$

For $n = 2$ there are constants (3), 8 functions (4), depending on one argument (x_1 or x_2), and 10 functions, depending on two arguments:

$$\begin{aligned} &y_{10} = x_1 \vee x_2,\ y_{11} = x_1 x_2,\ y_{12} = (x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2), \\ &y_{13} = x_1\bar{x}_2 \vee \bar{x}_1 x_2,\ y_{14} = \overline{x_1 x_2},\ y_{15} = \overline{x_1 \vee x_2},\ y_{16} = \bar{x}_1 \vee x_2, \\ &y_{17} = x_1 \vee \bar{x}_2,\ y_{18} = \bar{x}_1 x_2,\ y_{19} = x_1\bar{x}_2. \end{aligned} \tag{5}$$

There are also 64 functions depending on 2 arguments. They can be obtained by the superposition of the previous 20 functions or via the compilation of the function value tables and the consequent transition to analytical representation. For $n = 3$ CL functions include all previous functions depending on, at most, 2 arguments, and all functions essentially depending on 3 arguments. The most common ternary functions include the following:

Disjunction and conjunction (maximum and minimum):

$$y = x_1 \vee x_2 \vee x_3 = \max(x_1, x_2, x_3), \;\; y = x_1 x_2 x_3 = \min(x_1, x_2, x_3), \quad (6)$$

median and median negation (inversion):

$$\begin{aligned} y &= \operatorname{med}(x_1, x_2, x_3) = x_1 x_2 \vee x_1 x_3 \vee x_2 x_3, \\ y &= \overline{\operatorname{med}}(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 \vee \bar{x}_1 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3, \end{aligned} \quad (7)$$

Sheffer and Webb functions:

$$y = \overline{x_1 x_2 x_3}, \quad y = \overline{x_1 \vee x_2 \vee x_3}, \quad (8)$$

and elementary three-place disjunction and conjunction:

$$y = x_1 \vee \bar{x}_1 \vee x_2 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_3, \quad y = x_1 \bar{x}_1 x_2 \bar{x}_2 x_3 \bar{x}_3. \quad (9)$$

It is possible to obtain other functions depending essentially on 3 arguments by the superposition of the functions listed above or by compiling the table of values followed by a transition to analytical representation. Any set of CL functions of a great number of arguments is generated in the same way.

Notice that the number $P(n)$ of CL functions of n arguments grows with n sufficiently faster than the number $Q(n)$ of functions of binary logic. For example, $Q(0) = 2$, $Q(1) = 4$, $Q(2) = 16$, $Q(3) = 256$. Therefore we are not able to apply exhaustive techniques to the investigation of CL functions, as it is done with binary functions. Thus we should stick to the analysis of typical CL functions.

Basic operations of CL were analyzed by R. McNaughton; the general description of CL and its mathematical apparatus is elaborated on by S. A. Ginsburg, V. I. Levin and P. N. Shimbirev. A review of their work can be found in [1, 2, 4, 6, 7, 9, 11, 12, 14, 21].

3. Some Examples of Applications regarding Continuous Logic

Example 1 (geometry). Given a piece-wise linear function $y = f(x)$ formed of two linear functions $y = ax+b$ and $y = cx+d$, the first function is accepted on the left side of the intersection of graphs of these two functions; the second function works on the right side. Using graphs of $y = ax + b$ and $y = cx + d$ we can check that only two possibilities for the formation of $f(x)$ are correct: we can use either the lower of the lines (concave function $f(x)$) or the upper of the lines (convex function $f(x)$). Therefore we obtain an analytic form of the piece-wise linear function $f(x)$:

$$y = (ax + b) \begin{smallmatrix} \vee \\ \wedge \end{smallmatrix} (cx + d),$$

where the operation of CL disjunction $\vee$ is applied when the graph of $f(x)$ is concave and the operation of CL conjunction $\wedge$ is used when it is convex.

The analytical representation of piece-wise linear and piece-wise non-linear functions in terms of CL is developed in the work of E. I. Berkovich, S. A. Ginsburg and V. I. Levin [1, 6, 12].

Example 2 (theory of discrete automata). Let us consider an automaton with two binary inputs x_1, x_2 and one binary output y; the automaton implements a Boolean function:

$$y = x_1 \,\&\, x_2, \quad x_1, x_2, y \in \{0, 1\}.$$

The inputs are determined by the binary processes of the form:

$$x_1(t) = \begin{cases} 0, & t < a, \\ 1, & t \geq a, \end{cases} \quad x_2(t) = \begin{cases} 1, & t < b, \\ 0, & t \geq b. \end{cases}$$

Our purpose here is to analyze the binary process $y(t)$ on an automaton's output and its reaction on the given input processes. Let $1(A, B)$ be a binary process of impulses in the time interval (A, B). The reaction of an automaton equals the impulse $1(a, b)$ when $b \geq a$ and it is the constant 0 when $b < a$. We can interpret the constant 0 as an impulse with a coincidental end and beginning. Then the reaction can be written in the terms of CL as follows:

$$y(t) = \left\{ \begin{array}{rl} 1(a, b) & \text{if } b \geq a \\ 0 = 1(a, a) & \text{if } b < a \end{array} \right| = 1(a, a \vee b).$$

Analytical theory of the processes in discrete automata was developed by V. I. Levin in [2, 4, 6, 14].

Example 3 (optimization). Let us imagine three vacancies and three candidates to fill these vacancies. Let a_{ij} be an efficiency of ith candidate to jth position. Our aim here is to distribute the positions between the candidates in such a manner that all positions are occupied, all the candidates are accepted and an integral efficiency is maximal. Obviously, every distribution of the positions between the candidates has its own sum of elements regarding the matrix $A = \|a_{ij}\|$; this sum includes exactly one element from every column and every row. Thus, we need to find a maximal sum $A^{\vee}$ of the elements of the matrix A. The sum has the following general form:

$$A^{\vee} = (a_{11} + a_{22} + a_{33}) \vee (a_{11} + a_{23} + a_{32}) \vee (a_{12} + a_{21} + a_{33}) \vee \\ \vee (a_{12} + a_{23} + a_{31}) \vee (a_{13} + a_{21} + a_{32}) \vee (a_{13} + a_{22} + a_{31}).$$

An algorithm of exhaustive search can be employed. The expression simplified with the help of law (20) looks like this:

$$A^{\vee} = \{a_{11} + [(a_{22} + a_{33}) \vee (a_{23} + a_{32})]\} \vee \\ \vee \{a_{12} + [(a_{21} + a_{33}) \vee (a_{23} + a_{31})]\} \vee \\ \vee \{a_{13} + [(a_{21} + a_{32}) \vee (a_{22} + a_{31})]\}.$$

It is three operations less than the previous expression. Therefore, we reduced the complexity of an exhaustive search.

The methods of optimization using CL were designed by V. I. Levin in [6, 9, 14–20].

Numerous examples of these techniques can be found in [1] (approximation of functions and the design of electrical circuits), [3] (design of digital devices), [5] (set theory and decision theory), [6] (design of analogue and digital devices, simulation of tools, queue theory, pattern recognition), [7] (fault-tolerance and diagnostics, technical servicing), [8, 13] (control, decision making), [9] (simulation and optimization of economic systems), [10–12, 17, 19] (synthesis of functional generators, design of analogue and hybrid devices), [15, 16, 18, 20] (simulation of economic systems, social groups, societies and historical events).

4. Laws of Continuous Logic

There is a straightforward generalization regarding CL in the frame of DL for the continuous interval C:

$a \vee a = a, \ aa = a$	(Tautology)	(10)
$a \vee b = b \vee a, \ ab = ba$	(Commutative)	(11)
$(a \vee b) \vee c = a \vee (b \vee c), \ (ab)c = a(bc)$	(Associative)	(12)
$a(b \vee c) = ab \vee ac, \ a \vee bc = (a \vee b)(a \vee c)$	(Distributive)	(13)
$\overline{a \vee b} = \bar{a}\bar{b}, \ \overline{ab} = \bar{a} \vee \bar{b}$	(de Morgan)	(14)
$a \vee ab = a, \ a(a \vee b) = a$	(Absorption)	(15)

$$\bar{\bar{a}} = a \qquad \text{(Double negation)} \qquad (16)$$

$$aA = A,\ aB = a,\ a \vee A = a,\ a \vee B = B \qquad \text{(Operation with constants)} \qquad (17)$$

$$a\bar{a}(b \vee \bar{b}) = a\bar{a},\ a\bar{a} \vee (b \vee \bar{b}) = b \vee \bar{b} \qquad \text{(Kleene)} \qquad (18)$$

The laws of contradiction and the eliminated third of DL are replaced by

$$a\bar{a} = M - |a - M|,\ a \vee \bar{a} = M + |a - M|. \qquad (19)$$

As soon as the operations of CL are applied to a continuum it is quite reasonable to combine them with algebraic operations over continuous variables.

Instead of addition and multiplication we can use new distributive laws of CL, which combine either disjunction or conjunction with addition:

$$\begin{aligned} a + (b \vee c) &= (a + b) \vee (a + c),\ a + (b \wedge c) = (a + b) \wedge (a + c), \\ a - (b \vee c) &= (a - b) \wedge (a - c),\ a - (b \wedge c) = (a - b) \vee (a - c). \end{aligned} \qquad (20)$$

A similar law works when disjunction and conjunction are coupled with multiplication.

A law of descent of negation on addends is as follows:

$$\overline{a + b} = \bar{a} - b = \bar{b} - a \qquad (21)$$

The operations of CL are expressed with the help of addition and multiplication, as well as two auxiliary functions: $I(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$ and $|x|$. Therefore disjunction and conjunction in CL are expressed as

$$a \vee b = 0{,}5[a + b + |a - b|], \quad a \wedge b = 0{,}5[a + b - |a - b|]. \qquad (22)$$

The possibility to express CL operations via algebraic operations (see (1) and (22)) indicates a connection between algebra and logic.

If we consider the generalization of CL itself, CL will be a special case of distributive structure with a pseudo-complement, i.e. with the operation of negation, which is not a complement because contradiction and excluded middle laws do not take place. In this context the values of a continuous CL variable (which belongs to the interval $[A, B]$) can be interpreted similarly to a variable of DL: the boundary value $x = A$ ($x = B$) represents the statements "absolutely false" ("absolutely true"), and the intermediate values x, $A < x < B$, measure the truth values of other statements.

Some basic laws of CL for the particular case of $C = [0, 1]$ were indicated by R. McNaughton. These laws were investigated in a general manner by S. A. Ginsburg, V. I. Levin and E. I. Berkovich. A review of these results can be found in [1, 2, 4, 6, 9, 14–16, 21].

5. Enumeration and the Standardization of Continuous-Valued Logical Functions

The enumeration of all CL functions for a fixed number of arguments and the representation of CL functions in standard form are the two most typical problems of CL. The problem relating to the enumeration of all CL functions in algebra (2) requires a specification of the appropriate analytical expressions. This can be done using the following two steps:

1. Enumeration of the tables of function values (the tables for all functions have an identical order of the following of argument ordering variants $x_1, \ldots, x_n$ and their negations $\bar{x}_1, \ldots, \bar{x}_n$, with various distributions of function values equal x_i or $\bar{x}_i$);
2. Transition from the tables to appropriate analytical expressions by a method of uniting.

Unfortunately this approach is not appropriate for $n \geq 3$. Therefore in practice we are limited by the defined classes of CL functions, which can be selected from the corresponding similar functions of DL, simplicity in the deriving of formulas and their practical importance.

Being the standard forms of CL, the functions of algebra (2) obey disjunctive and conjunctive normal forms (DNF and CNF). These forms differ from similar forms of binary DL because their elementary conjunctions (disjunctions) may include, together with argument x_i, its negation $\bar{x}_i$. The transition from any analytical representation of CL function to its DNF or CNF is similar to the transformation in binary DL. In the case of DNF the transition consists of (1) descent of negations on more simple expressions according to laws (14), (16); (2) disclosure of brackets in agreement with law (13). For CNF we have (1) the same descent of negations; (2) placement of brackets in accordance to law (13).

Example 4. Let us transform the analytical representation of CL to its DNF:

$$
\begin{aligned}
(x_1x_2 \vee \bar{x}_2x_3)\overline{\bar{x}_1x_4} &= (x_1x_2 \vee \bar{x}_2x_3)(x_1 \vee \bar{x}_4) = \\
&= x_1x_2 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_4 \vee \bar{x}_2x_3\bar{x}_4 = \\
&= x_1x_2 \vee x_1\bar{x}_2x_3 \vee \bar{x}_2x_3\bar{x}_4.
\end{aligned}
$$

We can accept canonical forms of DNF and CNF as unambiguous standard forms of the functions of CL. DNF unambiguously represents the CL function if it is a deadlock disjunction of non-decomposable elementary conjunctions. In turn, the elementary conjunction is non-decomposable in a disjunction of conjunctions when it is fundamental, i.e. consistent (does not contain simultaneously x_i and $\bar{x}_i$) or inconsistent but contains all the arguments of a given function, in direct x_i or inverse $\bar{x}_i$ form. From this point we can apply the following algorithm for the reduction of some DNF of a function of CL to canonical DNF:

1. Select in DNF all fundamental conjunctions;
2. represent each non-fundamental conjunction k (i.e. a conjunction which is contradictory and does not contain all of the arguments of the functions) by a disjunction of the fundamental conjunctions (to do this we can combine its conjunction with a suitable disjunction $x_j \vee \bar{x}_j$ because it does not change the value of k that includes $x_i\bar{x}_i \leq M$, for $x_j \vee \bar{x}_j \geq M$) and to open the brackets;
3. eliminate the smaller conjunction of each pair of comparable (in the sense of ratio $\leq$) conjunctions of DNF. The resultant canonical DNF is similar in sense but not form to the complete DNF of a Boolean function.

Example 5. Let us transform DNF of a CL function to its canonical DNF:

$$y = x_2x_4 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee x_1x_2\bar{x}_2x_3\bar{x}_3.$$

The first two conjunctions are fundamental. The third conjunction is not: when multiplied as $x_4 \vee \bar{x}_4$ it is transformed to $x_1x_2\bar{x}_2x_3\bar{x}_3x_4 \vee x_1x_2\bar{x}_2x_3\bar{x}_3\bar{x}_4$. These newly emerged fundamental conjunctions are absorbed by the first two fundamental conjunctions of DNF of y. Eventually we have the canonical DNF:

$$y = x_2x_4 \vee x_1\bar{x}_2x_3\bar{x}_4.$$

Any CL function different from a fundamental conjunction is decomposable, i.e. a class of non-decomposable (elementary) CL functions in algebra (2) consists of only fundamental conjunctions.

The problems of representation and enumeration of CL functions were discussed by C. M. Clark, D. Dubois, H. Prade, A. Kandel, V. I. Levin, M. Mukaidono and P. N. Shimbiriev (see [3, 5, 6, 9, 11–13]). The standard representation of CL functions was tackled by F. P. Preparata, A. Kandel, D. Dubois, H. Prade, V. I. Levin and P. N. Shimbiriev (see reviews in [3, 6, 9, 11].

