

ML STUDIES IN LOGIC,
GRAMMAR AND RHETORIC

UNDER THE AUSPICES OF THE POLISH ASSOCIATION FOR LOGIC AND PHILOSOPHY OF SCIENCE

**Issues
of Decidability
and
Tractability**

Edited by WITOLD MARCISZEWSKI

**9 (22)
2006**

U N I V E R S I T Y O F B I A Ł Y S T O K

**ISSUES OF DECIDABILITY
AND TRACTABILITY**

Series: STUDIES IN LOGIC, GRAMMAR AND RHETORIC 9(22)

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

ISSUES OF DECIDABILITY AND TRACTABILITY

Guest Editor: Witold Marciszewski

UNIVERSITY OF BIAŁYSTOK
Białystok 2006

Series: STUDIES IN LOGIC, GRAMMAR AND RHETORIC 9(22)
http://logika.uwb.edu.pl/studies/

Edited by Halina Święczkowska
University of Białystok, Faculty of Law, Section of Semiotics

in collaboration with Kazimierz Trzęsicki
University of Białystok, Faculty of Mathematics and Physics
Chair of Logic, Informatics and Philosophy of Science – logika@uwb.edu.pl

Editorial Assistants:

Izabela Bondecka-Krzykowska
Adam Mickiewicz University, Poznań
Polish Association for Logic and Philosophy of Science

Katarzyna Doliwa
University of Białystok

Dariusz Surowik
University of Białystok

Editorial Advisory Board:

Jerzy Kopania, University of Białystok
Grzegorz Malinowski, University of Łódź
Witold Marciszewski (Chairman), University of Białystok
Roman Murawski, Adam Mickiewicz University, Poznań
Mieczysław Omyła, Warsaw University
Katarzyna Paprzycka, Warsaw School of Social Psychology
Jerzy Pogonowski, Adam Mickiewicz University, Poznań
Jan Woleński, Jagiellonian University, Cracow
Ryszard Wójcicki, Polish Academy of Sciences

Refereed by Jerzy Kopania
Cover design: Krzysztof Tur
Type-setting: Stanisław Żukowski

Publication supported by Polish Ministry of Science
Grant No. 2H01A03025
and by the University of Białystok, Faculty of Law

ISBN 978-83-7431-108-3
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

3188107p 296800

0229816/22

BIBLIOTEKA UNIwersYTECKA
im. Jerzego Giedroycia w Białymstoku



FUW0218806

CONTENTS

| | |
|---|-----|
| <i>Preface to this Volume as appearing in a new setting</i> | 7 |
| Witold Marciszewski <i>The Gödelian Speed-up and Other Strategies to Address Decidability and Tractability. Editorial Comment on this Volume</i> | 9 |
| Roman Murawski <i>The Present State of Mechanized Deduction, and the Present Knowledge of its Limitations</i> | 31 |
| Roman Matuszewski <i>On Computer-assisted Approach to Formalized Reasoning</i> | 61 |
| Kazimierz Trzęsicki <i>From the Idea of Decidability to the Number Ω</i> | 73 |
| Witold Marciszewski <i>Undecidability and Intractability in Social Sciences</i> | 143 |
| Anna Gomolińska <i>Rough Information Granules in Social Agent System Modelling</i> ... | 175 |
| <i>Appendix</i> | 207 |

**PREFACE TO THIS VOLUME
AS APPEARING IN A NEW SETTING**

The first issue of “Studies” appeared in 1980. This was done without a permission of the Ministry of Science, what under the communist regime in Poland of those times was a serious offence of law. However, due a small legal trick, that risky deed had passed unnoticed by authorities. In Editorial Introduction the bonds among the medieval Logic, Grammar and Rhetoric, as well as among their modern counterparts, were revealed – with a touch of enthusiasm about chances of reviving the venerable tradition. For instance, the opening essay by Kuno Lorenz (one who created dialogic logic together with Paul Lorenzen) as being entitled “Main ideas of dialogic logic” was to represent a modern sophisticated form of rhetoric, while N. G. de Bruin’s (Eindhoven) contribution “The AUTOMATH mathematics checking project” perfectly exemplified a modern synthesis of logic and grammar (syntax) in devising proof-checkers.

The opening of a new phase dates at 1997 year when “Studies” obtains features of both a journal, appearing annually, and a book series, each volume being a collection concerning a common subject announced by the volume title. Hence the double numbering, the first number signifying the position in the so established series. The first publication in the new shape was entitled “On Leibniz’s Philosophical Legacy in the 350th Anniversary of His Death”. The second one, 1998, was also an anniversarial publication to celebrate Emil Post’s Birth Centenary under the title “Emil Post and the Problem of Mechanical Provability”. It includes papers delivered at the conference held in 1997 in Białystok – the capital of the region in which Post was born in 1887; from this volume onward the Series is fully and freely available on-line (logika.uwb.edu.pl/studies/).

After nearly ten years we again enter a new period. Its visible sign is the enhanced cover design, while an essential novelty consists in starting a systematic collaboration with the *Polish Association for Logic and Philo-*

sophy of Science, what is signified with formula *under the auspices*. This is to mean that Association's representatives join the Editorial Team and the Advisory Board, the latter including now the Association's President Roman Murawski. This also means Association's help in gaining wider contacts for international collaboration. On the other hand, the journal provides Association with a forum for reporting its activities and for other publications.

The present volume fulfills the additional task of collecting contributions resulting from the research supported by the Polish Ministry of Science, concerning decidability and tractability (or their lack) in social sciences (Grant No. 2H01A03025). The unit responsible for the project is the University of Białystok acting through its Logic Section, hence the research participants, and thereby contributors to this volume, are mainly persons affiliated to this university. However, this is no deliberate policy. Hopefully, next volumes, like the previous ones, will host authors from various academic circles.

The moment like this provides an opportunity to mention those who in the past years, since the birth of the project, contributed to its development. Once upon a time, when I conceived the idea of a foreign language journal as a window to scholarly circles abroad, this was meant as a chance for a young academic institution in the North-Eastern corner of Poland. Obviously, this could not succeed without able efforts of my colleagues. Among them there is Jerzy Kopania as the first editor, Kazimierz Trzęsicki as the Head (since 2004) of Logic Section coresponsible for the journal, Roman Matuszewski who mastered the art of winning Ministry grants for editing purposes, Mariusz Giero as the webmaster of on-line edition, Stanisław Żukowski as an expert type-setter, the Law Faculty Authorities who supported the journal when it did not succeed in gaining support from other sources. Special appreciation is due to Halina Świączkowska as the Editor-in-Chief who runs the journal in a competent and diligent way since late 80's up to now.

In this new stage and new setting, the journal is to be addressed to a much wider circle of foreign recipients that it used to be so far. We hope to obtain sympathetic responses which should make this journal a significant forum of scholarly collaboration in modern counterparts of Logic, Grammar and Rhetoric.

Witold Marciszewski

Witold Marciszewski

University of Białystok

THE GÖDELIAN SPEED-UP AND OTHER STRATEGIES TO ADDRESS DECIDABILITY AND TRACTABILITY

Editorial Comment on this Volume

Gödelian speed-up is a handy concept when dealing with decidability and tractability, even if it did not become common in logical literature yet.¹ Since the phrase “decidability and tractability”, when uttered in one breath (as we shall often do), is a bit clumsy, let it be shortened for “D&T”. In this pair, the theoretical (or, in principle) possibility of solving a problem is what one calls *decidability*, while the practical one is called *tractability* (more on this distinction – in item 1.3 below).

No wonder that it is Gödel whom we shall follow as the great master of D&T issues. To him we owe deepest insights into the feedback between human minds and computers in their common enterprise of problem solving.² His idea of speed-up (though not the word itself which appeared later) is to the effect that one can obtain ever shorter proofs, hence ever shorter times of problem solving, with engaging ever more abstract languages. Abstraction is to mean in such a sense in which talking about sets is more abstract than talking about individuals, while a talk about sets of sets is still more

¹ This text, when introducing to this volume, at the same time sums up the Research Project “Nierozstrzygalność i algorytmiczna niedostępność w naukach społecznych” [Undecidability and Intractability in Social Sciences] Grant No. 2 H01A 030 25, supported by the Polish Committee for R&D, next by Ministry of Science, in years 2003-2006. When mentioning the Polish title of the Project, a terminological note concerning Polish terminology will be in order here. The phrase “niedostępność algorytmiczna” (that might be rendered as “algorithmic inaccessibility”), as used in the statement of the Project, is not the only one to be recommended as a counterpart of the English “intractability”. There is also in the Polish dialect of computer scientists the term “nieobliczalność praktyczna” (“practical uncomputability”) to express the same concept. May be, it is the latter which has more chances to become commonly accepted among Polish-speaking authors.

² There were no physical computers in the time of Gödel's revolutionary discoveries. However, the idea of mechanical problem-solving belongs to the very core of mathematical logic, especially in the inquiries of Hilbert, Turing and Gödel.

abstract, and so on (a more systematic explanation is to be given below in item 1.3). It is that feedback which brings about the speed-up in problem solving, contributing to a rapid growth of computational powers, by which the range of what can be treated in an algorithmic way gets ever greater.

It is worth while to mention social implications of the increase of computational power. That constant and rapid increase is what constitutes the essence of modern information society. A popular definition (in Wikipedia) says that an information society is a society in which the creation, distribution and manipulation of information is a significant economic and cultural activity. The vagueness of the adjective ‘significant’ makes the whole definition too vague. This can be remedied if significant activity is to mean ever more massive information-processing due to the dramatic (in some segments even exponential) increase of computational power which, in turn, has the computational speed-up among its main sources. This is why a thorough inquiry into the nature of that speed-up is needed, to find an efficient strategy of further social development. Related claims appear in EU documents postulating a dramatic growth of information society in Europe. Some documents of the Lisbon Strategy emphatically call for improvement of the knowledge management procedures. Such a progress should heavily draw on the feedback between minds and computers, and this again would lead to the D&T issues.

In a previous stage of research on D&T these properties had been studied with regard to theories of deductive sciences. It was a novelty when Stephen Wolfram [1985] raised the D&T question in physics. There were authors who followed that train of thought with regard to economy, e.g. Latsch [2003], and other social sciences.

A relevant quotation from Latsch’s [2003] abstract runs as follows. “A highly complex computational economy can evolve and self-organize but it also displays computational universality that means that many problems are not decidable. The inherent limits of computability become evident. This paper proposes incorporating a particular (constructive) non-computability into our view of economic behavior and processes. The paper defines constructively non-computational behavior, discusses its origins in Roger Penrose’s writings, and provides an application of this concept to the question of realistic counterfactuals in economic models.”

The title and content of this volume – *Issues of Decidability and Tractability* – is to serve as a productive challenge to researchers ready to engage themselves in such a study with respect to social sciences. No new results on D&T are presented here but some material is provided as a background to put a new question: *what about D&T in social sciences?* An incentive to such

a study is what is intended with this volume; it should offer a thought-provoking survey of some issues, statements and ideas being relevant to such a study.

1. D&T story in three chapters: Gödel, Boolos, a current research

1.1. The D&T story is foreworded with what historians call Hilbert Programme. It was Hilbert’s famous project that involved the task of solving the problem of decidability (*Entscheidungsproblem*) of first-order predicate logic (called by him *der engere Funktionenkalkül*). In Hilbert and Ackermann [1928: 73] we read what follows (ad hoc translation by WM).

“The decision problem gets solved if one knows a procedure which for a given logical expression allows to decide, with finitely many steps, about its validity or its satisfiability. The solution of the decision problem has a fundamental impact for all those theories whose statements are at all capable of being logically derived from finitely many axioms.”

The negative solution of this problem came soon with the famous results by Turing [1936/37], Church [1936] and Post [1936] regarding undecidability of first-order logic (FOL, for short).

Undecidability of FOL is also supported by the fact of incompleteness of arithmetic of natural numbers (ANN, for short), the property demonstrated by Gödel [1931]. To see this, we need a method of expressing any ANN formula in the language of FOL; such a method has been elaborated by Hilbert and Ackermann [1928]. Then the problem whether a formula can be proved within ANN reduces to the question whether it logically follows from ANN axioms. Let the conjunction of ANN axioms, written in FOL notation, be denoted as κ . We ask whether a formula ϕ is provable from κ . Were FOL decidable, that is, were there a universal and mechanical decision procedure to recognize whether $\kappa \Rightarrow \phi$ is valid or not, then the recognition of its validity would make ϕ provable, while the recognition of non-validity would evidence non-provability, and so ANN would turn out to be complete.³

³ Cp. Kneale and Kneale [1962: 736]. The transition from a valid formula of the form $\kappa \Rightarrow \phi$ to the inference from κ to ϕ is due the *converse deduction theorem* that holds for FOL. Cp. Surma [1981] and Marciszewski [1981: “Predicate Logic”, 6.1].

1.2. Let us consider the message about FOL limitations vis-à-vis the main problem of information society, to wit the question of how to increase its computational power. This should be viewed in a larger perspective of the destiny of our civilization, and this, in turn, in a still larger cosmological perspective.

Our ancestors, looking for fundamental elements of the universe, found them naively in earth, fire, air and water. Nowadays, in the same role of the basic constituents of the universe, we entertain the triad of Matter, Energy and Information, each being capable of highly productive transformations.

Such a cosmological insight yields us the broadest frame for understanding the development of civilization. It should be seen as succession of three eras, each of them being defined with respect to the skill of transforming an element of that triad. First, people acquired the skill of transforming matter through agriculture and use of various tools and machines, as turning cereal grains into wheat, and that into bread, or a piece of clay into a jar, etc. The next era started with the invention of engines to produce energy through its transformations, say, mechanical energy from its other forms, as chemical or electric. And the third era came with inventing machines to process information. The third coincides with the beginning of the space age which is due to the united forces of all the three technologies: those of transforming matter, energy and information.

The first space flights, as those performed in the 20. century, should be seen as small preliminary steps towards the titanic work of colonizing the universe and of cosmic engineering. We, as the human race, are bound to be farsighted enough to engage ourselves into such enterprises, taking into account that warning by Stephen Hawking: “I don’t think the human race will survive the next thousand years unless we spread into space.” The accomplishment of such projects will require unimaginable energies and not less computational powers. The latter would depend on the dramatic development of both software and hardware.

When considering perspectives of developing software, we at last come to the essential role of FOL (First-Order Logic). This is so because each computer program is based on an algorithm, and each algorithm presupposes a mathematical proof. This relationship is obvious regarding programs of computing; say, a program to compute $y = \sqrt{x}$ owes its efficiency to the fact that the underlying formula is provable from the axioms of arithmetic. As to any other program, for example, one translating texts from English into Chinese, its functioning is due to the encoding of instructions into strings of ones and zeros, hence some arithmetic expressions. Such expressions are again formulae which are provable in arithmetic, while the range of what

can be proved depends on the strength of logical rules of proof. The strength of FOL rules (let us recall) is limited – in that sense that in some cases they do not decide whether a formula does follow from relevant axioms.

Thus we come to realize the important fact: that reinforcing inference rules, be it possible, would extend the scope of problems being solvable through decision procedures, and thereby accordingly extend the scope of computing with the aid of software.

Is it in fact possible? It is Gödel to whom we are indebted an illuminating hint. Let us focus on his contribution and how it is continued nowadays, up to a recent mechanized deduction research.

1.3. On June 19, 1934, at the seminar run by Karl Menger in Wien, Kurt Gödel presented the communiqué entitled “Über die Länge von Beweisen” (on the length of proofs). The text appeared in 1936 in reports on the seminar (referred to as Gödel [1936: 23-24]). Its main point runs as follows (the numbering of items by WM).⁴

Thus, passing to the logic of the next higher order has the effect, not only of (1) making provable certain propositions that were not provable before, but also of (2) making it possible to shorten, by an extraordinary amount, infinitely many of the proofs already available.

Thus we attain at what one may call *the Gödelian speed-up in problem-solving* (as in the title of this essay). The term ‘speed-up’ renders the acceleration of the processes of proving due to reducing the number of steps.

This assertion has not been either demonstrated or exemplified by Gödel himself (examples to test the assertion were to come later, produced by other authors – see Section 1.4 below). This may be a reason why this theorem had not been much referred to in the first years after its publishing; not as much as the theorems on the completeness of first-order logic, the incompleteness of arithmetic, etc., although its import is comparable with that enjoyed by those famous results. However, one can find illuminating comments on its content, for instance in Kneale and Kneale [1962: 722], noticing the novelty of the approach suggested by Gödel in 1936. After discussing

⁴ Translation in Gödel [1986: 397] under the title *On the length of proofs*. It is in order here to quote the original statement for its historical significance. *Der Übergang zur Logik der nächst höheren Stufe bewirkt also nicht bloß, daß gewisse früher unbeweisbare Sätze beweisbar zu werden, sondern auch daß unendlich viele der schon vorhandenen Beweise außerordentlich stark abgekürzt werden können.*

the unprovability of Gödel's formula, referred to as G , and the possibility of strengthening ANN (Arithmetic of Natural Numbers) by simply adding G to ANN axioms, the authors appreciate the new approach in the following passage.

If, instead of merely adding G as a new axiom, we enlarge our formalism by adding some new general apparatus of proof which enables us to obtain [...] G as a theorem, we have an instrument which is more powerful for the ordinary purposes of arithmetic. The novelty here is the use of axioms containing variables of higher orders than those occurring in the number theory; and Gödel has shown that it not only enables us to prove formulae that were hitherto unprovable, but allows very much shorter proofs for many of the previously obtainable formulae. Within this new system it is possible, however, to construct a new undecidable formula, and so the whole process can be repeated *ad infinitum*.

The time in which this comment was made (1962) was not ripe enough to fully appreciate the import of the second part of Gödel's assertion, that concerning the shortening of proofs already available. Only just with the rise of the theory of computational complexity of algorithms (cp. Hartmanis and Stearns [1965]) people started to realize the significance of this part of assertion. The greater was becoming the computational power of hardware (increasing exponentially according to Moore's law), the more was growing the awareness of the import of the computational power of software. When we now are able to estimate the speed of processors available in the coming time, we can also estimate how fast algorithms we need in order to solve (with the united forces of hardware and software) the problem having a definite complexity. And so we learn messages like that: there are problems so hard that even most speedy computers, programmed with the fastest available algorithms, do not give us a hope of solving them even in some millions or more years.

What should reasonable people do in such a drawback? They should look for methods of *making it possible to shorten, by an extraordinary amount, infinitely many of the proofs already available* – as reads item 2 of Gödel's assertion (instead of 'proofs', now read 'algorithms'). When enjoying such realization due to Gödel, what we need bad is a paradigm of attaining such methods.

1.4. Before I report on such a serious argument to show the enormous difference between the efficiency of first-order and second-order logic, let me give just a rough idea of that difference. Consider the following reasoning

expressed in that part of ordinary English which corresponds to some part of second-order logic.

In a village there are three parental couples, each having two little children, and there are no more parental couples in that village. Hence there are exactly six little children in that village.

Even in such a childish problem, one resorts to the second-order ideas because of talking of sets (couples) as existing. If a radical nominalist like, say, Tadeusz Kotarbiński regards second-order logic as an unscientific metaphysics, he should express such a reasoning in the first-order language. Then in its conclusion he has to use six individual variables (say, x_1, \dots, x_6) in order to state about each individual the fact of being a little child, and then to say about any other individual denoted by a variable, say y , that if y were a little child, then it would be identical either with x_1 or [...] with x_6 . This would require using more than hundred symbols. (And what if one is to speak about millions of individuals?). The formalism needed to express the premiss would be even more cumbersome, involving binary relation of being, so to speak, parentally coupled, and ternary relation of being a child of each member of the couple in question.

The comparison of the above reasoning in the first-order logic and that in the second-order logic suggests that some mental mechanisms of reasoning in humans happen to function according to second-order pattern. Is it possible for computers to simulate and match humans also in this respect? This is the question. The one we shall address with tracing the way from Gödel's [1936] idea to recent research in mechanized deduction.

A milestone in this way is found in George Boolos' [1987] seminal paper "A Curious Inference".⁵

Let the inference in question be called BP (for Boolos' Proof). I postpone presenting the content of BP (to be roughly presented below, item 1.5), and focus on hinting at the scale of difference when one compares the lengths of first-order and second-order proofs. The latter, as performed by Boolos, is a short derivation taking no more than one page, while the number of symbols used in the FOL derivation is represented by the exponential stack of as many 2's as 64536. It is larger than any integer that might appear in science.

An important source of this difference is found in the fact that BP reduces to so short derivation owing to the use of comprehension axioms

⁵ There were earlier studies which resorted to Gödel's idea of shortening proofs, e.g. Parikh [1973], but none of them has made a comparable impact on clarifying the issue.

(typical second-order means). These are propositions being subsumed under the following schema:⁶

$$\exists x \forall y (y \in X \Leftrightarrow \phi(y)).$$

1.5. This fact is crucial for the next step of our inquiry into the feedback of minds and computers. This is the step of entering the realm of mechanized deduction. The most expert guide to that realm I could find so far is the survey by Christoph Benzmüller and Manfred Kerber [2001] entitled “A Challenge for Mechanized Deduction”.⁷ The authors start from the message that the art of general purpose automated theorem proving has been best developed for first-order logic. This was convincingly highlighted by the success of first-order theorem provers in producing the machine-generated proof of the Robbins problem.⁸ Their argument nicely fits into the idea I express with the phrase *the feedback between minds and computers*. They mean something like that when considering *facilities to combine interaction with automation*. Let the following quotation render such a course of thought.

Automating proof search in higher-order logic is a very challenging enterprise, such that the above systems all provide facilities to combine interaction with automation. The idea is that the interactive human provides the crucial proof steps while simple subgoals are handled automatically by the prover. Of course, many non-trivial proofs can be already automated in higher-order logic. [...] A well known example illustrating the expressiveness and elegance of automated higher order theorem proving is Cantor’s theorem, where the diagonalisation argument, in form of a lambda-term, is synthesised by higher-order unification.

However, 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*. [From Section 1; italics – WM.]

The question raised in the last sentence is fundamental. To give it a case study exemplification, the authors analyze difficulties involved in Boolos’ ‘curious inference’. It contains the following premises (1-5) and concludes with line 6. This conclusion may be reached in a short (one-page size) derivation in the second-order logic, being practically unattainable in the first-order logic (the notation as below is adjusted to a programming idiom).

1. FORALL n. f(n,1)=s(1)
 2. FORALL x. f(1,s(x))=s(f(1,x))
 3. FORALL n. FORALL x. f(s(n),s(x))=f(n,f(s(n),x))
 4. D(1)
 5. FORALL x. (D(x) -> D(s(x)))
- hence
6. D(f(s(s(s(1))),s(s(s(1))))))

This case yields opportunity to observe the importance of critical reflexion on the strategy in planning a proof. As the authors assert (in Section 5.1), it depends on the reasoner’s experience. The case in question requires enormous expertise, hence the following challenge for mechanized deduction: how to impart such a superior skill to a mechanical prover. This would mean constructing provers which together with planning proofs at object level would in parallel critically reflect on that planning, thus having a kind of meta-level understanding of their object-level activity. This challenge might be expressed using Emil Post’s maxim that “Symbolic Logic may be said Mathematics self-conscious”.⁹ The endeavour to make a mechanical prover critical about its own doings is something like to simulate a logical self-consciousness. This seems to be not only technical but also philosophical challenge.

An equally great challenge consists in simulating the process of concept-forming. It is specially perspicuous in Boolos’ inference in which so great a role is played by searching for comprehension axioms. Selecting

⁶ In the mentioned ‘naive’ form, this schema is exposed to danger of antinomies, but its sole purpose in this context is to suggest, in a possibly simplest form (without refinements), the very idea of comprehension through hinting at its second-order character.

⁷ 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.

⁸ A report on that event, compiled by the present author, can be read in *Mathesis Universalis*, No. 4, Autumn 1997. See www.calculemus.org/MathUniversalis/4/6com-rob.html.

⁹ This saying is taken from Post’s Diary entitled “Time Accounts”, being the Appendix to Post’s paper “Absolutely Unsolvable Problems and Relatively Undecidable Propositions – Account of an Anticipation” published by Davis [ed. 1994].

useful comprehension axioms is the same question (the authors notice) as to form interesting concepts. Another task consists in teaching a prover to resort to a model-based reasoning (this would require, if I properly guess, equipping the system with a large data-base concerning the domain which a given inference is concerned with).¹⁰

1.6. All that is highly interesting and amenable for the discourse on the development of information society. Mechanized deduction is essentially involved in knowledge management and expert systems as well as artificial society projects, the latter using the computer simulation of social phenomena cellular automata, agent programs, etc. All the listed enterprises, when undertaken at a large scale, are characteristic of information society. The input of mechanized deduction is in them obvious, as each of them includes reasonings.

Do researchers in the listed areas happen to recognize some encountered problems as likely to be either undecidable or intractable? Certainly, such awareness did appear in economics in the famous Hayek-Lange debate on socialist calculation (cp. Marciszewski [2002]). It was Oskar Lange who claimed all economic problems be tractable in the system of socialist central planning, thus endorsing the strong AI with regard to economic issues. It was Friedrich Hayek who claimed the intractability of central planning problems and advised therefore to resort to free market mechanisms as using better methods of data-processing, to wit taking advantage of parallel and interactive processing.

However, what would be the most interesting in that debate, has not been explicitly identified. It is the problem of the kind of logic as used by market agents. Let me direct our attention to a deep observation in the paper by Boolos [1987: 380]. It runs as follows.

The fact that we so readily recognize the validity of I [so is referred to by Boolos his famous inference] would seem to provide as strong a proof as could be asked for that no standard first-order logical system can be taken to be a satisfactory idealization of the psychological mechanisms or processes, whatever they might be, whereby we recognize logical consequences.

Thus it does not seem unlikely that free market agents in their reasonings use higher-order logics without any computer assistance, so taking advan-

tage of the Gödelian speed-up in their intuitive reasonings (like Boolos did in his short inference conducted in the second-order logic). On the other hand, a socialist planner (unlike a free market agent dealing with a highly restricted set of data) has to deal with so astronomically huge data amount that he is helpless without computer's assistance. However, that assistant is often doomed to fail in solving some problems at the second-order level – for the reasons discussed by Benz Müller and Kerber in their survey of various difficulties.

Let us look at the computational speed-up and tractability in the broad perspective of civilizational progress. The stage of information society we enter nowadays is but a preparatory step towards the age of biological engineering and cosmic engineering. Obviously, there will be enormous risks in such a course of events. However, let us assume that the human race will succeed (may be, in a distant future) in gaining the necessary moral potential to face the challenges. Then the rest would depend on a sufficiently large cognitive potential. Its core lies in mathematics as an indispensable tool of science and technology.

Now, let us look at the growing interaction, in the form of positive feedback, between the development of mathematics and the increase of computational power. That increase, apart from being due to ever greater hardware perfection (as claimed in Moore's law), heavily draws on software perfecting. The kind of software most in this case relevant consists of programs for mechanized deduction. It is in order here to recall the lesson due to Gödel, Boolos and, lastly, such researchers as Benz Müller and Kerber. To wit, if we like getting a necessary speed-up in mechanized deduction, we should resort to higher-order logics. Then some important problems, which were not tractable so far, become solvable in real practice.

A bit of reflection is due to European R&D policies regarding the challenges of information society, as considered specially in Lisbon Strategy. Among documents concerning that strategy there is one that reads as follows.

“The Union has today set itself a new strategic goal for the next decade: to become the most competitive and dynamic knowledge-based economy in the world. [...] Achieving this goal requires an overall strategy aimed at preparing the transition to a knowledge-base economy and society by better policies for the information society and R&D.”

So reads a passage in the document, dated 2004.01.05, *Lisbon Strategy – the European Agenda for competitiveness, employment and social cohesion* by Professor Mario João Rodrigues being in charge of the preparation of the Lisbon Summit as special advisor to the Prime Minister. (Cp. www.bsssc.com/section.asp?id=217&pid=79.)

¹⁰ These are just some examples from among a larger set of problems which the authors consider under the heading “How to find a solution?”. I discuss this particular report in such details, paying so much attention to it (as compared with other publications I mention here), for I see in it an illuminating example of how much advantage may take social researchers and philosophers of mind from expert reports on mechanized deduction.

There is no need to exaggerate the role of logic and tractability issues in such enterprises. Nevertheless, there is a place for them in that R&D segment in which knowledge management, expert systems, and the like, require mechanized deduction. And this in turn – to make problems tractable – needs enhancing with more sophisticated logical tools like those from the height of higher-order logics.

2. On how does this volume deal with D&T issues

2.1. There are two contributions which offer a general historical account in the role of introduction: one by Roman Murawski and one by Kazimierz Trzęsicki. It is the former which opens this volume as the Editor decided to emphasize what is expressed by its title: *The present state of mechanized deduction, and the present knowledge of its limitations*. It is a revised and updated version of the closing chapter in the book by Witold Marciszewski and Roman Murawski *Mechanization of Reasoning in a Historical Perspective* [1995]. This is in accordance with the narrative of the previous Section where mechanized deduction is shown as an important factor of information society: the one likely to enrich the body of mathematical results (being in turn applied in high-tech), and profiting from inquiries into logics of higher orders.

Murawski's contribution explains how mechanized deduction techniques derive from theoretical achievements in logic. Namely, the results of Skolem and Herbrand demonstrate that if a theorem is true, this can be proved with an algorithm, hence in a finite number of steps. However, this does not hold if the theorem in question lacks truth. Then either one can prove in some cases the falsity of the given statement or the process does not stop.¹¹ These results are discussed in an earlier chapter of the book mentioned to which the author makes references.

After the introductory remarks, the paper tells the story of early attempts of applying computers to prove theorems, in particular the results of Davis, Newell-Shaw-Simon, Gilmore, Gelernter, Hao Wang and Davis-Putnam. The next sections deal with resolution and unification algorithms of Prawitz and Robinson, and with their modifications. They proved crucial for the further research in mechanization and automatization of reasonings.

The author does not continue his narration up to the point marked by Boolos' problem with its implication for automated theorem-proving. To complete the story, so relatively much attention is paid to that problem in the preceding section of this text. Another line of progress is discussed in the next paper.

2.2. Roman Matuszewski's contribution "On computer-assisted approach to formalized reasoning" is also concerned with the both constituents of this volume's title – decidability and tractability. As for the latter, we find the message that, for instance, the problem of deciding validity of propositional sentences (in conjunctive normal form) belongs to the co-NP class where we consider a sequence of finite ever larger search spaces to eventually establish validity.

Even more interesting is to hear that a major difficulty in formalizing mathematics turns out to lie in its practical unfeasibility, that is, intractability, rather than the impossibility of formalizing all mathematical proofs. It is believed by most, if not all, mathematicians – the author says – that one is able to formalize most of the present day mathematics using a sufficiently strong system of set theory, say ZFC, while valid theorems which cannot be derived in such a strong system are supposedly uninteresting statements which would not occur in the mathematical literature. Let me comment, we encounter here a philosophically intriguing question of what is involved in the realm of undecidability. If there is so as is claimed by the mentioned mathematicians, then a practical impact of limitative theorems, like those of Gödel, turns out to be relatively small; undecidable statements do exist, but they would not belong to what is interesting in mathematics. This issue seems to require a careful study.

The utility of the paper consists in detailed information on some research projects which endeavour to make the enterprise of mechanized deduction as tractable as possible by doing this in two ways: with enhancing inferential mechanism and by widening the base of mathematical knowledge. This is mostly exemplified through experiences of the Polish project MIZAR (in which the author takes part). As to the former, the system of natural deduction created in 1934 by Stanisław Jaśkowski is discussed as a theoretical inspiration for MIZAR. As to the latter, the role of sufficiently strong systems of set theory is emphasized.

The author inserts some historical remarks which should be treated as means to demonstrate the import of the problem as having so significant historical antecedents. With such an intention, it is usual that some authors see rather similarities than differences between the past and the

¹¹ Examples can be found in Jerzy Pogonowski's on-line draft of a textbook on logic, Section III.1, page 36, referred to from the page www.logic.amu.edu.pl/posluga_pogon.html.

present. However, arguments of Euclid are not formalized, likewise Aristotle's logic is not one to suit Hilbert's project. It was not until Leibniz that the idea of formalized and mechanized proof emerged, but with Leibniz it was just the idea, not its accomplishment (thus there is bit of exaggeration in saying that "Leibniz developed a calculus of reasoning"). Anyway, such references make sense to the effect that they hint at the main evolutionary line. When sketching that line, the author rightly stresses the role of Gentzen's *Hauptsatz* as a milestone in the way towards mechanized deduction.

2.3. Kazimierz Trzęsicki contributed the extensive essay: *From the idea of decidability to the number Ω* . Among its merits there is the due appreciation of the role of David Hilbert in the history of computation, the role closely tied with his *Entscheidungsproblem*. The author gives a sympathetic account of Hilbert's ideas and activities, starting from his Paris address of 1900. The story includes many crucial ideas in the form of literal quotations, and numerous historical details, as much picturesque as instructive, concerning Hilbert and his contemporaries. Special attention is paid to the Turing-Church Thesis and the problem of a chance of going beyond the limitations it claims. Not less interesting are footnotes remembering the past times of Leibniz's projects compared with current state of issues. The vivid narration combined with precise historical workshop, e.g. in quotations (preserving also the original language) and references, make the reading both pleasantly entertaining and instructive.

The next part of the essay that encourages reading is that dealing with Gödel's results and its relation to Hilbert Programme. A point which seems worth special recommending yields evidence against the often repeated view that Gödel's so-called First and Second theorems totally ruined Hilbert Programme. One after another authors repeat that opinion, while any diligent reader of *Grundlagen der Mathematik* by Hilbert and Bernays, a monumental work written after Gödel's discoveries, can ascertain what follows: that study involves a quiet analysis and continuation of Gödel's results, without any feeling of dissonance with Gödel. Trzęsicki documents that relationship from Gödel's point of view. A quotation from Gödel found by Trzęsicki deserves to be re-quoted here. Here are Gödel's thoughts from his letter to Constance Reid.

"I would like to call your attention to a frequently neglected point, namely the fact that Hilbert's scheme for the foundation of mathematics remains highly interesting and important in spite of my negative results. [...] As far my negative results are concerned, I would see their importance primarily in the fact

that in many cases they make it possible to judge, or to guess, whether some specific part of Hilbert's program can be carried through on the basis of given metamathematical presuppositions."

Let this comment be read in the light of Gödel's communication on the length of proofs (as discussed in the Section 1 above) and Hilbert's program be construed as a demand that for every proof there should be a formalism to make it amenable to mechanized deduction, not necessarily a formalism within the first-order arithmetic. Such a claim may be supported by the fact that Hilbert and Ackermann [1928] convincingly encourage to adopt the second-order logic for mathematical purposes. The last chapter of the said book starts with the section entitled "On the indispensability of a higher-order logic" (*Notwendigkeit einer Erweiterung des Kalküls*). Hence the historical evidence mentioned by Trzęsicki appears to be in a nice accord with the contention of Section 1 of this text.

Unfortunately, this accord gets spoiled through the Author's opinion (shared by him with some other writers) that Hilbert believed in the following thesis: any human reasoning may be expressed in the first-order logic. In a previous page the Author is a bit more cautious as he adds in parentheses that this so-called Hilbert Thesis was not formulated by him explicitly. This is to mean the thesis be formulated implicitly. But the sole implicit evidence as given by Trzęsicki is limited to an exemplification which consists in Hilbert's formalizing of a geometrical argument, due to Pascal, with the means of the first-order logic. But it was due to the very content of this argument that the first-order logic proved sufficient to formalize it.

Hence the reasoning of those who attribute to Hilbert the said Thesis looks like a naive generalization: in at least *one* case Hilbert used FOL as a suitable tool of formalization, hence Hilbert regarded that in *any* case FOL were a suitable tool of formalization. Apart of that example, if one carefully reads the passages (in [1928] book) referred to, one does not find any formulation, even implicit, like the so-called Hilbert Thesis. On the contrary, the mentioned content of the last chapter of [1928] book provides the explicit Hilbert's statement that mathematics needs higher-order logics.

The continuation of the issue of decidability, entitled *Entscheidungsproblem* is again reach in descriptions of historical setting. To give an example, the Author fittingly applies Rudolf Carnap's concept of explication (as a concept-forming procedure) to Turing's analysis of computation.

A subject still more capturing our interest is treated in the section entitled *Beyond the Church-Turing Thesis*. After a thorough discussion of the meaning of the Church-Turing Thesis, we get an exhausting and vividly narrated survey of various chances of going beyond the said Thesis. When

appreciating the Author's approach, I see a point for a bit of polemics, namely the following statement by Trzęsicki.

The Church-Turing Thesis tells about the procedure of calculation carried out by a human being. It does not say anything about the "calculation" realized in the nature by physical or biological processes.

Assuming that human beings belong to the physical and biological world (a point by no means controversial), one has to infer from the first sentence that the Thesis tells about procedures realized in the biological world, contrary to what the second sentence says. A serious problem is whether humans may surpass the Universal Turing Machine owing to their biological nature. Something like that was suggested by John von Neumann by the end of his booklet *The Computer and the Brain* [1958] where he opposed the historically formed logic and mathematics to the logic and mathematics of the brain as two highly different systems.

At last the reader arrives at the ending section which tells about the number Omega mentioned in the title of the essay. In a natural way this account crowns the discussion on the idea of decidability for it is concerned, so to speak, with the furthest limits of undecidability. The Letter Ω represents a real number denoting halting probability, that is, the chance that a program running at Turing Machine will eventually stop. The digits of this number are distributed in such a random way that any attempt to find a rule for predicting them is doomed to failure. Thus, this number, consisting of infinitely many 0's and 1's, has no recognizable pattern. Such is the message carried and commented in the final passages. The comment made at the very end incites a philosophical question about decidability of empirical problems, that is, those forming a set which subsumes the class of social research issues. Chaitin's results on Omega are by Trzęsicki applied to the physical world in order to conclude that the physical world, like mathematics, has a random structure. However, one may philosophically believe that all the physical quantities we encounter in the world are measurable with computable numbers. Such a belief seems to be entertained by some physicists. Some other people may regard that such a world would be too miraculous to be real. Anyway, the question seems to remain open.

2.4. The title of Witold Marciszewski's contribution resembles that of Stephen Wolfram [1985]: "Undecidability and intractability in theoretical physics". This is not to mean that this paper parallels Wolfram's in listing and commenting results concerning either undecidability or intractability in scientific theories, in this case theories in social research. When imitating

Wolfram's title, with exchanging "theoretical physics" for "social sciences", the author intends to encourage social theorists to ask themselves: do they in their work encounter analogous metatheoretical problems?

The answer should be in the affirmative, though the cases of such meta-theoretical awareness are not frequent among social scientists. Such a case occurs, although not quite explicitly, in the famous debate on socialist economic calculation between the Austrian School (Ludwig von Mises, Friedrich Hayek and others) and some socialist economists, especially Oskar Lange (see Section 2.2, Example 2). The Austrians are convinced that the enormous complexity of economic problems makes them intractable at the scale of a state economy. On the other hand, the free market, seen as information-processing system, provides us with such efficient methods of handling data as is computing being both parallel and interactive. These features exist solely in a free-market setting, where many independent units parallelly process only those, relatively small, data amounts which they need for their individual purposes (Hayek's idea of *dispersed knowledge*). Moreover, otherwise than in central planning, their information processing activity is thoroughly interactive, so they can learn from experiences, and adjust their strategies to a changing environment. Oskar Lange was certain that the rapid development of computational power will enable handling even most complex problems of central planning. As he died in 1965, he could not check his views against the background of the theory of computational complexity which did not appear until about 1970. It is a challenge for his present followers (there are such ones, though not much numerous) to confront the socialist project with the current state of complexity research in order to estimate chances of this project. In the paper some rudiments of complexity theory are reported as a background to put the problem of decidability and tractability in economics with respect to the current state of complexity research.

Other prerequisites to ask about the said properties of economic and social theories are due to the theory of games, devised by von Neumann and Morgenstern [1944]. It provides social researchers with a fitting model of human interactions motivated by quest for gains, while its mathematical formalism makes sense of asking computability questions. More recently, such inquiries profit much from addressing two computational studies, those on agent programs and those on cellular automata. It is natural to treat players as agents behaving according to rules for cellular automata, adjusted to problems and strategies of a game (the theory of such automata was devised by Stanisław Ulam and John von Neumann). A favourite case discussed in the theory of games is called "prisoner's dilemma" for its anecdotic plot

being concerned with the issue of solidarity of two accused in their defence in a trial. This is to exemplify the core of the question of how in a game to reasonably compare profits of loyal collaboration with a partner with those which one may win being selfish.

The paper, following recent literature, offers some examples of tractability results for the theory of games combined with the computational model of cellular automata. The data offered are not much abundant, but they should help to draw the following methodological moral.

To start, let us realize that any talking of tractability makes sense just with respect to those theories which have mathematical models. Such are some economic and social theories using, for instance, either a model provided by the theory of games or that of cellular automata (or both); this is why such theories alone are considered in the paper. Notwithstanding, sociologists, social psychologists, etc. used to report on their experimental results in the tone of full certainty, as if they relied on a perfect algorithm. It is up to philosophers to make their learned colleagues aware of two things.

First, if one uses a mathematical model, there may be problems which are in that model intractable or, even, undecidable. Like the question of whether a given game strategy will gain a permanent predominance over the rest (cf. item 1.3). A researcher, when observing a finite sequence of moves may be satisfied with perceiving that in this set such a predominance was the case, and thus declare his observation as a final result; however, such a success would be due rather to his ignorance than to experimental skill.

Second, if no mathematical model comes into play, then any empirical proceeding consists in guesses, or in the method of error and trial. Once upon a time, for example in the Vienna Circle, people believed in the so-called logic of induction which should have yielded algorithms to infer scientific laws from observational data. However that dream has gone with wind, especially with the wind of Popperian criticism. This is not to mean that all guesses are equally devoid of reliability. There is a fact significant for scientific method, to wit that there are varying degrees of reliability, depending, e.g., on preciseness of operational definitions of theoretical concepts involved.

2.5. A concrete and significant piece of own research in modelling social interactions is reported in the paper by Anna Gomolinska "Rough information granules in social agent system modelling". The paper nicely fits into the frame of tractability issues and exemplifies a high degree of methodological awareness. The computational status of some problems of social

interaction and knowledge engineering is diagnosed as their being computationally hard (exactly, NP-hard). Hence, because of such intractability, the researchers look for heuristics which are computationally more attractive though they provide us with suboptimal and approximate solutions alone. These are the other strategies, alternative to the speed-up phenomenon, as mentioned in the title of this essay.

The paper deals with the modelling of social, cognitive, and communicating agents, either natural or artificial. Agents and their systems are dynamic complex objects which can be represented by rough information granules. Information granules, mentioned in the title as this research subject, are defined as follows. An agent – an autonomous entity able to act on behalf of others or for itself – is viewed as a complex structure built of rough information granules; the same refers to multiagent systems. "Rough" is taken in the sense of rough sets theory as developed by Z. Pawlak; in the paper the classical Pawlak model of rough sets and its two extensions are recalled. Rules of some knowledge representation language are used as labels for information granules and, on the other hand, they are themselves objects to form information granules. An agent's knowledge and belief, value and norm systems, judgment systems, classification procedures, and (inter)action modules can be modelled as information granules consisting of rules. Patterns of interaction and systems of agents can be viewed as such granules as well.

A considerable merit of the theory, as developed by Gomolińska and the international team she belongs to, lies in the fact that it considers a wide spectrum of agents' utilities including moral values, cooperative attitudes etc. This constitutes a reasonable and desirable extension as compared with the standard approach of game theorists.

*
* *
*

Let me put in a nutshell how the content of this volume relates to its title. The preceding part of this introductory comment recalls Gödel's claim that decidability and tractability are system-relative: in the sense that problems lacking these properties in a particular system can obtain them in a stronger one. Trzęsicki's paper discusses a high and unconquerable threshold of undecidability constituted by the number Omega. Both Trzęsicki and Murawski much contribute to the picture of how logic has developed towards the issue of decidability.

Once the issue is settled in the negative, logicians focus on partial solutions to find means of mechanical decision procedures to such an extent as it could be done. This story is told by Murawski and by Matuszewski,

the first narrated in a wider historical perspective, the second focussing on most recent developments and exemplifying issues of tractability.

A similar relation of completing general considerations with a piece of concrete research holds between the contributions by Marciszewski and by Gomolińska, respectively, each applying to social sciences. The former states the question and offers a set of concepts to address it, the latter shows some means to devise a working model of game interactions, a model which proves efficient in spite of hard intractability, due to the art of obtaining some suboptimal and approximate solutions.

Clearly, this survey of problems, ideas and results is far from being either complete or duly detailed. However, it seems to shed some light on crucial issues of current logic and informatics and, thereby, issues of our civilization.

References

- Benzmüller, Christoph and Kerber, Manfred. A Challenge for Mechanized Deduction. 2001. www.cs.bham.ac.uk/mmk/papers/01-IJCAR.html.
- Boolos, George. A Curious Inference. *Journal of Philosophical Logic*, vol. 16, 1-12, 1987.
- Church, Alonzo. A note on the Entscheidungsproblem. *J. of Symbolic Logic* 1, 40-41 and 101-102, 1936.
- Davis, M. (ed). *Solvability, Provability, Definability. The Collected Works of Emil L. Post*. Birkhäuser, Boston etc. 1994.
- Feferman, S. Are There Absolutely Unsolvable Problems? Godel's Dichotomy. *Philosophia Mathematica*, 14(2), June 1, 134-152, 2006.
- Gödel, Kurt. Über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme – I. *Monatshefte für Mathematik und Physik*, vol. 38, 173-198, 1931.
- Gödel, Kurt. Über die Länge von Beweisen. *Ergeb. Math. Kolloquiums*, vol. 7, 23-24, 1936.
- Gödel, Kurt. *Kurt Gödel Collected Works*, vol. 1. Oxford Univ. Press, New York, 1986.
- Hartmanis J. and Stearns R. On the computational complexity of algorithms. *Transactions of the AMS* vol. 117, 285-306, 1965.
- Hilbert D. und Ackermann W. *Grundzüge der theoretischen Logik*. Springer, Berlin 1928.
- Kneale W. and Kneale M. *The Development of Logic*. Clarendon Press, Oxford 1962.
- Latsch, Wolfram. Androids and agents: do we need a non-computational economics? *Journal of Economic Methodology*, vol. 10, no. 3/September, 375-396, 2003.
- Marciszewski, Witold (ed.). *Dictionary of Logic as Applied in the Study of Language. Concepts, Methods, Theories*. Nijhoff, The Hague etc. 1981.
- Marciszewski, Witold. Predicate Logic. In: Marciszewski (ed.) [1981].
- Marciszewski, Witold and Murawski, Roman. *Mechanization of Reasoning in a Historical Perspective*. Rodopi, Amsterdam 1995.
- Marciszewski, Witold. Hypercomputational vs. computational complexity. A challenge for methodology of the social sciences. In: H. Świączkowska (ed.) *Free Market and Computational Complexity. Essays in Commemoration of Friedrich Hayek (1899-1992)*. Series: *Studies in Logic, Grammar and Rhetoric*, vol. 5(18), 2002. On line: logika.uwb.edu.pl/studies/vol5.html
- Parikh R. J. Some Results on the length of proofs. *Transactions of the American Mathematical Society*, vol. 177, March 1973.
- Post, E. L. Finite combinatory process – Formulation I. *J. of Symbolic Logic* vol. 1, 103-105, 1936. Reprinted in: Davis (ed.) [1994].
- Shanker, Natarajan. Using Decision Procedures with a Higher-Order Logic. citeseer.ist.psu.edu/update/483966.
- Surma, Stanisław. Deduction Theorem. In: Marciszewski (ed.) [1981].
- Turing, Alan. On computable numbers, with an application to the Entscheidungsproblem. *Proc. of the London Math. Society*, Series 2, 230-265, 1936.
- Von Neumann, John and Morgenstern, Oskar. *Theory of Games and Economic Behavior*. Princeton University Press 1944.
- Von Neumann, John. *The Computer and the Brain*. Yale Univ. Press, New Haven 1958.
- Wang, Hao. On Physicalism and Algorithmism: Can machines think? *Philosophia Mathematica*, 1(2), 97-138, 1993.
- Wolfram, Stephen. Undecidability and Intractability in Theoretical Physics, *Physical Review Letters* 54, pp. 735-738, 1985. On line: www.stephen-wolfram.com/publications/articles/physics/85-undecidability/

Witold Marciszewski
 University of Białystok
 and Białystok School of Public Administration
 Section of the Methodology of Social Sciences
 Home page: www.calculemus.org/witmar-pol.html

Roman Murawski
Adam Mickiewicz University

THE PRESENT STATE OF MECHANIZED DEDUCTION, AND THE PRESENT KNOWLEDGE OF ITS LIMITATIONS*

1. Introduction

In 1936 Alan Turing and Alonzo Church proved two theorems which seemed to have destroyed all hopes of establishing a method of mechanizing reasonings. Turing in (1936–37) reduced the decidability problem for theories to the halting problem for abstract machines modelling the computability processes (and named after him) and proved that the latter is undecidable. Church (1936) solving Hilbert's original problem proved the undecidability of the full predicate logic and of various subclasses of it.

On the other hand results of Skolem and Herbrand (cf. Chapter 6 of Marciszewski and Murawski, 1995) showed that if a theorem is true then this fact can be proved in a finite number of steps – but this is not the case if the theorem is not true (in this situation either one can prove in some cases the falsity of the given statement or the verification procedure does not halt). This semidecidability of the predicate logic was the source of hope and the basis of further searches for the mechanized deduction systems. Those studies were heavily stimulated by the appearance of computers in early fifties. There appeared the idea of applying them to the automatization of logic by using the mechanization procedures developed earlier. The appearance of computers stimulated also the search for new, more effective procedures.

In the sequel we shall describe the history of those researches. In Section 2 the early attempts of applying computers to prove theorems will be

* The financial support of the Committee for Scientific Research (grant no 1 H01A 041 27) is acknowledge.

presented, in particular we shall tell about results of Davis, Newell-Shaw-Simon, Gilmore, Gelernter *et al.*, Hao Wang and Davis-Putnam. Section 3 will be devoted to resolution and unification algorithms of Prawitz and Robinson and to their modifications. They turned out to be crucial for the further development of the researches towards mechanization and automatization of reasonings. We will sketch them in Section 4.

All those studies led to the idea of automated theorem proving by which one means the use of a computer to prove non-numerical results, i.e., to determine their truth (validity). One can demand here either a simple statement “proved” (what is the case in decision procedures) or human readable proofs. We can distinguish also two modes of operation: (1) fully automated proof searches and (2) man-machine interaction proof searches.

Note that the studies of mechanization of reasonings and automated theorem proving were motivated by two different philosophies. The first one – which we shall call the logic approach – can be characterized “by the presence of a dominant logical system that is carefully delineated and essentially static over the development stage of the theorem proving system. Also, search control is clearly separable from the logic and can be regarded as sitting ‘over’ the logic. Control ‘Heuristics’ exist but are syntax-directed primarily” (cf. Loveland 1984, p. 3). The second philosophical viewpoint is called the human simulation approach. It is generally the antithesis of the first one. It can be characterized shortly by saying that “the thrust is obviously simulation of human problem solving techniques” (cf. Loveland 1984, p. 3). Of course the logic and human simulation approaches are not always clearly delineated. Nevertheless this distinction will help us to order the results and to consider the influence of each of the approaches on various particular systems.

In the last Section 5 we indicate some limitations of mechanized deduction and automated theorem proving. In particular we distinguish effective and feasible computability/decidability/deducibility and show on some examples the complexity of decision procedures for certain theories. Connections with the famous problem $P \stackrel{?}{=} NP$ are also indicated (and some information on the origins of this problem is presented). On the other hand we discuss the speed-up phenomenon that appears when moving from the logic of the level n to the logic of the level $n + 1$ and – following Boolos – give a simple example of a formula which can be proved (in the usual sense) in the second-order logic but which has no proof that could be written down in the first-order logic.

2. First mechanized deduction systems

The different philosophical backgrounds mentioned above can be spotted already in the first studies towards mechanization of reasonings and automated theorem proving. In 1954 Martin Davis wrote a program to prove theorems in additive arithmetic (this program was never published in a paper). It was developed on the computer “Johniac” in the Institute for Advanced Study and was a straight implementation of the classical Presburger’s decision procedure for additive number theory (i.e. for the theory of non-negative integers in the language with zero, successor and addition only) (cf. Presburger 1929). The complexity of this decision procedure is very high and therefore the program proved only very simple facts (e.g. that the sum of two even numbers is also an even number – this was the first mathematical theorem in the history proved by a computer!).

The Presburger prover of M. Davis was an example of the logic approach. The second achievement in the field of automated theorem proving called the logic theorist should be included among examples of the human simulation approach. We mean here the program of A. Newell, J. S. Shaw and H. A. Simon presented in 1956 at the Dartmouth conference (cf. Newell *et al.* 1956). This program could prove some theorems in the propositional calculus of *Principia Mathematica* of A. N. Whitehead and B. Russell. Its goal was to mechanically simulate the deduction processes of humans as they prove theorems of the sentential calculus. Two methods were used: (1) substitution into established formulas to directly obtain a desired result and, failing that, (2) to find a subproblem whose proof represents progress towards proving the goal problem. This program was able to prove 38 theorems of *Principia*.

The Geometry Theorem-proving Machine of Gelernter and others from 1959 (cf. Gelernter *et al.* 1959, 1960) is also an example of the second approach. It applied the idea of M. Minsky that the diagram that traditionally accompanies plane geometry problems is a simple model for the theorem that could greatly prune the proof search. The program worked backwards, i.e., from the conclusion (goal) towards the premises creating new subgoals. The geometry model was used just to say which subgoals were true and enabled to drop the false ones. It should be noticed that this program was able to prove most high school exam problems within its domain and the running time was often comparable to high school student time.

Simultaneously with the Geometry Theorem-proving Machine two new efforts in the logic framework occurred. We mean here works of Gilmore and Hao Wang. They used methods derived from classical logic proof procedu-

res and in this way rejected opinions that logical methods cannot provide a useful basis for automated theorem proving. Such opinions were rather popular at that time. They were founded on the fact that logic oriented methods were inefficient and on the fact that the methods of Newell, Shaw, Simon and Gelernter proved to be successful. The method of P. C. Gilmore was based on Beth's semantic tableau technique. It was probably the first working mechanized proof procedure for the predicate logic – it proved some theorems of modest difficulty (cf. Gilmore 1959, 1960).

In the summer 1958 Hao Wang developed the first logic oriented program of automated theorem proving of IBM and continued this work at Bell Labs in 1959–63 (cf. Wang 1960, 1960a, 1961). Three programs were developed: (1) for propositional calculus, (2) for a decidable part of the predicate calculus and (3) for all of predicate calculus. Those programs were based on Gentzen-Herbrand methods, the last one proved about 350 theorems of *Principia Mathematica* (they were rather simple theorems of pure predicate calculus with identity).

During the Summer Institute for Symbolic Logic held at Cornell University, USA, in 1954 Abraham Robinson put forward, in his short lecture, the idea of considering the additional points, lines and circles – which must be used in a search for a solution of a geometrical problem – simply as elements of the so called Herbrand universe. This should enable us to abandon the geometrical constructs and to use directly Herbrand's methods.

This idea turned out to be very influential and significant. One of the first programs that realized it was implemented in 1960–62 by M. Davis and H. Putnam (cf. Davis–Putnam 1960). By Herbrand's theorem, the question of validity of a predicate calculus formula Z can be reduced to a series of validity questions about ever-expanding propositional formulas. More exactly one should consider so called Herbrand disjunctions $A_1 \vee \dots \vee A_n$ (which can be effectively obtained from Z). It holds that Z is valid if and only if there exists an n such that the disjunction $A_1 \vee \dots \vee A_n$ is valid. The formulas A_i are essentially substitution instances over an expanded term alphabet of Z with quantifiers removed. So one can test now $A_1 \vee \dots \vee A_n$ for ever-increasing n for example by truth table and conclude that Z is valid if among formulas $A_1 \vee \dots \vee A_n$ a tautology was found. But truth tables happen to be quite inefficient. The procedure of Davis and Putnam tried to overcome this difficulty. In fact they were considering unsatisfiability (instead of validity) of formulas and worked with conjunctive normal forms. Such a form is a conjunction of clauses, each clause being a disjunction of literals, i.e. atomic formulas (atoms) or their negations. The Davis–Putnam procedure can be described now as follows: “[it] made optimal use of simplification by can-

cellation due to one-literal clauses or because some literals might not have their complement (the same atomic formula but only one with a negation sign) in the formula. A simplified formula was split into two formulas so further simplification could recur anew” (cf. Loveland 1984).

3. Unification and resolution

The procedure of Davis and Putnam described in the last section had some defects. The main one was the enumeration substitutions – prior to this point substitutions were determined by some enumeration scheme that covered every possibility. Prawitz (1960) realized that the only important substitutions create complementary literals. He found substitutions by deriving a set of identity conditions (equations) that will lead to contradictory propositional formula if the conditions are met. In this way one got a procedure of substituting Herbrand terms. It is called today unification.

M. Davis developed right away the idea and combining it with the procedure of Davis–Putnam implemented it in a computer program based on a so called Linked-Conjunct method (cf. Davis 1963). This program used conjunctive normal forms of formulas and the unification algorithm developed by D. McIlroy in November 1962 at the Bell-Telephone-Laboratories. It was the first program which overcame the weaknesses of Herbrand procedure and improved it just by using conjunctive normal form and the unification algorithm (cf. Davis 1963 and Chinlund *et al.* 1964).

Simultaneously at Argonne National Lab near Chicago a group of scientists (G. Robinson, D. Carson, J. A. Robinson, L. Wos) was working on computer programs proving theorems. They used methods based on Herbrand theorem recognizing their inefficiency. Studying papers of Davis–Putnam and Prawitz they came to the idea of trying to find a general machine-oriented logical principle which would unify their ideas in a single rule of inference. Such a rule was found in 1963–64 by John Alan Robinson and published in (1965). It is called the resolution principle and is today one of the most fundamental ideas in the field of automated theorem proving. Therefore we shall describe it now more exactly.

The resolution principle is applied to formulas in a special form called conjunctive normal form. Given a formula A of the language of predicate calculus we first transform it into prenex normal form, i.e., to the form $(Q_1x_1)(Q_2x_2)\dots(Q_nx_n)B$ where Q_i ($i = 1, \dots, n$) are universal or existential quantifiers and B is quantifier-free (B is called matrix). One can show that the prenex normal form of a formula A is logically equivalent to the

formula A . As an example consider the following formula (stating that the function f is continuous):

$$\forall \varepsilon \{ \varepsilon > 0 \longrightarrow \exists \delta \{ \delta > 0 \wedge \forall x \forall y (|x - y| < \delta \longrightarrow |f(x) - f(y)| < \varepsilon) \} \}.$$

Its prenex normal form is:

$$\forall \varepsilon \exists \delta \forall x \forall y [\varepsilon > 0 \longrightarrow \delta > 0 \wedge (|x - y| < \delta \longrightarrow |f(x) - f(y)| < \varepsilon)].$$

Observe that prenex normal form of a formula is not uniquely determined but on the other hand all prenex normal forms of a given formula are logically equivalent.

The next step of the transformation of formulas we need is skolemization (cf. Section 6.5 of Marciszewski and Murawski, 1995). Recall here only that it consists of dropping of all the quantifiers and of replacing every occurrence of each variable bound by an existential quantifier by a functional term containing the variables that in this formula are bound by universal quantifiers preceding the considered existential quantifier. A formula obtained in this way is said to be in a skolemized prenex normal form or in its functional form for satisfiability. The skolemized prenex normal form of the formula from our above example is the following:

$$\varepsilon > 0 \longrightarrow g(\varepsilon) > 0 \wedge (|x - y| < g(\varepsilon) \longrightarrow |f(x) - f(y)| < \varepsilon).$$

Note that the skolemized normal form of a formula is not equivalent to the given formula but it is satisfiable (inconsistent) if and only if the given formula is satisfiable (inconsistent).

The next step of getting the conjunctive normal form consists of the elimination of connectives \longleftrightarrow and \longrightarrow and moving all negation signs \neg to the atoms. We proceed according to the following rules:

$$\begin{aligned} A \longleftrightarrow B & \text{ is logically equivalent to } (A \longrightarrow B) \wedge (B \longrightarrow A) \\ A \longrightarrow B & \text{ is logically equivalent to } (\neg A \vee B) \\ \neg(A \wedge B) & \text{ is logically equivalent to } (\neg A \vee \neg B) \\ \neg(A \vee B) & \text{ is logically equivalent to } (\neg A \wedge \neg B) \\ \neg\neg A & \text{ is logically equivalent to } A. \end{aligned}$$

A formula obtained in such a way is said to be in the negation normal form. For example the negation normal form of the formula considered above is:

$$\neg \varepsilon > 0 \vee [g(\varepsilon) > 0 \wedge (\neg |x - y| < g(\varepsilon) \vee |f(x) - f(y)| < \varepsilon)].$$

Note that atoms and negated atoms are usually called literals.

The last step of the considered transformation consists of the application of the following distributivity laws:

$$\begin{aligned} A \vee (B \wedge C) & \longleftrightarrow (A \vee B) \wedge (A \vee C), \\ (A \wedge B) \vee C & \longleftrightarrow (A \vee C) \wedge (B \vee C). \end{aligned}$$

In this way the obtained formula is of the form of conjunctions of disjunctions of literals. Such a form is called the conjunctive normal form and the disjunctions of literals are called clauses. Clauses consisting of a single literal are called unit clauses. Clauses with only one positive literal are called Horn clauses. They correspond to formulas of the form $A_1 \wedge \dots \wedge A_n \longrightarrow B$. To illustrate the last step note that the conjunctive normal form of our formula stating that a function f is continuous is the following:

$$(\neg \varepsilon > 0) \vee [g(\varepsilon) > 0 \wedge \neg |x - y| < g(\varepsilon)] \vee [g(\varepsilon) > 0 \wedge |f(x) - f(y)| < \varepsilon].$$

Note that if $(A_1^1 \vee \dots \vee A_{n_1}^1) \wedge \dots \wedge (A_1^l \vee \dots \vee A_{n_l}^l)$ is a conjunctive normal form of a formula A then we write it sometimes also as: $\{A_1^1 \vee \dots \vee A_{n_1}^1, \dots, A_1^l \vee \dots \vee A_{n_l}^l\}$ or as $\{\{A_1^1, \dots, A_{n_1}^1\}, \dots, \{A_1^l, \dots, A_{n_l}^l\}\}$. This form is called the clause form. In this way we have shown that for any formula A of the language of predicate calculus there exists a formula A' in conjunctive normal form such that the formula A is satisfiable (inconsistent) if and only if the formula A' is satisfiable (inconsistent).