6. Minimization and the Decomposition of Continuous-Valued Logical Functions

The minimization of CL functions, similarly to the minimization of DL functions, aims to produce a form with a minimal number of variables.

A procedure of the minimization of the functions of CL in algebra (2) is developed only for such functions as represented in DNF. It deals with the search for the DNF with a minimal number of the entries of $x_i, \bar{x}_i$. The procedure of minimization regarding CL functions can be described in terms similar to the minimization of a Boolean function:

1. Make a search for all the fundamental conjunctions of the CL function f (these conjunctions play a role in the elementary conjunctions of the complete DNF of a Boolean function) and a representation f in canonical deadlock form;
2. search for all the simple implicants of the function (as usual, an implicant of the function f is thought of as an elementary conjunction k such that $k \leq f$; the implicant k is called simple if it is not absorbed by other implicants);
3. compute a minimal covering of the set of fundamental conjunctions by a set of simple implicants; for examples, with the help of tables of implicants.

Steps (1) and (2) are specific for CL functions. Step (1) is discussed in Section 7 of the paper. As for step (2), it is based on the content of consensus of elementary conjunctions k_i: if $k_1 = x_i a$, $k_2 = \bar{x}_i b$, where a, b are conjunctions of other characters, then consensus of k_1 and k_2 is represented in the sets of such contradictory conjunctions that (i) ab, (if it is inconsistent); (ii) conjunctions $x_i \bar{x}_i ab$, $i = \overline{1, n}$ (if ab is consistent). If k_1, k_2 are non-representable in indicated form with any i then consensus equals 0.

Example 6. For elementary conjunctions $k_1 = x_1\bar{x}_2x_3$, $k_2 = x_2\bar{x}_3$ the consensus is

$$\{x_1x_2\bar{x}_2, x_1x_3\bar{x}_3\}.$$

For elementary conjunctions $k_1 = x_1x_2x_3$, $k_2 = x_2\bar{x}_3$ the consensus is

$$\{x_1x_2x_3\bar{x}_3, x_1x_2\bar{x}_2, x_1\bar{x}_1x_2\}.$$

We can search for all the simple implicants of CL function f, represented in deadlock DNF $f = \bigvee_i k_i$ by using the following algorithm:

1. For some pair k_i, k_j a consensus is formed;

2. all conjunctions obtained at step 1 are added to the disjunction $\bigvee_i k_i$;
3. all conjunctions k_a included in other conjunctions k_b (i.e. $k_a \leq k_b$) are eliminated.

Steps 1–3 are repeated for new pairs k_i, k_j until the form of the function f remains unchanged. The final expression $f = \bigvee_i \tilde{k}_i$ in terms of conjunctions $\tilde{k}_i$ does contain all the simple implicants of the function f.

The number of CL functions grows enormously when the number of arguments increases, which is reflected in the complexity of minimization; therefore, the problem of decomposition regarding CL functions begins to play a very important role. The decomposition of the CL function $f(x)$, $x = (x_1, \ldots, x_n)$, represents f as a composition of several CL functions with a smaller number of arguments:

$$f(x) = F[f_m(x^m), \ldots, f_1(x^1), x^i], \quad x^i \subset x, \quad i = \overline{0, m}. \tag{23}$$

If the intersection of the sets x^i, $i = \overline{0, m}$ is empty, the decomposition is called a separating decomposition, otherwise this is a non-separating decomposition. The representation in (23) with $m = 1$ is called a simple decomposition. This is presently the only known algorithm in the search for simple decomposition of the CL function in algebra (2).

The problems of minimization and decomposition of CL functions were investigated by A. Kandel, D. Dubois, H. Prade, N. P. Shimbiriev (see [3, 9, 11, 12]).

7. Analysis and Synthesis of Continuous-Valued Logical Functions

Analysis and synthesis of CL functions is quite different from those problems relating to DL. Let be a range of values of a vector of arguments $x = (x_1, \ldots, x_n)$, D_f be a range of values of the CL function $f(x)$; there is also a one-to-one correspondence:

$$(x \in D_x) \Leftrightarrow (f(x) \in D_f). \tag{24}$$

The analysis of function f is converted to the following problem: given range D_f and function $f(x)$, find the range D_x in accordance with (24). The synthesis of the function $f(x)$ can be thought of as the following: given ranges D_x and D_f, construct the CL function f which realizes the correspondence in (24). Most methods of the analysis are developed for special cases when f is either a many-placed disjunction or conjunction, and D_f is a half-interval or interval. They are based on the following equivalencies:

$$\begin{aligned}
&\Big(\bigvee_{i=1}^{n} x_i \geq a\Big) \Leftrightarrow (x_1 \geq a \text{ or } \dots \text{ or } x_n \geq a);\\
&\Big(\bigvee_{i=1}^{n} x_i \leq b\Big) \Leftrightarrow (x_1 \leq b, \dots, x_n \leq b);\\
&\Big(\bigwedge_{i=1}^{n} x_i \geq a\Big) \Leftrightarrow (x_1 \geq a, \dots, x_n \geq a);\\
&\Big(\bigwedge_{i=1}^{n} x_i \leq b\Big) \Leftrightarrow (x_1 \leq b \text{ or } \dots \text{ or } x_n \leq b);
\end{aligned} \tag{25}$$

In general, with the arbitrary CL function f and its range D_f, we should apply formal methods: divide D_f on sub-range, half-intervals, make a decision for each appropriate inequality (see section 8) and unify the results. Sometimes the analysis of the function $f(x)$ may be understood as the search for given f and ranges $D_{x_1}, \dots, D_{x_n}$ for the arguments $x_1, \dots, x_n$ (components in aggregate range D_x) of the appropriate range D_f (24). This task is an inverse of the previous one; it is based on the following equivalencies:

$$\begin{aligned}
&(a \leq x_1 \leq b, c \leq x_2 \leq d) \Leftrightarrow (a \vee c \leq x_1 \vee x_2 \leq b \vee d, ac \leq x_1 x_2 \leq bd),\\
&(a \leq x \leq b) \Leftrightarrow (2M - b \leq \bar{x} \leq 2M - a)
\end{aligned} \tag{26}$$

The task of the synthesis of the CL function in a common case has no unique solution. An algorithm for the exact solution is unknown. One of the possible methods might be as follows:

1. Discard the requirement $x \leq D_x$;
2. select any standard function $f(x)$;
3. analyze $f(x)$ for the given condition $f \in D_f$, select an appropriate condition for $x : x \in D'_x$;
4. if $D_x \subseteq D'_x$, then $f(x)$ is a solution of this task; otherwise, make a transition to the following function $f(x)$ and continue.

Such an exhaustive search is unrealistic for large n, therefore we can reject the requirement $x \in D_x$, and set up the following problem of the synthesis: construct function $f(x)$ on the given range D_f such that $f(x) \in D_f$. But almost any (except constants) function $f(x)$ of CL with suitable x can accept any value in C. Therefore we have the following problem: Find, according to (24), the range D_x on the given range D_f and selected function $f(x)$.

8. Solution of Equations and Inequalities of CL

Equations and inequalities of CL bear the same sense as the equations and inequalities of DL. However they must be treated in a different way because they do relate to continuous sets. An equation of CL is:

$$f(a, x) \leqq F(a, x), \tag{27}$$

where f and F are given CL functions, $a = (a_1, \ldots, a_k)$ is a vector of parameters, $x = (x_1, \ldots, x_n)$ is a vector of unknowns. An individual solution of the inequality (27) names any vector x, for which the equality is fair. Equations and inequalities in CL are classified on the number of unknowns n and on the complexity of CL functions of the left and right parts represented in standard deadlock DNF. Now we can fill the inequality with one unknown in the standard form:

$$ax \vee a'\bar{x} \vee bx\bar{x} \vee c \leqq dx \vee d'\bar{x} \vee lx\bar{x} \vee e \tag{28}$$

A maximal number of unknowns and their negations in one elementary conjunction of a standardized inequality is called an order I of the inequality; thus, for example, $I = 2$ for the equation (28). The equations with $I = 1$ are called linear, and those with $I \geq 2$ are nonlinear. A general form of a simple equation with n unknowns in the standard form looks like this:

$$\left(\bigvee_{i=1}^{n} a_i x_i\right) \vee \left(\bigvee_{i=1}^{n} a_i' \bar{x}_i\right) \vee c \leq \left(\bigvee_{i=1}^{n} d_i x_i\right) \vee \left(\bigvee_{i=1}^{n} d_i' \bar{x}_i\right) \vee e \tag{29}$$

The inequalities of CL can be subdivided into ones containing negations of unknowns and those which do not contain them. The main method of the solution of inequalities of CL is the sequential partition of their right-hand and left-hand parts allowing replacing an input inequality by equivalent association of systems of the more simple equations and inequalities.

Example 7. Let us consider equation (27), where the last operation on the left-hand side is the disjunction of CL:

$$f_1(a, x) \vee f_2(a, x) \leqq F(a, x).$$

Using a definition of CL disjunction we can subdivide this equation into an equivalent union of two following systems of equations:

$$\left\{ \begin{array}{l} f_1(a, x) \geq f_2(a, x) \\ f_1(a, x) \leqq F(a, x) \end{array} \right\} \cup \left\{ \begin{array}{l} f_1(a, x) < f_2(a, x) \\ f_2(a, x) \leqq F(a, x) \end{array} \right\}$$

Here, the newly obtained equation is simpler than the original one because it contains fewer operations in one of its parts. The simplification can be continued with the right-hand part of the equation, etc. The process will be finished when we obtain indivisible equations and inequalities that represent a solution of a given equation. Cases when the last operation of the left-hand part or the right-hand part of a given equation is a conjunction can be considered by analogy.

The theory, together with the solution techniques, was developed by V. I. Levin. The most detailed investigation of the problem can be found in [2]. We also recommend the reviews in [4–6, 9].

9. Differentiation and Integration of Continuous-Valued Logical Functions

Functions of DL, defined on discrete arguments, cannot be differentiated or integrated. CL functions have continuous-valued arguments. Therefore they can be differentiated and integrated. However, it is difficult to differentiate the functions of CL because they always have some points at which they break, where a derivative does not exist. Let us call a CL function derived as a superposition of operations $\vee$ and $\wedge$ of the arguments x_i (their negations) as a function of the first and the second sort, respectively. Point $x = (x_1, \ldots, x_n)$ is called a half-regular point of the function of the first sort, if it has an ε-vicinity in the constant ordering of $x_1, \ldots, x_n$. The point $x = (x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n)$ is called a regular point of the function of the second sort, if it has an ε-vicinity in the constant ordering of $x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n$. It is necessary and sufficient for point x to have its coordinates strictly ordered in values to be a half-regular (or regular) point. The following theorems are very important:

1. Any CL function of the first sort has in each half-regular point the single derivative for any argument with values from the set $\{0, 1\}$;
2. any CL function of the second sort has in each regular point the single derivative for any argument with values from the set $\{1, -1, 0\}$;
3. any CL function f in each point of existence of its derivatives $\partial f / \partial x_i$, $i = \overline{1, n}$, has no more than one non-zero derivative.

The main method of differentiation of CL functions lies in their sequential partition with obtaining a collection of simpler expressions, correct in their sub-ranges, and their differentiation. If necessary, some general rules of differential calculus (as, e.g. a derivative of a sum and a product) can be used as well.

Some examples of the derivatives of CL functions are shown below:

$$\begin{aligned} &x'_x = 1, \quad (\bar{x})'_x = -1, \quad (x \vee \bar{x})'_x = 1(x - M) - 1(M - x), \\ &(x\bar{x})'_x = 1(M - x) - 1(x - M), \quad x \neq M; \\ &(x_1 \vee x_2)'_{x_1} = 1(x_1 - x_2), \quad (x_1 x_2)'_{x_1} = 1(x_2 - x_1), \quad x_1 \neq x_2. \end{aligned} \tag{30}$$

Here $1(x)$ is a single function. The condition $x \neq M$ excludes the irregular point $x = M$, where the third and fourth derivatives do not exist. The differential calculation in CL may be a source of new laws. They emerge, in particular, with differentiation of the laws of a CL algebra and can be considered as the differential equations defining various functions of CL. For example:

$$(x_1 \vee x_2)'_{x_1} + (x_1 x_2)'_{x_1} = 1, \quad (x_1 \vee x_2)'_{x_1} \cdot (x_1 x_2)'_{x_1} = 0. \tag{31}$$

The system in (31) of the two differential equations determines two functions of CL: disjunction and conjunction, which are solutions of the system.

When differentiating functions with a large number of arguments, it is reasonable to transform them to standard forms where a differentiate variable is selected. This form can be easily differentiated. Thus, for example, for standard forms (in the class of DNFs) with the selected variable

$$ax \vee d, \quad b\bar{x} \vee d \tag{32}$$

the derivatives are as follows:

$$\begin{aligned} &(ax \vee d)'_x = I(ax - d) \cdot I(a - x), \quad ax \neq d, \quad x \neq a; \\ &(b\bar{x} \vee d)'_x = I(bx - d) \cdot I(\bar{x} - b), \quad b\bar{x} \neq d, \quad \bar{x} \neq b. \end{aligned} \tag{33}$$

For functions of CL we can define the high-order derivatives, e.g. 2nd order and 3rd order derivatives. In this case any function of the 1st order has a derivative of higher orders, which equal 0; the same takes place for the 2nd order function.

It is possible to integrate CL functions as functions of continuous variables. The function may be decomposed into the collection of more simple expressions, correct in their sub-ranges, which are integrated. If necessary, such usual rules of integration as integral of a sum, subdivision of integration interval, etc., can be used. The obtained integrals always exist because CL functions are continuous.

Differential and integral calculus of CL functions was investigated in detail by E. I. Berkovich and V. I. Levin [12].

10. Completeness in Continuous-Valued Logic

In CL, as well as in DL, there is a problem of completeness. A system of CL functions $\{f_1, \dots, f_m\}$ is a complete system (basis) in R class, if any function from R can be represented by a superposition of the functions $f_1, \dots, f_m$. In contrast with DL, where R is a given and the basis is unknown, in CL the basis is usually a given, and R class has to be found. The following examples seem to be useful:

1. The system $\{\vee, \wedge\}$ is the basis for class R_1 of the functions $C^n \to C$ which accept the value of one of the arguments;
2. the system $\{\vee, \wedge, ^-\}$ is the basis for class R_2 of the functions $C^n \to C$ which accept the value of one of the arguments or its negation;

3. the systems $\{\overline{x_1 x_2}\}$ and $\{\overline{x_1 \vee x_2}\}$ are the basis for class R_1;
4. the systems $\{\overline{x_1 x_2}, {}^{-}\}$ and $\{\overline{x_1 \vee x_2}, {}^{-}\}$ are the basis for class R_2;
5. the system $\{\vee, \wedge, \supset\}$ is the basis for class R_3 of such functions $C^n \to C$ which can be represented in the following form:

$$y = \left[A \vee \left(b_0 + \sum_{i=1}^{n} b_i x_i\right)\right] \wedge B, \quad \text{where } b_0, \ldots, b_n \text{ are integer.} \tag{34}$$

The classes R_1, R_2, R_3 are different subsets of the continuous set of all CL functions. Mathematically, these classes are quite narrow. However their practical importance cannot be overestimated because the elementary operations of CL (disjunction, conjunction etc.) are similar to the processes of real systems. This adequacy together with the completeness of CL operations lies in the basis of numerous applications of CL in the investigation of mathematical, engineering, economical, social and other phenomena. The problems of completeness regarding CL functions were investigated by R. McNaughton, F. P. Preparata and V. I. Levin. An overview of the results can be found in [5, 6, 9, 21].