Having described the needed form of formulas we can introduce now the resolution principle. First observe that a formula B is a logical consequence of formulas A_1, \dots, A_n if and only if the formula $A_1 \wedge \dots \wedge A_n \wedge \neg B$ is inconsistent, i.e., unsatisfiable. Let \square denote an empty clause (i.e., a contradiction) and let formulas $A_1, \dots, A_n, \neg B$ be in conjunctive normal form. Hence to show that B follows logically from A_1, \dots, A_n it suffices to prove that \square is contained in the set \mathcal{S} of all clauses constituting $A_1, \dots, A_n, \neg B$ or that \square can be deduced from this set (the sense of the word 'deduce' will be explained below). This is the main idea of the method of resolution. Hence we can say that this method is a negative test calculus.

The resolution calculus introduced by J. A. Robinson (1965) is a logical calculus in which one works only with formulas in clause form. It has no logical axioms and only one inference rule (the resolution rule). The simplest version of this rule has the following form: if C_1 and C_2 are two clauses such that C_1 contains a literal L_1 and C_2 contains a literal L_2 which is inconsistent with L_1 (i.e., L_1 and L_2 are complementary literals) then one obtains a new clause C consisting of all literals of C_1 except L_1 and all literals of C_2 except L_2 . Symbolically it can be written as:

$$\begin{array}{l} \text{clause } C_1 : K_1 \vee \dots \vee K_n \vee L \\ \text{clause } C_2 : \frac{M_1 \vee \dots \vee M_m \vee \neg L}{K_1 \vee \dots \vee K_n \vee M_1 \vee \dots \vee M_m} \end{array}$$

The clause $K_1 \vee \dots \vee K_n \vee M_1 \vee \dots \vee M_m$ is called the resolvent of C_1 and C_2 and clauses C_1 and C_2 are called parent clauses. We say that L_1 and L_2 are the literals resolved upon when the resolvent exists. Observe that the resolvent is a logical consequence of the parent clauses.

Let \mathcal{S} be a set of clauses. A refutation (or a proof of unsatisfiability) of \mathcal{S} is a finite sequence C_1, \dots, C_n of clauses such that (1) any C_i ($i = 1, \dots, n$) either belongs to \mathcal{S} or there exist C_j and C_k , $j, k < i$ such that C_i is a resolvent of C_j and C_k and (2) the last clause C_n is \square . This explains what we meant above by saying that \square can be “deduced” from the set \mathcal{S} of clauses.

We give now two examples.

A. We show that U is a logical consequence of formulas $P \longrightarrow S$, $S \longrightarrow U$ and P . Indeed it suffices to show that the formula $(P \longrightarrow S) \wedge (S \longrightarrow U) \wedge P \wedge \neg U$ is unsatisfiable (inconsistent). Writing it in conjunctive normal form we get

$$(\neg P \vee S) \wedge (\neg S \vee U) \wedge P \wedge \neg U.$$

Its clause form is $\{\neg P \vee S, \neg S \vee U, P, \neg U\}$. Denote this set of clauses by \mathcal{S} . The following sequence of formulas is a proof of unsatisfiability of \mathcal{S} :

- (1) $\neg P \vee S$
- (2) $\neg S \vee U$
- (3) P
- (4) $\neg U$
- (5) $\neg P \vee U$ resolvent of (1) and (2)
- (6) U resolvent of (3) and (5)
- (7) \square resolvent of (4) and (6).

Hence U is a logical consequence of $P \longrightarrow S$, $S \longrightarrow U$ and P .

B. We show that the formula $\neg Q$ is a logical consequence of $P \longrightarrow \neg Q \vee (R \wedge S)$, $P, \neg S$. Hence consider the formula

$$(P \longrightarrow (\neg Q \vee (R \wedge S))) \wedge P \wedge \neg S \wedge Q.$$

Its clause form is

$$\mathcal{S} = \{\neg P \vee \neg Q \vee R, \neg P \vee \neg Q \vee S, P, \neg S, Q\}.$$

We have the following proof of unsatisfiability of \mathcal{S} :

- (1) $\neg P \vee \neg Q \vee R$
- (2) $\neg P \vee \neg Q \vee S$
- (3) P
- (4) $\neg S$
- (5) Q
- (6) $\neg P \vee \neg Q$ resolvent of (2) and (4)
- (7) $\neg P$ resolvent of (5) and (6)
- (8) \square resolvent of (3) and (7).

So far we considered the simplest form of the resolution principle and its application in the propositional calculus. In the case of formulas containing variables the whole situation is more complicated.

First we describe a substitution device called unification (we shall do it following Chang-Lee 1973). By a substitution we mean a finite set of the form $\{t_1/v_1, \dots, t_n/v_n\}$ where v_i are variables and t_i are terms different from v_i . An empty substitution will be denoted by ε . If $\theta = \{t_1/v_1, \dots, t_n/v_n\}$ is a substitution and E is a formula then by $E\theta$ we denote the formula $E(v_1/t_1, \dots, v_n/t_n)$. Observe that substitutions can be composed, i.e., if $\theta = \{t_1/x_1, \dots, t_n/x_n\}$ and $\lambda = \{u_1/y_1, \dots, u_m/y_m\}$ are two substitutions then by $\theta \circ \lambda$ we shall denote a composition of θ and λ and define it as a substitution obtained from the set $\{t_1\lambda/x_1, \dots, t_n\lambda/x_n, u_1/y_1, \dots, u_m/y_m\}$ by removing from it all the elements $t_j\lambda/x_j$ such that $t_j\lambda = x_j$ and all the elements u_i/y_i such that $y_i \in \{x_1, \dots, x_n\}$. Note that the composition \circ is associative and that ε is the left and right unit, i.e., $\varepsilon \circ \theta = \theta \circ \varepsilon = \theta$.

A substitution θ is said to be a unifier of a set of formulas $\{E_1, \dots, E_k\}$ if and only if $E_1\theta = E_2\theta = \dots = E_k\theta$. If there exists a unifier of a set $\{E_1, \dots, E_k\}$ then this set is said to be unifiable. A unifier σ of the set $\{E_1, \dots, E_k\}$ is called a most general unifier if and only if for any unifier θ of this set there exists a substitution λ such that $\theta = \sigma \circ \lambda$. J. A. Robinson showed that for any set \mathcal{S} of formulas there exists at most one most general unifier.

We shall describe now an algorithm of finding a most general unifier. It is called the unification algorithm.

Let \mathcal{S} be a nonempty set of formulas. A disagreement set of \mathcal{S} is defined as follows: one indicates the first (from the left) position such that there are two formulas from \mathcal{S} which differ on this position. Then for

every element E of \mathcal{S} we write its subformula beginning with the symbol being written on this position. The set of these subformulas is just the disagreement set of \mathcal{S} . For example the disagreement set of the set $\{P(x, f(y, z)), P(x, a), P(x, g(h(k(x))))\}$ is $\{f(y, z), a, g(h(k(x)))\}$.

The unification algorithm can now be given by the following rules. Let a set \mathcal{S} of formulas be given.

Step 1: $\mathcal{S}_0 = \mathcal{S}$, $\sigma_0 = \varepsilon$.

Step 2: If \mathcal{S}_k is a unit clause then STOP and σ_k is the most general unifier of \mathcal{S} . Otherwise let D_k be the disagreement set of \mathcal{S}_k .

Step 3: If there are v_k and t_k in D_k such that v_k is a variable not occurring in t_k then move to Step 4. Otherwise STOP: \mathcal{S} is not unifiable.

Step 4: Let $\sigma_{k+1} = \sigma_k \circ \{t_k/v_k\}$ and $\mathcal{S}_{k+1} = \mathcal{S}_k\{t_k/v_k\}$. (Observe that $\mathcal{S}_{k+1} = \mathcal{S}\sigma_{k+1}$.)

Note that the unification algorithm always halts when applied to a finite nonempty set of formulas. J. A. Robinson proved that if \mathcal{S} is a finite nonempty unifiable set of formulas then the unification algorithm halts on Step 2 and the last σ_k is the most general unifier of \mathcal{S} .

We give some examples. First find a most general unifier of the set $\mathcal{S} = \{Q(f(a), g(x)), Q(y, y)\}$. We proceed as follows:

1. $\sigma_0 = \varepsilon$, $\mathcal{S}_0 = \mathcal{S}$.
2. Since \mathcal{S}_0 is not a unit clause we find a disagreement set D_0 of \mathcal{S}_0 . We have $D_0 = \{f(a), y\}$. Hence $v_0 = y$, $t_0 = f(a)$.
3. $\sigma_1 = \sigma_0 \circ \{t_0/v_0\} = \sigma_0 \circ \{f(a)/y\} = \{f(a)/y\}$,
 $\mathcal{S}_1 = \mathcal{S}_0\{t_0/v_0\} = \{Q(f(a), g(x)), Q(f(a), f(a))\}$.
4. \mathcal{S}_1 is not a unit clause and we have $D_1 = \{g(x), f(a)\}$. By Step 3 we conclude that the set \mathcal{S} is not unifiable.

We give one more example.

Let $\mathcal{S} = \{P(a, x, f(g(x))), P(z, f(z), f(u))\}$. Find a most general unifier of \mathcal{S} . We proceed as follows:

1. $\sigma_0 = \varepsilon$, $\mathcal{S}_0 = \mathcal{S}$.
2. $D_0 = \{a, z\}$ and we have $v_0 = z$, $t_0 = a$.
3. $\sigma_1 = \sigma_0 \circ \{t_0/v_0\} = \{a/z\}$,
 $\mathcal{S}_1 = \mathcal{S}_0\{t_0/v_0\} = \{P(a, x, f(g(y))), P(a, f(a), f(u))\}$.
4. $D_1 = \{x, f(a)\}$ and we put $v_1 = x$, $t_1 = f(a)$.
5. $\sigma_2 = \sigma_1 \circ \{t_1/v_1\} = \{a/z, f(a)/x\}$,
 $\mathcal{S}_2 = \mathcal{S}_1 \circ \{t_1/v_1\} = \{P(a, f(a), f(g(y))), P(a, f(a), f(u))\}$.
6. $D_2 = \{g(y), u\}$ and we put $v_2 = u$, $t_2 = g(y)$.

$$\begin{aligned} 7. \sigma_3 &= \sigma_2 \circ \{t_2/v_2\} = \{a/z, f(a)/x, g(y)/u\}, \\ \mathcal{S}_3 &= \mathcal{S}_2\{t_2/v_2\} = \{P(a, f(a), f(g(y)))\}. \end{aligned}$$

By Step 2 the set \mathcal{S} is unifiable and σ_3 is the most general unifier of it.

We have to define two more notions to formulate at last the general form of the resolution rule. If two or more literals (with the same sign) of a clause C have a most general unifier σ then the clause $C\sigma$ is called a factor of C . Let C_1 and C_2 be two clauses which have no common variable and let L_1 and L_2 be two literals of C_1 and C_2 , resp. If L_1 and $\neg L_2$ have a most general unifier σ then the clause $(C_1\sigma - L_1\sigma) \cup (C_2\sigma - L_2\sigma)$ is called a binary resolvent of C_1 and C_2 . The clauses C_1 and C_2 are called parent clauses and we say that L_1 and L_2 are literals resolved upon. A resolvent of two parent clauses C_1 and C_2 is now defined as one of the following resolvents: (1) binary resolvent of C_1 and C_2 , (2) binary resolvent of C_1 and a factor of C_2 , (3) binary resolvent of C_2 and a factor of C_1 , (4) binary resolvent of a factor C_1 and a factor of C_2 .

Now we can define a resolution deduction. Let a set \mathcal{S} of clauses and a clause A be given. A resolution deduction of A from \mathcal{S} is a finite sequence C_1, \dots, C_n of clauses such that: (1) C_n is A , (2) for any i , $1 \leq i \leq n$, C_i is either a member of \mathcal{S} or there exist $j, k < i$ such that C_i is a resolvent of C_j and C_k (i.e., C_i is obtained from C_j and C_k by the resolution rule). A resolution deduction of the empty clause \square from \mathcal{S} is called a refutation (or a proof of unsatisfiability) of \mathcal{S} .

We shall give some examples (cf. Chang-Lee 1973).

1. Show that the formula $B =: \exists x(S(x) \wedge R(x))$ is a logical consequence of formulas $A_1 =: \forall x[P(x) \rightarrow (Q(x) \wedge R(x))]$ and $A_2 =: \exists x(P(x) \wedge S(x))$. It suffices to show that the formula $A_1 \wedge A_2 \wedge \neg B$ is unsatisfiable. Transform the given formulas $A_1, A_2, \neg B$ into conjunctive normal form. We get the following formulas, resp.:

$$\begin{aligned} &(\neg P(x) \vee Q(x)) \wedge (\neg P(x) \vee R(x)), \\ &P(a) \wedge S(a), \\ &\neg S(x) \vee \neg R(x). \end{aligned}$$

We can construct now the following resolution deduction:

$$\begin{array}{l} (1) \neg P(x) \vee Q(x) \\ (2) \neg P(x) \vee R(x) \\ (3) P(a) \\ (4) S(a) \\ (5) \neg S(x) \vee \neg R(x) \end{array} \left. \begin{array}{l} \} \\ \} \\ \} \\ \} \\ \} \end{array} \begin{array}{l} \text{from } A_1 \\ \text{from } A_1 \\ \text{from } A_2 \\ \text{from } A_2 \\ \text{from } \neg B \end{array}$$

- (6) $R(a)$ resolvent of (2) and (3)
 (7) $\neg R(a)$ resolvent of (4) and (5)
 (8) \square resolvent of (6) and (7).

Hence we have shown that B is a logical consequence of A_1 and A_2 .

2. We show that the formula $B =: \exists x(P(x) \wedge R(x))$ is a logical consequence of the following formulas:

- $A_1 =: \forall x[Q(x) \wedge \neg T(x) \longrightarrow \exists y(S(x, y) \wedge R(y))]$,
 $A_2 =: \exists x[P(x) \wedge Q(x) \wedge \forall y(S(x, y) \longrightarrow P(y))]$,
 $A_3 =: \forall x(P(x) \longrightarrow \neg T(x))$.

Transforming the formulas A_1, A_2, A_3 and $\neg B$ into conjunctive normal form we get the following clauses:

- | | | |
|---|---------------|------------|
| (1) $\neg Q(x) \vee T(x) \vee S(x, f(x))$ | } | from A_1 |
| (2) $\neg Q(x) \vee T(x) \vee R(f(x))$ | | |
| (3) $P(a)$ | } | from A_2 |
| (4) $Q(a)$ | | |
| (5) $\neg S(a, y) \vee P(y)$ | | |
| (6) $\neg P(x) \vee \neg T(x)$ | from A_3 | |
| (7) $\neg P(x) \vee \neg R(x)$ | from $\neg B$ | |

The needed deduction of \square can be the following:

- (8) $\neg T(a)$ resolvent of (3) and (6)
 (9) $T(a) \vee R(f(a))$ resolvent of (2) and (4)
 (10) $R(f(a))$ resolvent of (8) and (9)
 (11) $T(a) \vee S(a, f(a))$ resolvent of (1) and (4)
 (12) $S(a, f(a))$ resolvent of (8) and (11)
 (13) $P(f(a))$ resolvent of (5) and (12)
 (14) $\neg R(f(a))$ resolvent of (7) and (13)
 (15) \square resolvent of (10) and (14).

The method of resolution has an important property called the refutation completeness. One can prove that a set \mathcal{S} of clauses is unsatisfiable if and only if there exists a resolution deduction of the empty clause \square from \mathcal{S} . The proof of this fact uses semantical trees and Herbrand's theorem. Hence the method gives a semidecidability of the predicate logic: if a given formula is a logical consequence of a given set \mathcal{S} of formulas then by a systematic application of the resolution rule we get in a finite number of steps the empty

clause \square , but if it is not a logical consequence then sometimes one can decide it after a finite number of steps but in general the procedure does not halt. On the other hand the considered method does not give a procedure of finding a formal proof of a formula B on the basis of formulas A_1, \dots, A_n in the case when B is a logical consequence of A_1, \dots, A_n .

Recall that we used in the resolution most general unifiers, i.e., most general substitutions that allow the equality of literals. This guarantees the elimination of branching of search due to different possible substitutions that equate those atoms but lead to different clauses. Therefore the method of resolution is simple, elegant and powerful. But the world is not so perfectly beautiful – this method has also some defects. If one generates from a given set of clauses new clauses by the method of resolution then they accumulate at a rapid rate. Indeed given a set \mathcal{S} of clauses one can obtain new clauses systematically using the level-saturation method which is described by the following equations: $\mathcal{S}^0 = \mathcal{S}$, $\mathcal{S}^{n+1} = \{\text{resolvents of } C_1 \text{ and } C_2 : C_1 \in \mathcal{S}^0 \cup \dots \cup \mathcal{S}^n, C_2 \in \mathcal{S}^n\}$, $n = 1, 2, \dots$. In this way we get all the resolvents of all pairs of elements of \mathcal{S} , add them to \mathcal{S} , further we calculate all the resolvents of elements of this new set, etc., till we come to the empty clause \square . Among clauses generated in this procedure there are many irrelevant ones and the total number of clauses grows rapidly. Hence the idea of improving the method of resolution by finding restrictions and strategies to control the growth of the number of clauses. We shall tell here only about some of the proposed improvements (Loveland 1978 summarizes in Appendix twenty five such improvements but more exist). By a restriction of resolution we mean a variant for which some clauses generated by the basic resolution procedure are not generated. A strategy of resolution only rearranges the order of generation to get likely useful clauses earlier.

One of the earliest refinements of resolution is unit-preference introduced in Wos-Carson-Robinson (1964). This strategy guarantees that the resolvent is shorter than the longer parent clauses. In the same paper L. Wos, D. F. Robinson and G. A. Carson introduced the set-of-support restriction. It can be described as follows: one chooses a subset \mathcal{T} (called a support) of a given set \mathcal{S} of clauses and then two clauses from $\mathcal{S} - \mathcal{T}$ are never resolved together. This means that every resolvent has in its deduction history some clause of \mathcal{T} . In practice \mathcal{T} is usually chosen to be a (sub)set of the clauses special to the considered problem.

J. A. Robinson in (1965a) introduced a restriction called hyperresolution. It restricts resolutions to where one parent clause contains only positive literals and any resolvent containing a negative literal is immediately used in all permitted resolutions and then discarded.

D. W. Loveland (1970) and D. Luckham (1970) proposed another restriction called linear resolution (Luckham used the name “ancestry-filter form”). It constrains the deduction so that a new clause is always derived from the preceding clause of the deduction by resolving against an earlier clause of the deduction. In this way one is always seeking to transform the last clause obtained into a clause closer to the goal clause. This method was further developed by R. Anderson and W. W. Bledsoe (1970). R. Yates, B. Raphael and T. Hart (1970), R. Reiter (1971), D. W. Loveland (1972) and R. Kowalski and D. Kuehner (1971).

D. W. Loveland (1968) and (1969) introduced a procedure which is not a resolution procedure but is close to a very restricted form of linear resolution – it is called model elimination. R. Kowalski and D. Kuehner (1971) provided the translation of it into a very restricted linear resolution format called SL-resolution. It was used by A. Colmerauer and P. Roussel to an early version of a programming language Prolog.

There appeared a number of resolution refinements which reduce multiple derivations of the same clause by ordering literals in clauses. An example of this type of procedures is the method called locking or lock-resolution introduced by R. S. Boyer (1971). Its main idea is to use indices to order literals in clauses from a given set of clauses. The occurrences of literals are indexed by integers. Then resolution need be done using only the lowest indexed integer of each clause.

We should mention here also the semantic resolution of J. R. Slagle (1967) which generalizes hyperresolution of J. A. Robinson (1965a), resolution with renaming of B. Meltzer (1966) and the set-of-support restriction of L. Wos, G. A. Robinson and D. F. Carson (1965).

All types of refinements of resolution given above are refutation complete. There exist also two forms of resolution restrictions which are incomplete. We mean here unit clause resolution and input clause resolution. The former was introduced by L. Wos, D. F. Carson and G. A. Robinson (1964). It permits resolution only when one parent is a unit clause, i.e., it consists of one literal. The input clause resolution was introduced by C. L. Chang (1970). It is a restricted form of linear resolution where one parent is always an input (given) clause. Chang proved that the unit clause and input clause resolutions are of equal power – they are complete over the class of Horn formulas.

Most mathematical theories contain among its symbols the equality symbol and among its axioms – the equality axioms. The immediate application of the usual resolution procedure generates in this case a lot of undesired clauses. Hence L. Wos and G. A. Robinson (1970) introduced

a procedure called paramodulation. This is the equality replacement rule with unification. It replaces all equality axioms except certain reflexivity axioms for functions. When paramodulation is restricted to replacement of the (usually) shorter term by the longer term with no instantiation allowed in the formula incorporating the replacement then one uses the term demodulation (cf. Wos-Robinson-Carson-Shalla 1967). It should be mentioned that there are also two other systems treating equality: system introduced by E. E. Sibert (1969) and E-resolution introduced by J. B. Morris (1969).

To finish this section we should add that almost simultaneously with Robinson’s invention of resolution, J. Ju. Maslov in the USSR introduced a proof procedure very close to resolution in spirit. His method is called the inverse method and it is a test for validity rather than unsatisfiability (cf. Maslov 1964, 1971 and Kuehner 1971).

4. Development of mechanized deduction after 1965

The last section was devoted to the method of resolution and to its refinements. This approach to the mechanization and automatization of reasonings was dominating in the sixties. Nevertheless there were developed also other methods. In this section we shall discuss them briefly and sketch further development of the researches in the field after 1965.

We should begin with the Semi Automated Mathematics (SAM) project that spanned 1963 to 1967. It belongs to the human simulation approach (cf. Section 1). In the framework of this project a succession of systems designed to interact with a mathematician was developed. They used many sorted ω -order logic with equality and λ -notation. The system SAM I was a proof checker but the theorem proving power continued to increase through SAM V which had substantial automatic capability. Only a part of the work in this project got recorded in the literature – cf. J. R. Guard-F. C. Oglesby-J. H. Bennett-L. G. Settle (1969).

Another project belonging to the human simulation approach developed in the mid-sixties was ADEPT, the Ph. D. thesis of L. M. Norton (cf. Norton 1966). It was a heuristic prover for group theory.

The dominant position of the resolution methods brought sharp criticism from some researchers. Their main argument was that there cannot be a unique procedure which would suffice to realize (to simulate) the real intelligence. They stressed the necessity of using many components. One of those critics was M. Minsky from MIT. In 1970 C. Hewitt, a Ph. D.

student at MIT wrote a dissertation on a new programmic language called PLANNER (cf. Hewitt 1971). Its goal was to structure a theorem prover system in such a way that locally distributed knowledge could be represented at various positions of the proving program. In fact it was not a theorem prover *per se*, but a language in which a “user” was to write his own theorem prover, specifically tailored to the problem domain at hand. This language was never fully implemented, only a subset of it was realized (microPLANNER).

About the same time another effort in human simulation approach was undertaken by A. Nevins (cf. Nevins 1974, 1975, 1975a). He built in fact at least two provers that were able to prove theorems which most resolution provers could not touch. For example Nevins could prove fully automatically that:

$$x^3 = e \longrightarrow f(f(a, b), b) = e$$

where $f(x, y) = xyx^{-1}y^{-1}$. This result is much harder than the implication: “ $x^2 = e \longrightarrow$ the group is abelian” which constituted the limit of capability of ADEPT.

We should mention also works of the group of scientists gathered around W. W. Bledsoe at the University of Texas which proved to be very important. They were working not for single uniform rule of inference for the whole mathematics but were seeking specific methods for particular domains of mathematics such as analysis, set theory or nonstandard analysis (cf. Bledsoe 1983, 1984).

So far we spoke about studies in seventies of the automated theorem proving which could be classified as human simulation approach. It does not mean that this was the only direction. There were also some new ideas within the logic approach. We should say here first of D. E. Knuth and P. B. Bendix (1970). They used the idea of rewrite rules – a device familiar to logicians. It is a replacement rule and allow to replace the left hand side by the right hand side at any occurrence of the left hand side. In equational theories one converts equations just to rewrite rules. Those rules enable us to reduce terms and to equate them. Knuth and Bendix proposed an algorithm which for a class of equational theories gave a complete set of rewrite rules – i.e., a set of rewrite rules sufficient to check the truth of every equation of the theory by demanding that equal terms reduce to the same normal form. An example of a theory to which the algorithm applies is the theory of groups (while the theory of abelian groups is not). There is an open problem connected with this algorithm: what equational theories have complete sets of rewrite rules?

Recall that the unification algorithm (described in the previous section) permits computation of a most general substitution for variables to make atomic formulas identical. Working with special theories, usually equational theories, one can simplify the procedure given by unification and resolution by introducing special unification. Its main idea is that the usual unification is augmented by equations or rewrite rules obtained from axioms of the considered theory. This idea was first formalized by G. D. Plotkin (1972) and further developed by M. E. Stickel (1981), (1985) and M. Livesey and J. Siekmann (1976). The idea of theory resolution of Stickel can be summarized as follows: since the resolution rule enlarges the whole number of steps, it is desirable to find macrorules in which certain sequences of steps could be performed in one step.

Another example of results which should be classified as logic oriented is the system of R. Overbeek developed later by S. Winker, E. Lusk, B. Smith and L. Wos and named AURA (Automated Reasoning Assistant). It was based on the old (i.e., coming from the sixties) ideas of unit preference, set-of-support for resolution, paramodulation and demodulation to which hyperresolution as well as more flexibility in demodulation and preprocessors for preparation of input from a variety of formats were added.

Around 1973 there appeared a very interesting effort different from the resolution approach and the strongly human oriented prover of Bledsoe. We mean here the Computational Logic Theorem Prover of R. S. Boyer and J. S. Moore (1975), (1979), (1981). This system uses the language of quantifier-free first order logic with equality and includes a general induction principle among the inference rules. It can be used to work within traditional mathematics (e.g., number theory) as well as to prove properties of programs and algorithms (so called proofs of correctness).

We should tell also about graph representation and about prover for systems of higher order. The former is based on the idea of enriching the structure of basic data with additional information, e.g. by representing the potential resolution steps in the graph structure. Literals or clauses and possible complementary literals form vertices of graphs which are connected by edges. This approach was introduced by R. Kowalski (1975) (cf. also S. Sickel 1976, R. E. Shostak 1976 and P. Andrews 1976).

The first proving system for higher order logics was developed by a group of scientists working under the direction of J. R. Guard in the early sixties. The studies were continued by W. E. Gould (1966), G. P. Huet (1975) and D. C. Jensen and T. Pietrzykowski (1976). The most important and influential group of people working in this direction is today at the Carnegie-Melon University (its chief is P. B. Andrews). They developed

in the late seventies a theorem prover for type theory (TPS) (cf. Andrews 1981 and Miller, Cohen, Andrews 1982). It can prove for example Cantor's theorem as well as numerous first order theorems.

We shall finish this survey of activities in the field of mechanization and automatization of reasonings in the sixties and seventies by mentioning an ambitious theorem prover now being developed at the University of Karlsruhe and named Markgraf Karl Refutation Procedure (cf. K. Bläsius, N. Eininger, J. Siekmann, G. Smolka, A. Herold, C. Walther 1981).

5. Some limitations

Having described so far the positive achievements in the field of mechanized deduction and automated theorem proving let us turn now to the discussion of some (essential) limitations in this process.

It is a trivial observation that an automatic theorem prover would have wide application if it operated effectively enough. Hence it is useful to distinguish effective and feasible computability (decidability). Both are intuitive (nonformal) concepts. Recall that a problem is said to be effectively decidable (a function is effectively computable) if and only if there exists a definite mechanical procedure which can solve in a finite number of prescribed steps every instance of the problem (calculate the value of the function for any given arguments). It is believed that the concept of recursive computability (or equivalently of Turing machine computability) is an adequate mathematical formalization (counterpart) of this concept – this is stated by Church's Thesis. On the other hand by "feasible" one means "computable in practice" or "computable in the real world". As a mathematical model of this intuitive notion one can consider the concept of polynomial time computability, i.e., computability by a deterministic Turing machine in the time bounded by a polynomial of the size of the input, hence by a deterministic Turing machine that needs a number of steps bounded by a polynomial of the size of the input. Denote by P the class of predicates (problems) recognizable (solvable) in a polynomial time. A closely related class is the class NP of problems recognizable in nondeterministic polynomial time. There is an open problem whether problems in NP are feasibly decidable or are polynomial time decidable – it is shortly denoted as $P = ?NP$ and is a central problem in the contemporary computer science.

It seems that this problem was stated (in a certain sense) for the first time in a letter of Kurt Gödel to John von Neumann from 20th March 1956 (cf. Gödel 2003, p. 373–375). Gödel was thinking about computational

complexity of Turing machine computations and asked von Neumann about the computational complexity of a problem (which is in fact an NP complete problem; note that we are using the modern terminology, in Gödel's letter the problem is not referred to as NP complete) about proof systems and wondered if it could be solved in linear or possibly quadratic time. He asked how hard is it (computationally) to decide if a statement has a proof of length n in a formal system (it is of course a question about the fundamental nature of mathematics). It is worth quoting here some fragments of Gödel's letter:

Obviously, it is easy to construct a Turing machine that allows us to decide, for each formula F of the restricted functional calculus and every natural number n , whether F has a proof of length n [length = number of symbols]. Let $\psi(F, n)$ be the number of steps required for the machine to do that, and let $\varphi(n) = \max_F \psi(F, n)$. The question is, how rapidly does $\varphi(n)$ grow for an optimal machine? It is possible to show that $\varphi(n) \geq Kn$. If there really were a machine with $\varphi(n) \sim Kn$ (or even just $\sim Kn^2$) then that would have consequences of the greatest significance. Namely, this would clearly mean that the thinking of a mathematician in the case of yes-or-no questions could be completely¹ replaced by machine, in spite of the unsolvability of the Entscheidungsproblem. n would merely have to be chosen so large that, when the machine does not provide a result, it also does not make any sense to think about the problem. Now it seems to me to be quite within the realm of possibility that $\varphi(n)$ grows so slowly. For 1.) $\varphi(n) \geq Kn$ seems to be the only estimate obtainable by generalizing the proof of the unsolvability of the Entscheidungsproblem; 2.) $\varphi(n) \sim Kn$ (or $\sim Kn^2$) just means that the number of steps when compared to pure trial and error can be reduced from N to $\log N$ (or $\log N^2$). Such significant reductions are definitely involved in the case of other finitist problems, e.g., when computing the quadratic remainder symbol by repeated application of the law of reciprocity. It would be interesting to know what the case would be, e.g., in determining whether a number is prime, and how significantly *in general* for finitist combinatorial problems the number of steps can be reduced when compared to pure trial and error.

To formulate the problems raised by Gödel more clearly let us think of first-order predicate logic formalized in one of the usual Hilbert-style systems with a finite set of axiom schemata and modus ponens and generalization as the only rules of inference. Let the symbol $\vdash_n \varphi$ denote that φ has a first-order proof of $\leq n$ symbols. Gödel was asking about the difficulty of answering questions of the form " $\vdash_n \varphi$ ". Let $A = \{\langle \varphi, 0^n \rangle : \vdash_n \varphi\}$.

¹ Except for the formulation of axioms.

Gödel's question is now: is the set A recognizable on a (multitape) Turing machine in time $O(n)$ or in time $O(n^2)$. It can be shown that there is an effective algorithm for deciding membership in A and that the set A is in fact NP -complete (it is NP -complete even for propositional logic – cf. Buss 1995).

It is interesting that Gödel was thinking in his letter about problems related to $P = ?NP$ well before these classes were widely stated (cf. Hartmanis 1989 for the discussion of Gödel's letter). Note also that Gödel treated linear or quadratic time computability as corresponding to feasible computation and did not realize the importance of polynomial time as a mathematical model of feasible computability.

The problem $P = ?NP$ belongs to the most famous open problems in computer science. On the other hand it is known today that even in the case of simple mathematical theories the decision procedures are of high complexity. The following theorems can serve as examples indicating the measure of the complexity.

Let F be a function defined in the following way:

$$F(n, 1) = 2^n, \quad F(n, m + 1) = 2^{F(n, m)}.$$

Denote by $l(\varphi)$ the length of a formula φ , i.e., the number of (logical and nonlogical) symbols occurring in φ .

It has been shown that:

- (Meyer 1975) Let $f(n)$ be the function

$$F(n, [dn]).$$

There exists a constant $d > 0$ such that for any Turing machine P deciding the weak second-order monadic theory of one successor WS1S there exist infinitely many sentences φ with the property that the machine P needs more than $f(l(\varphi))$ steps to decide whether $\varphi \in \text{WS1S}$ or not.

- (Meyer 1975) The complexity of the theory of linear order is at least $F(n, [dn])$ for a certain positive constant d , i.e., to decide a formula of the length n one needs at least $F(n, [dn])$ steps.
- (Fisher and Rabin 1974) A decision procedure for Presburger arithmetic is of the complexity at least $2^{2^{cn}}$ for a certain constant $c > 0$, i.e., to decide whether a formula φ of the length n is a theorem of Presburger arithmetic one needs at least $2^{2^{cn}}$ steps.
- (Fisher and Rabin 1974) The complexity of the theory $Th(\langle \mathbb{N}, \cdot \rangle)$ is at least $F(cn, 3)$, i.e., $2^{2^{2^{cn}}}$ for a certain constant $c > 0$.

So one sees that even for such simple theories as $Th(\langle \mathbb{N}, + \rangle)$ or $Th(\langle \mathbb{N}, \cdot \rangle)$ decision procedures are of exponential complexity. Add also that the decision procedure for the theory of real numbers is doubly exponential in the number of quantifier blocks (cf. Heintz *et al.* 1989) and that even deciding first-order sentences for the ordered group $\langle \mathbb{R}, +, < \rangle$ is exponentially hard (cf. Fisher *et al.* 1974).

On the other hand if one moves from the level of first-order logic to higher systems (or generally from the level n to the level $n + 1$) the complexity of proofs (their lengths) can be reduced. This observation was made by Gödel in (1936). Having already shown that a logic S_{n+1} of a higher order could prove formulas that a logic S_n of a lower order could not prove, in this abstract he considered the question of formulas that can be proved in both the weaker and the stronger logics. He stated that if the length of a proof is defined to be the number of lines in it, there are formulas that can be proved in both S_n and S_{n+1} but that have a proof in S_{n+1} much shorter than their shortest proof in S_n . This speed-up can be by arbitrary function computable in S_n , i.e., for any function F computable in S_n there exist infinitely many formulas φ such that if k is the length of a shortest proof of φ in S_n and l is the length of a shortest proof of φ in S_{n+1} then $k > F(l)$. Hence “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” (Gödel 1936).

Gödel did not give a proof of his result. An analogous result (taking the length of a proof to be its Gödel number rather than the number of lines in it) was given by Mostowski in (1952). Similar results were also proved by Ehrenfeucht and Mycielski (1971) and by Parikh (1971). Statman in (1978) has shown that there is no function F provably recursive in second-order arithmetic such that whenever a first-order formula φ is derivable in a certain standard system of second-order logic with length $\leq l$ then φ is derivable in a certain standard system of first-order logic with length $\leq F(l)$.

Note that the problem $P = ?NP$ discussed above can be thought of as a speed-up question.

A nice illustration of the phenomenon indicated by Gödel was given by Boolos in (1987). He considered there the following set of axioms (in the language with function symbols F and S and a unary predicate D):

- (1) $\forall n F(n, 1) = S(1)$,
- (2) $\forall x F(1, S(x)) = SS(F(1, x))$,

- (3) $\forall n \forall x F(S(n), S(x)) = F(n, F(S(n), x))$,
 (4) $D(1)$,
 (5) $\forall x [D(x) \longrightarrow D(S(x))]$.

In the intended interpretation of the above formulas the variables range over the natural numbers, 1 denotes the number one and S is the successor function. There is no particular interpretation intended for D . By this interpretation F denotes an Ackermann-style function $f : \mathbf{N} \times \mathbf{N} \rightarrow \mathbf{N}$ defined by the following equations: $f(1, x) = 2x$, $f(n, 1) = 2$ and $f(n + 1, x + 1) = f(n, f(n + 1, x))$. This is a rapidly growing function. In fact one can easily show that $f(1, x) = 2x$, $f(2, x) = 2^x$, $f(3, x) = 2^{2^{\dots^2}}$ (the value of a stack of x 2's), $f(4, 1) = 2$, $f(4, 2) = 4$, $f(4, 3) = 2^{2^{\dots^2}}$ (= 64 K), $f(4, 4) = 2^{2^{\dots^2}}$ (64 K of 2's in all).

In second-order logic one can deduce from the given set of axioms that

$$D(F(SSSS(1), SSSS(1))).$$

The usage of the second-order logic is essential here (one uses, e.g., the comprehension principle). The proof can be found in (Boolos 1987, Appendix). This formula can be also proved in a first-order system but any derivation of it must contain at least $f(4, 4)$ symbols (!) – details of the proof of this (metatheoretical) statement can be found in Appendix to (Boolos 1987).² Hence it is impossible to write down such a proof, no actual or conceivable creature or device could do it. There are simply far too many symbols in any such derivation. It shows that first-order logic is in a certain sense practically incomplete. Boolos says even in (1987, p. 135) that “no standard first-order logical system can be taken to be a satisfactory idealization of the psychological mechanisms or processes, whatever they might be, whereby we recognize (first-order!) logical consequences. “Cognitive scientists” ought to be suspicious of the view that logic as it appears in logic texts adequately represents the whole of the science of valid inference.” We can add that this thesis should be taken into account not only by cognitive scientists but also by specialists in mechanized deduction and automated theorem proving (as well as in the artificial intelligence).

² There is another example of a similar result: H. Friedman has shown (cf. Nerode and Harrington 1984) that a certain “finitization” of a combinatorial theorem due to J. Kruskal concerning embeddings of trees can be proved in Zermelo-Fraenkel set theory with the axiom of choice (ZFC) in a few pages but not in the system of second-order arithmetic called ATR (for Arithmetic Transfinite Recursion) in under $f(3, 1000)$ pages.

6. Final remarks

In the previous sections we have presented the recent period of the history of efforts to find an automated theorem prover. They were stimulated by the appearance of computers which made possible the practical realization of earlier ideas. The emphasis was put on the resolution procedure and the unification algorithm and their modifications because they proved to be the most influential ideas.

As was proved by Turing, Gödel and Church there exists no universal automated theorem prover for the whole mathematics – and even more, there are no such provers for most mathematical theories. Proving theorems in mathematics and logic is too complex a task for total automation because it requires insight, deep thought and much knowledge and experience. Nevertheless the semidecidability of mathematical theories was a sufficient motivation for looking for weaker theorem provers. We have described those efforts in the previous sections.

What does one expect from an automated theorem prover? First of all one obtains a certain unification of reasonings and their automatization. Having that one can shift the burden of proof finding from a mathematician and a logician to the computer. In this way we are also assured that faulty proofs would never occur. Are such automated theorem provers more clever than people? Of course they can proceed quicker than a human being. But can they discover new mathematical results? The answer is YES. Some open questions have been answered in this way within finitely axiomatizable theories. For example S. Winker, L. Wos and E. Lusk (1981) answered positively the following open question: does there exist a finite semigroup which simultaneously admits of a nontrivial antiautomorphism without admitting a nontrivial involution? The progress in more complex theories such as analysis or set theory is slower, but there are also provers being able to prove some nontrivial theorems such as for example Cantor's theorem stating that a set has more subsets than elements (cf. P. Andrews, D. A. Miller, E. L. Cohen, F. Pfenning 1984) and various theorems in introductory analysis. The latter includes limit theorems of calculus such as

- the sum, product and composition of two continuous functions is continuous,
 - differentiable functions are continuous,
 - a uniformly continuous function is continuous,
- as well as theorems of intermediate analysis (on the real numbers) such as
- Bolzano-Weierstrass theorem,

- if the function f is continuous on the compact set S then f is uniformly continuous on S ,
- if f is continuous on the compact set S then $f[S]$ is compact,
- intermediate value theorem

(cf. Bledsoe 1984).

All those achievements can be treated as partial realizations and fulfillments of Leibniz's dreams of *characteristica universalis* and *calculus ratiocinator*. They are still far from what Leibniz did expect but they prove that a certain progress in the mechanization and automatization of reasonings and generally human thought has been made. On the other hand one should be aware of some limitations indicated above.

References

- Anderson, R. and W. W. Bledsoe, 1970, A Linear Format for Resolution with Merging and a New Technique for Establishing Completeness, *Journal of the Association for Computing Machines*, 17, 525–534.
- Andrews, P., 1976, Refutations by Matings. *IEEE Trans. on Computers*, C-25, 801–807.
- Andrews, P., 1981, Transforming Matings into Natural Deduction Proofs. In: *Proc. Fifth Conf. on Automated Deduction*, ed. by W. Bibel and R. Kowalski, *Lecture Notes in Computer Science 87*, Berlin – Heidelberg – New York: Springer-Verlag, pp. 281–292.
- Andrews, P., D. A. Miller, E. L. Cohen and F. Pfenning, 1984, Automating Higher-Order Logic. In: *Automated Theorem Proving. After 25 Years*, ed. by W. W. Bledsoe and D. W. Loveland, *Contemporary Mathematics*, 29, AMS, Providence, Rhode Island, pp. 169–192.
- Bläsius, K., N. Eininger, J. Siekmann, G. Smolka, A. Herold and C. Walther, 1981, The Markgraf Karl refutation procedure. In: *Proc. Seventh Intern. Joint Conf. on Artificial Intelligence*, pp. 511–518.
- Bledsoe, W. W., 1983, Using Examples to Generate Instantiations for Set Variables. In: *Proc. Intern. Joint Conf. on Artificial Intelligence*.
- Bledsoe, W. W., 1984, Some Automatic Proofs in Analysis. In: *Automated Theorem Proving. After 25 Years*, ed. by W. W. Bledsoe and D. W. Loveland, *Contemporary Mathematics*, 29, AMS, Providence, Rhode Island, pp. 89–118.
- The Present State of Mechanized Deduction, and the Present Knowledge...
- Boolos G., 1987, A curious inference. *Journal of Philosophical Logic* 16, 1–12.
- Boyer, R. S., 1971, *A Restriction of Resolution*. Ph. D. Thesis, University of Texas at Austin, Texas.
- Boyer, R. S. and J. S. Moore, 1975, Proving Theorems about LISP Functions. *Journal of the Association for Computing Machines*, 22, 129–144.
- Boyer, R. S. and J. S. Moore, 1979, *A Computational Logic*. New York: Academic Press.
- Boyer, R. S. and J. S. Moore, 1981, A Verification Condition Generator for FORTRAN. In: *The Correctness Problem in Computer Science*, ed. by R. S. Boyer and J. S. Moore, London.
- Buss S.R., 1995, On Gödel's theorem on lengths of proofs II: Lower bounds for recognizing k symbol provability. In: *Feasible Mathematics II*, eds. P. Clote and J. Remmel, Birkhäuser, 57–90.
- Chang, C. L., 1970, The Unit Proof and the Input Proof in Theorem Proving. *Journal of the Association for Computing Machines*, 17, 698–707.
- Chang, C. L. and R. Lee, 1973, *Symbolic Logic and Mechanical Theorem Proving*. New York – San Francisco – London: Academic Press.
- Chinlund, T. J., M. Davis, G. Hineman and D. McIlroy, 1964, *Theorem Proving by Matching*. Bell Laboratories.
- Church, A., 1936, A Note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1, 40–41, 101–102.
- Davis, M., 1963, Eliminating the Irrelevant from Mechanical Proofs. *Proc. Symp. Applied Mathematics*, 25, 15–30.
- Davis, M. and H. Putnam, 1960, A Computing Procedure for Quantification Theory. *Journal of the Association for Computing Machines*, 7, 201–215.
- Ehrenfeucht A. and Mycielski J., 1971, Abbreviating proofs by adding new axioms, *Bulletin of the American Mathematical Society* 77, 366–367.
- Fisher, M. J. and Rabin, M. O., 1974, Super Exponential Complexity of Presburger's Arithmetic. In: Karp, R. (Ed.) *Complexity of Computation, Proc. SIAM-AMS Sympos. Appl. Math.* 7, pp. 27–41.

- Gelernter, H., 1959, Realization of a Geometry Theorem-Proving Machine. In: *Proc. Intern. Conf. on Information Processing*, Paris: UNESCO House. pp. 273–282. Also in: *Computers and Thought*, ed. by Feigenbaum and Feldman. McGraw-Hill.
- Gelernter, H., J. R. Hanson and D. W. Loveland, 1960, Empirical Explorations of the Geometry-Theorem Proving Machine. In: *Proc. Western Joint Computer Conf.*, pp. 143–147. Also in: *Computers and Thought*, ed. by Feigenbaum and Feldman. McGraw-Hill.
- Gilmore, P. C., 1959, A Program for the Production of Proofs for Theorems Derivable within the First Order Predicate Calculus from Axioms. In: *Proc. Intern. Conf. on Information Processing*. Paris: UNESCO House.
- Gilmore, P. C., 1960, A Proof Method for Quantification Theory: its Justification and Realization. *IBM Journal Research and Devel.* 28–35.
- Gödel K., 1936, Über die Länge von Beweisen, *Ergebnisse eines mathematischen Kolloquiums* 7, 23–24. Reprinted with English translation: ‘On the length of proofs’ in: Gödel K., *Collected Works*, vol. I, eds. Feferman S. et al., Oxford University Press, New York, and Clarendon Press, Oxford 1986, 396–399.
- Gödel K., 2003, *Collected Works*, vol. V, eds. Feferman S. et al., Clarendon Press, Oxford.
- Gould, W. E., 1966, A Matching Procedure for ω -Order Logic. *Air Force Cambridge Research Laboratories, Report 66-781-4*.
- Guard, J. R., F. C. Oglesby, J. H. Bennett and L. G. Settle, 1969, Semiautomated mathematics. *Journal of the Association for Computing Machines*, 16, 49–62.
- Hartmanis J., 1989, Gödel, von Neumann and the $P = ? NP$ problem, *Bulletin of the European Association for Theoretical Computer Science (EATCS)* 38, 101–107.
- Heintz, J., M. F. Roy and P. Solerno, 1989, Complexité du principe de Tarski-Seidenberg, *C. R. Acad. Sci. Paris, Sér. Math.* 309, 825–830.
- Hewitt, C., 1971, *Description and theoretical analysis (using schemata) of PLANNER: a language for proving theorems and manipulating models in a robot*, Ph.D. Thesis, MIT.
- Huet, G. D., 1975, A Unification Algorithm for Typed λ -Calculus. *Theoretical Computer Science.* 27–57.
- Jensen, D. C. and T. Pietrzykowski, 1976, Mechanizing ω -Order Type Theory through Unification. *Theoretical Computer Science.* 123–171.

- Knuth, D. E. and P. B. Bendix, 1970, Simple Word Problems in Universal Algebra. In: *Combinatorial Problems in Abstract Algebras*, ed. by Leech, Pergamon – New York, pp. 263–270.
- Kowalski, R., 1975, A Proof Procedure Using Connection Graph. *Journal of the Association for Computing Machines*, 22.
- Kowalski, R. and D. Kuehner, 1971, Linear Resolution with Selection Function. *Artificial Intelligence*, 2, 227–260.
- Kuehner D. G., 1971, A Note on the Relation between Resolution and Maslov’s Inverse Method. In: *Machine Intelligence*, 6, ed. by Meltzer and D. Michie. New York, pp. 73–76.
- Livesey M. and J. Siekmann, 1976, Unification of A+C-terms (bags) and A+C+I-terms (Sets). *Universität Karlsruhe, Interner Bericht Nr. 5/76*, Karlsruhe.
- Loveland, D. W., 1968, Mechanical Theorem Proving by Model Elimination. *Journal of the Association for Computing Machines*, 15, 236–251.
- Loveland D. W., 1969, A Simplified Format for the Model Elimination Procedure. *Journal of the Association for Computing Machines*, 16, 349–363.
- Loveland, D. W., 1970, A Linear Format for Resolution. In: *Proc. IRIA Symp. on Automatic Demonstration*, Versailles, France 1968, LNM 125, Berlin – New York: Springer – Verlag, pp. 147–162.
- Loveland, D. W., 1972, A Unifying View of some Linear Herbrand Procedures. *Journal of the Association for Computing Machines*, 19, 366–384.
- Loveland, D. W., 1978, *Automated Theorem Proving: A Logical Basis*. Amsterdam – New York – Oxford: North-Holland Publ. Comp.
- Loveland, D. W., 1984, Automated Theorem-Proving: A Quarter-Century Review. In: *Automated Theorem Proving. After 25 Years*. Eds. W. W. Bledsoe and D. W. Loveland, *Contemporary Mathematics*, 29, AMS, Providence, Rhode Island, pp. 1–46.
- Luckham, D., 1970, Refinements in Resolution Theory. In: *Proc. IRIA Symp. on Automatic Demonstration*, Versailles, France 1968, LNM 125, Berlin – New York: Springer-Verlag, pp. 163–190.
- Marciszewski W. and R. Murawski, 1995, *Mechanization of Reasoning in a Historical Perspective*, Editions Rodopi, Amsterdam/Atlanta, GA.
- Maslov, S. Ju., 1964, An Inverse Method of Establishing Deducibility in Classical Predicate Calculus. *Dokl. Akad. Nauk SSR.* 17–20.

- Maslov, S. Ju., 1971, Proof-Search Strategies for Methods of the Resolution Type. *Machine Intelligence*, 6, ed. by B. Meltzer and D. Michie, New York, pp. 77–90.
- Meltzer, B., 1966, Theorem-Proving for Computers: Some Results on Resolution and Renaming. *Computer Journal*, 8, 341–343.
- Meyer, A. R., 1975, The Inherent Complexity of Theories of Ordered Sets. In: *Proceedings of the International Congress of Mathematicians, Vancouver 1974*, vol. 2, pp. 477–482.
- Miller, D. A., E. L. Cohen and P. B. Andrews, 1982, A Look at TPS. In: D. W. Loveland (Ed.), *Proc. Sixth Conf. on Automated Deduction*. Lecture Notes in Computer Science 138. Berlin – Heidelberg – New York: Springer-Verlag, pp. 60–69.
- Morris, J. B., 1969, E-resolution: Extension of Resolution to Include the Equality. In: *Proc. Intern. Joint Conf. Artificial Intelligence*. Washington D.C., pp. 287–294.
- Mostowski A., 1952, *Sentences Undecidable in Formalized Arithmetic: An Exposition of the Theory of Kurt Gödel*. North-Holland Publishing Company, Amsterdam.
- Nerode A. and L. A. Harrington, 1984, The work of Harvey Friedman, *Notices of the American Mathematical Society* 31, 563–566.
- Nevins, A. J., 1974, A Human Oriented Logic for Automatic Theorem Proving. *Journal of the Association for Computing Machines*, 21, 606–621.
- Nevins, A. J., 1975, Plane Geometry Theorem Proving Using Forward Chaining. *Artificial Intelligence*, 6, 1–23.
- Nevins, A. J., 1975a, A Relaxation Approach to Splitting in an Automatic Theorem Prover. *Artificial Intelligence*, 6, 25–39.
- Newell, A., J. C. Shaw and H. A. Simon, 1956, Empirical Explorations of the Logic Theory Machine: A Case Study in Heuristics. In: *Proc. Western Joint Computer Conf.*, 15, pp. 218–239. Also in: Feigenbaum and Feldman (Eds.), *Computers and Thought*. McGraw-Hill 1963.
- Norton, L. M., 1966, *ADEPT – a Heuristic Program for Proving Theorems of Group Theory*, Ph. D. Thesis, MIT.
- Parikh R., 1971, Existence and feasibility in arithmetic, *The Journal of Symbolic Logic* 36, 494–508.
- Plotkin, G. D., 1972, Building-in Equational Theories. In: B. Meltzer and D. Michie (Eds.), *Machine Intelligence*, 7, New York, pp. 73–90.
- Prawitz, D., 1960, An Improved Proof Procedure. *Theoria*, 26, 102–139.
- Presburger, M., 1929, Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In: *Sprawozdanie z I Kongresu Matematyków Krajów Słowiańskich – Comptus Rendus, I Congres des Math. des Pays Slaves*. Warszawa, pp. 92–101, 395.
- Reiter, R., 1971, Two Results on Ordering for Resolution with Merging and Linear Format. *Journal of the Association for Computing Machines*, 18, 630–646.
- Robinson, J. A., 1965, A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the Association for Computing Machines*, 12, 23–41.
- Robinson, J. A., 1965a, Automated Deduction with Hyperresolution. *International Journal Comput. Math.*, 1, 227–234.
- Shostak, R. E., 1976, Refutation Graphs. *Artificial Intelligence*, 7.
- Sibert, E. E., 1969, A Machine Oriented Logic Incorporating the Equality Relation. In: B. Meltzer and D. Michie (Eds.), *Machine Intelligence*, 4, New York, pp. 103–134.
- Sickel, S., 1976, Interconnectivity Graphs. *IEEE Trans. on Computers*, C-25.
- Slagle, J. R., 1967, Automated Theorem Proving with Renamable and Semantic Resolution. *Journal of the Association for Computing Machines*, 14, 687–697.
- Statman R., 1978, Bounds for proof-search and speed up in the predicate calculus, *Annals of Mathematical Logic* 15, 225–287.
- Stickel, M.E., 1981, A Complete Unification Algorithm for Associative-Commutative Functions. In: *Proc. Fourth Intern. Joint Conf. on Artificial Intelligence*, Tbilisi, USSR. Also in: *Journal of the Ass. for Computing Machines*, 28, 423–434.
- Stickel, M. E., 1985, Automated Deduction by Theory Resolution. *Journal of Automated Reasoning*, 1, 333–356.
- Turing, A. M., 1936–1937, On Computable Numbers, with an Application to the Entscheidungsproblem. *Proc. of the London Mathematical Society*, Series 2, vol. 42, pp. 230–265.
- Wang, H., 1960, Toward Mechanical Mathematics. *IBM Journal Research and Devel.*, 2–22. Also in: *Logic, Computers and Sets*, Chelsea, New York, 1970.

Roman Murawski

- Wang, H., 1960a, Proving Theorems by Pattern Recognition. Part I. *Commun. Assoc. Comput. Mach.*, 3, 220–234.
- Wang, H., 1961, Proving Theorems by Pattern Recognition. Part II, *Bell System Technical Jour.*, 40, 1–41.
- Winker, S., L. Wos and E. Lusk, 1981, Semigroups, Antiautomorphisms and Involutions: A Computer Solution to an Open Problem, I. *Math. of Computation*, 533–545.
- Wos, L., G. A. Robinson, D. F. Carson and L. Shalla, 1967, The Concept of Demodulation in Theorem Proving. *Journal of the Association for Computing Machines*, 14, 698–709.
- Wos, L., D. F. Carson and G. A. Robinson, 1964, The Unit Preference Strategy in Theorem Proving. *AFIPS Conf. Proc. 26*, Washington D.C., pp. 615–621.
- Wos, L., G. A. Robinson and D. F. Carson, 1965, Efficiency and Completeness of the Set of Support Strategy in Theorem Proving. *Journal of the Association for Computing Machines*, 12, 536–541.
- Wos, L. and G. A. Robinson, 1970, Paramodulation and Set-of-Support. In: *Proc. IRIA Symp. on Automatic Demonstration*, Versailles, France 1968, LNM 125, Berlin – New York: Springer-Verlag, pp. 276–310.
- Yates, R., B. Raphael and T. Hart, 1970, Resolution Graphs. *Artificial Intelligence*, 1, 257–289.

Roman Murawski

Adam Mickiewicz University
Faculty of Mathematics and Comp. Sci.
ul. Umultowska 87
61-614 Poznań, Poland
e-mail: rmur@amu.edu.pl

Roman Matuszewski
University of Białystok

ON COMPUTER-ASSISTED APPROACH TO FORMALIZED REASONING*

1. Automated Reasoning for Mathematics – a Survey

The idea of the automation of reasoning has been around since Euclid. Aristotle, Leibniz, Peano, Boole, Frege, Hilbert, many of the greatest mathematicians and computer scientists of this century including the inventors of the first computers had this dream in mind. For hundreds of years, many great thinkers have envisioned the automation of reasoning and have worked toward the realization of that goal. Most of the historical information on the topic can be found in Martin Davis' survey article [Dav83], in [Shu97] and in [MarMur95].

The reasoning algorithms invented by Euclid, such as the division algorithm, are early examples of some types of reasoning that were mechanized by the ancients. Aristotle worked on a system of mechanical rules that he believed would allow their user to draw any true conclusion from a set of assumptions.

Leibniz is famous for inventing the differential and integral calculus. By using this calculus, a machine or person could complete calculations in a few minutes that took Kepler years to figure. Thus, it might be said that human intelligence is no longer required to solve these problems. Leibniz also developed a *calculus of reasoning*. He wanted disputes in human affairs to be answered by what he called *calculators*, which were people who knew how the algorithm for reasoning worked. The question would be stated in the formal language and then the disputants would simply say:

* The research reported in this paper is a part of the project entitled *Undecidability and Algorithmic Intractability in the Social Sciences* supported by the Polish Ministry of Science, grant no. 2 H01A 030 25.

calculemus. He recognized the three basic elements needed for automatic reasoning:

- a formal language,
- formal rules of inference, and
- knowledge.

Leibniz worked on a formal, unambiguous language along with a complete set of rules for drawing conclusions (inference rules). He hoped that the combination of these would allow for any question to be answered finally and unquestionably. His work in this area was left unfinished.

In the nineteenth century, George Boole developed the propositional calculus (in his book *The Laws of Thought*) that provided a language and set of inference rules in which much ordinary common sense reasoning can be expressed. An advantage to his language was that there was a procedure that would determine whether any sentence in the language was true or false in a finite amount of time. As early as 1869, a machine was constructed that could verify statements in propositional logic. The language of propositional logic is, unfortunately, not expressive enough to answer most interesting questions.

In 1879, Gottlob Frege expanded the language to full first-order logic, which allows much more complex statements to be expressed and verified. In fact, the language of first-order logic can be used to express almost any question. Frege introduced some rules of inference but did not develop them fully.

Peano's work was similar to Frege's. He invented a similar formal language and said:

"I think that the propositions of any science can be expressed by these signs of logic alone, provided we add signs representing the objects of that science."

Russell and Whitehead showed that this was true. They laid a foundation for all of mathematics (and thus physics and other sciences and fields of reasoning) in logic.

Hilbert was deeply involved in end of XIX century in this topic and initiated a program of research that had among its goals the discovery of a systematic procedure that would decide the truth or falsity of any statement in Frege's first-order logic. So his question could be stated: *is there a procedure that can always decide whether a statement is a consequence of some base of knowledge?* Given a base of knowledge (which is called a *theory*) a statement that follows logically from that knowledge is called a *theorem* of the theory. Verifying the truth of a statement using inference

rules is called proving a theorem. Hilbert began a program to find a procedure (algorithm) that could decide whether a sentence in first-order logic was true or false given some assumptions. Many logicians (e.g., Gentzen, Beth, Goedel, Church, Turing, Skolem and Herbrand) began working on this problem.

Gentzen and Beth developed algorithms for proving theorems. Their algorithms have the property that they detect if a sentence is true in a finite amount of time.

In the 1930s, Church and Turing independently discovered, based on the work of Goedel, that there is no procedure that will decide whether any given statement in first-order logic is true or false. There are, however, procedures that will detect if a statement in first-order logic is true. A procedure with this property is called a *complete* procedure or a *semi-decision* procedure. Unfortunately, complete procedures may never halt if the statement in question is not true; and if the statement is true, there is no telling how long the procedure will take to detect that. So if a complete procedure is followed in an attempt to decide some question, after some time passes, we will not know whether we should give up and guess that the statement is false or whether we would find the statement to be true.

But all of these facts **apply equally to people and computers**. The problem of proving theorems can be difficult and there is no easy answer. Still people do it reasonably well all the time and so there seems to be nothing to prohibit computers from doing it. Many of the great mathematicians and logicians of this century continued to pursue this possibility. Goedel, in 1944, talking about the difficulties of solving "mathematical problems systematically" said, *But there is no need to give up hope*. Indeed, in the late 1950s and early 1960s, computer implementations of proof procedures began to be reported. In the last forty years tremendous progress has been made and computers have proved several theorems that have been puzzling mathematicians [MCu97].

Even with all of the work that has been done, the problem of reasoning remains difficult for computers. Hao Wang, a logician who was aware of the theoretical difficulty of the problem of the automation of reasoning, was optimistic. He expected that computers would come into common use as proof assistants for working mathematicians just as they are used today to solve numerical problems. While great success has been attained, there has been much frustration and disappointment. The lion's share of the realm of reasoning has not been touched by the computer. One very basic skill that computers have yet to master is that of using successfully an enormous base of knowledge. This is something that humans do rather well. When given

a large base of knowledge we can expect big progress. And now we have a good beginning – the Mizar Mathematical Library.

2. Mechanized Mathematics

The term *mechanization* in a mathematical context can be viewed in two ways. In a broader sense it is synonymous with *formalization*, the latter meaning the existence of a mechanical (algorithmical) procedure to check validity of a proof. The use of the adjective *mechanical* is usual in contexts dealing with *formalization*. In the narrower sense, *mechanization* denotes a state in which *formalization* is suited for a definite device. This means that there exists such a device and software to operate on it in order to process sentences according to an algorithm involved in the given *formalization*. If the device in question is a computer, then one may speak of *computerization of reasoning*. This concept of *mechanization* is found in the work of Hao Wang *Toward mechanical mathematics* (1970).

The goal of *mechanized mathematics* is to produce computer systems that support and improve the mathematical process. *Mechanized mathematics* systems come in two major types. Computer algebra systems, like Maple or Mathematica, enable the user to perform many kinds of calculations, both symbolic and numeric. The second type of systems are theorem provers and proof checkers, which assist the user in developing mathematical models in the form of axiomatic theories.

Interactive provers there are systems mentioned so far were interactive to some extent. Programs that are designed to cooperate with the user are thereby able to prove **more theorems** than programs that work completely independently of human help. Some programs of this type are called “proof checkers” rather than “theorem provers”. In these cases, the user writes out a proof, with or without all of the details, and the prover makes sure that the proof is correct or finds some correct proof using the information entered by the user. These are sometimes called “interactive provers” or “semi-automatic theorem provers”. Probably most automated reasoning programs lie somewhere between the extremes of completely automatic theorem proving and proof checking.

Automated deduction is the branch of computer science and artificial intelligence which deals with the use of computers to decide the validity of logical sentences. Although this decision problem is undecidable in general, there are several non-trivial theories in which the validity of sentences is decidable. For instance, propositional logic is decidable. Also, first-order

logic is semi-decidable and therefore one can implement algorithms which terminate on valid sentences, though they may not halt on invalid ones. This is usually done by searching for a proof since checking whether a proof derives a particular theorem is decidable.