11. Conclusions

We may predict that in the future the main attention regarding the theory of CL will be, apparently, given to the development of new generalizations of CL. The good old traditional tasks should not be overlooked either. In the field of enumeration of CL functions, in search for their representations and minimal forms, more effective solutions will be found. Great progress is expected in the applications of CL concerning operation research, simulation of complex economic systems and neuronal structures, description and the analysis of the processes in sociology and history.

It should be noted that in addition to CL, discussed here in this paper, there is another continuous logic. This is the $\aleph_0$-valued logic of Lukasiewicz, defined during the interval $[0, 1]$. This logic has basic operations which are similar to those of CL. However, as it was demonstrated by R. McNaughton [21], but not for all tuples of arguments, for which CL functions are defined, it is possible to determine corresponding functions of Lukasiewicz logic. In the works of V. I. Levin [2, 3, 6, 7, 9] it was proved that by simulating applied systems with CL we must define the operations on the interval $[A, B]$, where $A < 0$, $B > 1$. Therefore, the theoretical and applied potentials of CL are wider than those of Lukasiewicz logic.

The further particulars on the subjects, mentioned in our review, can be found in the publications [1–13]. A lot of applied results were discussed at the conferences held in Penza and Ulianovsk [14–20]. Let us now briefly discuss some Russian works in the field [1, 2, 4, 6–12]. The early book [1] considers the basic principles of continuous-valued logic applied to problems of function approximation, synthesis of functional schemes and the design of electrical circuits. A theory of CL, including equations and inequalities of CL and their application to automata theory and the design of digital devices is proposed in [2, 4]. The book [6] considers CL, its generalizations and application to automata theory, information processing, reliability theory, decision theory, and optimization. A theory of the reliability of engineering systems, based on CL, is built in [7]. The collection of papers [8] includes various fuzzy logics and their application to artificial intelligence. The mathematical apparatus of CL, its generalization and its application to scheduling, optimization and simulation of economic systems can be found in [9]. The hybrid systems, derived from CL and DL and algebraic structures, form a subject of the monograph [11]; here we can also find the minimization of CL functions and its application to functional synthesis. Some basic results of CL and its applications in mathematics, economic, engineering, system theory and biology are presented in [12].

References

1. Ginsburg S. A. (1968). Mathematical Continuous Logic and Image of Functions. – Moscow, Energia (In Russian).
2. Levin V. I. (1975). Introduction to the dynamic theory of finite automata. – Riga: Zinatne (In Russian).
3. Kandel A., Lee S. C. (1979). Fuzzy switching and automata. Theory and application. – New York. Grain, Russak and Co.
4. Levin V. I. (1980). Dynamics of logical devices and systems. – Moscow, Energia (In Russian).
5. Koffman A. (1982). Introduction to the theory of fuzzy sets. – Moscow, Radio and Svjaz (In Russian).
6. Levin V. I. (1982). Infinite-valued logic in the tasks of cybernetics. – Moscow, Radio and Svjaz (In Russian).
7. Levin V. I. (1985). The logical theory of reliability of complex systems. – Moscow, Energoatomizdat (In Russian).
8. Pospelov D. A. (Editor) (1986). Fuzzy sets in models of control and artificial intelligence – Moscow, Nauka Publishers (In Russian).

9. Levin V. I. (1987). Structure-logical methods of a research of complex systems with applications of the computer. – Moscow, Nauka (In Russian).
10. Volgin L. I. (1989). Synthesis of devices for processing and conversion of the information in element basis of relators. – Tallinn: Valgus (In Russian).
11. Shimbirev P. N. (1990). Hybrid continuous-logical devices. – Moscow, Energoatomizdat (In Russian).
12. Volgin L. I., Levin V. I. (1990). Continuous logic. The theory and applications. – Tallinn: Academy of Sciences of Estonia (In Russian).
13. Terano A., Asai E. and Sugeno M. (Editors) (1993). Application of fuzzy systems – Moscow, Mir (In Russian).
14. Levin V.I. (Editor) Proc. 1st All-Russian Conf. "Continuous logic and its application in engineering, economics and sociology", Penza, 1994. (in Russian).
15. Levin V. I. (Editor) Proc. Int. Conf. "Continuous logical methods and models in a science, engineering and economics" Penza, 1995. (in Russian).
16. Levin V. I. (Editor) Proc. Int. Conf. "Continuous and similar logics in engineering, economics and sociology". Penza, 1996 (In Russian).
17. Volgin L. I. (Editor) Proc. Int. Conf. "Continuous-logical systems, models and algorithms", Ulyanovsk. 1995. (in Russian).
18. Levin V. I. (Editor) Proc. All-Russian Conf. "Continuous and related logics in computer science, economics and sociology" Penza, 1997 (in Russian).
19. Volgin L. I. (Editor) Proc. Int. Conf "Relator and continuous logical networks and models" Ulyanovsk, 1998 (in Russian).
20. Levin V. I. (Editor) Proc. Int. Conf. "Logic-mathematical methods in engineering, economics and sociology", Penza, 1998 (In Russian).
21. McNaughton R. A theorem about infinitive-valued sentential logic. J. Symb. Logic. 16. No 1. 1951.

Mariusz Giero
University of Białystok

QUERYING TEMPORAL DATABASE WITH THE LANGUAGE OF FIRST-ORDER TEMPORAL LOGIC[1]

1. Introduction

A temporal database [Etz, Ste, Tan] is defined as a database maintaining object histories, i.e., past, present, and possibly future data. There are numerous application domains dealing with temporal data: Medical Systems (e.g. patient's records), Computer Applications (e.g. history of file back ups), Archive Management Systems (e.g. sporting events, publications and journals), Reservation Systems (e.g. when was a flight booked) and many others [Sno]. Support for time-varying data within a traditional relational database is not straightforward. There have been more than two dozens extended relational data models proposed [JenSno]. Time-varying data is commonly represented by timestamping values [JenSno, Jen]. Timestamps can be time points, intervals or a set of intervals and can be added to tuples or attributes. There are also different considerations of what time stamps represent: valid time, i.e., time when data (tuple) is true in the universe of discourse, transaction time, i.e., time when data is stored in a database or both time references together.

In this paper we consider a temporal database model with tuple timestamping. Tuples are timestamped by a set of intervals which represent valid time.

[1] The research reported in this paper is part of the project entitled *Temporal Representation of Knowledge and Its Implementation in the Computer Systems of Medical Conducts* supported by the Polish Ministry of Science and Higher Education, grant no. 3 T11F 011 30.

ISBN 978–83–7431–158–8 ISSN 0860-150X

2. Structure of Time

In this paper, we assume that the flow of time $(T, <)$ is a linear, discrete and ordered structure with no end points. T is a set of time points and $<$ is a binary order relation defined on T which satisfies the following conditions:

- transitivity $\forall x, y(x < y \wedge y < z \rightarrow x < z)$
- irreflexivity $\forall x \neg(x < x)$
- totality $x = y$ or $x < y$ or $y < x$, where $x, y, z \in T$

3. Relational Database

The relational data model was introduced in the 1970s by E. F. Codd [Cod, Dat]. Currently, it is the most widespread data model used for database applications. Formally, it can be defined as follows:

Definition 1.
A *relational database schema* is a quintuple $S = (R, A, D, \mathrm{attr}, \mathrm{dom})$, where:

- $R = \{R_1, \ldots, R_k\}$ is a set of relation names,
- $A = \{A_1, \ldots, A_n\}$ is a set of attribute names,
- $D = \{D_1, \ldots, D_m\}$ is a set of domains,
- $\mathrm{attr} : R \rightarrow \mathrm{TUP}(A)$, where $\mathrm{TUP}(A)$ denotes a set of finite tuples of different elements of A, is a mapping that assigns to each relation name a tuple of attribute names,
- $\mathrm{dom} : A \rightarrow D$ is a mapping that assigns to each attribute name a domain.

Definition 2.
An *instance of relational database* (or just a *relational database*) for schema $S = (R, A, D, \mathrm{attr}, \mathrm{dom})$ is a set $DB = \{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_k\}$ where $\boldsymbol{R}_i$ is a relation instance (or just a relation) over the relation name $R_i \in R$, i.e.,

$$\boldsymbol{R}_i \subseteq \mathrm{dom}(A_1) \times \ldots \times \mathrm{dom}(A_l),$$

where $\mathrm{attr}(R_i) = (A_1, \ldots, A_l)$, $l \leq n$ and $\times$ is the Cartesian product operator.

Example 1.
Let us consider a database storing data about the patients at a hospital. For simplicity, we will use only two attributes and one relation name.

$R = \{\text{PATIENTS}\}$, $A = \{\text{ID}, \text{NAME}\}$, $D = \{\text{N}, \text{CHAR}\}$[2],
$\text{attr}(\text{PATIENTS}) = (\text{ID}, \text{NAME})$,
$\text{dom}(\text{ID}) = \text{N}$, $\text{dom}(\text{NAME}) = \text{CHAR}$.

$DB = \{\mathbf{PATIENTS}\}$
$\mathbf{PATIENTS} = \{(1, \text{Kowalski}), (2, \text{Kozłowski}), (3, \text{Piasecka})\}$

The fact $(x, y) \in \mathbf{PATIENTS}$ means that a person named y with an identifier x is a patient at a specific hospital. A database can also be represented (not formally, for the sake of readability) as a set of tables. A table represents a relation. In this example:

PATIENTS

ID	NAME
1	Kowalski
2	Kozłowski
3	Piasecka

4. Temporal Database

Definition 3.

An *instance of temporal database* (or just a *temporal database*) for schema $S = (R, A, D, \text{attr}, \text{dom})$ over the flow of time $(T, <)$ is a set $TDB = \{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_k\}$ where $\boldsymbol{R}_i$ is a temporal relation instance (or just a temporal relation) over the relation name $R_i \in R$, i.e.,

$$\boldsymbol{R}_i(\text{dom}(A_1) \times \ldots \times \text{dom}(A_l)) \times 2^T,$$

where $\text{attr}(R_i) = (A_1, \ldots, A_l)$.

Example 2.

Let S be the same schema as in Example 1. We take the flow of time to be that of days $T = \{\ldots, 2007\text{–}03\text{–}01, 2007\text{–}03\text{–}02, 2007\text{–}03\text{–}03, \ldots\}$.

$TDB = \{\mathbf{PATIENTS}\}$
$\mathbf{PATIENTS} =$

$\{((1, \text{Kowalski}), [2007\text{–}02\text{–}01, 2007\text{–}02\text{–}25]$[3] $\cup [2007\text{–}03\text{–}15, 2007\text{–}03\text{–}16]),$
$((2, \text{Kozłowski}), [2007\text{–}02\text{–}25, 2007\text{–}03\text{–}01]),$
$((5, \text{Piasecka}), [2007\text{–}02\text{–}20, 2007\text{–}03\text{–}05] \cup [2007\text{–}04\text{–}01, 2007\text{–}04\text{–}16])\}$

[2] N denotes a set of natural numbers, CHAR a set of character sequences.
[3] $[a, b] = \{x : x \in T, a \leq x \leq b\}$.

The set of time points (stamps) associated to a tuple describes when data represented by the tuple are true in modelled reality, i.e., in this example, when a person is (or was or is going to be) a patient at the hospital. The temporal database can also be represented as a set of tables:

PATIENTS

ID	NAME	
1	Kowalski	[2007–02–01, 2007–02–25] ∪ [2007–03–15, 2007–03–16]
2	Kozłowski	[2007–02–25, 2007–03–01]
3	Piasecka	[2007–02–20, 2007–03–05] ∪ [2007–04–01, 2007–04–16]

5. The Query Language for Temporal Database

Let $S = (R, A, D, \mathrm{attr}, \mathrm{dom})$ be a relational database scheme, $(T, <)$ be the flow of time and $TDB = \{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_k\}$ be a temporal database for S over $(T, <)$. The *query language* (QL) for TDB is based on the language of first-order temporal logic [Gab, ChoTom]. It has the following categories of basic symbols:

- Domain variables: $x_1, x_2, \ldots$;
- Domain constants: $c_1, c_2, \ldots$;
- time variables: $t_1, t_2, \ldots$;
- time constants: $e_1, e_2, \ldots$;
- elements of R as predicate symbols: $R_1, R_2, \ldots, R_k$ and a predicate symbol *time*;
- equality symbol: $=$;
- logical connectives: $\neg, \wedge$;
- existential quantifier: $\exists$;
- temporal connectives: U, S;
- punctuation symbols: $(,)$.

Syntax

A term is either a constant or a variable. The atomic formulas of the language are of the form:

- $a_i = a_j$, where a_i and a_j are terms of the same sort, i.e., either domain terms or time terms,
- $R_i(a_1, a_2, \ldots, a_n)$, where n is the length of the sequence $\mathrm{attr}(R_i)$ and a_j is a domain variable (constant) that ranges over (is element of) the domain $\mathrm{dom}([\mathrm{attr}(R_i)]_j)$,
- *time*(a), where a is a time term.

Formulas of QL are finite strings of basic symbols defined in the following recursive manner:
(1) Any atomic formula is a formula,
(2) if φ, ψ are formulas, so also are $\neg\varphi$, $\varphi \wedge \psi$, $\exists a\, \varphi$, $\mathsf{U}(\varphi, \psi)$, $\mathsf{S}(\varphi, \psi)$, where a is any variable x_i.