The development is guided and checked by formally proving that certain conjectures are theorems of the theory being developed. Set theory has been the most popular foundation for mathematics, mathematicians understand it quite well, and all serious mathematics practitioners are familiar with it to some degree. In [Fri97] Harvey Friedman wrote that there is no problem in formalizing mathematics in standard formal systems of axiomatic set theory. Most researchers feel that they know as much as they ever want to know about how one can reduce natural numbers, integers, rationals, reals, and complex numbers to sets, and prove all of their basic properties. However, the formalization of mathematics is inconvenient in any of the current formalisms. There are number of theorem provers and proof checkers based on set theory:

- The Mizar system based on Tarski-Grothendieck set theory [RudTry99],
- Metamath proof verifier based on ZF [Meg],
- M. Gordon’s augmentation of HOL with ZF axioms [GorMel93],
- The EVES program verification system based on ZF [KroPasMei95],
- The Isabelle generic theorem prover with implementation of ZF [Pau93],
- Formalization of von-Neumann-Bernays-Gödel set theory in the Otter resolution theorem prover [Ott86].

The purpose of a proof checking system is to check the correctness of a formal proof, which can be found by a human, machine, or by a combined effort from both. Modern proof checkers are usually called proof development systems, or theorem proving environments, because they can contribute more to the formalization process than to just proof checking. Proof checkers include a number of decision and semi-decision procedures for particular theories to prove certain steps automatically, and a number of proof procedures to automate a sequence of non-trivial inferences. Since proof checking systems are in general not expected to find proofs themselves, the deductive systems they implement are usually not search-oriented. On the other hand, they are expected to formalise a variety of mathematical concepts and therefore they are based on rather rich and expressive foundational systems.

The reliability of the proofs accepted by proof checkers is an important issue. In order to maximize this, some proof checkers are designed so that the correctness of their proofs depends only on a small fragment of their code. This fragment is usually small and simple enough to be well understood

so that the possibility of programming errors is minimized (like in AUTOMATH and Coq).

The complexity of the decidable decision problems mentioned above is however very high. The problem TAUT of deciding the validity of propositional sentences (in conjunctive normal form) is in *co-NP* where one considers a sequence of finite search spaces, one larger than the other, in order to ensure that the validity of a sentence is eventually established.

In order to be efficient, automated deduction systems are based on deductive systems whose proofs can be *easily* found by mechanical means. We can refer to such deductive systems as *search-oriented*, and usually require the following two properties:

- the lengths of proofs in these systems are short,
- complete proof search strategies are not faced with too much non-determinism.

An ideal deductive system which satisfies the above properties does not seem to exist, however a number of systems have been developed in which proofs of non-trivial theorems can be found in a relatively short time. Despite the inherent difficulty of automated deduction, a number of difficult problems in mathematics have been solved by such proof search systems. A recent example is the proof of the Robbins problem which was open for more than fifty years and a successful proof for this problem was found by the EQP theorem prover (MCu97). The machine generated proof was logged and then checked using a proof checker implemented in NuPrl.

To conclude it, in *mechanization of mathematics* the domain of information processing involves numbers, and the domain of data processing involves numerals. In reasoning, the former consists of propositions, the latter of sentences, and operations involved in data the processing are *formal rules of inferences*. The term *formal* is to hint at their dealing merely with formal transformations, i.e. those concerned with the form, or the structure of the strings of characters, that is data, and not with the content. About a reasoning, which proceeds solely according to formal rules, we say that it is *formalized*. Should such a reasoning be carried out by a machine, we call it *mechanized*.

The pioneering usage of formal mathematical language in formalizing large amounts of mathematical knowledge, with proof checking, was the AUTOMATH system [Jut77] – encoded in the 1970s Landau's *Grundlagen der Analysis*, and is the Mizar system – encoded in the end of the 1970s a part of Grzegorzczuk's *Outline of Theoretical Arithmetic* and also in the 1990s large part of *Compendium of Continuous Lattices* [Ban98].

3. Formal Mathematical Language

The idea to have a mathematical, standardized language is an old one. Now with the advances in computer science there is a need of mathematical texts which would be appropriate for processing by computer. The mathematical process consists of formulating mathematical models and then exploring them by stating and proving conjectures and by performing calculations.

A language is formal if its syntax and semantics are unambiguously defined. Similarly we refer to the development of mathematics in an informal, though rigorous, language as informal mathematics. A language for the formalization of mathematics must be rich enough to express mathematical objects, statements about them and valid reasoning involving these statements. Such valid reasoning can be expressed as a number of logical rules manipulating the statements concerning the mathematical objects.

The motivations for formalizing mathematics include the ability to achieve a higher degree of correctness and precision than that found in informal mathematics. The ability to express valid mathematical reasoning by symbolic manipulations implies that the validity of an argument can be checked in a mechanical fashion. This is believed to be more reliable than accepting an informal, but convincing, argument.

A substantial amount of effort was put in using symbolic manipulations to express mathematical reasoning during the nineteenth and twentieth centuries. Boole (in 1848) developed a formal system for propositional logic in which reasoning can be performed through mechanical calculations rather than through the interpretation of the symbolic statements. Frege (in 1879) included quantifiers in the formal logical system he developed which was aimed at expressing the whole of mathematics, and Peano (in 1897) focused on the implementation of mathematics of his period in a formal symbolic form whose notation is closer to informal mathematics than that of Frege. Russell included types in his logic to avoid inconsistencies in Frege's deductive system.

Although the proof checking algorithm of a theorem proving environment can be based on a very simple deductive system, the input language which is used for the formalization, and in particular in the implementation of proofs, can (and usually will) be more expressive. Simple statements in the input language can correspond to the application of several primitive inferences in order to simplify the theorem proving task of the user. Examples of such derived rules include a term rewriting system, procedures for numeric calculations, and a number of decision procedures. Similarly,

constructs for the straightforward definition of recursive types, primitive recursive functions, inductive relations, and other objects, are also provided.

The major difficulty in formalizing mathematics, however, turned out to be its practical unfeasibility, rather than the impossibility of formalizing all mathematical truths. It is believed by most, if not all, mathematicians that one can in theory formalize most of the present day mathematics using a sufficiently strong axiomatisation such as the ZFC set theory. The valid statements which cannot be derived in such a strong system are probably uninteresting statements which would not occur in the mathematical literature. Using a computer system to check and even find formal proofs can however, relieve the practical difficulty of formalized mathematics.

As it has been stated before, most systems are based on the set theory. Besides the logical background also important is the problem of how to build a logical system that would agree with the practice of using suppositions in proving theorems. In 1934 Polish logician Stanisław Jaśkowski introduced *the method of supposition* [Jas34] instead today's term *natural deduction*. In 1934, independently of Gentzen, Jaśkowski initiated the method of natural deduction. However Gentzen's work has won much wider renown. This may be partly due to the fact that Gentzen's system of rules was more conspicuous and elegant with its two symmetric sets of rules, those for the introduction of logical constants, and those for the elimination of constants. Another explanation can be found in the impact of context, namely, in the same paper the presentation of natural deduction was accompanied by the introduction and discussion of the famous *Hauptsatz*, which contributed so much to the development of modern logic, also in its computerized form. The system of natural deduction is very simple, has only about a dozen of intuitively obvious inference rules, which suffices to prove whatever is probable in classical logic. In particular, the basic rules of inference can be used to prove the derived rules of inference and thus justify the most frequently met schemes of reasoning used in informal arguments.

Jaśkowski's goal was to identify the methods used by mathematicians in their proofs, namely *to put those methods under the form of structural rules and to analyze their relation to the theory of deduction*. Investigation shows, that he has achieved his goal – proof authors indeed frequently use the proof structures of his logical system. Then, the theory of Jaśkowski was adapted by Fitch [Fit52], and by Ono [Ono62], and was named the method of subordinate proofs.

The Mizar language investigated in this paper, and its style of proof is based on the Jaśkowski *composite system of logic*. It is typical of Jaśkowski's system that at the right margin of each proof line one makes reference to

both the premises from which this line derives and the rule which justifies the derivation. This method is very natural and convenient. However, if a formalization is to reconstruct natural inferences as truly as possible, the method of referring to rules should somehow be altered. This requires a new kind of formalization in which making use of inference rules would more resemble those linguistic means which occur in the mathematical vernacular.

Another very interesting solution in the *composite system of logic* is, that there are no free variables (also called *real variables*) in predicate logic. In [Mar93] is explained, that this feature is related to the above mentioned method of referencing rules. This method does not consist of putting rule names outside proof lines as metalinguistic comments. Below is a relevant comment of Jaśkowski as far as the concept of variable is concerned:

Symbols of variables which are not apparent variables do not merit the name of variable at all. We deal with such a term as with a given constant, though it is neither a primitive term nor a defined one. It is a constant, the meaning of which, although undefined, remains unaltered through the whole process of reasoning. In practice, we often introduce such undefined constants in the course of a proof. For example, we say: *Consider an arbitrary x* and then we deduce propositions which can be said to belong to the *scope of constancy* of the symbol x . This process of reasoning will be applied in our system. (p. 28)

In [Mar93] W. Marciszewski explained, that this procedure consists of the operation of eliminating a quantifier, and this amounts to replacing a bound variable with a constant. However, instead of referencing a rule in the metalinguistic style, one inserts an expression in the language of the proof itself, viz, the word *consider*. For proofs formalized in this system, Jaśkowski suggests replacing this word with the symbolic constant T (from *Term*) which is analogous to the constant S , for *suppose*, introduced at the level of propositional logic. T was chosen to hint at the fact of substituting a *Term* for a variable. The scope of constancy of the term T is called its *domain*. In this domain the meaning of the term is to be regarded as fixed.

4. Bibliography

- [Ban98] Bancerek G., *Mizar – a work environment for a mathematicians: an experiment with continuous lattices*, Proceedings of the International Congress of Mathematicians ICM98, Birkhauser, Berlin, 1998.
- [Dav83] Davis M., *The Prehistory and Early History of Automated Deduction*, Automation of Reasoning, Vol. 1, Springer Verlag, 1983.

- [Fit52] Fitch F., *Symbolic Logic. An Introduction*, NY: Roland Press, 1952.
- [Fri97] Friedman H., *The Formalization of Mathematics*, <http://www.math.ohio-state.edu/foundations/manuscripts.html>
- [GorMel93] Gordon M., Melham T., *Introduction to HOL: a theorem proving environment for higher order logic*. Cambridge University Press, 1992.
- [Jas34] Jaśkowski St., *On the Rules of Suppositions in Formal Logic*, Series: Studia Logica, Nr. 1, University of Warsaw, 1934.
- [Jut77] van Benthem Jutting L., *Checking Landau's "Grundlagen" in the AUTOMATH System*, PhD thesis, Eindhoven University of Technology, 1977.
- [KroPasMei95] Kromodimoeljo S., Pase B. and Meisels I., *Software Manual for EVES Version 3.0.*, ORA Canada, 1995.
- [Mar93] Marciszewski W., *A Jaśkowski-Style System of Computer-Assisted Reasoning*, Philosophical Logic in Poland, Kluwer, 1993.
- [MarMur95] Marciszewski W. and Murawski R., *Mechanization of Reasoning in a Historical Perspective*, Rodopi, 1995.
- [Mat84] Matuszewski R., *On Automatic Translation of Texts from Mizar-QC language into English*, Studies in Logic, Grammar and Rhetoric, 1984.
- [Mat94] Matuszewski R., *Mizar-MSE – a computer aided course for elementary logic of quantifiers*, Proceedings of International Congress of Mathematicians, ICM 94, Zurich, 1994.
- [Mat95] Matuszewski R., *Mathematical Proof-Checked Journal Automatically Translated into English*, Proceedings of Conference "Language and Technology", European Union, Poznań, 1995.
- [Mat98] Matuszewski R., *Journal of Formalized Mathematics – a collection of mechanically checked articles*, Proceedings of the International Congress of Mathematicians, ICM 98, Birkhauser, Berlin, 1998.
- [Mat99] Matuszewski R., *On Natural Language Presentation of Formal Mathematical Texts*, Studies in Logic, Grammar and Rhetoric, Vol 3(16), 1999.
- [Mat99a] Matuszewski R., *Mizar Texts Automatically Translated into English*, Proceedings of Calculemus Workshop, CADE 99, Trento, 1999 (position paper).
- [Mat99b] Matuszewski R., *A Structural Approach to Human Oriented Presentation of Formal Proofs*, Studies in Logic, Grammar and Rhetoric, Vol 3(16), 1999.
- [Mat99c] Matuszewski R., *Automatic Translation of Machine-Checked Mathematical Texts into English*, Proceedings of the 11th International Congress of Logic, Cracow, 1999.
- [MCu97] McCune W., *Solution of the Robbins Problem*, Vol. 19, no. 3, 1997.
- [Meg] Megill N., *Metamath web site*, <http://www1.shore.net/~ndm/java/mm.html>
- [Ono62] Ono K., *On a Practical Way of Describing Formal Deductions*, Nagoya Mathematical Journal, Vol. 21, 1962.
- [Ott86] Boyer R., Lusk E., McCune W., Overbeek R., Stickel M. and Wos L., *Set theory in first-order logic: Clauses for Gödel's axioms*, J. Automated Reasoning, 2(3): 287–327, 1986.
- [Pau93] Paulson L., *Set Theory for Verification*, Journal of Automated Reasoning, Vol. 11, 1993.
- [RudTry99] Rudnicki P., Trybulec A., *On Equivalentents of Well-foundedness – An Experiment in Mizar*, Journal of Automated Reasoning, Vol 23, Issue 3, 1999.
- [Shu97] Shults B., *Discoveries and Experiments in the Automation of Mathematical Reasoning*, PhD dissertation, University of Texas, Austin, 1997.

Roman Matuszewski
University of Białystok
romat@lasek.waw.pl

Kazimierz Trzęsicki
University of Białystok

FROM THE IDEA OF DECIDABILITY TO THE NUMBER Ω^*

*It is not once nor twice but times without number that
the same ideas make their appearance in the world.*
(Aristotle, *On the Heavens*)

Abstract. Information science is based on the idea of computation. In the 20th century that idea was developed in connection with the problem of decidability. David Hilbert's formulation of the so-called *Entscheidungsproblem*, i.e. classical problem of decidability, produced a plethora of ideas that – in particular – gave rise to information science and is still abundant opening new horizons to philosophy, mathematics and computer science. The paper will discuss methodological and cognitive premises of posing the question of decidability as well as the ideas that have been born since the appearance of the problem of decidability. The last part of the paper will be devoted to speculations on a computer that could cross the barriers of the Turing machine. Nevertheless, even quantum and biological computers – if possible – would not be able to cross the barriers of the most random number Ω .

1. Introduction

Practical and theoretical limitations of computers pose a question of a possibility of overcoming them. Due to the progress in technology, intractability¹ could be lowering but new tasks and tendencies to make compu-

* The work has been supported by Grant 2 H01A 030 25: Nierozstrzygalność i algorytmiczna niedostępność w naukach społecznych (Undecidability and Intractability in Social Sciences).

¹ *Webster's Dictionary* defines:

Tractable

- (a) easily managed or controlled; docile; yielding.
- (b) easily worked, shaped, or otherwise handled; malleable.

Intractable

- (a) not docile or manageable; stubborn
- (b) hard to shape or work with: an intractable metal.
- (c) hard to treat, relieve, or cure.

Tractable derives from Latin *tractabilis*, from *tractare*, to handle, to manage, frequentative of *traho*, to draw, to drag. **Intractable** is from Latin *intractabilis*, from *in-*, “not”

ters “more friendly” result in even more complex programs, consequently leading to its increasing. The progress in the performance of computers is lower than the growth of time and hardware resources need to execute ever more complicated tasks.

The variety of approaches to high performance computing may be divided into three main directions, cf. (Burgin 1999):

1. hardware-oriented (aimed mostly at the development of computational elements and their networks),²
2. software-oriented (focused on the advancement of computational methods and procedures),
3. pointed at the evolution of computed structures (such as data and knowledge).

There is a connection between these approaches, in particular, advances in software and structures depend on the hardware solutions.

The main topic of the author’s considerations will be the second approach. Universal computation is one of the foundation concepts in computer science. Any computation that can be carried out by one general-purpose computer can also be carried out on any other general-purpose computer. In order to avoid having to refer to different computers, it is defined as a model of computation that can simulate all computations. The most popular model of a universal computer is the Turing machine, cf. (Akl 2005).

The idea of computer is based on the concept of calculus, which has a very long history with its origins in antiquity. At the beginning of the 20th century the problem of decidability posed by Hilbert gave rise to the

+ *tractabilis*, “manageable”, from *trahere*, “to draw (along), to drag, to pull”. A programming task is considered as tractable if it can be accomplished in a reasonable period of time or with a reasonable supply of physical resources (usually space). Otherwise it is intractable. The study of tractability has a theoretical and a practical aspect, yielding theoretical and practical definitions of terms.

² In 1965 Gordon Moore, a co-founder of Intel and co-inventor of the integrated circuit, observed that the number of transistors per square inch on integrated circuits would double every year since the integrated circuit was invented (monthly “Electronics”, 19 April 1965). The current definition of Moore’s Law predicts that the data’s density doubles every 18 months. Most experts expect Moore’s Law to hold for at least another two decades. Moore’s Law is the key-defining trend of the technology age. In order to understand the future in the exponential world, Kurzweil in his famous book *The Age of Spiritual Machines: When Computers Exceed Human Intelligence* (1992) tells the story of the price that was promised to the inventor of the chess game by the emperor of China. The inventor asked the emperor to place a grain of rice on the first square of a chess board, and double the number of grains thereafter on each of 64 subsequent squares. To fill the entire board with rice it requires 18,446,744,073,709,551,615 grains. If a grain was counted out every second, it would take 584 billion years to count them all. The age of the planet Earth is only 4.5 billion years. The story goes that either the emperor lost his kingdom, or the inventor lost his head. In any way, the consequences were severe.

question of definition of calculus. The Turing-Church thesis (conjecture) claims that an intuitive notion of calculus is adequate to the notion of the Turing machine (or equivalent notions such as: recursive functions, the lambda-calculus by A. Church, the canonical systems by E. Post, the normal algorithms by A. A. Markov, the Minsky machines, the random access machines (RAM), the Kolmogorov algorithms, etc.). The Turing machine is a theoretical device that computes all the functions that are computable in any reasonable sense. Conversely, it is also believed that if a computation cannot be performed by the Turing machine, then it cannot be computed at all. In other words, the thesis states that all effective computational models are equivalent to, or weaker than, the Turing machine, cf. (Shoenfield 1991, p. 26). It has long been assumed that the Turing machine is able to do all that computers equipped with recursive algorithms do and all that can be done by the Turing machine may be executed by such computers (if they have enough time³). Thus the Turing-Church thesis states that any computation solvable by a precisely stated set of instruction (an algorithm) can be run on the Turing machine or a digital process computer. In recent years the number of people who maintain that the Turing machine cannot capture the entire spectrum of applications of computers has been growing. There are important constraints on the ability of the Turing machine. Moreover, some of these limitations do not concern physical restrictions. There are well-stated problems that are not computable by the Turing machine. Generally speaking, there are problems related to tractability, commensurability and computability. Thus the question arises if there are possible devices which can compute more than the Turing machine.

The attempts both to conceive a new notion of calculus and to abolish the Turing-Church thesis are still ineffectual. It would be a mistake to dismiss such theorizing as idle speculation. Moreover, the study of more powerful models of computation is of considerable importance, with many far-reaching implications in computer science, mathematics, physics and philosophy. The inability to develop ways in which we could build machines with more computational power than the Turing machine is not good evidence that this is impossible. Perhaps the path oriented on enhancing and enriching the already existing methods and procedures is deceptive. Maybe

³ The notion of the “speed of computation” makes little sense in the classical understanding of mathematics. Its importance has been recognized with the advent of modern computers and their applications, especially such that need to be accomplished in real time.

only inventing of extremely original computational structures will perform a real breakthrough. It can also be the case that the task exceeds the ability of our intuition. Perhaps we are not able to accomplish it only in our inner world. Thus we have to look for other ideas searching for them in the external world. What we require is a radically new technique and/or even new developments in physics and biology.

Computers are used to forecast natural phenomena. For Plato, the world has a soul and God speaks through mathematics. Leibniz's saying: *Cum Deus calculat et cogitationem exercet, fit mundus* (when God thinks things through and calculates, the world is made)⁴ could be, in this context, assumed to mean: computers' usage is effective because the nature calculates (Galileo's principle of natural computation), too. Today, we may omit "Deus" from this precept: via calculation we create everything from the neutron bomb to computers or to the human genome map project. There is a parallelism between a computer and the nature. If so, the question of calculus is no longer purely theoretical. We may ask about natural computers, i.e. about those phenomena that "calculate" (in nature). The nature has inspired Turing. The deliberation on brain and its way of operation provokes many ideas in computer science. According to Fredkin⁵ and Zuse,⁶ the universe is a kind of computational device – it is a cellular automaton.⁷ Per-

⁴ This sentence is on the margin of *Dialogus*, (1846–1863, pp. 190–193).

⁵ Edward Fredkin (1934–), one of the first modern computer programmers and hackers. Despite a quite non-traditional style of thinking and even without being a graduate student, in the sixties of the XX century he was invited to become a professor of MIT. For more about the life and ideas of Fredkin see (Wright 1988a, Wright 1988b). See also Fredkin's website <http://www.digitalphilosophy.org/>

⁶ Konrad Zuse (1910–1996), a German civil engineer and painter. He invented the first programmable computer (Z1) and the first high-level programming language. In 1936 Zuse applied for a patent, which he did not get. In spite of it, Zuse decided to build the machines.

⁷ Fredkin is its originator. Zuse (1967, 1969) arrived at that idea independently of Fredkin. "A cellular automaton is a collection of 'colored' cells on a grid of specified shape that evolves through a number of discrete time steps according to a set of rules based on the states of neighboring cells. The rules are then applied iteratively for as many time steps as desired. Von Neumann was one of the first people to consider such a model, and incorporated a cellular model into his 'universal constructor.' Cellular automata were studied in the early 1950s as a possible model for biological systems (Wolfram 2002, p. 48). Comprehensive studies of the cellular automata have been performed by S. Wolfram starting in the 1980s, and Wolfram's fundamental research in the field culminated in the publication of his book *A New Kind of Science* (2002) in which Wolfram presents a gigantic collection of results concerning automata, among which are a number of groundbreaking new discoveries." See (Weisstein 2005). Pythagoras, an ancient philosopher, maintained that "the whole thing is a number" and that "everything can be calculated". In the information era this thought has been repeated by Stephen Wolfram: the universe is an enormous computer.

haps we have to think about computers starting from the idea of "universe as a computer".⁸ We may ask then if the nature-computer is more powerful than the Turing machine. It is rather an empirical than a theoretical problem. From a philosophical angle, there are good reasons to hope that the answer is positive and – if so – it is not excluded that the future computer programs will not be constrained by the Turing machine.

In the theory of computation,⁹ the Church-Turing thesis has a position comparable to that of the fifth of Euclid's axioms in the 18th and 19th centuries. When in 1817 Carl Friedrich Gauss, the prince of mathematicians, began to study the consequences of dropping of the fifth axiom, he did not publish his results fearing the reactions of the mathematical community. Non-Euclidean geometries of János Bolyai and Nikolas Lobachevski were considered to be in obvious breach of the real-world geometry that the mathematics sought to model and thus an unnecessary and fanciful exercise. The new ideas and results were applied in Einstein's general theory of relativity according to which the real geometry of our universe was non-Euclidean. Nowadays the idea of the non-Turing or super-Turing or hyper-Turing (a restricted form of super-Turing) machine seems to be a mere object of speculation. Nevertheless, there are some good reasons to believe in its applicability in quantum physics and molecular biology. If hypercomputation¹⁰ is possible, we may ask whether there are any limitations of (hyper)computability. The answer is positive. If the universe has a random structure, then no matter how efficient the future computer would be, there would remain some non-(hyper)computable numbers.

⁸ The idea of universe as a computer has many supporters. For them "the universe indeed may be some kind of universal computational device, or, to say the least, there may be some advantage to look at the Universe as if it is a computer". See eg. (Petrov August 30, 2003, Petrov n.d.a, Petrov n.d.b).

⁹ The original name was *recursion theory*, since the mathematical concept claimed to cover exactly the computable functions called *recursive function*. That name was changed into *computability theory* during the last years. In many titles the term "recursion theory" still occurs.

¹⁰ Hypercomputation sometimes is defined as a method of computation of non-computable functions. Apparently, there is a *contradictio in adiecto*. Hypercomputation was first introduced by Turing (1939), which investigated systems with an oracle to compute a single arbitrary (non-recursive) function from naturals to naturals. Here we are not interested in a device that stops for an input for that the Turing machine does not, as e.g., the Turing machine in which an oracle is available. What we aim to discuss is the notion of computability and the possibility of abolishing the Church-Turing Thesis.

2. Hilbert's concept of mathematical knowledge

While searching for the origins of the problem of decidability, it is necessary to take into account the address *Mathematische Probleme* (1900) that David Hilbert (1862–1943)¹¹ delivered at the second International Congress of Mathematicians in Paris, 8th August 1900.¹² The questions of the lecture were consulted with Hilbert's friends and most distinguished mathematicians, Hermann Minkowski (1864–1909)¹³ and Adolf Hurwitz (1859–1919).¹⁴

¹¹ By 1900 Hilbert had emerged as a leading mathematician. He solved the major problems of invariant theory. He was famous for his great *Zahlbericht* (1897) that was written for the request of Deutsche Mathematiker-Vereinigung (the German Mathematical Society). In Hurwitz's opinion the implications of *The Foundations of Geometry* (1899) reached far beyond its immediate field. In a letter to Hilbert he wrote: "You have opened up an immeasurable field of mathematical investigation which can be called the "mathematics of axioms" and which goes far beyond the domain of geometry." Cf. (Gray 2000b).

For Otto Blumenthal, Hilbert's biographer and his first student, Hilbert was a man of problems (Hilbert 1932–1935, vol. 3, p. 405): "In the analysis of mathematical talent one has to differentiate between the ability to create new concepts that generate new types of thought structures and the gift for sensing deeper connections and underlying unity. In Hilbert's case, his greatness lies in an immensely powerful insight that penetrates into the depths of a question. All of his works contain examples from far-flung fields in which only he was able to discern an interrelatedness and connection with the problem at hand. From these, the synthesis, his work of art, was ultimately created. Insofar as the creation of new ideas is concerned, I would place Minkowski higher, and of the classical great ones, Gauss, Galois, and Riemann. But when it comes to penetrating insight, only a few of the very greatest were the equal of Hilbert." For Hermann Weyl (1944, p. 612) "No mathematician of equal stature has risen from our generation."

¹² At the Congress because of time constraints Hilbert presented only ten of the twenty-three problems. For more details and about the twenty fourth problem (it asks for the simplest proof of any theorem), see (Thiele 2003). Thiele (2003, p. 1) writes: "for a century now, the twenty-fourth problem has been a Sleeping Beauty." Let us remark that Polish logicians obtained many results concerning economy and simplicity in expressing (e.g., a language with only one functor – Leśniewski, Łukasiewicz) and formulation of formal theory (e.g., with only one axiom – Łukasiewicz).

¹³ About his friendship with two-year-younger Minkowski, Hilbert wrote: "Seit meiner Studienzeit war mir Minkowski der beste und zuverlässigste Freund, der an mir hing mit der ganzen ihm eigenen Tiefe und Treue. Unsere Wissenschaft, die uns das liebste war, hatte uns zusammengeführt; sie erschien uns wie ein blühender Garten. [...] Gern suchten wir dort auch verborgene Pfade auf und entdeckten manche neue, uns schön dünkende Aussicht, und wenn der eine dem anderen sie zeigte und wir sie gemeinsam bewunderten, war unsere Freude vollkommen. [...] Er war mir ein Geschenk des Himmels, wie es nur selten jemand zuteil wird, und ich muss dankbar sein, dass ich es so lange besaß. [...] Jäh hat ihn der Tod von unserer Seite gerissen. Was uns aber der Tod nicht nehmen kann, das ist sein edles Bild in unserem Herzen und das Bewusstsein, dass sein Geist in uns fortwirkt." About Minkowski himself he wrote, (1971): "Er war mir ein Geschenk des Himmels, wie es nur selten jemand zuteil wird, und ich muß dankbar sein, daß ich es so lange besaß. Jeder, der ihm näher stand, empfand die Harmonie seiner Persönlichkeit und den Zauber seiner Genialität; sein Wesen war wie der Klang einer Glocke, so hell in dem Glück bei der Arbeit und der Heiterkeit seines Gemütes, so voll in der Beständigkeit und Zuverlässigkeit, so rein in seinem idealen Streben und seiner Lebensauffassung. Wie er gelebt hat, so starb er – als Philosoph."

¹⁴ Hurwitz was three years older than Hilbert. Hilbert remembered him: "Damals noch

Weyl (1944, p. 613) wrote about the role that friendship had played in Hilbert's intellectual development:

More decisive than any other influence for the young Hilbert at Königsberg was his friendship with Adolf Hurwitz and Minkowski. He got his thorough mathematical training less from lectures, teachers or books, than from conversation.

In a letter of 5th January 1900 Minkowski (1973)¹⁵ advised Hilbert to seize the moment, writing:

Most alluring would be the attempt to look into the future and compile a list of problems on which mathematicians should test themselves during the coming century. With such a subject you could have people talking about your lecture decades later.

Mathematical research has been influenced by that address, which continues to have a tremendous importance for all mathematicians.¹⁶ Already by September 1900, George Bruce Halsted had written that Hilbert's beautiful paper on the problems of mathematics "is epoch-making for the history of mathematics" (Halsted 1900, p. 188). A substantial part of Hilbert's fame rests on it. Hilbert spoke on the problems of mathematics to such effect that it has been remarked by Hermann Weyl (1951, p. 525):

We mathematicians have often measured our progress by checking which of Hilbert's questions had been settled in the meantime.

Hilbert and his twenty-three problems have become proverbial. Throughout the 20th century the solution to the problem was the occasion for praise and celebration. The Hilbert problems have made their impact, just as Minkowski predicted. It would take more than a single book to describe the results

Student, wurde ich bald von Hurwitz zu wissenschaftlichem Verkehr herangezogen und hatte das Glück, in der mühelosesten und interessantesten Art die geometrische Schule von Klein und die algebraisch-analytische Berliner Schule kennenzulernen. Dieser Verkehr wurde um so anregender, als auch der geniale Hermann Minkowski zu unserem Freundschaftsbund hinzutrat: Auf zahllosen, zeitweise Tag für Tag unternommenen Spaziergängen haben wir damals während acht Jahren wohl alle Winkel mathematischen Wissens durchstößert, und Hurwitz mit seinen ebenso ausgedehnten und vielseitigen wie festbegründeten und wohlgeordneten Kenntnissen war uns dabei immer der Führer."

¹⁵ See also (Reid 1970, p. 69).

¹⁶ During the Millennium Meeting in Paris in May 2000, the Clay Mathematics Institute of Cambridge, Massachusetts, identified seven Millennium Prize Problems, for each of which it has put up a one million dollar prize for a solution. It has been declared that the problems "are not intended to shape the direction of mathematics in the next century" (The Clay Mathematics Institute http://www.claymath.org/prize_problems/html).

produced in this mushrooming field in the 20th century, cf. (Gray 2000b). Hilbert's impact on the world of mathematicians is summed up by Weyl (1944, p. 132):

I seem to hear in them from afar the sweet flute of the Pied Piper that Hilbert was, seducing so many rats to follow him into the deep river of mathematics.

According to Hilbert,

... jedes Zeitalter eigene Probleme hat, die das kommende Zeitalter löst oder als unfruchtbar zur Seite schiebt und durch neue Probleme ersetzt.

... every age has its own problems, which the following age either solves or casts aside as profitless and replaces by new ones.¹⁷

In the introduction to the Paris lecture the future problems that are expected for the new generation of mathematicians are characterized first of all on the ground of methodological and epistemological premises.