Semantics

We define interpretation Θ as follows: $\Theta(R_i) = \boldsymbol{R}_i$, $\Theta(c_i) \in \bigcup D$, $\Theta(e_i) = T$, for every i. An assignment v is a mapping that associates every domain variable x_i with a domain value $v(x_i) \in \bigcup D$ and every time variable t_i with a time point $v(t_i) \in T$. It is convenient to extend an assignment over constants by making $v(c_i) = \Theta(c_i)$ and $v(e_i) = \Theta(e_i)$, for every i. We define a formula φ to be true in TDB at time t under assignment v (denoted by $TBD, v, t \models \varphi$) by induction on the structure of the formula:

Definition 3a.
(1) $TBD, v, t \models R_i(a_1, \ldots, a_s)$ iff $((v(a_1), \ldots, v(a_s)), \tau) \in \Theta(R_i)$,
where $\tau \subseteq T$ and $t \in \tau$,
(2) $TBD, v, t \models a_i = a_j$ iff $v(a_i) = v(a_j)$,
(3) $TBD, v, t \models \mathsf{time}(a_i)$ iff $v(a_i) = t$,
(4) $TBD, v, t \models \neg\varphi$ iff not $TBD, v, t \models \varphi$,
(5) $TBD, v, t \models \varphi \wedge \psi$ iff $TBD, v, t \models \varphi$ and $TBD, v, t \models \psi$,
(6) $TBD, v, t \models \exists x_i\, \varphi$ iff $TBD, v^*, t \models \varphi$,
where v^* is an assignment which agrees with the assignment v on the values of all variables except, possibly, on the values of x_i,
(7) $TBD, v, t \models \mathsf{U}(\varphi, \psi)$ iff
there exists a $t_1 \in T$ with $t < t_1$ and $TBD, v, t_1 \models \varphi$
and for every $t_2 \in T$ such that $t < t_2 < t_1$ holds $TBD, v, t_2 \models \psi$,
(8) $TBD, v, t \models \mathsf{S}(\varphi, \psi)$ iff
there exists a $t_1 \in T$ with $t_1 < t$ and $TBD, v, t_1 \models \varphi$
and for every $t_2 \in T$ such that $t_1 < t_2 < t$ holds $TBD, v, t_2 \models \psi$.

For convenience, we will introduce additional symbols: $\vee, \rightarrow, \leftrightarrow$ (other logical connectives), $\forall$ (universal quantifier) and $\mathsf{F}, \mathsf{P}, \mathsf{G}, \mathsf{H}, \mathsf{X}, \mathsf{Y}$ (other temporal connectives, (see Fig. 1)) defined as:

Definition 3b.
(1) $\varphi \vee \psi \quad \equiv_{\text{def}} \quad \neg(\neg\varphi \wedge \neg\psi)$
(2) $\varphi \rightarrow \psi \quad \equiv_{\text{def}} \quad \neg\varphi \vee \psi$
(3) $\varphi \leftrightarrow \psi \quad \equiv_{\text{def}} \quad (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$
(4) $\forall x\, \varphi \quad \equiv_{\text{def}} \quad \neg\exists x\, \neg\varphi$
(5) $\mathsf{F}\varphi \quad \equiv_{\text{def}} \quad U(\varphi, \top)$

(6) $G\varphi \equiv_{\text{def}} \neg F \neg \varphi$
(7) $P\varphi \equiv_{\text{def}} S(\varphi, \top)$
(8) $H\varphi \equiv_{\text{def}} \neg P \neg \varphi$
(9) $X\varphi \equiv_{\text{def}} U(\varphi, \bot)$ (if the flow of time is discrete)
(10) $Y\varphi \equiv_{\text{def}} S(\varphi, \bot)$ (if the flow of time is discrete)

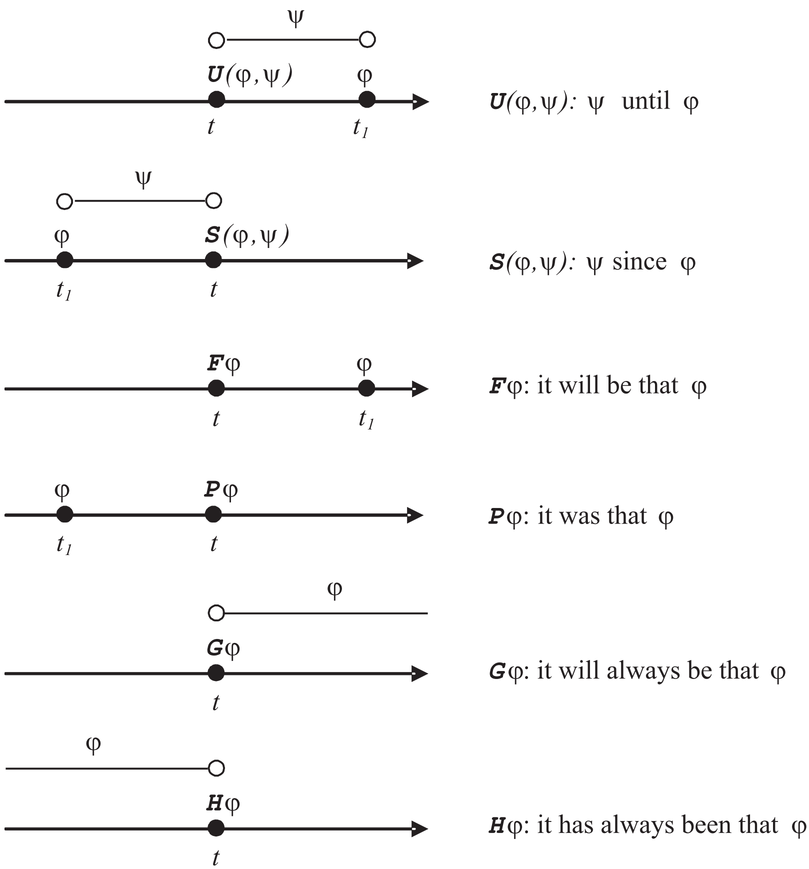


Fig. 1. Graphical Representation of Temporal Connectives

Definition 4.

A *temporal database query* is a formula of QL with at least one free variable. The *answer of the query* φ (denoted by $\varphi(TDB)$) is the temporal relation it generates in the database:

$$\varphi(TDB) = \{((v(x_1), \ldots, v(x_s)), \tau) : TBD, v, t \models \varphi \text{ and } t \in \tau\},$$

where $x_1, \ldots, x_s$ are all free variables of the formulae φ.

6. Queries

We will formulate four queries over the temporal database presented in Example 2 (we will assume that today is 2007–04–03).

Query 1.

Find those who were (but are no longer) patients at the hospital

$$\varphi = (\mathsf{P}\,\mathrm{PATIENTS}(x_1, x_2)) \wedge \neg\mathrm{PATIENTS}(x_1, x_2) \wedge \mathsf{time}(2007\text{–}04\text{–}03)$$

According to definition 4 the answer of the query is the set:

$$\begin{aligned}\varphi(TDB) = \{((v(x_1), v(x_2)), \tau) : TBD, v, t \models (\mathsf{P}\,\mathrm{PATIENTS}(x_1, x_2))\\ \wedge \neg\mathrm{PATIENTS}(x_1, x_2)\\ \wedge\, \mathsf{time}(2007\text{–}04\text{–}03) \text{ and } t \in \tau\}.\end{aligned}$$

From definition 3a and 3b, we have:
$TBD, v, t \models (\mathsf{P}\,\mathrm{PATIENTS}(x_1,x_2)) \wedge \neg\mathrm{PATIENTS}(x_1,x_2) \wedge \mathsf{time}(2007\text{–}04\text{–}03)$

$\Updownarrow$ (def. 3a, p. 5)

(a) $TBD, v, t \models \mathsf{P}\,\mathrm{PATIENTS}(x_1, x_2)$
(b) and $TBD, v, t \models \neg\mathrm{PATIENTS}(x_1, x_2)$
(c) and $TBD, v, t \models \mathsf{time}(2007\text{–}04\text{–}03)$

(a) $TBD, v, t \models \mathsf{P}\,\mathrm{PATIENTS}(x_1, x_2)$

$\Updownarrow$ (def. 3b, p. 7)

$TBD, v, t \models \mathsf{S}(\mathrm{PATIENTS}(x_1, x_2), \top)$

$\Updownarrow$ (def. 3a, p. 8)

there exists $t_1 \in T$ with $t_1 < t$
and $TBD, v, t_1 \models \mathrm{PATIENTS}(x_1, x_2)$
and for every $t_2 \in T$ such that $t_1 < t_2 < t$ holds $TBD, v, t_2 \models \top$

$\Updownarrow$

there exists $t_1 \in T$ with $t_1 < t$ and $TBD, v, t_1 \models \mathrm{PATIENTS}(x_1, x_2)$

$\Updownarrow$ (def. 3a, p. 1)

there exists $t_1 \in T$ with $t_1 < t$ and $(v(x_1), v(x_2), \tau) \in \mathbf{PATIENTS}$,
where $\tau \subseteq T$ and $t_1 \in \tau$

(b) $TBD, v, t \models \neg\mathrm{PATIENTS}(x_1, x_2)$

$\Updownarrow$ (def. 3a, p. 4)

not $TBD, v, t \models \text{PATIENTS}(x_1, x_2)$

$$\Updownarrow \text{(def. 3a, p. 1)}$$

$(v(x_1), v(x_2), \tau) \notin \textbf{PATIENTS}$, where $\tau \subseteq T$ and $t \in \tau$

(c) $TBD, v, t \models \mathit{time}(2007\text{–}04\text{–}03)$

$$\Updownarrow \text{(def. 3a, p. 3)}$$

$t = 2007\text{–}04\text{–}03$

(a), (b) and (c) are satisfied by:

$t_1 = 2007\text{–}03\text{–}01$,
$\tau = [2007\text{–}02\text{–}25, 2007\text{–}03\text{–}01]$,
$v(x_1) = 2$, $v(x_2) = \text{Kozłowski}$

and

$t_1 = 2007\text{–}03\text{–}16$,
$\tau = [2007\text{–}02\text{–}01, 2007\text{–}02\text{–}25] \cup [2007\text{–}03\text{–}15, 2007\text{–}03\text{–}16]$,
$v(x_1) = 1$, $v(x_2) = \text{Kowalski}$, therefore,

$$\begin{aligned}\varphi(TDB) = \{&((1, \text{Kowalski}), [2007\text{–}02\text{–}01, 2007\text{–}02\text{–}25] \cup \\ &\qquad [2007\text{–}03\text{–}15, 2007\text{–}03\text{–}16]), \\ &((2, \text{Kozłowski}), [2007\text{–}02\text{–}25, 2007\text{–}03\text{–}01])\}\end{aligned}$$

Query 2.

Find those who stayed at the hospital more than once

$$\begin{aligned}\varphi = \mathsf{P}\,(&\text{PATIENTS}(x_1, x_2) \wedge \\ &\mathsf{P}\,(\neg\text{PATIENTS}(x_1, x_2) \wedge \mathsf{P}\,\text{PATIENTS}(x_1, x_2))) \wedge \mathit{time}(2007\text{–}04\text{–}03)\end{aligned}$$

$$\begin{aligned}\varphi(TDB) = \{&((1, \text{Kowalski}), \\ &[2007\text{–}02\text{–}01, 2007\text{–}02\text{–}25] \cup [2007\text{–}03\text{–}15, 2007\text{–}03\text{–}16])\}\end{aligned}$$

(It can be shown in an analogous way to the previous query).

Query 3.

When did Kowalski ($id = 1$) stay at the hospital? (in other words: show the past history of the tuple $(1, \text{Kowalski})$)

$\varphi = \text{PATIENTS}(1, x)$

$$\begin{aligned}\varphi(TBD) = \{&((1, \text{Kowalski}), \\ &[2007\text{–}02\text{–}01, 2007\text{–}02\text{–}25] \cup [2007\text{–}03\text{–}15, 2007\text{–}03\text{–}16])\}\end{aligned}$$

Query 4.

Find those who were admitted to hospital between 2007–01–01 and 2007–04–03

$$\varphi = \mathsf{time}(2006\text{–}12\text{–}31) \wedge \neg \mathrm{PATIENTS}(x_1, x_2) \wedge \\ \mathsf{F}(\mathrm{PATIENTS}(x_1, x_2) \wedge \mathsf{F}(\mathsf{time}(2007\text{–}04\text{–}04)))$$

$$\varphi(TBD) = \{((1, \mathrm{Kowalski}), \\ [2007\text{–}02\text{–}01, 2007\text{–}02\text{–}25] \cup [2007\text{–}03\text{–}15, 2007\text{–}03\text{–}16]), \\ ((2, \mathrm{Kozłowski}), [2007\text{–}02\text{–}25, 2007\text{–}03\text{–}01]), \\ ((5, \mathrm{Piasecka}), \\ [2007\text{–}02\text{–}20, 2007\text{–}03\text{–}05] \cup [2007\text{–}04\text{–}01, 2007\text{–}04\text{–}16])\}$$

References

[ChoTom] Chomicki J, Toman D.: *Temporal Logic in Information Systems.* In *Logics for Databases and Information Systems*, Kluwer Academic Publishers, 1998, pp. 31–70.

[Cod] Codd E. F.: *A Relational Model of Data for Large Shared Data Banks.* Communications of the ACM, 1970, 13(6), pages 377–387.

[Dat] Date C. J.: *Introduction to Database System*, Addison-Wesley, 2003.

[Etz] Etzion, Jajodia, Sripada. *Temporal Databases: Research and Practice.* Springer 1998.

[Gab] Gabbay D. M., Reynolds M. A., Finger M.: *Temporal Logic, Mathematical Foundations and Computational Aspects*, *Vol 2.* Clarendon Press. Oxford 2000.

[Jen] Jensen, C. S.: *Temporal Database Management*, PhD Thesis, Department of Computer Science, Aalborg University, 2000 [http://www.cs.aau.dk/~csj/Thesis/].

[JenSno] Jensen C. S., Snodgrass R. T.: *Semantics of Time-Varying Information*, Information Systems, 21(4), 1996, pp. 311–352.

[Sno] Snodgrass R., editor. *The TSQL2 Temporal Query Language.* Kluwer Academic Publishers, 1995.

[Ste] Steiner A.: *A Generalisation Approach to Temporal Data Models and Their Implementations*, PhD Thesis, Departement Informatik, ETH Zurich, Switzerland, 1997.

[Tan] Tansel A. U., Clifford J., Gadia S., Jajodia S., Segev A., Snodgrass R.: *Temporal Databases: Theory, Design, and Implementation*, Benjamin-Cummings Publishing Co., Inc., 1993.

Dariusz Surowik
University of Białystok

A FEW REMARKS ON QUERYING LISTS, TREES AND DAGS A TEMPORAL-LOGIC APPROACH[1]

Abstract. This paper discusses various conceptions of query languages for database systems (object oriented), in which objects are lists, trees and directed acyclic graphs. I use temporal logic as a modelling tool for the query languages under consideration. In the case of query language regarding lists, I will discuss temporal logic constituting an extension of linear time temporal logic, whereas in the case of query language regarding trees and directed acyclic graphs, I will discuss temporal logics constituting extensions of the branching–time temporal logic, the so called computation tree logic (CTL).

Key words: temporal logic, data models, query language

1. Introduction

Lists, trees and graphs are among the most important data structures in informatics. Computer linguistics or text databases are typical fields of database applications where trees, for example, are useful as a data type. In these applications, trees can serve as a tool for modelling a description of integrated concepts (for example, as a result of syntactic analysis) or for representing document structure. Although these structures are very important as tools for modelling, they have hardly been used so far as data types in existing databases. Standard database systems offer only sets of suitable relations as a way of representing unit sets and simple data types, such as whole numbers or strings, for representing unit attributes.