Science needs unsolved problems. Due to them it is alive and develops. Hilbert says:

Solange ein Wissenszweig Überfluß an Problemen bietet, ist er lebenskräftig; Mangel an Problemen bedeutet Absterben oder Aufhören der selbstständigen Entwicklung.

As long as a branch of science offers an abundance of problems, so long it is alive; a lack of problems foreshadows extinction or the cessation of independent development.¹⁸

A good mathematical problem has to be clear and understandable. What it means was explained by Hilbert as follows:

Ein alter französischer Mathematiker hat gesagt: Eine mathematische Theorie ist nicht eher als vollkommen anzusehen, als bis du sie so klar gemacht hast,

¹⁷ All fragments quoted here are from the lecture translated into English by David Joyce, Clark University.

¹⁸ In 1947 that Hilbert's thought was repeated by André Weil. He said: "Great problems furnish the daily bread on which the mathematician thrives." And turning to Hilbert's famous list of problems he singled out the 5th problem on Lie groups, the Riemann hypothesis, and the problem of generalizing the theorems of Kronecker's Jugendtraum which "still escapes us, in spite of the conjectures of Hilbert himself and the efforts of his pupils". Bourbaki, the group of French mathematicians (André Weil, Jean Dieudonné), shared Hilbert's view of mathematics. Till now the supply of problems in mathematics is inexhaustible, and as soon as one problem is solved numerous others come forth in its place. Contemporary mathematicians produce approximately two hundred thousand theorems a year (multiplying the number of journals by the number of yearly issues by the number of papers per issue by the average number of theorems per paper).

daß du sie dem ersten Manne erklären könntest, den du auf der Straße triffst. Diese Klarheit und leichte Faßlichkeit, wie sie hier so drastisch für eine mathematische Theorie verlangt wird, möchte ich viel mehr von einem mathematischen Problem fordern, wenn dasselbe vollkommen sein soll; denn das Klare und leicht Faßliche zieht uns an, das Verwickelte schreckt uns ab.

An old French mathematician said: A mathematical theory is not to be considered complete until you have made it so clear that you can explain it to the first man whom you meet on the street. This clearness and ease of comprehension, here insisted on for a mathematical theory, I should still more demand for a mathematical problem if it is to be perfect; for what is clear and easily comprehended attracts, the complicated repels us.

The "clearness and ease of comprehension" is the first feature of a good mathematical problem. The second feature of it – that also attracts – is its difficulty. Hilbert wrote:

Ein mathematisches Problem sei ferner schwierig, damit es uns reizt, und dennoch nicht völlig unzugänglich, damit es unserer Anstrengung nicht spotte; es sei uns ein Wahrzeichen auf den verschlungenen Pfaden zu verborgenen Wahrheiten – uns hernach lohnend mit der Freude über die gelungene Lösung. Moreover a mathematical problem should be difficult in order to entice us, yet not completely inaccessible, lest it mock at our efforts. It should be to us a guide post on the mazy paths to hidden truths, and ultimately a reminder of our pleasure in the successful solution.

Where is the source of mathematical problems? It is the next question posed by Hilbert. First of all, mathematical problems may have been found in the physical world and empirical experience. They are in mechanics, astronomy and physics. By the end of the address the question of unity of mathematics was discussed. According to Hilbert:

Der einheitliche Charakter der Mathematik liegt im inneren Wesen dieser Wissenschaft begründet; denn die Mathematik ist die Grundlage alles exacten naturwissenschaftlichen Erkennens. Damit sie diese hohe Bestimmung vollkommen erfülle, mögen ihr im neuen Jahrhundert geniale Meister erstehen und zahlreiche in edlem Eifer erglühende Jünger!

The organic unity of mathematics is inherent in the nature of this science, for mathematics is the foundation of all exact knowledge of natural phenomena. That it may completely fulfil this high mission, may the new century bring it gifted masters and many zealous and enthusiastic disciples!

The source of problems is also in human mind. In some cases, even without any influence of external world, mathematical problems result from logical operations such as comparison, generalization, specification, analysis and

grouping of notions. There is a continuous interchange between thinking and experience.

According to Hilbert, to solve a mathematical problem, methods of deduction and strictness should be used. A solution for a problem has to be based on a finite number of premises and a finite number of inferences. The premises should be precisely formulated. This is the requirement of strictness of proving, which corresponds to general philosophical expectations of human mind. Only in such a way the fullness of meaning of a problem and its fruitfulness are revealed. Mathematics and any other science needs inventing sharper tools and simpler methods.

The strictness of proof is not an enemy of its simplicity. On the contrary, many examples show that problems which demand strictness of proof can be solved unexpectedly easily or without much effort. Such a case may be true about all sciences. Due to sharper tools and simpler methods a unity of mathematics would be preserved.

Symbols have to associate with their objects of denotation. Arithmetical symbols are geometrical figures and geometrical figures are drawn formulas. Geometrical representation is indispensable for mathematicians. Hilbert shares Leibniz's view,

Drawing is a very useful tool against the uncertainty of words.

A similar notion of intuition and geometrical knowledge could be found in Kant, to whom Hilbert appealed explicitly. For Kant the intuition directly produces knowledge of a general geometric theorem: a requisite geometrical proof is one diagram, and constructing the diagram in intuition provides knowledge of the theorem, cf. (Zach 2001, p. 153).

An application of geometrical figures as means of strictly proving assumes the exact knowledge and complete mastery of axioms that are fundamental to the theory of these figures. For these reasons their intuitively perceived content has to be strictly axiomatized.¹⁹ For Hilbert (1922a):

Diese Zahlzeichen, die Zahlen sind und die Zahlen vollständig ausmachen, sind selbst Gegenstand unserer Betrachtung, haben aber sonst keinerlei Bedeutung.

¹⁹ Axiomatization and its importance is the topic of *Axiomatisches Denken*. Hilbert believes that any possible subject of scientific knowledge, if it is mature for theoretical study, is subjected to axiomatic method and indirectly to mathematics. The unity of science is based on a leading role of mathematics, (1918, p. 415). For more see (Hilbert n.d., p. 93), cf (Thiele 2003, p. 19):

Axiomatics is the rhythm that makes music of the method, the magic wand that directs all individual efforts to a common goal.

These digits (signs of numbers), that the numbers are and the numbers completely express, themselves are a subject of our perception, do not have any meaning.

By the end of December 1899, in a letter to Frege, Hilbert wrote (Frege 1976, p. 66):

Wenn sich die willkürlich gesetzten Axiome nicht einander widersprechen mit sämtlichen Folgen, so sind sie wahr, so existieren die durch die Axiome definierten Dinge. Das ist für mich das Criterium der Wahrheit und Existenz. When arbitrary chosen axioms do not contradict to the totality of consequences, so they are true, so the things defined by the axioms exist. It is for me the criterion of truth and existence.²⁰

The axiomatic method – an area of interest and a research program, which Hilbert pursued till the end of his activity – began with his work on axiomatic geometry and the publication of *Grundlagen der Geometrie*.²¹ Axioms provide implicit definitions of non-logical terms.²² In a famous remark:

Man muß jederzeit an Stelle von 'Punkte, Geraden, Ebenen' 'Tische, Stühle, Bierseidel' sagen können. It must be possible to replace the words 'point, line, plane' with 'table, chair, beer mug'.

already in 1891 the main idea of *Grundlagen der Geometrie* was expressed.²³ It was in contrast with Euclid's view of axioms. Euclid conceived

²⁰ Later a quite similar thought was expressed by Henri Poincaré (1914, p. 137):

In der Mathematik kann das Wort: 'existieren' nur einen Sinn haben: es bedeutet 'widerspruchsfrei sein'.

Frege asks Hilbert to consider the following example: Suppose we know that the propositions

1. *A* is an intelligent being
2. *A* is omnipresent
3. *A* is omnipotent

together with their consequences did not contradict one another; could we infer from this that there was an omnipotent, omnipresent, intelligent being? (Frege to Hilbert, 6/1/1900 (Frege 1980b)).

²¹ Hilbert's work in geometry has the greatest influence in that area after Euclid. First published in 1899 *Grundlagen der Geometrie* has appeared in many new editions. Its impact on the axiomatic approach to mathematics throughout the 20th century is hard to overestimate.

²² Logical terms are topic-neutral. Learning a system of logic involves learning how to use these terms correctly.

²³ See (Blumenthal 1935, p. 402f), (Toepell 1986, p. 42).

axioms as propositions that embody intuitive truth.²⁴ Later, in 1921, in a paper of a programmatic character “The new grounding of mathematics” Hilbert (1922*b*) wrote:

If logical inference is to be certain, then these objects [certain extra-logical discrete objects, which exist intuitively as immediate experience before all thought] must be capable of being completely surveyed in all their parts, and their presentation, their difference, their succession (like the objects themselves) must exist for us immediately, intuitively, as something that cannot be reduced to something else. [...] The solid philosophical attitude that I think is required for the grounding of pure mathematics – as well as for all scientific thought, understanding, and communication – is this: In the beginning was the sign.²⁵

The idea that (in mathematics) there is nothing that is not given explicitly even logic which should be reduced to operations on symbols (things)²⁶ is significant for the development of information science and especially for artificial intelligence (AI). The process of computation by a computer is an abstract principle and it is independent of a physical form or mechanism used (i.e., symbol manipulation is as accurately accomplished with tables, chairs, beer mugs as with CPUs and memories). Computers process signals, thus for computers mathematics is a game played according to certain simple rules. Formal theories are physical objects of a specific kind: they all can be implemented as programs of digital computers. The concept of mechanization (in Turing’s sense) is equivalent as to its scope with that of formalization (in Hilbert’s sense). Mechanization and formalization differ

in their pragmatic roles. The Turing machine is a mathematical model of a mathematician who acts according to Hilbert’s formalistic programme, cf. (Marciszewski 2003, p. 79). Computers are “engines” that by means of a program generate theorems without involving human skills, intuitions, etc. Turing maintained that (Hodges 1992, p. 361):

... if a machine is expected to be infallible, it cannot also be intelligent.

Hence, as it is said in a contemporary handbook of mathematics:

The ultimate goal of mathematics is to eliminate all need for intelligent thought, (Graham, Knuth & Patashnik 1989, p. 56).

By the end of the introductory part of his lecture, Hilbert expressed a very characteristic view that (according to his opinion that view was shared by all mathematicians):

daß ein jedes bestimmte mathematische Problem einer strengen Erledigung notwendig fähig sein müsse ...
that every definite mathematical problem must necessarily be susceptible of an exact settlement ...²⁷

The belief in the solvability of every problem is not a peculiarity of mathematics. It is a general law inherent in the nature of the mind. The similar thought could be found in Wittgenstein’s *Tractatus Logico-Philosophicus*, 6.5:

For an answer which cannot be expressed the question too cannot be expressed.
The *riddle* does not exist.
If a question can be put at all, then it *can* also be answered.

To solve a problem, we have to know it. Thus, for us there is a difference between the conviction of solvability of every problem and the conviction that we are able to achieve complete knowledge.²⁸ This difference is not clear when Hilbert writes:

²⁷ Let us remark that there is a difference between the thesis that all mathematical problems are solvable and the thesis of decidability. It may be assumed that this difference was not clear for Hilbert. The belief that every mathematical question is solvable in principle has been called the Hilbert axiom. See (Thiele 2003).

²⁸ The difference is clearer if we consider decidability and omniscience. For an omniscient being, an undecidable problem remains undecidable. We can also imagine (it is not self-contradictory) that everything is decidable but there are truths that we are not able to know. If ‘to know’ means ‘to be axiomatizable’, then, according to Gödel (1906–1978), our knowledge is convicted to be incomplete. Some of true propositions (e.g., Ramsey theorem) that are independent of the Peano Arithmetics may be proved in ZFC. But what

²⁴ Frege argued with Hilbert. In a letter to him, Frege wrote (Frege to Hilbert, 6/1/1900, (Frege 1980*b*)):

Given your definitions, I do not know how to decide the question whether my pocket watch is a point.

In debate with Hilbert, Frege held that axioms express determinate truths, fundamental facts of intuition. See (Kambartel 1975, Resnik 1974, Resnik 1980). If so, then they must be facts which follow from concepts which we already have. The logical concepts are the only ultimate intuitive concepts. Thus mathematical concepts should be reduced to purely logical ones. See Frege (1884).

²⁵ See (Hilbert 1922*b*, p. 202); repeated almost verbatim (Hilbert 1925), (Heijenoort 1967, p. 376). This is the text of a talk given in Hamburg, July 25–27, 1921. Cf. (Zach 2001, pp. 8, 115–116). Discussing with Hilbert, Oskar Becker (1927) maintained that signs that did not have meaning were not signs.

²⁶ Hilbert’s *Grundlagen der Geometrie* (1899) is the first work where (at least implicitly) logic in a model theoretic conception is involved. Frege for whom logic was a language (not as for Hilbert a calculus) criticized Hilbert from that universalistic position. Hilbert partly took it into account in the second edition of *Grundlagen der Geometrie* in 1903. Cf. (Müller 2001, p. 19).

Diese Überzeugung von der Lösbarkeit eines jeden mathematischen Problems ist uns ein kräftiger Ansporn während der Arbeit; wir hören in uns den steten Zuruf: *Da ist das Problem, suche die Lösung. Du kannst sie durch reines Denken finden; denn in der Mathematik giebt es, kein Ignorabimus!*

This conviction of the solvability of every mathematical problem is a powerful incentive to the worker. We hear within us the perpetual call: *There is the problem. Seek its solution. You can find it by pure reason, for in mathematics there is no ignorabimus.*

The “ignorabimus” of the Paris address referred to famous agnostic catchwords “Ignoramus et ignorabimus” [We are ignorant and we shall (always) be ignorant]²⁹ of Emil du Bois-Reymond’s fashionable academic pessimism. Hilbert insisted that in mathematics:

Wir müssen wissen. Wir werden wissen.
We can know, and we shall know.³⁰

Hilbert even believed:

That there is no *ignorabimus* in mathematics can probably be proved by my theory of logical arithmetic. See (Thiele 2003).

This epistemological optimism was not restricted to mathematics. Hilbert (1932–1935, p. 292) in his Paris report speaks about “pre-established

about propositions independent of ZFC? The continuum hypothesis is (for Gödel and other platonists) true or false, but – as it has been proved by Cohen in 1963 – it is independent of ZFC. By the end of life Gödel (1970) was searching for the answer. If we agree with the thesis that the limits of axiomatization coincide with the limits of mathematics itself, then the Gödel’s incompleteness theorem states that the (mathematical) world is incomplete. In that interpretation the Gödel’s theorem is an argument against platonism. As an argument against the possibility of complete knowledge, the Skolem-Löwenheim theorem can also be used. The first-order (countable) language is too weak to describe models that are uncountable.

²⁹ It was given currently by Emil du Bois-Reymond (1831–1889), a German physiologist, in his *Über die Grenzen des Naturerkennens* [On the limitations of knowledge in the natural sciences] of 1872, (Du Bois-Reymond 1912). His brother Paul Du Bois-Reymond was a mathematician. Problems of natures of matter and the origin of motion were examples of unsolvable questions.

³⁰ These famous six words ended the address that Hilbert gave at the meeting of the Society of German Scientists and Physicians in Königsberg in 1930, on the occasion of presentation to him of an honorary citizenship of the town. *Wir müssen wissen. Wir werden wissen* is engraved on his tombstone in Göttingen. See <http://www.psych.uni-goettingen.de/home/ertel/ertel-dir/morehome/4gallerypast/4.04gravesofrenownscientists.html> “Reide-meister and Szego made arrangements for Hilbert to repeat the last part of his speech over the local radio station; a record of this talk pronounced at the broadcasting studio exists and was recently acquired by Victor Katsnelson. He kindly supplied his colleague Victor Vinnikov with the transcript of Hilbert’s address, which was then diligently translated to English by Amelia and Joe Ball.” See <http://www.math.ucsd.edu/williams/motiv/hilbert.html>; <http://math.sfsu.edu/smith/Documents/HilbertRadio/HilbertRadio.pdf>; (Vinnikov 1999).

harmony” between mathematics and physics. After 30 years, enriched by the experience of the participation in elaboration of the general theory of relativity and quantum mechanics, he speaks again about this “pre-established harmony”, the most magnificent and most wonderful example of “which is the general theory of relativity and quantum mechanics” (Hilbert 1930, p. 961). In the Königsberg address Hilbert stresses the special role of mathematics in science. It is as Galileo said:

Die Natur kann nur der verstehen der ihre Sprache und die Zeichen kennengelernt hat, in der sie zu uns redet.

Only one who has learned the language and signs in which nature speaks to us can understand nature.

After Kant, Hilbert repeated:

Ich behaupte, daß in jeder besonderen Naturwissenschaft nur so viel eigentliche Wissenschaft angetroffen werden kann, als darin Mathematik enthalten ist.

I maintain that, in any particular natural science, genuine scientific content can be found only in so far as mathematics is contained therein.

Hilbert (n.d., p. 95), (1931, p. 485) believed in parallels between nature and thought:³¹

Between thought [Denken] and event [Geschehen] there is no fundamental and no quantitative difference. This explains the pre-established harmony [between thought and reality] and the fact that simple experimental laws generate ever simpler theories.³²

³¹ See (Thiele 2003, p. 18).

³² The question of simple principles that generate theory is discussed by Leibniz. In his *Discourse on Metaphysics* (2005, sections 5–6) we read:

When the simplicity of God’s way is spoken of, reference is specially made to the means which he employs, and on the other hand when the variety, richness and abundance are referred to, the ends or effects are had in mind. Thus one ought to be proportioned to the other, just as the cost of a building should balance the beauty and grandeur which is expected. It is true that nothing costs God anything, just as there is no cost for a philosopher who makes hypotheses in constructing his imaginary world, because God has only to make decrees in order that a real world come into being; but in matters of wisdom the decrees or hypotheses meet the expenditure in proportion as they are more independent of one another. The reason wishes to avoid multiplicity in hypotheses or principles very much as the simplest system is preferred in Astronomy. [...] Thus we may say that in whatever manner God might have created the world, it would always have been regular and in a certain order. God, however, has chosen the most perfect, that is to say the one which is at the same time the simplest in hypotheses and the richest in phenomena, as might be the case with a geometric line, whose construction was easy, but whose properties and effects were extremely remarkable and of great significance. I use these comparisons to picture a certain imperfect resemblance to the divine wisdom, and to point out that which may at least raise our minds to conceive in some sort what cannot otherwise be expressed. I do not pretend at all to explain thus the great mystery upon which depends the whole universe.

Mathematics is a tool that joins thoughts and practice. All our culture is based on the intellectual exploration of the nature and the exploitation of the nature has its fundamentals in mathematics, cf. (Hilbert 1930).

It is clear that Hilbert was in duty to justify his epistemological optimism. This optimism was grounded in his concept of mathematics. At the International Congress of Mathematicians in Bologna in 1928 Hilbert (1929) added a new problem of completeness³³ to the old problem of consistency.³⁴ Hans Hahn (1879–1934) communicated Hilbert's program to the Vienna Circle. In 1929 the problem of the completeness of the first order logic was presented. It also provided the young Kurt Gödel with his dissertation topic. Gödel (1930a) in his Ph.D. dissertation demonstrated that the first-order logic is complete, i.e. every valid formula can be derived from the axioms based on the *Principia Mathematica* (1910, 1912 and 1913) by Whitehead and Russell.³⁵ In the same year the problem of decidability (Entscheidungsproblem) was also raised in the famous textbook of logic *Grundzüge der theoretischen Logik* (1928) by Hilbert and Ackerman that proved to be extremely influential.

The questions of completeness, consistency and decidability may be formulated as follows:

1. is mathematics complete, in the sense that every truth about a given subject matter is inferable by means of a finite number of well-definite steps from a well-definite (recursive) set of axioms,³⁶

Hilbert also was convinced that: there is a realm beyond phenomena, and the universe is governed in such a way that a maximum of simplicity and perfection is realized, cf. (Thiele 2003, p. 18).

The idea of simplicity governed Einstein's scientific work and his searching for unitary field theory, see (Norton 2000).

³³ The problem was already mentioned in (Hilbert 1899). The question was also a subject of academic lectures (Hilbert 1905). In 1929, Mojżesz Presburger provided a partial solution to Hilbert's problems. He proved that natural number arithmetic with only addition and no multiplication is consistent (void of contradiction) and complete (capable of proving all valid statements).

³⁴ This problem was formulated for the first time in Hilbert's lecture at the Congress in Paris as the second problem of proving the consistency of axioms of arithmetic (number theory and analysis): "Any contradiction in the deductions from the geometrical axioms must thereupon be recognizable in the arithmetic of this field of numbers. In this way the desired proof for the compatibility of the geometrical axioms is made to depend upon the theorem of the compatibility of the arithmetical axioms.

On the other hand, a direct method is needed for the proof of the compatibility of the arithmetical axioms. The axioms of arithmetic are essentially nothing else than the known rules of calculation, with the addition of the axiom of continuity." (Hilbert 1900)

³⁵ The completeness theorem was a step towards the resolution of Hilbert's Entscheidungsproblem.

³⁶ It means that we can mechanically check for any given statement if it is an axiom

2. is mathematics consistent, in the sense that only valid formulas may be proved,
- and

3. is mathematics decidable, i.e. there exists a definite procedure which, in principle, be applied to any formula and which guarantees producing a correct decision as to whether that formula is valid or not?

It is believed that Hilbert expected the answer to each question to be "yes".

If an axiomatic system is complete and consistent, i.e. all valid and only valid formulas are its theorems, as is the case of the first-order logical calculus, 3 may be replaced by:

4. there is a method that in the case of any formula in a finite number of well-definite steps allows to decide whether this formula is a theorem or not.

If an infinite set of axioms is allowed, then the condition 3 is equivalent to the conjunction of 1, 2 and 4: all valid and only valid formulas can be taken as the axioms. An inconsistent system is decidable in the sense 4 though not necessarily in the sense 3. The difference between 3 and 4 is apparent in the case of an incomplete system.

On 7 September 1930 (one day before Hilbert's address) at the second Meeting of Epistemology, September 5–7, 1930 (Zweiten Tagung für Erkenntnislehre der exakten Wissenschaften, a meeting of members of Wiener Kreis and Berlin Society for Empirical Sciences that also took place in Königsberg) Gödel lectured on his incompleteness theorem (the First Incompleteness Theorem).³⁷ In the proof³⁸ Gödel combined two insightful ideas. First, formulas were associated with natural numbers, to allow statements in arithmetic to refer to other statements in arith-

as well as the validity of proofs so that there can be no doubt that a theorem follows from the starting list of axioms. In theory, such a proof can be checked by a computer.

³⁷ Gödel spoke on the third, last day of the conference. Perhaps it was not tiredness that was the reason for the lack of interest in Gödel's lecture (being 25 years old, he was not known yet). Even in the conference materials there is a lack of information about it. John von Neumann (He met David Hilbert on a visit to Göttingen in 1926, after which he was offered a position of a Privatdozent, an unsalaried lecturer, at the University of Berlin and then at the University of Hamburg. In 1930 he visited the United States, accepting a salaried lecturership at Princeton University, a move which would shape the rest of his life) was the only participant that immediately understood and conceived the significance of Gödel's result: *genius recognized genius*. After Gödel's talk he had a long discussion with him asking about details of the proof. See (Hahn et al. 1930, Gödel 1930b).

³⁸ (Gödel 1931) is arguably the most important mathematical paper of the 20th century and one of the greatest and most surprising results in the whole history of mathematics. The results of Gödel, however, have been also achieved – but not published – by Zermelo and Post.

metic.³⁹ Second, the predicate 'is provable in *Principia Mathematica*' was translated into a statement in arithmetic. Thus it was allowed to construct a statement that defined itself as not provable within *Principia Mathematica*. He proved that the leading formalization of mathematics, *Principia Mathematica*, was either an incomplete or inconsistent theory of the natural numbers. In other words, he stated that there are propositions in the first-order language of arithmetic that are either true but unprovable within *Principia Mathematica* or false but provable. Since consistency is required of all serious proof systems, Gödel's result is considered a proof of the incompleteness of *Principia Mathematica*. The completeness of the axioms of arithmetic can never be proved. If a system of sufficient power to express arithmetic is consistent, then it must be incomplete.⁴⁰ More generally, the first theorem can be paraphrased as:

there is no all-encompassing axiomatic system which is able to prove all mathematical truths, but no falsehoods.

Thus the condition 1 cannot be fulfilled. It was the first nail to coffin of Hilbert's program.⁴¹ While the proof of the incompleteness of *Principia Mathematica* was finished and unquestionable, the incompleteness

³⁹ The idea of arithmetization of theorems as well as rules applied in the proof by Gödel is the base of von Neumann architecture of computers (instructions are stored as binary values).

⁴⁰ Gödel remembered: "In summer 1930 I began to study the consistency problem of classical analysis [...] I reached the conclusion that in any reasonable formal system in which provability in it can be expressed as a property of certain sentences, there must be propositions which are undecidable in it." (Wang 1996)

⁴¹ In the 1920s, David Hilbert proposed a research program with the aim of providing mathematics with a secure foundation. It was one of the grandest research projects in the philosophy of mathematics. A number of mathematical principles, such as impredicative definitions, the axiom of choice, and the law of excluded middle for infinite totalities, were charged contradictory, false, or at least were unfounded assumptions. Hilbert distinguished between the unproblematic, finitistic part of mathematics and the infinitistic (ideal) part that needed justification. Infinitistic mathematics should be justified by finitistic methods. Only they are trustful. The ideal part of mathematics has to be formalized. In the proof theory only the finitistic methods could be used. Hilbert wanted to reduce all of mathematics to finite reasoning from a set of self-evident axioms. The purpose of Hilbert's program was to show that mathematics when the actual infinity is allowed is safe and free of any inconsistencies: *No one shall expel us from the paradise that Cantor has created for us* (Aus dem Paradies, das Cantor uns geschaffen, soll uns niemand vertreiben können), see (Hilbert 1925). In pursuing this objective Hilbert finally approached Kronecker's finitism. To a certain extent his metamathematics closely corresponded to finitary mathematics 'à la Kronecker. Due to Cantor the concept of the actual infinity for the first time became available for strict formal-logical (certainly, in the sense of classical Aristotle's logic) and mathematical analysis. In the field of mathematics Hilbert tried to answer the old question of philosophy: whether and how the limited human brain is able to handle the infinite?

of all formal systems was dependent upon an interpretation of 'finite means'.⁴²

Though Gödel's theorems are not the main subject of interest here, it is worth saying something more about them for the better understanding of further considerations. Roughly speaking, in the first theorem Gödel has proved that for any consistent first-order formal system that is sufficiently strong to axiomatize the arithmetic of natural numbers there are true sentences, moreover one can construct such a formula, that can neither be proven nor disproven within the system itself. There is not a complete list of axioms. Each time a statement is added as an axiom, if the set of axioms is recursive, there will always be another statement (an undecidable statement) out of reach. Mathematical knowledge is always imperfect; starting from a recursive consistent set of axioms we cannot prove everything that is to be known. In the axiomatic approach, the "tree" of all formulas growing from axioms is not only extremely expansive but even disconnected. In the theorem on the unprovability of consistency (today called Gödel's Second Incompleteness Theorem) it is stated that such a formal system, if used as its own proof theory, cannot be used to prove both its own consistency and completeness.⁴³ It does not depend on the power of induction that is allowed within the system. It means that the condition 2 of page 89 is not feasible even if non-finitistic induction is allowed. It is worth mentioning that Church (1936a) showed that Gödel's results cannot be obtained by finite means. To prove the incompleteness of a system, we have to prove its consistency. However, according to the second incompleteness theorem, the proof of consistency of a system containing arithmetics is not possible without using stronger logical means that are available in the system itself. In such a system one can construct a formula that:

1. can be neither proven nor disproven within that system;
- or
2. can be both proven and disproven within that system.

⁴² The goal of Hilbert's problem progressed into an effort to show that mathematics is reducible to logic. The school of thought that assumes that is referred to, alternatively, as positivism, logical positivism or logicism, see (Corbeil 1997). Bertrand Russell and Alfred North Whitehead using a particular set of axioms of arithmetic that seemed promising, commonly labeled as Peano's axioms of arithmetic, proceeded to construct a set theory and a number theory. The result was *Principia Mathematica*, a monumental three-volume tome spanning thousands of pages of small print. When Gödel published his incompleteness theorems, Russell and Whitehead were working on the fourth part of *Principia Mathematica* (on geometry). The volume was never completed.

⁴³ For technical as well as philosophical and historical information on Gödel's theorems see e.g., (Murawski 1999).

The Gödel theorems do not state that a theory of arithmetic is inconsistent or incomplete, only that it cannot be proven under given conditions. Gerhard Gentzen (1936, 1938, 1974) has showed that it is possible to prove both the consistency and completeness of a formal system, only if induction of strictly greater order is used.⁴⁴ Thus such a proof with induction of an equivalent order is not possible. It is proven that the arithmetic with finitistic induction is both consistent and complete by using non-finitistic induction to ϵ_0 .⁴⁵ Gödel's theorems are consequences of Gentzen's theorem.

Gödel's results confirmed Cantor's belief that there are no foundations of mathematics without metaphysics, i.e. without infinite methods.

Gödel (1931, p. 197) openly referred to Hilbert's programme:

It must be expressly noted that [these theorems...] represent no contradiction of the formalistic standpoint of Hilbert. For this standpoint presupposes only the existence of a consistency proof effected by finite means, and there might conceivably be finite proofs which **cannot** be stated in [Peano arithmetic].

The thought is repeated in a letter to Constance Reid:

I would like to call your attention to a frequently neglected point, namely the fact that Hilbert's scheme for the foundation of mathematics remains highly interesting and important in spite of my negative results. What has been proved is only that the specific epistemological objective which Hilbert had in mind cannot be obtained [...] As far as my negative results are concerned, apart from the philosophical consequences mentioned before, I would see their importance primarily in the fact that in many cases they make it possible to judge, or to guess, whether some specific part of Hilbert's program can be carried through on the basis of given metamathematical presuppositions. Cf. (Thiele 2003).

The incompleteness theorems do not only affect the philosophy of mathematics. They are pertained to knowledge in general. J. I. Austin, being informed about Gödel results that not all truth can be proved, asked:

Who would have ever thought otherwise?

While they are only a limitation for the formalization process, they were taken by many to argue that there are statements in arithmetic whose truth is unknowable. Gödel's theorems are thus usually considered to be a major limitation on the power of reasoning. For some philosophers Gödel's theorems

⁴⁴ Independently of Gentzen the consistency of arithmetic was proved by Ackerman (1940).

⁴⁵ ϵ_0 is a transfinite (though countable) ordinal. ϵ_0 is defined as the smallest ordinal ϵ such that $\omega^\epsilon = \epsilon$ or as the limit of the sequence $\omega, \omega^\omega, \omega^{\omega^\omega} \dots$

are sufficient reasons for the conclusion that human thought is uncomputable. This claim is addressed by Lucas (1961) and his argument is already classical. It has been reported by Marvin Minsky that Kurt Gödel told him personally that he believed that human beings had an intuitive, not just computational, way of arriving at truth and that therefore his theorem did not limit what can be known to be true by humans.⁴⁶ Gödel himself believed that it was not a limitation for human reasoning, see (Dawson 1997). Arguments based on Gödel's theorems and their implications are indispensable in contemporary considerations of the concept of mind and questions of human knowledge. These arguments are the main subject of many publications.⁴⁷ One of the most discussed is Roger Penrose's *The Emperor's New Mind* (1989) and *Shadows of the Mind* (1994), where he claims that minds depend on quantum mechanical phenomena that cannot be reproduced by computation. If human mind is able to transcend limitations proved by Gödel but computers are not able to do that, then the application of computers is limited and the artificial intelligence is fundamentally weaker than a human mind.