The present paper discusses lists, trees and directed acyclic graphs, as well as formal tools offered by temporal logic for searching the aforemen-

[1] The research reported in this paper is part of the project entitled *Temporal Representation of Knowledge and Its Implementation in the Computer Systems of Medical Conducts* supported by the Polish Ministry of Science and Higher Education, grant no. 3 T11F 011 30.

ISBN 978–83–7431–158–8 ISSN 0860-150X

tioned data structures. Applying temporal logic allows one to easily construct queries, which would be difficult to express without taking a temporal context into account. For example, a temporal modality such as "always" can be used to express the dynamic utterance "The wages of the workers on the list never decrease". One can, of course, express the same in a sense in a 'static' way as, for example, "the workers' wage list is ordered according to wages". It seems, however, that the use of temporal context makes the utterance more interesting.

2. Querying lists

In order to correctly define the syntax of temporal logic used for formulating queries in structures which have a form of lists, we must specify what we mean by the notion of an *atomic formula*. The correct definition of this concept turns out to be problematic, however. The problem lies in the fact that the elements of a list may have a very complex structure. What an atomic formula actually is depends therefore on the type of data to which the list has been applied. Thus, the definition of an atomic formula should be different when the elements of the list are some numerical values and different when the list has sets or families of sets as its elements. An *atomic formula* is a predicate which can be directly evaluated by referring to an individual element of a list. I am aware of the fact that this is not a precise definition of the concept in question. An atomic formula is a syntactic notion and should be described in syntactic terms. Unfortunately, the only thing we can say at this point about an atomic formula from a syntactic point of view is that it does not comprise logical operators[2]. An example of an atomic formula is the formula $(\circ < 2)$. This formula is true if the value of the current element of the list is smaller than 2.

In the language $\boldsymbol{L}$ of our logic we have the connectives $\neg, \vee$ temporal operators S (Since), U (Until), $\circ$ (Next) and $\bullet$ (Previous), and the binder operator $\uparrow$.

Well formed formulas we may define as follows:

- Atomic formula is well formed formula,
- If α, β are well formed formulas, then so are $\neg\alpha$, $\uparrow\alpha$, $\circ\alpha$, $\bullet\alpha$, $\alpha \vee \beta$, $\alpha U \beta$, $\alpha S \beta$.

[2] In some temporal logic systems an atomic formula is simply a formula which does not contain temporal operators.

SYNTAX

Definition 1

A list is a string of n objects $L[1], L[2], \ldots, L[n]$, where $n > 0$.

The value of the i-th element of the list will be marked as $L[i]$, whereas the i-th element of the list will have the symbol $\langle L, i\rangle$. In order to refer to the current position on the list, I will use the symbol $\downarrow$.

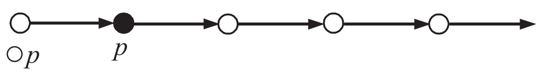
The truthfulness of the formula is defined as the truthfulness at the i-th point of the list L.

Definition 2

For any list L, for any $1 \leq i \leq n$ and for any formula α holds:

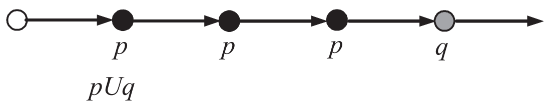
1) $\langle L, i\rangle \models \alpha \equiv$ α is an atomic formula and evaluates to true for $L[i]$,
2) $\langle L, i\rangle \models \neg\alpha \equiv \langle L, i\rangle \not\models \alpha$,
3) $\langle L, i\rangle \models (\alpha \vee \beta) \equiv \langle L, i\rangle \models \alpha$ or $\langle L, i\rangle \models \beta$,
4) $\langle L, i\rangle \models \circ\alpha \equiv$ if $1 \leq i < n$, then $\langle L, i+1\rangle \models \alpha$,
5) $\langle L, i\rangle \models (\alpha U \beta) \equiv \exists_{j \geq i}$ such that $\langle L, j\rangle \models \beta$ and $\forall_k$ (if $i \leq k < j$, then $\langle L, k\rangle \models \alpha$),
6) $\langle L, i\rangle \models \bullet\alpha \equiv$ if $1 < i \leq n$, then $\langle L, i-1\rangle \models \alpha$,
7) $\langle L, i\rangle \models (\alpha S \beta) \equiv \exists_{j \leq i}$ such that $\langle L, j\rangle \models \beta$ and $\forall_k$ (if $j < k \leq i$, then $\langle L, k\rangle \models \alpha$),
8) $\langle L, i\rangle \models \uparrow x\alpha \equiv \langle L, i\rangle \models \alpha^x_{L[i]}$.

Operators $\circ$, $\bullet$ are the operators of the next and the previous respectively. The notation $\langle L, i\rangle \models \circ\alpha$ is understood as follows: *a formula α is fulfilled for the $i+1$-st element of the list*. The graphic interpretation of the notation $\langle L, 1\rangle \models \circ p$ is shown below.



In the language of the logic under consideration, the sentence *the follower of the current element of the list is negative* can be written as $\circ(\downarrow < 0)$.

Operators U and S are known operators *until* and *since*. The formula $\alpha U \beta$ is understood as follows: *formula β is true at a certain moment in the future, let us call this moment j, whereas formula α is true from the* *current moment till the moment j*. The graphic interpretation of the notation $\langle L, 2\rangle \models (p\,U q)$ is shown below.



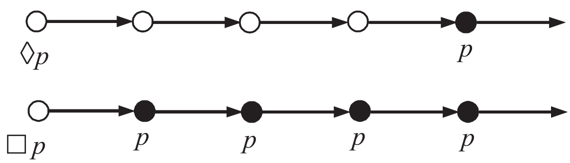
The notation $L \models \alpha$ will be used as an abbreviation for the notation $\langle L, 1\rangle \models \alpha$. If $L \models \alpha$ obtains, we will say that list L is the model for formula α. Let us note that operators $\circ, \bullet$ are weak operators, i.e., formula $\circ\alpha$, $(\bullet\alpha)$ is true in the last (first) element of the list, regardless of the form of formula α. This is the so called empty fulfilment, since, as follows on from the definition of fulfilment respectively for operators $\circ, \bullet$ in the first (last) element of the list the predecessor of the corresponding implication is false.

With the help of the previously defined operators, we can introduce additional specific temporal operators.

Definition 3

1) $\bar{\circ}\alpha \;\equiv\; \neg\circ\neg\alpha$ (strong next)
2) $\bar{\bullet}\alpha \;\equiv\; \neg\bullet\neg\alpha$ (strong previous)
3) $\Diamond\alpha \;\equiv\; true\ U\alpha$ (eventually in the future)
4) $\blacklozenge\alpha \;\equiv\; true\ S\alpha$ (eventually in the past)
5) $\Box\alpha \;\equiv\; \neg\Diamond\neg\alpha$ (always in the future)
6) $\blacksquare\alpha \;\equiv\; \neg\blacklozenge\neg\alpha$ (always in the past)

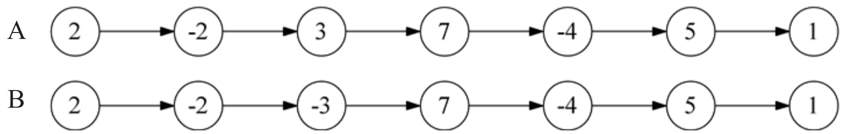
Examples of the graphic interpretations of the chosen operators are presented below.



Formula $\bar{\circ}\alpha$ is true at the i-th point of list L if the i-th point of the list is not the last point and α is fulfilled at point $i + 1$.

Example 1

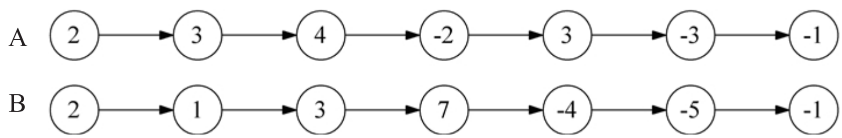
a) The sentence *The list does not contain two neighbouring negative elements* can be written symbolically as: $G((\downarrow < 0) \Rightarrow \circ(\downarrow \geq 0))$



The model for the formula under consideration is list A; while list B is not the model for this formula due to the value of the second and third element of the list. In the language of first order classical logic the above formula can be written as follows:

$$\neg\exists_{1\leq i<n}\,(L[i] < 0 \wedge L[i+1] < 0)$$

b) The expression $(\downarrow > 0) \wedge \bar{\circ}((\downarrow > 0)\, U \Box(\downarrow < 0))$ is a formal notation of the statement that
The first element of the list is positive. The following elements are also positive until a certain element is reached whose value is negative. All the elements following the element that has negative value, also have negative value. (We can thus state that the change in the value sign of the elements of the list has occurred only once).



List A is not a model for this formula, because the change of the sign of the elements of the list occurs twice. List B, on the other hand, constitutes the model for the formula under consideration since the first element of the list is positive and the change in the sign of the elements of the list occurs only once. In the language of first order classical logic the formula under consideration can be written as follows:

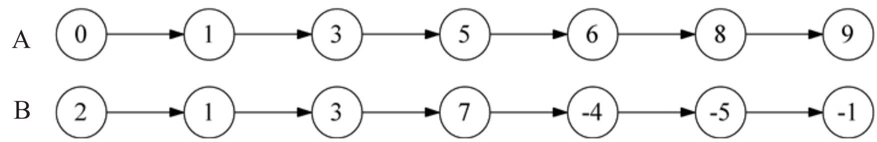
$$L[1] > 0 \wedge \exists_{j>1}\,(\forall_{1\leq i<j}\; L[i] > 0 \wedge \forall_{i>j}\; L[i] < 0)$$

So far we have considered expressions in which individual list values are compared to some constant arbitrarily chosen value. However, in the case of a query of the type *Is the list monotonously growing?*, we have to compare a number of values of the elements of the list to one another. To this end we shall use the operator $\upharpoonleft$. By formula $\alpha^{x}_{L[i]}$ we understand a formula obtained from formula α by replacing all free occurrences of variable x in formula α with the value $L[i]$. According to definition 2, we should understand formula $\upharpoonleft x\alpha$ in the same way.

Example 2

a) Formula $\Box(\uparrow x\circ(\downarrow> x))$ is a formal notation of the statement that *the list is monotonously growing.* (In the language of first order classical logic the above formula can be written as follows: $\forall_{1\leq i<n}\ (L[i+1] > L[i])$).

b) Formula $\Box(\uparrow x\neg\circ\Diamond(\downarrow= x))$ is understood as: *the list does not contain two identical elements.* ($\neg(\exists_{1\leq i\leq n}\ \exists_{1\leq j\leq n}\ (i\neq j \wedge L[i] = L[j]))$ in the language of first order classical logic).



List A is a model for the formula $\Box(\uparrow x\circ(\downarrow> x))$, while both list A and list B are examples of models for the formula $(\Box(\uparrow x\neg\circ\Diamond(\downarrow= x))$.

3. Querying trees

Let us consider an unordered tree $T = (V, E, \lambda)$. V is a set of nodes of tree T, E is a set of edges of tree T, while λ is a function assigning to each node v of tree T a value from a certain domain D and thus $\lambda : V \to D$. Analogously to lists, $T[v]$ denotes a value in node v of tree T, while by $\langle T, v\rangle$ we understand node v of tree T.

In order to analyse lists with the help of temporal logic, it is sufficient to adopt notions of temporal logic regarding linear time (PLTL). In linear time there is only one future (just like in list type structures) and each element can have only one follower. It is different in the case of tree-like structures, whose elements can have several followers. The future in such structures can branch. Hence, for tree analysis one should use notions created for the purpose of temporal logic of branching time. A temporal logic which, after small modifications, can be used to construct the language of queries for tree-like structures is CTL* logic. The main idea of CTL* logic was introducing the so called path operators E, A. Formula $E\alpha$ should be understood as *There is a path starting at the current node such that formula α is fulfilled in this path.* Formula $A\alpha$, on the other hand, should be understood in the following way: *In all the paths starting at the current node formula α is fulfilled.*

A temporal logic which was to serve the purpose of tree analysis was proposed by Peter Becker [1]. Because it is a construct very similar to the known

CTL* logic, the author of the system in question uses the symbol CTL$^*_{DB}$ for his logic.

Just as in the case of lists, this logic assumes that there is a set of atomic formulae. In CTL$^*_{DB}$ logic, however, we deal with two types of formulae, i.e. *point formulae* and *path formulae*. Point formulae are connected to nodes and their truthfulness or falsehood is verified with reference to a given node. Path formulae, on the other hand, are connected to paths and their truthfulness or falsehood is verified in relation to a given path.

The syntax of CTL$^*_{DB}$ is as follows:

Definition 4

a) State formulas
   1. Each atomic formula is a state formula,
   2. If α and β are state formulas, then $\alpha \vee \beta$, $\neg\alpha$ are state formulas,
   3. If α is a patch formula, then $E\alpha$, $A\alpha$ are state formulas.

b) Patch formulas
   1. Each state formula also is a patch formula,
   2. If α and β are patch formulas, then $\alpha \vee \beta$, $\neg\alpha$ are patch formulas,
   3. If α and β are patch formulas, then so are: $\alpha\, U \beta$, $\alpha\, S \beta$, $\circ\alpha$, $\bullet\alpha$.

c) The set of state formulas generated by the above rules forms the language of CTL$^*_{DB}$ logic.

Before presenting the semantics of CTL$^*_{DB}$ logic, let us define the notion of maximal path for a given node.

Definition 5

Let v be a node of tree T. Path $p = (v, v_1, v_2, \ldots, v_{r_p})$ of tree T is called the *maximal path for node v* if and only if v_{r_p} is a leaf.