Gödel proved that the idea of consistent and complete axiom system is not feasible, i.e. to both the questions 1 of page 88 and 2 of page 89 the answer is negative. Nevertheless the question 3 of page 89 still remained unanswered.⁴⁸ However, Gödel's theorems say nothing about decidability. Gödel assumed that the system was consistent but incomplete. With a decision method, we can assume a consistent system that is incomplete, but still attempts to use the decision method to check for theorems. That aspect was addressed by Turing and Church.

3. The idea of decidability

The idea of mechanical method of solving problems is old.⁴⁹ Usually Lullus⁵⁰ is pointed out as its originator. The founder of contemporary logic,

⁴⁶ See http://www.absoluteastronomy.com/encyclopedia/G/G/Gödel's_incompleteness_theorem.htm.

⁴⁷ For more see (Krajewski 2003).

⁴⁸ If we ask about 4, a procedure to decide whether a sentence was or was not a theorem, we might still hope that there would be possible to produce a general algorithm that for a given statement determines whether it is undecidable or not.

⁴⁹ The problem of mechanization of reasoning in a historical perspective is a subject of (Marciszewski & Murawski 1995).

⁵⁰ Raimundus Lullus (1232–1316), a theologian, philosopher and logician. He was influenced by Christian, Muslim and Jewish cultures. Lullus wrote over 280 works in Latin and Arabic. In *Ars Magna* (1305–08) he presented the idea of mechanization of reason.

Leibniz,⁵¹ was inspired by *Ars Magna* of Ramon Llull, although he criticized the author because of the arbitrariness of his categories and his indexing. Leibniz distinguished between two different versions of *Ars Magna*. The first version, *ars inveniendi*, finds all true scientific statements. The other, *ars iudicandi*, allows one to decide whether any given scientific statement is true or not. In *Dissertatio de arte combinatoria* (1923) Leibniz cites the idea of Hobbes that all reasoning is just a computation: *cogitatio est computatio*.

William Stanley Jevons was next after Lullus to build a logical machine,⁵² which he did in 1869.

In the seventeenth century Gottfried Wilhelm von Leibniz, after having constructed a successful mechanical calculating machine, dreamt of building a machine that could manipulate symbols in order to determine the truth values of disputed statements.⁵³ He dreamt about times when political and economic questions could be settled, not by disputes, but by a sort of reckoning (*calculemus*) through which it would be possible for all people concerned to agree at least in principle about the issues at stake. *Calculemus!*⁵⁴ (Let's calculate!) reflects Leibniz's conviction that all human

ning. He constructed a logical machine: on seven wheels different notions were written, rotation of the wheels resulted in different associations of the notions. Lullus had the proper intention to demonstrate all the truths of the Catholic church. The machine was used in his missionary work in North Africa and Asia Minor where Lullus taught the pacifist doctrines of St Francis of Assisi. Lullus used logic and mechanical methods involving symbolic notation and combinatorial diagrams to relate all forms of knowledge. The underlying goal of the Lullus text machine was the formulation of universal rules, which could cross boundaries and cultures. His great invention was the idea of producing the totality of human wisdom by a mechanical method of combining a strictly limited quantity of signs. Lullisten, the followers of Lullus, studied and copied his works though they were forbidden for many centuries. (In 1376 Pope Gregory XI charged Lullus with confusing faith with reason and condemned his teachings.) In the 17th century the Lullus's idea of an universal language had a special attention. Cf. (O'Connor & Robertson 2005), http://de.wikipedia.org/wiki/Raimundus_Lullus. Werner Künzel, with his colleague Heiko Cornelius (1991) translated Lullus's *Ars Magna* into the 'higher language' of Cobol.

⁵¹ Leibniz is among the authors who are often thought to have done more than they actually did. In Peckhaus (1999, s. 436) opinion:

The development of the new logic started in 1847, completely independent of earlier anticipations, e.g., by the German rationalist Gottfried Wilhelm Leibniz (1646–1716) and his followers; (Peckhaus 1994), (Peckhaus 1997, ch. 5). In that year the British mathematician George Boole (1815–1864) published his pamphlet *The Mathematical Analysis of Logic* (1847).

See also (Marciszewski & Murawski 1995).

⁵² In the case of a logical machine the output is YES or NO. In the case of calculating machine the output is – roughly speaking – a number.

⁵³ Leibnizian schemes are satirized in Jonathan Swift's *Voyage to Balnibarbi*.

⁵⁴ CALCULEMUS is an international research network. The goal of the group is the development of a new generation of mathematical assistance systems based on the integration of the deduction power of deduction systems and the computational power of computer algebra system. See <http://www.calculemus.net/>

reasoning may be turned into an object of mathematical demonstration and in such a way, any controversial truth can obtain the evidence of $2 + 2 = 4$.

... quando orientur controversiae, non magis disputatione opus erit inter duos philosophos, quam inter duos Computistas. Sufficiet enim calamos in manus sumere sedereque ad abacos, et sibi mutuo (accito si placet amico) dicere: c a l c u l e m u s.⁵⁵ (Leibniz 1890, vol. 7, p. 200)

Actually, when controversies arise, the necessity of disputation between two philosophers would not be bigger than that between two accountants. It would be enough for them to take the quills in their hands, to sit down at their abaci, and to say (as if inviting each other in a friendly manner): Let's calculate! (Calculemus!)

Leibniz was the first to realize that a comprehensive and precise symbolic language *characteristica universalis* (the perfect language which would provide a direct representation of ideas along with a calculus for the reasoning) is a prerequisite for any general problem solving method and much of his subsequent work was directed towards that goal. With the help of the language and a formal calculus (*calculus ratiocinator*) it would be possible to verify human thoughts like it is possible to verify an arithmetic calculation.⁵⁶ Leibniz's ideas were taken up again by Frege who defined the first

⁵⁵ Similar expressions occur in other texts of the same volume, e.g. p. 64–65, 125.

⁵⁶ The impact of Leibniz's ideas on computer science is discussed in *The Universal Computer: The Road from Leibniz to Turing* (2000) by Martin Davis. Leibniz's contributions to computer science are many-sided. E.g., he is the inventor of the binary system: ENIAC (**E**lectronic **N**umerical **I**ntegrator **A**nd **C**alculator) based on the decimal system (Stern 1981, p. 133); EDVAC (**E**lectronic **D**iscrete **V**ariable **A**utomatic **C**omputer) due to von Neumann was based on binary system (Goldstine 1972, p. 182):

Several states in each of which the combination can exist indefinitely, without any outside support, while appropriate outside stimuli (electric pulses) will transfer it from one equilibrium into another. These are the so-called trigger circuits, the basic one having two equilibria. ... The trigger circuits with more than two equilibria are disproportionately more involved. Thus, whether the tubes are used as gates or as triggers, the all-or-none, two equilibrium arrangements are the simplest ones. 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." See "First Draft of a Report on the EDVAC" (von Neumann 1981).

The idea of the binary system was in connection with Leibniz's ontology. From only two absolutely simple concepts, God and Nothingness, all other concepts may be constructed, the world, and everything within it. And then Leibniz came to see that what is crucial in what he had written is the alternation between God and Nothingness. And for this, the numbers 0 and 1 suffice. According to Chaitin (2004):

[...] all of information theory derives from Leibniz, for he was the first to emphasize the creative combinatorial potential of the 0 and 1 bit, and how everything can be built up from this one elemental choice, from these two elemental possibilities. So, perhaps not entirely seriously, I should propose changing the name of the unit of information from the bit to the leibniz!

formal language in his famous work *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens* (1879). The first complete calculus for this language was presented by Hilbert (1928) in 1927. Hilbert's calculus was proved to be complete by Gödel in 1930 (cf. page 88 of this work).

The problem of decidability was already present in Hilbert's Paris lecture in the question of Diophantine equations.⁵⁷ In the 10th problem Hilbert asked about a finite procedure of finding an answer whether a given Diophantine equation had or did not have a solution.⁵⁸ The opinion that in the 10th problem the idea of Entscheidungsproblem⁵⁹ was present (e.g. Penrose) could be justified by the fact, that

it is a corollary of the methods used to give a negative solution to Hilbert's tenth problem that the question of whether any given Turing machine will eventually halt, and hence the Entscheidungsproblem can be encoded as a Diophantine problem. (Davis, Matijasevic & Robinson 1976)

There is one-to-one correspondence of the Turing machines and the Diophantine equations.⁶⁰

Decidability pertains to a non-finite countable class of questions that are characterized by a finite amount of information. Such a class is decidable if and only if there exists a finitely described (mechanical) procedure that in a finite number of steps enables to answer any question of the class YES or NO.

The substance of the decidability relays on the existence of the only one method that is applicable to any question of a considered class. In the case of Diophantine equations since the times of Diophant the answer

⁵⁷ A Diophantine equation is an equation $P(x_1, \dots, x_k) = 0$, where P is a polynomial with integers coefficients and the variables x_i range over integers. As an example we could use a famous Fermat's equation: $x^n + y^n = z^n$, where $x, y, z, n \in \mathfrak{N}$ – Fermat claimed that for any $n > 2$ the equation is unsolvable.

⁵⁸ Hilbert's 10th problem (is there a finite process which determines if a polynomial equation is solvable in integers?) has been solved in negative by a young Russian mathematician Yuri Matiyasevich (1970). He used some earlier results of Martin Davis, Hilary Putnam and Julia Robinson. Matiyasevich and Robinson have – despite the Cold War – collaborated on the problem. Julia Robinson was close to the end result. See (Davis 1973, Davis & Hersh 1973, Davis 1982, Matiyasevich 1993).

⁵⁹ German for “decision problem”. The first documented case of using the term “Entscheidungsproblem” was by Behmann (1921, p. 47) in his announcement at the meeting of mathematical society in Göttingen in May 1921.

⁶⁰ Matiyasevich's solution that there is a Diophantine equation that has not got a solution and that this fact cannot be proved is equivalent to the Gödel theorem. The interrelations between the Universal Turing Machine and Diophantine equations lead Chaitin towards discovering random structure of mathematics.

YES or NO has been given for many subclasses and for many particular equations. Such an algorithm does exist for the solution of the first-order Diophantine equations. Moreover, there is an algorithm of finding solutions of any of the Diophantine equations. The problem of decidability of the class of the Diophantine equations concerns finding a method that enables to give an answer YES or NO to the question: has a given equation got a solution at all (not: the solution). This question has been raised as the 10th problem.

A formal system is decidable if and only if there is a mechanical procedure that in the case of any sentence in the language of the system after a finite number of steps enables to give an answer YES or NO to the question if the sentence is a theorem of the system. The problem of decidability of a formal system was stated in works of Schröder (1895), Löwenheim (1915) and Hilbert (1918).

The first though vague formulation of the problem of decidability appeared in the works of Hilbert (1900, 1918). Directly the problem was formulated by Behmann (1922, p. 166):

Es soll eine ganz bestimmte allgemeine Vorschrift angegeben werden, die über die Richtigkeit oder Falschheit einer beliebig vorgelegten mit rein logischen Mitteln darstellbaren Behauptung nach einer endlichen Anzahl von Schritten zu entscheiden gestattet, oder zum mindesten dieses Ziel innerhalb derjenigen – genau festzulegenden – Grenzen verwirklicht werden, innerhalb deren seine Verwirklichung tatsächlich möglich ist.

A completely definite general procedure should be given when after a finite number of steps it is possible to decide about truth or falsehood of any given statement that could be formulated by using logical means, or at least that goal may be realized in – precisely stated – frames where its realization is indeed possible.

The question of decidability of the first-order logical calculus, i.e. the classical problem of decidability, what presently is denoted as “*Entscheidungsproblem*”, was formulated in 1928 by Hilbert in *Grundzüge der theoretischen Logik* (1928), a book written together with Ackerman.⁶¹ In the part entitled “Das Entscheidungsproblem im Funktionenkalkül und seine Bedeutung” (The decidability problem in functional calculus and its significance) we read (1928, p. 72):

⁶¹ In the year of book's publication there took place a congress of mathematicians in Bologna. In his lecture at the congress Hilbert presented a question of completeness. Nothing was said about *Entscheidungsproblem*.

Das Entscheidungsproblem ist gelöst, wenn man ein Verfahren kennt, das bei einem vorgelegten logischen Ausdruck durch endlich viele Operationen die Entscheidung über die Allgemeingültigkeit⁶² bzw. Erüllbarkeit erlaubt.⁶³

The problem of decidability is solved if a procedure is known when for any given logical expression after a finite number of operations it is possible to decide about validity⁶⁴ or satisfiability.⁶⁵

The importance of decidability of the first-order logical calculus is based on Hilbert's thesis (not explicitly formulated by him) that all mathematics could be expressed in the first-order language. The Pascal theorem serves as an example. Hilbert showed how that theorem and questions concerning its logical connections in geometry would be reduced to a problem of the first-order logical calculus. Geometrical notions are replaced by implicit definitions and in result symbols are devoid of their geometrical meaning, see (Hilbert & Ackerman 1928, p. 74–76). After the discussion of the example he wrote (Hilbert & Ackerman 1928, p. 77):

Änliche Überlegungen gelten natürlich für jedes beliebige Axiomensystem. Similar considerations are valid for any axiomatic system.

In the conclusion it is clearly stated that:

... das Entscheidungsproblem muß als das Hauptproblem der mathematischen Logik bezeichnet werden.

the Entscheidungsproblem has to be appointed as the central problem of mathematical logic.

Hilbert's point was that if we came to possessing of such an effective procedure applicable to any first-order formula, then ignorance would be banished

⁶² Bei dem Problem der Allgemeingültigkeit handelt es sich um die folgende Frage: *Wie kann man beim einem beliebigen vorgelegten logischen Ausdruck, der keine individuellen Zeichen enthält, feststellen, ob der Ausdruck bei beliebigen Einsetzungen für die vorkommendem Variablen eine richtige Behauptung darstellt oder nicht?*

⁶³ Bei dem Problem der Erüllbarkeit handelt es sich um die Frage (Hilbert & Ackerman 1928, p. 73): *ob es überhaupt eine Einsetzung für die Variablen gibt, so daß durch den betreffenden Ausdruck eine richtige Behauptung dargestellt wird.*

⁶⁴ In the problem of validity there arises the following question: *How can man for any given logical expression, that do not contain individual symbols, establish if this expression for any substitution for occurring variables is a true sentence or not?*

⁶⁵ In the problem of satisfiability there appears the following question: *is there at all any substitution for the variables such that the corresponding sentence is true.* – Let us remark that the notions of validity and satisfiability are dual: a sentence is valid iff its negation is not satisfiable. The notion of validity is preferred in proof theory but the notion of satisfiability is preferred in model theory.

from mathematics forever. The view was shared by Bernays and Schönfinkel (1928):

Das zentrale Problem der mathematischen Logik, welches auch mit den Fragen der Axiomatik im engsten Zusammenhang steht, ist das Entscheidungsproblem.

The central problem of mathematical logic, which is also most closely related to the questions of axiomatics, is the Entscheidungsproblem.

Bearing in mind Leibniz's idea of *Ars Magna*, the Entscheidungsproblem may be characterized as follows (Börger, Grädel & Gurevich 1997, p. 4):

In the framework of first-order logic, an *ars inveniendi* exists: the collection of valid first-order formulae is recursively enumerable, hence there is an algorithm that lists all valid formulae. The classical decision problem can be viewed as the *ars iudicandi* problem in the first-order framework. It can be sharpened to a yes/no question: Does there exist an algorithm that decides the validity of any given first-order formula?

The important difference between the earlier stated problem of decidability, e.g., by Löwenheim (1915) and Behmann (1922), and Hilbert's *Entscheidungsproblem* consists in generality. Hilbert (1928, pp. 77–78) directly pointed out the known particular solutions of the decidability. If Hilbert's thesis was supposed to be solving *Entscheidungsproblem*, it would comprise all the questions of decidability and

Auch Fragen der *Widerspruchsfreiheit* würden sich an Hand des Entscheidungsverfahrens lösen lassen. (Hilbert & Ackerman 1928, p. 76)

By the decidability procedure the question of consistency could be solved, too.

Hilbert's belief that science is alive as long as there are unsolved problems seems to be in contradiction with the idea of decidability: if everything can be solved by calculation, then in science the development is fictitious. Already Leibniz remarked that:

It is unworthy of excellent men to lose hours like slaves in the labor of calculation which could safely be regulated to anyone else if machines were used.

To answer this question, we have to take into account the fact that besides sentences there are notions. Hilbert (1992, p. 8) wrote:

... vielmehr zeigt sich, daß die Begriffsbildungen in der Mathematik beständig durch Anschauung und Erfahrung geleitet werden, so daß im großen und ganzen die Mathematik ein willkürfreies, geschlossenes Gebilde darstellt.

... moreover it turns out that in mathematics formation of concepts is directed by intuition and experience, hence in size and whole mathematics is a free of arbitrariness, closed composition.

The mathematical cognition is accomplished in the process of forming of concepts:

Es bilden also die verschiedenen vorliegenden mathematischen Disziplinen notwendige Glieder im Aufbau einer systematischen Gednakenentwicklung, welche von einfachen, naturgemäß sich bietenden Fragen anhebend, auf einem durch den Zwang innerer Gründe im wesentlichen vorgezeichneten Wege fortschreitet. Von Willkür ist hier keine Rede. Die Mathematik ist nicht wie ein Spiel, bei dem die Aufgaben durch willkürlich erdachte Regeln bestimmt werden, sondern ein begriffliches System von innerer Notwendigkeit, das nur so und nicht anders sein kann. (Hilbert 1992, p. 9)

4. Entscheidungsproblem

One can characterize the problem of decidability as follows: Given a collection of assumptions A stated in some logical system, and a statement s , is it possible to decide on the basis of some universal computation method (a decision method) whether s is or not inferrable from A ? If for a given system S such a universal computational method exists, then the system is decidable. For many logical systems, several such procedures existed before Church's and Turing's publications, but not for the system of the first-order logic.⁶⁶

The first-order logic is complete. If we suppose – what Hilbert did – that any human reasoning may be expressed in this logic, then the decidability of the first-order logic means that all human reasoning can be reduced to a calculation. For this reason the Entscheidungsproblem has been characterized by Hilbert as the fundamental problem of mathematical logic.

In order to solve the question of the Entscheidungsproblem, a concept of calculus or – in other words – an effective (a mechanical) procedure has to be defined. Solving the Entscheidungsproblem (and the 10th problem of Diophantine equations) required a precise formalization of what mechanical computation is. An attempt to define an intuitive notion rises the following questions:

1. which already definite (in order to avoid *ignotum per ignotum*) notions should be used to be as close as possible to the intuition of the notion of effective procedure?
2. is there only one or more intuitively acceptable notions of effective procedure?

In the beginning, in order to answer these questions, let us consider a methodological problem of the procedure that shall be used. In formal sciences by definition one basically understands a nominal definition.⁶⁷ Nominal definitions tell us about the correct usage of names. There are two kinds of nominal definitions. The meaning of the defined term of a certain language \mathcal{L} is reported in this language. In this case the defined expression is definable in the language \mathcal{L} . In another case a nominal definition introduces to a language \mathcal{L} a new word to replace (usually as a short) an expression of \mathcal{L} . This consists in introducing a new symbol or notation by assigning a meaning to it. A real definition or conceptualization is a description of an object in such a way that it is a description of this and only this object. Neither the case of nominal definition nor the case of real definitions are taken into account here. The exclusion of real definition needs some explanation. The real definition is conceived in such a way that a real definition of an object presupposes the existence of the object. Perhaps an argumentation is possible that there is an object “effective procedure” that is independent of our cognition. Nevertheless, we have to take into account the fact that a real definition characterizes or not a defined object, *tertium non datur*. Such a definition is right or wrong. Thus, the correctness of a real definition does not depend in any way on our conventions or on our methods of cognition. It is not the case with the concept of effective procedure.

Besides nominal definition and conceptualization there is a third procedure that may be used to precise concepts, namely, explication. It seems that explication is the proper procedure to analyze calculus. Roughly speaking, an explication is a detailed analysis of a concept used to explicate means to unfold; to give a detailed explanation of; to develop the implications of; and to analyze logically.⁶⁸

Explication was described by Carnap (1950, p. 3) as follows:

By the procedure of explication we mean the transformation of an inexact, prescientific concept, the *explicandum*, into a new exact concept, the *explicatum*. Although the *explicandum* cannot be given in exact terms, it should be made as clear as possible by informal explanations and examples.

⁶⁷ For the first time rules of definition were formulated by Stanisław Leśniewski (1931).

⁶⁸ See e.g. (Trzėsicki 2000, pp. 366–369).

⁶⁶ See (Kneale & Kneale 1962, pp. 724–737).

The task of explication consists in transforming a given more or less inexact concept into an exact one or, rather, in replacing the first by the second. We call the given concept (or the term used for it) the *explicandum*, and the exact concept proposed to take the place of the first (or the term proposed for it) the *explicatum*. The *explicandum* may belong to everyday language or to a previous stage in the development of scientific language. The *explicatum* must be given by explicit rules for its use, for example, by a definition which incorporates it into a well-constructed system of scientific either logico-mathematical or empirical concepts.

As an example of explication the definition of truth by Tarski⁶⁹ is pointed out (Carnap 1950, p. 5).

I am looking for an explication of the term 'true', not as used in phrases like 'a true democracy', 'a true friend', etc., but as used in everyday life, in legal proceedings, in logic, and in science, in about the sense of 'correct', 'accurate', 'veridical', 'not false', 'neither error nor lie', as applied to statements, assertions, reports, stories, etc. This explanation is not yet an explication; an explication may be given by a definition within the framework of semantical concepts, for example, by Tarski's definition of 'true' [...].

Carnap (1950, p. 5) indicates the following requirements that a concept must fulfil to be an adequate explicatum for a given explicandum:

1. similarity to the explicandum,
2. exactness,
3. fruitfulness,
4. simplicity.

These conditions in more detailed form are as follows (Carnap 1950, pp. 6–7):

1. The explicatum is to be similar to the explicandum in such a way that, in most cases in which the explicandum has so far been used, the explicatum can be used; however, close similarity is not required, and considerable differences are permitted.
2. The characterization of the explicatum, that is, the rules of its use (for instance, in the form of a definition), is to be given in an exact form, so

as to introduce the explicatum into a well-connected system of scientific concepts.

3. The explicatum is to be a fruitful concept, that is, useful for the formulation of many universal statements (empirical laws in the case of a non-logical concept, logical theorems in the case of a logical concept).
4. The explicatum should be as simple as possible; this means as simple as the more important requirements 1, 2 and 3.

Ch. Morris divided semiotics, a general theory of sign, into:

1. syntactics,
 2. semantics
- and
3. pragmatics.

Exact terms do not need explication. Since the explicated term is inexact, its explication cannot be evaluated as right or wrong. A proposed solution may be less or more satisfactory than another one.

Syntactics concerns relations between signs, semantics has as its subject relations between signs and reality. Pragmatics analyzes relations between sign and its user. The concept of (formal) proof is fundamental for syntactics. Semantics is based on the notion of truth. In classical semantics Tarski's definition of truth is assumed. The notion of effective procedure characterizes our ability of operating signs, thus it is basic for pragmatics.

What is expected in the case of the Entscheidungsproblem is the explication of the effective procedure. The truth table test is such a method for the propositional calculus. Different models of computation were introduced. 1936 saw an independent development of the three influential models of computation, aimed at doing the following: the lambda calculus, Turing machines and recursive functions. An explication of effective procedure was done by Church (1935, 1936b, 1936a) whose solution presupposed the definition of the intuitive notion of "mathematical function". Church's explication has been based on the notion of "lambda-definable-function" introduced by Church (1932, 1936b, 1936a) and Stephen Kleene (1935). A few months later, independently of Church, it was done by Turing (1936–37).⁷⁰ Another

⁶⁹ The idea that the notion of effective procedure is the same kind notion as truth, function, limit and so on is exploited by Mendelson in his argumentation for the Church-Turing thesis.

My viewpoint can be brought out clearly by arguing that CT is another in a long list of well-accepted mathematical and logical "theses" and that CT may be just as deserving of acceptance as those theses. Of course, these theses are not ordinarily called "theses", and that is just my point. See (Mendelson 1990).

⁷⁰ A fascinating biography of Turing by Andrew Hodges (1983) may be recommended. Turing, a co-founder of computer science, is showed as a man and scientist. Turing was much more than a mathematician, he was a specialist in electronics and signal processing even while the subjects were still unrecognised. There is a website maintained by Hodges: <http://www.turing.org.uk/> It is worth remarking that as Hilbert's problem gave stimulus on theoretical as codebreaking of Enigma on practical level of computer science. (Enigma – used by the Nazis an improved, military version of the commercial machine created by a German electrical engineer Scherbius in 1918.) A contribution of Polish mathematicians to solving the mystery of ENIGMA has to be pointed out here.

definition of the notion of “effective procedure” was done by Kleene and Post (1936) and by others later, e.g. by Markov (1955). The concepts of “recursive functions”, studied by Jacques Herbrand (1932) and Kurt Gödel (1934), gave rise to the idea of general recursiveness that was the base of the explication proposed by Herbrand, Gödel and Kleene. The definition starting from the notion of binormality was proposed by Post. Each proposal was well-defined and each seemed to correspond with what we would intuitively regard as an effective process (Copeland 2005, Deutsch 1985, Deutsch 1997, Kleene 1967, Marciszewski 2005). Unlike others, Turing’s concept was as much an instruction manual on how such a device might be built as it was a formalism for studying computation. That link between abstract computability and physical computability made the Turing machines quickly become the standard model of computation.⁷¹

Alan Turing was first introduced to Hilbert’s problem during the spring term of 1935 while attending a lecture on the foundations of mathematics given by Max Newman at Cambridge. Turing was extremely intrigued by the problem and immediately set out to prove that there was no algorithm that could satisfy the Entscheidungsproblem in an interesting and ingenious manner. Turing’s draft paper arrived on Max Newman’s desk at the same time as a copy of the *American Journal of Mathematics* containing “An Unsolvable Problem in Elementary Number Theory” by Alonzo Church. In his article “On Computable Numbers, with an Application to the Entscheidungsproblem” (1936–37) Turing explicitly referred to Gödel (1931), Church (1936*b*), and the 1931 edition of (Hilbert & Ackerman 1928). There was no rivalry between Turing and Church. Turing came to Princeton and some of his best work was accomplished there under Church’s direction.⁷²

Turing’s analysis of computation aimed at determining the mental processes fundamentally required by an individual performing a computation. Turing focused on the actual thought-process that took place when an individual attempted to develop an algorithm. He utilized an example of a simple mathematical calculation to create a case study for the actual thought processes taking place when a person performed a calculation. In *On Computable Numbers with an Application to the Entscheidungsproblem* Turing noted (1936–37, p. 117):

⁷¹ The Turning machine inspired the first real computer in 1943, called *Colossus*, and later the modern computer.

⁷² Turing arrived to Princeton for a two-year stay. Shortly before, in 1936, Alonzo Church founded *The Journal of Symbolic Logic*. Kurt Gödel, S. C. Kleene, and J. B. Rosser were all to be found in Princeton, New Jersey. The United States had become a world center for cutting-edge research in mathematical logic. See (Davis 1995).

We may compare a man in the process of computing a real number to a machine...

That was the point that Turing was to emphasize, in various forms, again and again. These words resonated with Wittgenstein’s words (italics in original):

These (Turing’s) machines are *humans* who calculate. (Wittgenstein 1980, 1096)

According to Shanker (1987), that remark summarized the key feature of Wittgenstein’s reaction to Turing.⁷³ Later Wittgenstein wrote (1994, p. 234):

If calculating looks to us like the action of a machine, it is the human being doing the calculation that is the machine.

Turing (1950, pp. 454–455) asserted that the essential constituents of computational procedure performed by a person during a calculation could be replicated by a machine. He believed that the true nature of human mind is mechanical. According to him:

The ‘skin of an onion’ analogy is also helpful. In considering the functions of the mind or the brain we find certain operations which we can express in purely mechanical terms. This we say does not correspond to the real mind: it is a sort of skin which we must strip off if we are to find the real mind. But then in what remains, we find a further skin to be stripped off, and so on. Proceeding in this way, do we ever come to the ‘real’ mind, or do we eventually come to the skin which has nothing in it? In the latter case, the whole mind is mechanical.

According to him it was easy to imagine that a person performing a computation could be replaced by a machine. He noted (1950):

The idea behind digital computers may be explained by saying that these machines are intended to carry out any operations which could be done by a human computer.

An effective procedure can be performed by an idealized, infinitely patient mathematician working with an unlimited supply of paper, pencils and time

⁷³ Turing attended a few lectures of Wittgenstein’s course on the foundations of mathematics. The status of a contradiction and the nature of a proof were the subject of disputes.

– but without insight. Turing’s paper *Computing Machinery and Intelligence* (1950) can be taken as saying that even a mathematician working with insight cannot exceed the power of the Turing machine. Turing experienced with construction of a computer.⁷⁴ He was much more than a mathematician, he was a specialist in electronics and signal processing even while the subjects were still unrecognized.

The Turing machine⁷⁵ was one of the simplest results. It was considered to be a formal counterpart to an effective procedure or an algorithm. Church (1937) highlighted that Turing:

proposes as a criterion that an infinite sequence of the digits 0 and 1 be ‘computable’ that it shall be possible to devise a computing machine, occupying a finite space and with working parts of finite size, which will write down the sequence to any desired number of terms if allowed to run for a sufficiently long time. As a matter of convenience, certain further restrictions are imposed on the character of the machine, but these are of such a nature as obviously to cause no loss of generality – in particular, a human calculator, provided with a pencil and paper and explicit instructions, can be regarded as a kind of Turing machine.

Hodges, the biographer of Turing, in the updating to (1983)⁷⁶ holds that Turing (AMT) “might well have been disappointed by the lack of interest in his work on the part of the mathematical world in general, but it is worth adding that Church was wholehearted in recommending and adopting AMT’s definition of computability. Given that AMT was a young unknown outsider crashing into Church’s field this was not something he could have taken for granted. As regards the tricky question of priority, Church wrote:”

In an appendix, the author [i.e. AMT] sketches a proof of the equivalence of ‘computability’ in his sense and ‘effective calculability; in the sense of the present author [i.e. Church’s definition using the lambda-calculus.] The author’s result concerning the existence of uncomputable sequences was also anticipated, in terms of effective calculability, in the cited paper [i.e. Church’s paper]. His work was, however, done independently...

⁷⁴ The first computer was named *Colossus* (1943). *Colossus* was a secret enterprise of the World War II located at Bletchley Park (‘Station X’). Max Newman and Bill Tutte were primarily responsible for its construction, while Turing was a consultant for programming. 11 *Colossus* machines were working on German codes. For more, refer to <http://www.codesandciphers.org.uk/lorenz/colossus.htm>

⁷⁵ By Turing the machine was named a logical computing machine, or “a-machine” (automatic machine). It has subsequently become known as the “Turing machine”. Turing used the term “machine”, while “Turing machine” was first used by Church (1937).

⁷⁶ See <http://www.turing.org.uk/book/update/part3.html>

In a letter (22 April 1937) to Church Bernays wrote:

He [Turing] seems to be very talented. His concept of computability is very suggestive and his proof of equivalence of this notion with your λ -definability gives a stronger conviction of the adequacy of these concepts for expressing the popular meaning of ‘effective calculability’.

In *Grundlagen der Mathematik* there are two references to Turing. In the first one we read (1970, p. 356):

Bei den Kriterien der Widerlegbarkeit, die wir aus dem Herbrandschen Satz entnommen haben, wurde der Allgemeinbegriff der berechenbaren Funktion benutzt.