The definition of truthfulness for CTL$^*_{DB}$ logic is the following:

Definition 6

a) STATE FORMULAS
   For each tree T, for each node v and for each state formulas α, β
   1) $\langle T, v\rangle \models \alpha \quad \equiv \quad \alpha$ evaluates to *true* for $T[v]$,
   2) $\langle T, v\rangle \models \neg\alpha \quad \equiv \quad \langle T, v\rangle \not\models \alpha$,
   3) $\langle T, v\rangle \models (\alpha \vee \beta) \quad \equiv \quad \langle T, v\rangle \models \alpha$ or $\langle T, v\rangle \models \beta$,
   4) $\langle T, v\rangle \models E\alpha \quad \equiv \quad \exists_{p=(v,v_1,v_2,\ldots,v_{r_p})}\ p$ is maximal path, such that $\langle T, p, 1\rangle \models \alpha$,
   5) $\langle T, v\rangle \models A\alpha \quad \equiv \quad \forall_{p=(v,v_1,v_2,\ldots,v_{r_p})}\ p$ is maximal path, such that $\langle T, p, 1\rangle \models \alpha$.

b) PATH FORMULAS
For each tree T, for each maximal path $p = (v, v_1, v_2, \ldots, v_{r_p})$, for each $1 \leq i \leq r$ and for each path formulas α, β holds:

1) $\langle T, p, i\rangle \models \alpha \quad \equiv \quad \langle T, v_i\rangle \models \alpha$,
2) $\langle T, p, i\rangle \models \neg\alpha \quad \equiv \quad \langle T, p, i\rangle \not\models \alpha$,
3) $\langle T, p, i\rangle \models (\alpha \vee \beta) \quad \equiv \quad \langle T, p, i\rangle \models \alpha$ or $\langle T, p, i\rangle \models \beta$,
4) $\langle T, p, i\rangle \models \circ\alpha \quad \equiv$ if $1 \leq i < r$, then $\langle T, p, i+1\rangle \models \alpha$,
5) $\langle T, p, i\rangle \models (\alpha\, U \beta) \quad \equiv \quad \exists_{j \geq i}$ such that $\langle T, p, j\rangle \models \beta$ and $\forall_k$ (if $i \leq k < j$, then $\langle T, p, k\rangle \models \alpha$),
6) $\langle T, p, i\rangle \models \bullet\alpha \quad \equiv$ if $1 < i \leq r$, then $\langle T, p, i-1\rangle \models \alpha$,
7) $\langle T, p, i\rangle \models (\alpha\, S \beta) \quad \equiv \quad \exists_{j \leq i}$ such that $\langle T, p, j\rangle \models \beta$ and $\forall_k$ (if $j < k \leq i$, then $\langle T, p, k\rangle \models \alpha$),

Example 3

In the language of CTL^*_{DB} logic, the formulation:

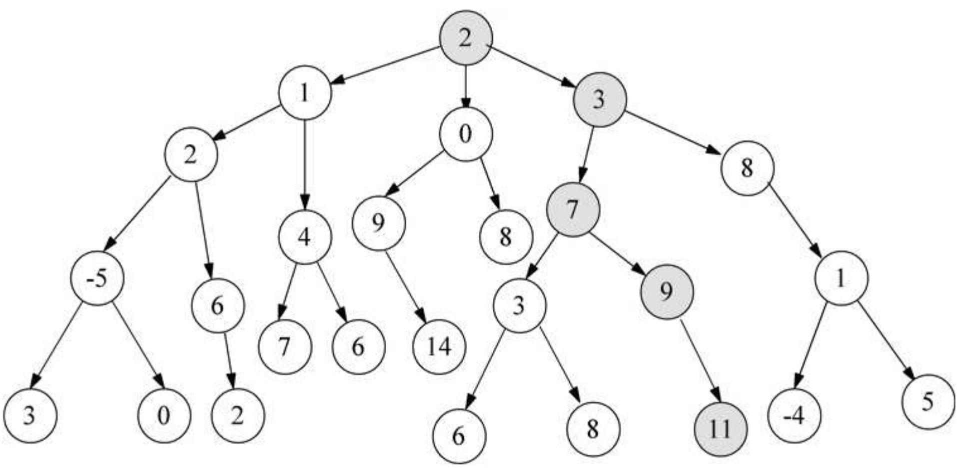
a) *There is a monotonously growing path* can be written symbolically as:

$$E\,(\Box\,(\uparrow\! x \circ (\downarrow > x))).$$

In the language of the first order logic the formula under consideration can be written as follows:

$$\exists_{p=(v_1, v_2, \ldots, v_{r_p}) \in T}\ \forall_{1 \leq i < r}\ (T[p, v_i] < T[p, v_{i+1}]).$$

An example of a tree-like structure, which is a model of the formula under consideration, is presented in the drawing below.



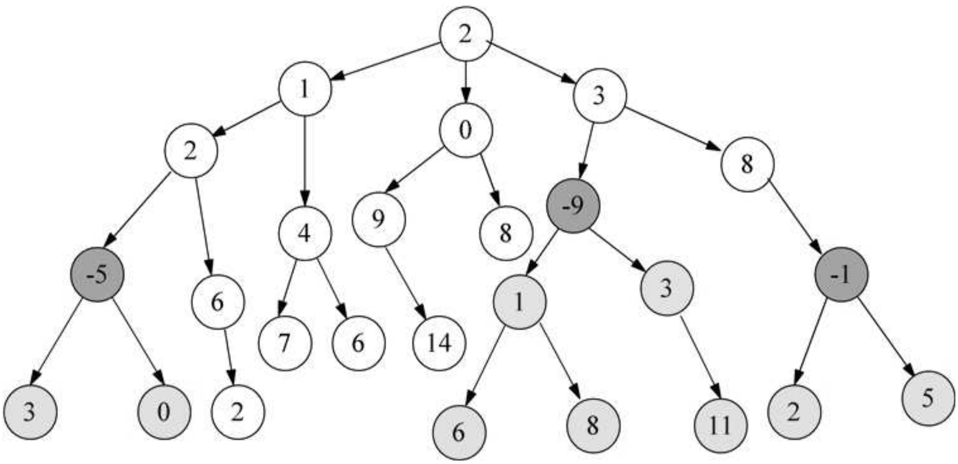
b) The statement that *Every node with a negative value has only non-negative children* can be written symbolically in the formula

$$A\,(\Box\,(\downarrow< 0) \rightarrow A(\circ(\downarrow\geq 0)))).$$

In the language of the first order logic the formula under consideration can be written as follows:

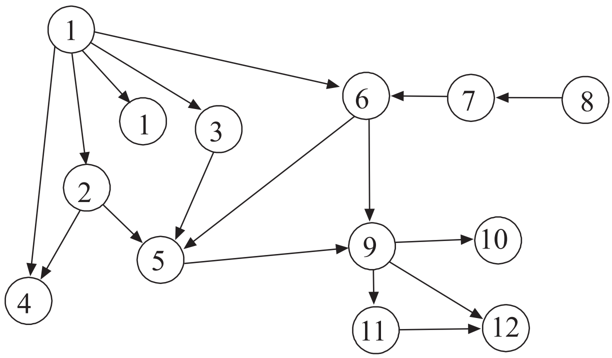
$$\forall_{p=(v_1,v_2,\ldots,v_r)\in T}\ ((\exists_{1\leq i<r}\ T[p,v_i] < 0) \Rightarrow$$
$$\Rightarrow \forall_{p'=(v_{1'},v_{2'},\ldots,v_{r'})\in T}\ ((\exists_{1\leq j<r'}\ v_i = v_j(\in p')) \Rightarrow (T[p',v_{j+1}] \geq 0))).$$

An example of a tree-like structure, which is a model of the formula under consideration, is presented in the drawing below:



4. Querying directed acyclic graphs

Analogical constructions can be created for the purpose of graph analysis. If a graph is coherent and undirected, it is equal to a tree, hence, in order to analyse such a graph we can use the tools discussed in the previous part of the paper. In the case of *directed graphs* (i.e. structures of the form $G = \langle V, A\rangle$ where: V is a set of vertices, A is a set of ordered pairs of various nodes from set V, called *directed edges*, or *arcs*: $A = \{(u,v), u, v \in V\}$), we will consider only *acyclic graphs*, i.e., directed graphs that do not have cycles. An example of such a graph is presented below.



Considering only this type of graph guarantees that there is a finite number of maximal paths. Unfortunately, contrary to trees, where the number of maximal paths from a given node to any leaf was limited by the number of nodes of a given tree, in graphs, the number of such paths grows exponentially. For this reason in the construction of the language of queries for directed acyclic graphs, one can use only the notions employed in CTL logic. It is only point formulae that are considered in this logic. A temporal logic which was to serve the purpose of directed acyclic graph analysis was proposed by Peter Becker [1], who marks the logic in question as CTL_{DAG}.

Unfortunately there are several problems connected with effective model verification for formulae of CTL_{DAG}. If, for example, G is an acyclic digraph with n nodes, then model verification for formula $\Phi\alpha$ of CTL_{DAG} logic (where Φ is any operator of this logic) may be performed by testing formula α for at least n nodes.

5. Conclusion

This paper discussed selected constructions of temporal logic systems, which allow forming queries for the purpose of analyzing lists, trees or acyclic digraphs. The constructions of these systems were based on CTL [3] logic or on CTL* logic. The use of temporal logic formalism for the analysis of the aforementioned data structures has its justification in well developed formal tools connected to temporal logic. The formalism of temporal logic expresses intuitive notions such as ‘always’, ‘possibly’, ‘from now on...’ well. This formalism, however, has certain limitations, which is especially visible when one considers issues relating to graph analysis, but not only. Even in the case of simpler data structures, such as lists, not everything can be expressed by means of temporal logic. For example, it is impossible to formulate a query regarding only those elements of a list which occupy even positions [6].

Moreover, as we showed, some of queries are more complicated in the language of the first order classical logic in compare to the formulation in the language of temporal logic. From the other hand, there are some queries, which are simpler in the language of the first order classical logic.

Bibliography

[1] Becker P., *A temporal Logic based approach fro querying lists, trees and dags in databases*, vol. 978 of Lecture Notes in Computer Science, pp. 293–302, Springer, 1995.

[2] Cattel R. G. G., editor, *The object database standard*, ODMG-93. Morgan Kaufmann, 1994.

[3] Emerson E. A., *Temporal and modal logic*, Handbook of theoretical computer science, Elsevier, 1990.

[4] Richardson J., *Supporting lists in a data model (a timely approach),* Proceedings of the 18$^{\text{th}}$ VLDB Conference, 1992.

[5] Subramanian B., Zdonik B., Leung T. Vandenbergy S., *Ordered types in the AQUA Data Model*, Proc. 4$^{\text{th}}$ Intl. Workshop on Database Programming Languages, 1993.

[6] Wolper P., *Temporal logic can be more expressive*, Information and control, vol. 56, 1983.

[7] Yamamoto M, Tanabe Y, Takahashi K., Hagiya M., *Abstraction of Graph Transformation Systems by Temporal Logic and Its Verification.*

Witold Marciszewski
University of Białystok

COMPUTATIONAL DYNAMICS OF COMPLEX SYSTEMS

A NEW WAY OF DOING SCIENCE

A comment on *Thinking in Complexity. The Computational Dynamics of Matter, Mind and Mankind* by Klaus Mainzer, Springer-Verlag 2004 (4th Edition). Polish translation: *Poznawanie złożoności. Obliczeniowa dynamika materii, umysłu i ludzkości*, Wydawnictwo UMCS, Lublin 2007, translated by a team run by Marek Hetmański.

There is a good message for those Polish philosophers who realize the enormous philosophical import of *thinking in complexity*: the mentioned book by Klaus Mainzer has become more available to Polish readers owing to the translation done by a team of philosophers and translators at UMCS (Maria Curie Skłodowska University) in Lublin, Poland.[1]

To account for why I regard this message so good, let me recall the following. The way of perceiving the world which in the book is aptly called *thinking in complexity* is something like a creeping revolution both in science and in philosophy. It is a revolution, indeed, as it radically changes our world perspective. It is creeping as no single great event announced its start, and even no single branch of learning might be named as its proper terrain, hence a specialist in one field alone may overlook its emergence and significance.

[1] At the start, let me hint at the intention of these comments. I do not intend to offer a paper in which the book under review gets examined for the correctnes of its statements and methods. This would require from the reviewer a level of expertise comparable with that found in the book, and this is not the case in question. Instead, I take the attitude of an appreciating reader who wishes to encourage fellow philosophers to make acquaintance with the book, as well as address the Author with certain issues. So my text distinguishes some items being attractive from a philosophical point of view, and puts some questions concerning philosophy of mind and epistemology. A special item might be devoted to the quality of Polish translation, but such an assessment should be made (and will, hopefully, be) in Polish.

ISBN 978–83–7431–158–8 ISSN 0860-150X

The revolutionary initiative is due to mathematical logic at this point from which theoretical computer science (informatics) has emerged, that is, the discoveries concerning computability contributed by Gödel, Turing, Post and companions. This, so to speak, software complexity is deeply entangled with the complexity of hardware, the latter meaning dynamic systems changing in time as studied by physics, technology, biology etc.

Quite a number of Polish philosophers and mathematicians are familiar with the problems of computability, owing to contributions to this field as made by their natives Alfred Tarski, Andrzej Mostowski, Andrzej Grzegorczyk and others. That is to say, they are fairly familiar with the software side of the complexity. However, a knowledge of complex dynamic systems, and of how these are related to computability studies, is far from being eminent among our philosophers. Thus the translation of Professor Mainzer's book – to be, hopefully, duly disseminated in academic circles – should assist Polish scholars in their endeavours to keep in line with current science.

In what follows, (1) I introduce some key concepts in a way which should fit into the Polish philosophical audience's interests and conceptual equippment. Then, (2) I briefly survey the book content to let prospective readers know what they may expect from it. At last (3) I put some questions of how the study of complexity can profit from some promising research in automated theorem proving.

1. Some notions of complexity historically explained

1.1. Klaus Mainzer belongs to that circle of scholars who claim the rise of the new science of complexity. That circle includes Stephen Wolfram whose monumental book (2002) bears the much speaking title *A New Kind of Science* to resemble Galileo's phrase *nuove scienze* (in the title of his famous treatise of 1638). Such a claim alludes to the notions of *new paradigm* and *scientific revolution*, as introduced by Thomas Kuhn, which entered the vernacular of historians and philosophers of science. Hence it is reviewer's task to scrutinize to what extent such a far-reaching claim can be justified with reliable evidence. This involves an assessment of the essential point, opening the Author's Preface to the Fourh Edition; it runs as follows.

> The first edition of this book, published in 1994, began with the statement that the science of complexity would characterize the scientific development of the 21th century. In the first decade of this century, the prediction has been confirmed by overwhelming new empirical results and theoretical insights of the physical and biological sciences, cognitive and computer sciences, and the

> social and economic sciences. Complexity and nonlinearity are still prominent features in the evolution of matter, mind and human society. Thus, the science of complexity still aims at explanation for the emergence of order in nature and mind and in the economy and society by common principles.

This opening paragraph puts in a nutshell what the book is to tell. It should be read jointly with the book's title. In this title *Complexity*, as the category *in* which one should *think* about the world, is linked with *Computational Dynamics* of the universe, which is meant as involving *Matter, Mind, and Mankind.*

Before going deeper into the subject, it will be in order to address some conceptual troubles in our thinking about complexity.[2]

This is the very concept of 'complex system' as liable to be burdened with too many roles and a kind of redundancy. When speaking of a *system*, one thinks about an object having a number of elements which are in a way interrelated, hence a complex object. Then the phrase 'complex system' would mean 'complex complex object'. To avoid such a redundancy, let us distinguish the two following notions of complexity.

(1) The most general notion refers to any system *qua* system, that is, any set having elements interrelated with one another. A system may be either (A) abstract and static, as are the domains of mathematical theories, computer programs, etc, or (B) dynamic, that is, changing in time as are bodies (except, presumably, elementary particles), ecosystems, minds, societies, etc.

(2) A more specific notion refers only to dynamic systems (mentioned above in 1 as B), and – moreover - just those among them which behave in a way that we call nonlinear; this is a feature that makes a system complex in this more restricted sense.

Note, besides those listed in (2) there are dynamic systems that possess the feature of linearity, hence behave in a regular and predictable way, like our old good earth, also its companion the moon (this is why we happen to be so successful in predicting eclipses); those do not deserve to be called complex in this special sense. As being systems, they are complex in that

[2] *'When I make a word do a lot of work like that', said Humpty Dumpty, 'I always pay it extra.'*. This was the replay to Alice's: *The question is whether you can make words mean so many different things.* (Lewis Carroll, *Through The Looking Glass*). This is exactly what happened to the term *complexity* in academic vernacular. It has got overworked with too many meanings, so there is a need to distinguish among them. A specially needed distinction is between the complexity of a dynamic systems due to its nonlinearity, and the complexity of a static system which depends on a number of elements and their interrelations. In the latter case (here under discussion) it is convenient to make use also of the comparative form, and so be allowed to talk about greater or smaller complexity.

most general sense mentioned in (1), hence we need a term to distinguish their type of complexity from that characteristic of nonlinear systems. For this purpose, we may devise the phrase *rudimentary complexity*.

In what follows, I am to use the single term *system* to refer to objects characterized by complexity in the most general sense as defined in (1) above, that is, embracing both rudimentary complexity and that possessed by nonlinear systems. Let the phrase *dynamic system* refer to any systems changing in time, and let the phrase *complex dynamic systems* denote those dynamic systems which are nonlinear.

1.2. Since I wish to encourage philosophers to take advantage of Mainzer's work let me start from their favourite phrase 'already the ancient Greeks' to mention two ancient insights, that of the Atomists and that of the Stoics.

Either of them contributed to the truth that the dynamics of the universe depends on two factors, to wit hardware and software. And each of them, when contributing one half of this truth to the picture, at the same time, ignored the other part. While the Atomists explained the universe in terms of hardware (atoms and space) alone, with Stoics the whole dynamics of the universe was explained by what might be compared to a software.

According to the Stoics, there is a ubiquitous system of non-physical units called in Greek *logoi spermatikoi*, in Latin (Augustinus) *rationes seminales*, what is being rendered in English as 'seeds of reason', 'germinating ideas', or else 'seminal plans'; the last phrase is the fittest for this discussion. While human-made things are produced according a plan devised outside these things themselves, to wit by a human mind, the things in Nature possess the designs of their construction and evolution inbuilt somehow inside. Such an idea might have come from observing seeds of plants, as suggested by the adjective 'seminal'. The tenets of either side can be summed up as follows.

(1) Atomists: the more complex a system is, that is, the more it involves elements and their interrelations as a physical object, the more problems it is able to solve.[3]

(2) Stoics: the ability of problem-solving depends on the kind and size of the seminal plan that is responsible for the development of the system in question, that is, the processes of solving its vital problems.

[3] As to the problem difficulty, it may be measured with the number of partial problems, down to single lines of a program or a proof, whose solutions are steps in the way towards the target solution, as can be seen in proving theorems. Thus the degree of difficulty reduces to the multiplicity of problems.

Now let us compare either position with the following statements.

(1*) The more complex a processor – *hardware* – is, that is, the more it involves elements and their interrelations, the more problems it is able to solve (as can be seen in the example of high-scale integration devices).

(2*) The ability of problem-solving depends on the kind and size of program – *software* – that is responsible for the solving of problems by the system in question.

When comparing 1 with 1*, we realize that the Atomists grasped a rudimentary level of complexity, the same which is exemplified with such structures as electronic chips. What they did not succeed to grasp will be discussed a bit further.

When comparing 2 with 2*, we realize that the Stoics did not imagine any quantitative estimations of the power of plan relative to its complexity. This can be expressed by recalling that the great German pioneer of computer science Konrad Zuse introduced the concept of program under the German name *Plankalkül*. Thus we notice that the Stoics had an idea of plan, without having any idea of calculus. This sheds light on the giant distance between modern science and its philosophical anticipations.

From the Atomists up to the second half of the 20th century their rudimentary conception of complexity dominated both in philosophy and in science. Physical systems being complex in that manner, easily tractable with linear equations, were what Newtonian physics dealt with. A more refined concept appeared with inquiries into the role of feedbacks in dynamic systems; this led to paying special attention to non-linear dependencies. Let the link between positive feedback loops and the phenomenon of non-linearity be explained with the following example. It exemplifies how some unpredictable processes emerge in systems which previously behaved in a predictable way.[4]

> Imagine a microphone which induces a loud squeal from a speaker when the microphone gets too close to the speaker. The positive feedback occurs because the sound picked up from the microphone is amplified, sent out through the speaker and returns to the microphone to be picked up louder than before. Now imagine a system consisting of many microphones randomly connected with wires, as well as many speakers. The probability of the emergence of powerful feedbacks increases as more elements are added, more interconnections are

[4] The example is taken from the text *Multicellular Computing: Dynamic Complexity*, see: evolutionofcomputing.org/Multicellular/DynamicComplexity.html.

> added, or the elements themselves become more complex and therefore can interact with others in more complex ways. Thus, any change to the system that increases the number of possible feedback loops increases the probability of such an emergent phenomenon.

This is a nice example since it exemplifies both feedback loops and transition from linear to non-linear process. Let us note, there is an interval of distances between a speaker and his microphone in which the squeal does not appear, and thus there holds a linear dependence between the distance in question and the sound power: the closer the microphone, the more intense is the sound, but without any unexpected events – up to a certain point beyond which a squeal emerges, and gets more and more intense; thus the process, so far having been linear, starts to be non-linear.

2. A survey of contents

2.1.[5] The introductory chapter emphasises the novelty of the theory of non-linear complex systems as well as its successes in problem solving in natural and in social sciences. The novelty consists in discovering and explaining the feature of 'emergence of certain macroscopic phenomena via nonlinear interactions of microscopic element in complex systems'. This new approach opposes the paradigm of reductionism which the Author exemplifies, with regard to mental and social phenomena, by mechanistic explanation as offered by Hobbes, Lamettrie etc.

According to the traditional paradigm of science, including the Newtonian mechanics, our macroscopic world would be – as a rule – linear, while nonlinearities would be negligible exceptions. Contrary to that view, recently we start to realize (what Poincare anticipated a century ago) that as a rule we deal with nonlinear systems, while linear ones are exceptional. In the macroscopic world of quanta, in spite of certain aspects of linearity, the quantum world is not linear in general. Thus, the picture of the whole science as seen by the Author is like Stephen Wolfram claim (see 1.1 above) that we enter into the age of new science.

2.2. In the next chapter we encounter a feature of the book, characteristic of the next chapters too, namely a combine of philosophical interpretations, merged in the history of philosophy, with technical discussions involving

[5] In the numbering in this section, each second digit corresponds to the so numbered chapter in the book.

a considerable knowledge from various fields of science: not only Newtonian physics but also quantum physics, thermodynamics, chemistry, biology, economics etc., supported by a necessary piece of mathematical apparatus. This is a reason, indeed, to appreciate the Author's expertize, and at the same time to caution philosophers, who as a rule hardly share such a competence, that the reading may appear a bit stressing. However, this by no means should discourage prospective readers, since a substantial lot of knowledge and understanding can be obtained from the book as a whole, in spite of local difficulties.

This chapter, entitled 'Complex Systems and the Evolution of Matter' starts from a historical background, dealing first with philosophical anticipatory insights of Aristotle and Heraclitus concerning the question: 'how can order arise from complex, irregular, and chaotic states of matter?' Another historical survey outlines the picture of deterministic and linear universe as found in Newton, Einstein and Laplace. The next sections reveal various departures from determinism and linearity.

A section most attractive from a philosophical point of view deals with the question of the emergence of order in cosmic evolution. No definitive answer is available at the current stage of research, but we learn from this section about several alternative models which were considered, as Hoyle's stationary universe, Linde's idea that our universe is involved in a fractal multiverse (a set of universes), and the string theory – an attempt to unify the four basic kinds of interactions as generated by oscillating strings. The last alternative is optimistic, giving a chance to avoid the loss of information (as stored in the strings) in black holes.

2.3. In the Chapter entitled 'Complex Systems and the Evolution of Life' we find illuminating remarks concerning the first integration of the idea of information, physics (thermodynamics) and biology (theory of evolution) elaborated by Boltzmann (1844-1906). It should be of special interest for philosophers as those who look for a synthetic picture of the world. Boltzmann's synthesis anticipated the very foundations of modern scientific philosophy.

Highly ordered complex systems, such as plants and animals, are most highly improbable forms in the light of the second law of thermodynamics. This law says that entropy (to be roughly equated with disorder and minimal information) grows with time, hence the intelligent life, as being an apex of order, should not have arisen in our universe. In the story in which this paradox gets explained, it is the sun which plays the role of the main hero. From it the earth receives energy to compensate the spontaneous increase of entropy in organisms. This is a crucial idea to elucidate the origins of life and its further evolution.

This idea sheds light on a universal feature of life and intelligence. There may be surprisingly many forms of life in the universe, many of them lacking any similarity to our familiar earth life, but no one can escape this dependency on the input of energy. Thus Boltzman's ideas have contributed to the philosophically fascinating issue of how energy and information are related to each other.

2.4. The chapter 'Complex Systems and the Evolution of Brain' deals with the emergence of brain and mind. Let me focus on one chosen point to recommend it for philosophical reflexion. It is found in the following passage concerning the neuronal basis generating consciousness (p. 172).

> Concerning consciousness [...], it is assumed that global activation of cell population, as exerted by the recticular formation on the cortex, would generally increase the probability of assemblies being formed. [...] The production rate of cell assemblies determines the amount, complexity, and duration of representations of sensory patterns from the outer world, for instance. *Consciousness is a self-referential state of self-reflexion. Thus, a conscious state is based on a cell assembly representing an internal state (and not only a state of the outer world.* [Italics – WMF] For example, I not only have the impression of a green tree, but I am conscious that I am looking at this tree.

Two comments will be in order, one to hint at the weakness of behaviourism, and one to exemplify the nonlinear dynamics and emergence in mental processes. Some examples in these comments are not taken from the chapter reported; however, they are meant to shed light on certain points in it that deserve special attention from a philosophical point of view.

According to behaviourists, getting rid of the concept of consciousness is necessary for any progress in psychology – for two reasons. First, the concept of consciousness is empty, it is a product of metaphysicians' phantasy. Second, even if fictional entities might sometimes be of use in science, this one proves of no use. The job of psychology consists in registering how certain stimuli are invariably associated by observable reactions. Behaviourists happen to refuse studying the nervous system, they remain satisfied with the results of their experiments corelating no more than stimuli and reactions.

This was a dogmatic attitude ignoring facts and losig chances to understand actual interactions. Among the ignored facts there is that mentioned in the quotation given above (the point italicized). Let X be an assembly of cells mapping a state of the outer world (e.g., the redness of a rose), and $M(X)$ be the state of X which consists in the mapping in question. Then there is another assembly of cells, say Y, to map the internal state $M(X)$, and this internal state of Y may be mapped by another internal state (one of a higher order, so to speak), and so on.

Some processes of development of intelligence consist in positive feedbacks between cell assemblies as those described above. When dynamic elements in a system, like those named above as X and Y, interact with other dynamic elements, they generate positive feedback loops. Then qualitatively much different and surprising new phenomena may emerge.

Analogous interactions, let me add here, occur in social processes of the evolution of intelligence. Consider, for instance, the development of logic from Socrates' teachings, twined with the progress of mathematics, up to the current theory of computability as a basis of computer science. I refer to that historical process as being well-known from numerous sources, but something like that appears in joint processes of learning logic, mathematics and computer science by an individual mind/brain.

Socrates dialogues are illuminating as a practice of reasoning combined with his ad hoc comments regarding correctness of the reasoning in question. In such comments there arises a primary logical consciousness. Suppose that Plato, owing to these beginnings of logical theory, improved such a practical art of reasoning so, that Aristotle had at his disposal a rich repertoire of forms of reasoning practised in Academia, and then he reflected upon them in a systematic way. Thus the first systematic logical theory has been born.

About the same time Euclid erected a wonderful edifice – a parennial paradigm of axiomatic mathematical practice. For centuries it provided logicians with an abundant material for logical reflexion. Owing to both streams of reflexion, that on Aristotle's theory and that on mathematical practice (e.g. a refined use of quantifiers in Calculus), mathematical logic emerged with Boole and Frege. This helped to improve mathematical practice, as methods of investigating consistency, completeness, decidability etc. Problems of decidability, in turn, gave rise to the study of computability, as initiated by Gödel and Turing, and that blossomed with the idea of computer from which a new period of civilization begins. This is, indeed, an emergence of something surprisingly new, a new civilization, from a cycle of positive feedbacks involving at the start the Greek passion for arguing and the ancient mathematical practice, both events fairly remote from the final result.

Such a historical process happens to be mapped by an individual development of an individual philosophical mind. His education should include an intense practice of reasoning which reinforces and gets reinforced by a theory of logic. It should include too a portion of mathematics to supply a material for logical reflexion, which in turn deepens the understanding of mathematics, science and cognition.

The moral to be drawn from such stories is to the effect that emergent phenomena are due to strings of positive feedbacks, and so result from a nonlinear growth. This is also a practical receipe for success in business, national economy, cultural development, etc. One has to deliberately employ such factors that they will interact in the positive feedback mode. Then emerge results which would astonish us as miracles, were we not conscious of their being natural in the nonlinear world of brain/mind and society.

2.5. From a philosophical perspective (outlined below in Section 3), the chapter 'Complex Systems and the Evolution of Computability' is central to this book. Its opening paragraph runs as follows.

> (1) *The evolution of complexity in nature and society can be understood as the evolution of computational systems.* In the beginning of modern times, Leibniz already had the idea that the hierarchy of natural systems from stones and plants up to animals and humans corresponded to natural automata with increasing degrees of complexity. (2) The present theory of computability enables us to distinguish complexity classes of problems, meaning the order of corresponding functions describing the computational time of their algorithms or computational programs. But we can also consider the size of a computer program when defining the algorithmic complexity of symbolic patterns. [Numbering and emphasis – WM.]

The chapter consists of four sections whose contents can be put in a nutshell as follows. Point (1) is discussed in the first section of the chapter, point (2) in the second section. The third section deals with information entropy in complex systems; since the approach in terms of such information dynamics does not account for human knowledge processing, it is asked whether a higher efficiency of information processing might be brought with quantum computing. The fourth (last) section, after recalling Leibniz's idea of automaton, presents John von Neumann's cellular automata (CA) framework as capable to account for chaos and randomness in complex systems.

In section 1 the first sentence (italicized in the quotation above) may figure as a motto for the whole book as it explains the key phrase *computational dynamics* in the title of the book. The phrase is to mean that the dynamics of the universe (i.e., matter, mind and mankind) possesses the essential feature of being *computational*.

What should it mean? Let me suggest that this dynamics consists in a rapid increase of computational power of the universe as a processing information device – somehow like devices referred to by the famous Moore's Law. In fact, the computational power of the universe does dramatically expand with the rise of life, then the appearance of brain, then of society, civilization, esp. information civilization, and so on. It is in order here to refer to what Barrow and Tippler endorse as *Final Anthropic Principle.*[6]

> *Intelligent information-processing must come into existence in the Universe, and, once it comes into existence, it will never die out.*

Even if one does not share the optimistic 'will never die out' (a view connected with a specific approach of the authors), the picture of such a dynamic evolution of information-processing (that is, computing) activity may us help in realizing a cosmological content of the notion 'computational dynamics'.

To contribute to the discussion of this chapter, let me sketch the question of limits of computational power. The limits of the Universal Turing Machine are well-defined by Church-Turing Thesis. Now, the crucial question is the following: are they the same for the human mind? Klaus Mainzer is cautious in proposing a definite answer, but an inclination of him may be read from his summary of the chapter in question as found in the introducing chapter. He puts a question in which it is assumed for granted that there are limitations to the analogies of computers with human mind and brain. The question based on this assumption is to the effect: do these limitations result from Gödel's and Turing's results of incompleteness and undecidability? No direct answer is found in this passage but an indirect one may be inferred from what the Author says of Stephen Wolfram results concerning CA. He refers to the result that all kinds of nonlinear dynamics can be simulated by CA.

For a final conclusion two more premisses are needed, one of them being already known; as to the second, it can be reasonably guessed to be held by the Author. The known one is to the effect that CA are equivalent to universal Turing machines (hence limited by Church-Turing Thesis), though in practice much more efficient. The premiss to be guessed would say that the nonlinear activity of the brain, that gives it so great advantages ower electronic computers, is just of the kind studied with Wolfram's research on CA. Now, since brains are CA, and those are equivalent to universal Turing machines, it would follow that brains are equivalent to universal Turing

---

[6] John D. Barrow and Frank J. Tipler, *The Anthropic Cosmological Principle*, Oxford University Press, 1996. See p. 23.

machines; and, since they are like CA, they share their enormous efficiency with those von Neumann's creatures.

This is a hypothetical line of reasoning whose conclusion I am to use to put the following question. Let us assume (as I guess to be assumed by the Author) that our brains/minds are CA: does this assumption accounts for the phenomenon of mathematical creativity? Emil Post who designed an abstract computing machine, which has proved equivalent to the universal Turing machine, firmly refused to believe that such a machine could simulate human creativity in proving theorems. Here is one of his numerous utterances in this question.[7]

> The logical process is essentially creative. This conclusion [...] makes of the mathematician much more than a kind of clever being who can do quickly what a machine would do ultimately. We see that a machine would never give a complete logic; for once the machine is made we would prove a theorem it does not prove.

In Section 3 I am to tell about some experiences which shed light on the phenomenon of creativity, as referred to by Post, and on its relation to algorithmic complexity. Since the point is crucial from the perspective here adopted, let it be the last item of this review. However, this has to be at the cost of giving up a discussion of the three ending chapters: the 6th concerning complexity in the evolution of artificial life and artificial intelligence, the 7th concerning complexity of evolutionary social and economic processes, and the 8th dealing with some philosohical issues. They offer such a reach supply of facts, ideas and problems that it will be more advisable to handle them in a special paper which I feel likely to write in a due time.

3. Insights producing algorithms to reduce complexity Are such insights themselves produced by algorithms? Exemplification through the 'Curious Inference' story

3.1. The question in the title above could be restated in the form: is the dynamics of mind in a full manner computational? The latter form refers to the *computational dynamics of mind* as called in the title of the book discussed. Should it be answered in the affirmative, then the whole dynamics of mind,

[7] Emil Post, *Absolutely Unsolvable Problems and Relatively Undecidable Propositions – An Account of an Anticipation* in: *Solvability, Provability, Definability. The Collected Works of Emil L. Post*, edited by Martin Davis, Birkhäuser, Boston (etc.) 1994. See p. 428, footnote 101.

not only when using algorithms, but also when eliciting insights, would entirely result from some algorithms – recorded, presumably, in a brain code.

Suppose, there exists a hard problem, that is such that no algorithmic (mechanical) intelligence is able to handle it, while it gets solved efficiently with an insight of human intelligence.

Before I discuss such a situation, let me suggest some handy terminological devices. Let the algorithmic intelligence be called a *robot*, and the insightful one – a *daemon* (in Unix slang, the name to mean a process runnig under its own account, as 'behind the scene').[8] In the enterprise of proving theorems, especially in mathematics, robots produce formalized proofs, i.e., such that their checking for logical correctness is carried strictly according to an algorithm. Let us call them *algorithmic proofs*, and those which require an insight let be called *intuitive proofs*. Now robots can be defined as entities whose proving ability is restricted to algorithmic proofs while daemons are capable of intuitive proving as well.

The story named in the last line of this section's title tells about experimenting with a problem which so far has proved too hard for any robot (i.e., any prover system), and easily solvable for a human daemon. The story goes back as far as 1936, when Kurt Gödel published a short comment regarding the complexity of algorithmic (formalized) proofs. Such a complexity is identified either with the number of steps (operations) in the course of proving or with the number of symbols occurring in a proof; the latter is the case here. It is crucial that the proof be formalized (within a system of formal logic), since only then no symbols are likely to be omitted (while in intuitive proofs some steps are omitted, if obvious for a supposed class of daemons). Owing to such, so to speak, typographical completeness, the number of symbols can be adopted as a measure of complexity. The paper offered by Gödel bears the title *Über die Länge von Beweisen* – On the Length of Proofs, where the length means a number of symbols, that is, the kind of complexity which in the moment we have in mind.[9]

Gödel discovered the phenomenon which nowadays is called *speedup* in the efficiency of proving; let me call it the *Gödelian speedup*. The respective Gödel's statement (English translation) runs as follows (italics – WM).

---

[8] More discussion on Unix daemon as a model of intuitive processes is offered in my paper *Rational Belief as Produced by Computational Processes* in: *Foundations of Science*, vol. 2, no. 1, 1997, Kluwer Academic Publishers.

[9] Kurt Gödel, *Über die Länge der Beweisen* in: *Ergebnisse eines mathematischen Kolloquiums* Heft 7, Franz Deuticke, Leipzig und Wien 1936. From among three forms of this term, 'speed up', 'speed-up' and 'speedup', I choose the last which is reported in Webster (though in a different meaning, occurring in economics, see www.merriam-webster.com/dictionary/), and also in some texts concerning data processing.

> Thus, passing to the logic of the next higher order has the effect, not only of making provable certain propositions that were not provable before, but also of *making it possible to shorten, by an extraordinary amount, infinitely many of the proofs already available.*

The first part of this statement says what is involved in Gödel's groundbreaking paper of 1931, while the second (italicized) tells something new, which is of great practical import for computer science. Let us dwell a while on the method of reducing complexity as suggested by that statement. Its significance lies in the fact that it opens the following questions.

- A. How great is that extraordinary amount by which a proof gets shortened, i.e., its complexity gets reduced, owing to the passing to the next higher order? How much is such a reduction significant practically?
- B. Is this method (of complexity reducing) (B1) obtainable in the same degree for human provers and mechanical provers (the latter dealing solely with algorithmic proofs)? If not the same, then (B2) how big may there be such a difference of degree, and (B3) how should it be accounted for?
- C. Provided its availability for mechanical provers, how great would be the difference when comparing complexities of the shortest proofs of the same theorem, one produced by a human prover (daemon) and one by a mechanical prover (robot)?

3.2. Question A has been answered after a half century hiatus by George Boolos in his seminal paper of 1987, attractively titled *A Curious Inference* (from now on cited as BP – for Boolos' Proof), which offered an enlightening case study. I can feel free from reporting wider on BP, since its contents is summed up in a preceding, easily available, volume of this journal's electronic version.[10] Let me just mention what counts most: that Boolos' proof of an arithmetical theorem (concerning a property of Ackermann function) when performed in the second-order logic takes one page alone. On the other hand, the number of symbols occurring in the first-order derivation is represented by the exponential stack of as many 2's as 64536, that is, larger than any integer that might appear in science. This is a dramatic result, indeed, which nicely exemplifies the Gödelian speedup.

[10] See logika.uwb.edu.pl/studies/vol9.htm. Roman Murawski, *The Present State of Mechanized Deduction, and the Present Knowledge of its Limitations* in: Witold Marciszewski (Ed.), *Issues of Decidability and Tractability*, vol. 9 (22), 2006 of the journal *Studies in Logic, Grammar and Rhetoric.* A sketchy discussion on this subject is also found in the same volume, in the paper by Witold Marciszewski: *The Gödelian Speed-up and Other Strategies to Address Decidability and Tractability.*

The same result answers the second part of Question A. Since a proof in the language of first-order logic is so much intractable, the adopted method of reducing the proof to such a short, so easily tractable, version is of enormous practical import. At the same time, Boolos to some extent paved the way to answering question B2 as discussed below.

The next two decades after BP have brought forth some answers to the remaining questions, due to an intense research in automated theorem-proving. As for B1, the answer is decidedly in the negative. In point B2 it is continued to the effect that differences of degree are colossal, and the source of them (asked in B3) lies in the fact that a great deal of creative intuition is necessary, which so far is a priviledge of humans, unattainable to robots. Both Boolos and automated reasoning researchers refer to the set-theoretical axiom of comprehension, essential for the proof under study, as one whose applications much require creative insights. This axiom (equivalent to a formula in the second-order language) makes it possible to introduce new concepts needed for the proof in question.

To win more understanding, let me give a voice to experts in automated reasoning research. The most illuminating guide to our problem I could find so far is the survey by Benzmüller and Kerber [2001] entitled *A Challenge for Mechanized Deduction.*[11] In the report on their research they conclude as follows.

> Boolos' example perspicuously demonstrates the limitations of current first-order and higher-order theorem proving technology. With current technology it is not possible to find his proof automatically, even worse, automation seems very far out of reach. Let's first give a high-level description why this is so. Firstly, Boolos' proofs need comprehension principles to be available and it employs different complex instances of them. [...] Secondly, the particular instances of the comprehension axioms cannot be determined by higher-order unification but are so-called Heurika-steps which have to be guessed. However, the required instantiations here are so complex that it is unrealistic to assume that they can be guessed. [...] Here it is where human intuition and creativity comes into play, and *the question arises how this kind of creativity can be realised and mirrored in a theorem prover.* [Emphasis – WM]

[11] See Christoph Benzmüller and Manfred Kerber, *A Challenge for Mechanized Deduction*, 2001 (www.cs.bham.ac.uk/ mmk/papers/01-IJCAR.html). A considerable number of other studies on this subject can be found with the search: citeseer.ist.psu.edu/. There may be of special interest the paper by Natarajan Shanker *Using Decision Procedures with a Higher-Order Logic* which refers to excellent surveys of higher-order logics as offered by S. Feferman, J. van Benthem, etc.

3.3. As for question C (regarding complexity differences between algorithmic and intuitive proofs), its relevance can be noticed with the following consideration. What may be a tractable size of a mathematical proof carried by a human mathematician? A highly interesting evidence is due to the fact that the famous proof of Fermat's great theorem by Andrew Wiles (published in *Annals of Mathematics*, May 1995) required 200 manuscript pages, that is, approximately, 400000 bits (symbols), and must have been divided into six parts to become readable for each of the six reviewers asked by the editor to critically read and comment on the proof. It is no formalized proof, hence it must have gaps to insightfuly be filled up by a competent reader.

Certainly, after filling up such gaps according to the rigours of algorithmic methods, the proof has to get longer. How much longer? Would have such a difference a practical relevance concerning time and memory size for data processing? Unfortunately, we have no answer in this individual instance, but we can consider a case having already been subject to a study, and so obtain instructive analogies. Such is the research on automated proof checking reported by Christoph E. Benzmüller (the University of Cambridge and Universität des Saarlandes) collaborating with Chad E. Brown (Universität des Saarlandes).[12] The authors possess a unique expertise in comparing *automatic proving* (which they call full automation) with *automatic proof-checking* as performed by systems like Mizar and OMEGA. As for automatic proving, they conclude, thus confirming the conclusion of the previously quoted paper (by Benzmüller and Kerber, 2001), with the following statement.

> The full automation of Boolos curious inference seems not to be in reach and it will be a challenge problem to automated theorem proving for a long time to come.

However, owing to the procedure of the automated checking (for logical correctness) of Boolos' inference, we can fancy a minimum size (hence a minimum complexity) of the same proof if produced by a robot (here, a system being an automatic prover). Let us suppose that a proof written by a human in a language specially devised for proof-checking (as Mizar, OMEGA, etc.)

[12] Christoph E. Benzmüller and Chad E. Brown, *The Curious Inference of Boolos in Mizar and OMEGA* in: Roman Matuszewski and Anna Zalewska (Eds.), *From Insight to Proof. Festschrift in Honour of Andrzej Trybulec*, vol. 10 (23) of the journal *Studies in Logic, Grammar and Rhetoric*, University of Białystok, Białystok (Poland) 2007. See also: logika.uwb.edu.pl/studies/. A. Trybulec, let me add, is the author of the system Mizar referred to in the mentioned title.

will be roughly of like size as the proof of the same theorem produced by an automatic prover. Thus we obtain (according to the paper by Benzmüller and Brown) the following sizes of compressed (using gzip) files.
— Boolos' proof sketch: 637 bytes;
— Mizar article: 2310 bytes;
— OMEGA article (in one of the versions produced): 2602 bytes.
Obviously, an article being a proof (of the same theorem) which would be produced by an automatic prover might considerably differ in size from those mentioned above. However, we can reasonably estimate that the order of quantity would be preserved.

Thus we can realize that the carrying of a proof in the second-order language dramatically lowers its complexity (as compared with first-order procedures) in any case; that is, independently of technology adopted, be it the old-fashioned pencil-and-paper technology, be it the modern computer technology.

What makes the problem hard for computer is that its solution requires a creative insight which so far is not obtainable for robots. This brings the problem of choosing between two philosophical options which are as follows. (α) Is there so that insights belong to a category of cognitive acts entirely disjoint with the class of algorithms? Or, rather, (β) any insight is due to a hidden algorithm? (Hidden in the sense that we cannot recognize this fact for limitations of our current knowledge.)

If the latter is the case, then a brain as that of Boolos should contain a second-order algorithm somehow recorded in a brain code (being like machine code in a computer). As being algorithmic, it does not exceed capabilities of a robot, to wit a universal Turing machine. Then no daemon is needed, to do the job of, for instance, inventive using the axiom of comprehension.

This statement of options leads to a final comment on the book under discussion. It takes the form of the following question. The title of the book contains the term *the Computational Dynamics of Mind* ('computational' amounts to 'algorithmic'). Should it mean that the whole dynamics of mind is computational? Or, that there is a kind of mental dynamics which is computational, while another one is not computational, and it is the former which belongs to the subject matter of the book (the latter being outside the scope of the Author's interests)?

The remaining items, Matter and Society (Mankind) might be addressed with an analogous disjunction. However, this would require a special discussion, at least as detailed as that devoted here to the computability of mental dynamics. Let it be left to another opportunity.