For Gödel, who was unconvinced by Church’s paper, Turing’s proposal was very well-justified. He considered Turing’s work as a successful analysis of “mechanical procedure”. Gödel (1965, p. 2) stated unequivocally that Turing had analyzed “mechanical procedure” in a satisfactory way:

The most satisfactory way, in my opinion, is that of reducing the concept of finite procedure to that of a machine with a finite number of parts, as has been done by the British mathematician Turing.

For someone with as exacting standards of correctness as Gödel had, this was a high praise.

Gurevich (1995, p. 130) esteems Turing’s argumentation as very convincing. For him it is “a beautiful piece of speculative philosophy”. The idea of computability executed by the Turing machine is very fertile and has originated many ideas such as von Neumann’s classical computer. In his paper there appeared very fruitful notions of “input-output”, “memory”, “kompi-ler/interpreter”, “finite-state machine”, “coded program”, and “algorithm”. Turing’s definition of computability remains a classic paper in the elucidation of an abstract concept into a new paradigm. It means that the question 1 posed on page 101 is answered.

Alonzo Church’s lambda calculus and Steven Kleene’s recursive functions were arguably more elegant, but it was the mechanical action of Turing’s machines that most agreed intuitively about how people calculate. It also makes the Turing machines a natural object for studying even more powerful models of computation. Church (1937, pp. 42–43) reviewed Turing’s paper comparing the Turing machine to other concepts:⁷⁷

⁷⁷ Church omitted Post’s concept of binormality. For Post (1965, pp. 408, 419) himself

As a matter of fact, there is involved here the equivalence of three different notions: computability by a Turing machine, general recursiveness in the sense of Herbrand-Gödel-Kleene, and the λ -definability in the sense of Kleene and the present reviewer. Of these, the first has the advantage of making the identification with effectiveness in the ordinary (not explicitly defined) sense evident immediately – i.e., without the necessity of proving preliminary theorems. The second and third have the advantage of suitability for embodiment in a system of symbolic logic.

For Post (1944, p. 462), the new notion was worth studying:

But apart from the question of importance, these formalisms bring to mathematics a new and precise mathematical concept, that of the general recursive function of Herbrand-Gödel-Kleene, or its proved equivalents in the developments of Church and Turing. It is the purpose of this lecture to demonstrate by example that this concept admits of development into a mathematical theory much as the group concept has been developed into a theory of groups.

It quickly turned out that not only the proposals of Turing and Church but other formalisms for describing effective computability (register machines, Emil Post's systems, combinatory definability, Markov algorithms, formal grammars, μ -recursive functions) were functionally equivalent to each other. They differed only in how functions were to be computed. It was proved that all different models of computation defined the same class of computable functions. It means that the question 2 on page 101 is answered, too. Church and Turing have proved – roughly speaking – that there are such formulas of the first-order that are not recursive (Church) or computable by the Turing machine (Turing). Roughly speaking, it has been proved that for the predicate calculus there is no procedure as the truth table test for the propositional calculus. Neither of these theorems alone answer Hilbert's Entscheidungsproblem.

Let us recall here that the Entscheidungsproblem is the problem of an effective mechanical procedure that applies to a formula of the first-order language after a finite number of steps gives YES or NO as the answer to the question: is the given formula valid? To solve this problem, two facts have to be established:

it was only a working hypotheses:

[...] for full generality a complete analysis would have to be made of all the possible ways in which the human mind could set up finite processes for generating sequences. [...] we have to do with a certain activity of the human mind as situated in the universe. As activity, this logico-mathematical process has certain temporal properties; as situated in the universe it has certain spatial properties.

1. a precise formal notion(s) of effective procedure,
2. a thesis that links the notion(s) with the Entscheidungsproblem.

The notion of computability by the Turing machine fulfills all the conditions of correct explication (page 102). Thus it is necessary to discuss the fact 2. It should be stated that the notion of computability by the Turing machine (or its equivalent) is the notion of the effective procedure of the Entscheidungsproblem. In other words, the adequacy of computability by the Turing machine with the intuitive notion of effective method in Hilbert's decidability problem it should be stated. The fact is out of reach for any formal proof. It only may and should be justified.⁷⁸

The Church-Turing thesis⁷⁹ in its most common form states that every effective computation can be carried out by the Turing machine or – in other words – any computation solvable by a precisely stated set of instruction (i.e., an algorithm) can be run on the Turing machine (or a digital process computer). The thesis states that the Turing computability precisely captures the intuitive notion of computability. It may be paraphrased as saying that the notion of effective or mechanical method is captured by the Turing machine.⁸⁰ In *On Computable Numbers...* a different formulation of the thesis may be found, e.g.:

[T]he “computable numbers” include all numbers which would naturally be regarded as computable. (1936–37, p. 249)

It is my contention that these operations [the primitive operations of a Turing machine] include all those which are used in the computation of a number. (1936–37, p. 232)

⁷⁸ For a survey of the discussion and solutions of the question see (Murawski 2005).

⁷⁹ For the first time the question of adequacy, which is the subject of the thesis emerged in a talk between Church and Gödel in Princeton in 1934. It was formulated for the first time on a session of the American Mathematical Society by Church on 19 April, 1935 (Church 1936b, §7). Turing's mathematically equivalent formulation was published, together with the notion of the Turing machine, in (Turing, 1936–37). It was dubbed “thesis” by Kleene (1936, p. 232):

So Turing's and Church's theses are equivalent. We shall usually refer to them both as *Church's thesis*, or in connection with that one of its ... versions which deals with “Turing machines” as *the Church-Turing thesis*.

This terminology is used also in (Kleene, 1943) and (Kleene 1952, p. 317). Cf. (Murawski 2005).

⁸⁰ Strictly speaking, it is Turing's thesis. Church's thesis states the same about Church's concept of effective method. For more about the Church-Turing thesis and its evolution from its modest origins to its current elevated status see (Copeland 2005).

For J. Martin (2003, p. 353):

To say that the Turing machine is a general model of computation is simply to say that any algorithmic procedure that can be carried out at all (by a human, a team of humans, or a computer) can be carried out by a TM [Turing machine].

Turing proved that there were functions that could not be computed by the machine invented by him. It is clear when we consider that the set of functions from \mathbb{N} to \mathbb{N} is uncountable while the set of Turing machines is countable. Both Turing and Church showed the examples of functions which could not be computed. The most known and discussed of these is Turing's halting function which takes a pair: a natural number representing a Turing machine and its input, returning 1 if the machine halts on its input and 0 if it does not. From this, we also get the existence of a specific uncomputable set (the halting set): $\{n \mid n \text{ represents a Turing machine/input pair that halts}\}$.⁸¹

There is a strong connection between Gödel's and Turing's results. Gödel was looking for the type of model to represent formal systems. In 1936 he praised the Turing machine as allowing a "precise and unquestionably adequate definition of the general concept of [a] formal system" (1965). A formal system can be specified as Turing machine that semi-computes a set of formulas provable in this system. A formal system may be conceived as a recursively enumerable set of axioms with recursively enumerable rules of inference. Gödel's Incompleteness Theorem can therefore be completely specified, stating that no consistent formal system of this type can prove all truths of arithmetic, or, that the set of true formulas of arithmetic is not recursively enumerable. Gödel, in his Incompleteness Theorem, proved that each consistent formal system had its own unprovable statement. It was not excluded that the statement could be proved in another system. Turing (and Church) pointed out the 'absolutely' undecidable function whose values could be proven by no consistent formal system. Thus Turing's proof of the uncomputability of the halting function also extended Gödel's Incompleteness Theorem.

⁸¹ The halting function is partially computable. In general, a function, $f: X \rightarrow \{0, 1\}$ is semi-computable (recursively enumerable) by the Turing machines if and only if there is a Turing machine which transforms input, x , to $f(x)$ whenever $f(x) = 1$ and either returns 0 or diverges when $f(x) = 0$. A function, $f: X \rightarrow \{0, 1\}$ is co-semi-computable (co-recursively enumerable) by Turing machines if and only if there is a Turing machine which transforms input, x , to $f(x)$ whenever $f(x) = 0$ and either returns 1 or diverges when $f(x) = 1$.

The thesis and the theorems by Church and Turing should not be confused, though. Instead they should be taken together for they are relevant to the Entscheidungsproblem. The Church-Turing thesis is essential to prove that certain mathematical functions/sets/reals are uncomputable.

Let us remark that the Church-Turing thesis has been a subject for many speculations and in many cases it is misunderstood. Jack Copeland (1997) refers to many papers and books where the Church-Turing thesis is misstated.

The Church-Turing thesis tells about finitistic formalism, thus it does not pertain to formalism in general. The infinitistic methods are beyond the thesis. Hilbert aimed to restrict the methods of proof theory to finitistic ones. He allowed non-finitistic methods in mathematics in general. For example, by using transfinite induction Gentzen has proved the consistency of arithmetic, cf. page 92.

Because all the formal definitions of effective computability have been shown to describe essentially the same set of functions, it is now generally assumed that the Turing-Church thesis is correct.⁸²

Both the theorem, claiming that there are problems uncomputable by the Turing machine, and the thesis, stating that the notion of computability by the Turing machine is the correct accurate rendering of the effective procedure,⁸³ solve the Entscheidungsproblem in negative. Though Turing and Church discovered the result independently of one another, Turing's solution is more satisfying than that of Church's. It was acknowledged by Church himself.

The theorem of Turing (or Church) does not say that there are any unsolvable questions. The theorem only says that there is not an effective procedure (if we suppose that the notion of the effective procedure is exhausted by the concept of the Turing machine) that makes possible to solve any question. Thus we may hope that any given question may be solved (earlier or later). But we do not suppose that one method will be found that will solve any question. Gödel, who proved the incompleteness theorem, believed that in case of any formula we would be able to prove or disprove it. There is no effective procedure that works for any formula of the first-order calculus but it is not excluded that there are decidable classes of formulas,

⁸² Any formal description of the effective method is accompanied by a proper formulation of the thesis. See (Kleene 1943, Markov 1955, Boolos & Jeffrey 1980, Kalmár 1959, Kreisel 1970, Hofstadter 1979, Lewis & Papadimitriou 1981, Savage 1998, Wood 1987). About the arguments that are used to support the thesis and against it, see (Murawski 2005, 2004a, Cooper 2005).

⁸³ Cf. (Turing 1969, p. 7).

e.g. the one-place predicate calculus is decidable. It was also proved that there exist many classes of problems, few of which are very easily defined, which do not take an effective procedure to solve them.⁸⁴

The negative solution of the Entscheidungsproblem in conjunction with Gödel's theorems was seen as a sufficient reason for the abolishment of Hilbert's program in mathematics. Hilbert's program towards a decision procedure for all fields of mathematics was proved to be impossible. The Hilbertian dream remains in total ruins. The original hopes for Hilbert's programme lie in tatters. However, even if it is true, a vast amount was learned about the fundamental nature of computation. One of the greatest accomplishments of Turing's work is that he has answered Hilbert's question developing innovative ideas related to the development of a universal computing machine. Alan Turing's creation of a theoretical computing machine served as a theoretical framework for the modern computer. Before Turing three components – machine, program and data – were distinguished.

Turing's universal machine showed that the distinctness of these three categories is an illusion. (Davis 2000, p. 165)

Thus, it is the development of a system that allows of the integration and "fluidity" among these three components. It has become fundamental to the contemporary computer science, cf. (Davis 2000, p. 165).

Gödel, Church and Turing set limits to Hilbert's program, but there remains much of value in continuing Hilbert-like programs (e.g. nonstandard analysis). The development of computer science has led to a rebirth of Hilbert's proof theory, where its methods play a significant role. A modified Hilbert's program is a base for the development of proof theory, metamathematics, and decision theory (computability theory). The theory of algorithms (recursive functions) enables us to give the exact definition of the "formal system": a system is formal if and only if there is an algorithm for checking correctness of inferences in this system. In formal systems the standards of inferences must be described precisely enough to enable checking of proofs by a computer.

The consequences of Hilbert's program, unexpected by himself, are summarized by Chaitin (2004):

As I said, formal systems did not succeed for reasoning, but they succeeded wonderfully for computation. So Hilbert is the most incredible success in the world, but as technology, not as epistemology.

⁸⁴ For the most comprehensive treatment available in a book form of the classical decision problem of mathematical logic and of the role of the classical decision problem in modern computer science see (Börger et. al. 1997).

[...]

To end, let me quote from a posthumous collection of essays by Isaiah Berlin, *The Power of Ideas*, that was just published: "Over a hundred years ago, the German poet Heine warned the French not to underestimate the power of ideas: philosophical concepts nurtured in the stillness of a professor's study could destroy a civilization." So beware of ideas, I think it's really true. Hilbert's idea of going to the limit, of complete formalization, which was for epistemological reasons, this was a philosophical controversy about the foundations of mathematics – are there foundations? And in a way this project failed, as I've explained, because of the work of Gödel and Turing. But here we are with these complete formalizations which are computer programming languages, they're everywhere!

The period between the two World Wars was a remarkable time in philosophy and particularly in logic and the foundations of mathematics. Hilbert was the one to raise the most important and fruitful problems for the future of mathematics. Let us refer to the beginning of the text where Hilbert was called a man of problems (Hilbert 1932–1935, vol. 3, p. 405). Though simple, Hilbert's starting points were always important. David Hilbert died in 1943. He lived to see the end of the great mathematical dynasty at the Georg-August University of Göttingen. His funeral was attended by fewer than a dozen people, only two of whom were his fellow academics. He had never seen a computer. What is more, he never imagined how big the consequences of his program of finitistic foundation of mathematics were to be.

5. Beyond the Church-Turing Thesis

Church and Turing have formally defined a boundary to what is possible to calculate or compute algorithmically, but they did not necessarily define an ultimate boundary. Turing himself was the first person who had recognized the limitations of the Turing machine (Copeland & Proudfoot 1999). His choice machine (*c*-machine) (Turing 1936–37) and unorganized machine (*u*-machine) (Turing 1969) did not survive later scrutiny. In Turing's Ph.D. thesis (1939) the *a*-machine (the Turing machine) is augmented with an oracle, a function which determines the value of a function that cannot be computed by the Turing machine (*o*-machine).⁸⁵

⁸⁵ For Davis (2004):

It is perfectly plain in the context of Turing's dissertation, that *O*-machines were introduced simply to solve a specific technical problem about definability of sets of natural numbers. There is not the faintest hint that Turing was making a proposal about a machine to be built. ... It makes no sense to imagine that he was thinking about actual machines to compute the uncomputable. Cf. (Cooper 2005).

The Turing machine still leaves room for the speculation about a possibility of a quantum-mechanical, non-mechanical, or super-mechanical computation. Everybody who tries to cross the boundary set by the Church-Turing thesis may feel encouraged by Hilbert's words:

It is by the solution of problems that the investigator tests the temper of his steel; he finds new methods and new outlooks, and gains a wider and freer horizon. (Hilbert 1900)

For many years, attempts to find more powerful models than the Turing machines gave no positive results. The first efforts to weaken the Church-Turing thesis were unsuccessful. The discussion was focused on logical and mathematical problems. Laszlo Kalmár's argument is an example of one of the most known and widely discussed such attempts, cf. (Kalmár 1959, pp. 72, 79). The thesis was also questioned by Peter Rózsa (1959), Jean Porte (1960), G. Lee Bowie (1973), Elliot Mendelson (1963) and others. Several models were proposed to replace the Turing Machine as a universal model of computation. Under discussion there remain models that communicate with the outside world during their computations, models that appeal to the laws of physics, models that can manipulate real numbers, and parallel computers. Artificial neural networks, cellular automata, evolutionary computing, L-systems (multicellular organisms), swarm computing, molecular computing, and many others are still under considerations.

If we were to find out that the Church-Turing thesis was true (or false), it would not say anything about what is computable by mathematicians through insight or the types of computation that are achievable through natural processes.

Turing aimed to describe the process of calculation by an idealized mathematician. Here the "idealized" means that the mathematician does not take into account the meanings of things that are the subject of simple operations and that his action does not depend on his will, desire and emotions. It is supposed that any operation needs the same amount of time and that these operations form a finite sequence. In other words, the Turing machine is equivalent to the finitistic formalization in the Hilbertian sense. The processes of calculation that are performable by contemporary digital computers are of that kind.⁸⁶ The Church-Turing thesis does not say

⁸⁶ We do not take into account the fact that there is a difference between what is theoretical and that what is physical (practical). Even if the nature "calculates" mathematics as it is performed by the Turing machine, there could be a difference when a chance of some faults is not excluded. The Turing machine is a theoretical device, our computers are not, they are physical devices. Though operating as the Turing machine, a compu-

anything about how a human being thinks, though Turing maintained that his machine is as powerful as the human mind. Moreover, it says nothing about the formalization in general. Thus, the question whether Leibniz's and others' dream of the substitution of human thinking by a calculation is performable by the Turing machine is open as long as it is not proved (logically, psychologically, empirically?) that all (logically) sound reasonings are simulable by the Turing machine. It is a question of creation of artificial intelligence. There are two types of approaches to the task (one does not exclude another):

1. the development of software that would be able to perform those activities that are normally thought to require intelligence,
2. the construction of a machine having this ability.

Any of the approaches depends on the Turing-Church thesis. It is not excluded that to achieve our goal either of the ways the thesis has to be rejected.

Turing believed in the possibility of the construction of a true thinking machine. He generally shied away from metaphysical questions but not in case of that issue. Arguing that the creation of such a machine would be similar to the creation of a soul, he wrote (1950):

In attempting to construct such machines we should not be irreverently usurping his [God] power of creating souls, any more than we are in the procreation of children, [...] Rather we are, in either case, instruments of his will providing mansions for the souls that he creates.

The Church-Turing thesis is strongly influenced by the philosophical and scientific environment of the time it was formulated in.

The contemporary electronic computer is a great achievement of engineers. There are good reasons to believe that the progress in the construction of computers will not be restricted to the electronic scheme. Calculation that is performable by the contemporary computers harnesses the phenomenon of electricity. Are there any reasons to reject to harness other natural phenomena to perform calculation? The other natural processes should not be excluded as phenomena that are able to calculate. Nevertheless, one may repeat after Davis, see (Cooper 2005):

Of course, even assuming that all this really does correspond to the actual universe in which we live, there is still the question of whether an actual device to take advantage of this phenomenon is possible.

ter as the output may have data that differs from what is calculated by a (theoretical) Turing machine. Some natural processes could be such that the cause-result relation is non-recursively calculated. Such a process is not simulable by the Turing machine. Such a "computer" is able to do more than the Turing machine can.

Till now the problem of an interface⁸⁷ is not solved but it does not mean that the problem will not be solved in the future. Of course, a lot of ingenuity needs to be put into making natural phenomena compute more than now. We may as well agree with Davis: “But the theoretical question is certainly of interest.” To support this idea, Cooper (2005) gives an analogy:

The domestication of horses around five or six thousand years ago brought a revolution in transportation, only achieved through a creative interaction between humans and the natural world. At that time, trying to understand the principles underlying the equine organism in order to synthesise an artificial horse was unthinkable. But a few thousand years later there was enough understanding of scientific basics to underpin the invention of the “iron horse”, leading, amongst other things, to the opening up of many previously isolated parts of the world to people with no riding skills whatsoever.

Even if it is true that Turing machine does what an idealized mathematician does, it does not mean that the Turing machine calculates everything what is calculable. There are several examples of computations that cannot be performed by the Turing machine (Copeland 1998, Siegelmann 1995, *Hypercomputation* 2006, Burgin 1999, 2004b, Copeland 2002b, Copeland & Proudfoot 2005).

However, our way of thinking may be changed. We may say that to calculate means the same as to be (deterministically) processed by the Nature. Generally speaking, this kind of argumentation is based on the scheme: natural processes are effective; the notion of effectiveness of the Church-Turing thesis is not adequate, since the Turing machine is not able to simulate all of them. It is argued that there are effective processes of the physical world that are not restricted to those that meet a notion of “effectiveness” as being embodied in Turing-computability, these processes are not fully describable by algorithms that fulfil the Church-Turing thesis. It is unsound to argue the other way round: that one first formulates some definition of effectiveness and then argues backwards that the physical world ought to conform to it, e.g., Rosen (1999, p. 160)⁸⁸ maintains that:

⁸⁷ David Pearson (1996) comments on the manner in which we are accustomed to electronic computers:

[...] if we had a crystal-lattice computer today, we simply wouldn't know how to use it. Programming is so well-developed under the von Neumann model that is virtually impossible to remove it from our thinking. Obviously we can simply program a CA [cellular automaton] to emulate a standard von Neumann machine [...] that will certainly be done [...]. But that does not take advantage of the added power the CA has. To use it efficiently, we must be prepared to change our programming model.

⁸⁸ Robert Rosen was Professor Emeritus of Biophysics at Dalhousie University and the author of books including *Life Itself* (compiling twenty articles on the nature of life and on

Indeed, we shall take the view that material object-systems, as things in the external world, and whose behaviors are governed by webs of causal entailments, constitute the true province of effective processes. That is, the notion of effectiveness has to get imported into language via modeling relations arising in material nature, through encodings/decodings from these. Accordingly, any attempt to characterize effectiveness independent of these, from within language (e.g., mathematics) alone, and most particularly, in terms of syntax alone, is perilous. But that is exactly the substance of propositions such as Church's Thesis – namely, that causal entailments in the external world must conform to some notion of effectiveness drawn from inside the language itself. However, it is the other way around.

Are there any (deterministic) processes of the universe that are more powerful than the Turing machine? The answer to the question is not the subject of explication, it is rather a subject of empirical exploration of processes that are performable by the nature. Models of computation that compute more than the Turing machine are assertions about the nature of physics and their truth or falsity rests in the underlying structure of our universe rather than being claims in philosophy or mathematics.

The Church-Turing Thesis tells about the procedure of calculation carried out by a human being. It does not say anything about the “calculation” realized in the nature by physical or biological processes. The idea that the universe is a big Turing machine is interesting for many reasons. But the idea that the universe is more powerful than the Turing machine is even more interesting. There are some good reasons to believe in the nature hypercomputes, i.e. that nature computer exceeds the Turing machine not only in size.

The idea of supercalculation or supermachine is based on questioning some of the attributes of the Turing machine.⁸⁹ Different concepts of supercomputation are a result of combination of omitting various attributes of the Turing machine. Two types of supercomputation may be distinguished:

1. based on the difference between real or ideal human thinking and the Turing machine,
2. based on the difference between real computer and Turing machine or the idea that some (quantum- or biological)-processes “calculate” not only more efficiently but also differently than the Turing machine.

the objective of the natural sciences, this remarkable book complements Robert Rosen's groundbreaking *Life Itself* – a work that influenced a wide range of philosophers, biologists, linguists, and social scientists.), *Principles of Mathematical Biology, and Principles of Measurement*.

⁸⁹ For more on super- and hypercomputation see (Davis 2004, Copeland 2002a, Copeland 2002b, *Hypercomputation* 2006).

Under discussion there are machines with various expanded abilities, possibly with the ability to compute directly on real numbers, the ability to carry out uncountably many computations simultaneously, or the ability to carry out computations with exponentially higher complexity. Theoretical models such as probabilistic, oracle, and quantum computers are being studied.

The Turing machine performs one deterministic step at a time. The steps form a sequence and the time needed for execution is a linear function of their number. This model resembles, on a basic level, the von Neumann computers in use today. John von Neumann is one of the first scientists that considered cellular automata, too. Cellular automata are abstract mathematical models for computation that, unlike the Turing machines, operate in full parallelism. Neither the sequentiality of steps nor the constant time of operations seems to be an indispensable attribute of calculus. The calculation may be performed in parallelism and/or the time needed for a step may (exponentially) decrease with the number of steps. If each step is performed in half the time taken by the previous step, the computation will last twice as long as the first step (Copeland 1998, Copeland 2002a). A cellular automaton might operate by manufacturing a clone of itself and similarly it could increase its speed exponentially. Though parallel computers have been promised for many years, there are significant problems to overcome in their development. "Conventional machines have always overtaken the parallel computers in speed before these problems could be overcome."⁹⁰ Is a cellular automaton only a theoretical device or – as it is the case of the Turing machine – there are possible empirical devices (constructed or natural) that simulable it?⁹¹

At this stage, none of such devices seem physically plausible, and so hypercomputers are likely to remain a mathematical fiction. The question whether the hypercomputation is a proleptic computer science, or it is of mainly philosophical interest still seems to be open. In the next section we will speculate on the idea of hypercomputation based on quantum physics and molecular biology.⁹² Maybe a quantum- or DNA-hypercomputer

⁹⁰ See (Pearson 1996), cf. (Petrov 2005).

⁹¹ To be precise, let us point out that the Turing machine is also a cellular automaton; depending on the structure, a cellular automaton is n -dimensional. According to the Kuratowski (1930) theorem for planar graphs, the minimum number of spatial dimensions that is sufficient to represent any possible graph is equal to 3) it follows that any n -dimensional cellular automaton may be equivalently reduced to at most 3-dimensional one (the universe is a 3-dimensional cellular automaton). The Turing machine is an 1-dimensional cellular automaton. See (Petrov 2005).

⁹² Under discussion there are also solutions based on optical realms (Li, Pan & Zheng 1998) and chemical reactions (Siehs & Mayer 1999).

would be able not only to compute the Entscheidungsproblem.⁹³ The Church-Turing thesis is questioned from the angle of quantum mechanics and biology.

In 1982 Richard Feynman (1982)⁹⁴ remarked that the computer technology was unable to simulate quantum systems effectively. He showed that simulation of quantum mechanical system on the Turing machine caused exponential slowdown of operations.

In 1985 David Deutsch (1985) deliberated over a theoretical model of computer based on quantum mechanics. He suggested that such a computer would be able to calculate problems that were not calculable by a traditional computer. That idea found a larger interest in 1994 when Peter Shor (1994)⁹⁵ discovered a new quantum algorithm of factorization of big numbers.⁹⁶ The initial promises in the nineties towards quantum computation are now far from their realization. Since the time a little progress has been made.

At present the quantum computing is still the subject of interest of many scientists and institutes. Paul Davies maintains that:⁹⁷

The nineteenth century was known as the machine age, the twentieth century will go down in history as the information age. I believe the twenty-first century will be the quantum age.

Let us only note that almost all important papers in the field of quantum information and computation can be found at the on-line archive arxiv.org under the Quantum Physics (quant-ph) subject group. Among many institutes which encourage the growth and development of the emerging field of quantum information science there are:

⁹³ The fact that for the Turing machine the Entscheidungsproblem is solved in negative does not contradict to the possibility that for any formula of the first-order predicate logic there is a procedure of deciding whether this formula is valid (satisfiable) or not.

⁹⁴ A physicist, Nobel Prize laureate, one of the founders of modern quantum mechanics.

⁹⁵ Peter Shor, a research and computer scientist at AT&T's Bell Laboratories in New Jersey.

⁹⁶ In a quantum computer, the fundamental unit of information, a quantum bit or qubit, is not binary but rather more quaternary in nature. A qubit can exist not only in a state corresponding to the logical state 0 or 1 as in a classical bit, but also in states corresponding to a blend or simultaneously as both 0 and 1, with a numerical coefficient representing the probability for each state. Shor's algorithm harnesses the power of quantum superposition to rapidly factor very large numbers of ~ 10200 digits and greater in a matter of seconds.

⁹⁷ See foreword to (Milburn 1997).

1. Institute for Quantum Information, Caltech;⁹⁸
<http://www.iqi.caltech.edu/>
2. Institute for Quantum Computing, University of Waterloo;
<http://www.iqc.ca/>
3. Quanta, MIT;
<http://www.media.mit.edu/quanta/>
4. Centre for Quantum Computation, UK;
<http://www.qubit.org/>
5. Group of Applied Physics, Geneva;
<http://www.gap-optique.unige.ch/>

The site of Institute for Quantum Information *Caltech* present the following:

Quantum information science (QIS) is a new field of science and technology which draws upon the disciplines of physical science, mathematics, computer science, and engineering. Its aim is to understand how fundamental physical laws can be harnessed to dramatically improve the acquisition, transmission, and processing of information.

The inspiration for QIS is the discovery that quantum mechanics can be exploited to perform important and otherwise intractable information-processing tasks. Quantum effects have already been used to create fundamentally unbreakable cryptographic codes, to teleport the full quantum state of a photon, and to compute certain functions in fewer steps than any classical computer can.

Even aside from its technological implications, QIS is an intellectually stimulating basic research field. Fundamental questions such as "What is the computational power of Nature?", "Can measurement be reversed?" and "How much information can we learn?" continue to drive the field and inspire new research directions. We expect that QIS will have an extensive impact on how science is taught at the college and secondary level. We also expect that QIS paradigms will enable quantum physics to be understood better by a broad segment of the lay public. See: <http://www.iqi.caltech.edu/qis.html>

The idea of a biological computer was already considered by Turing. The advances both in molecular biology and in information science have stimulated the merging of two great discoveries of the 20th Century. In 1958 at the University of Illinois the Biological Computer Laboratory (BCL)

was founded by Heinz von Foerster.⁹⁹ "BCL left behind a rich legacy. In its day, it was one of the few education institutions teaching cybernetics. Between 1958 and 1975, operating under 25 grants, the laboratory produced 256 articles and books, 14 master's theses, and 28 doctoral dissertations. The topics covered epistemology, logic, neurophysiology, theory of computation, electronic music, and automated instruction. Most likely, the first parallel computers were built and exhibited there." (See <http://www.ece.uiuc.edu/pubs/centhist/six/bcl.htm>)

In 1994 Adelman (1994)¹⁰⁰ suggested that DNA (deoxyribonucleic acids) could be used to solve mathematical problems. According to him the Hamiltonian path problem (the traveling salesman problem) may be encoded in DNA sequences. Each city was encoded as its own DNA sequence. The DNA sequences were set to replicate and create trillions of new sequences based on the initial input sequences in a matter of seconds (called DNA hybridization).

There are many advantages of DNA computers. Input, output and "software" are all composed of DNA, the material of genes, while DNA-manipulating enzymes are used as "hardware". They have the potential to take computing to new levels, picking up where Moore's Law leaves off. DNA computers perform calculations parallel to other calculations. Hence they are able to solve complex problems in hours, whereas electrical computers might take hundreds of years to complete them. More than 10 trillion DNA molecules can fit into an area no larger than 1 cubic centimeter. With this, a DNA computer could hold 10 terabytes of data and perform 10 trillion calculations at a time. DNA computer is incredible energy-efficient resource – the DNA sequences are created by simply "just add water" to initiate the "computation" – and cheap – as long as there are cellular organisms, there will be a supply of DNA, biochips can be made cleanly. Ordinary computers need absolutely correct information, a biological computer will come to the correct answer based on partial information, by filling in the gaps itself. Biological computers may "think for itself" because neurons are able to form their own connections from one to another (silicon computers only make the connections they are told to by the programmer).

⁹⁹ Born 1911 in Vienna. From 1962 to 1975 he was Professor of Biophysics and 1958–75 director of the Biological Computer Laboratory. Together with Warren McCulloch, Norbert Wiener, John von Neumann, and others, Heinz von Foerster was the architect of cybernetics.

¹⁰⁰ Leonard Adelman is a mathematician and computer scientist, biologist and information scientist. He was one of the inventors of the RSA (Rivest, Shamir, Adelman) public-key encryption system.

⁹⁸ This institute is sponsored by the National Science Foundation (NSF) in order to "recognizes the importance of QIS [Quantum Information Science] to the scientific, economic, and technological strength of our nation."

The original version of the biomolecular computer capable of performing simple mathematical calculations, was introduced by Shapiro and colleagues (Weizmann Institute of Science, Israel: <http://www.weizmann.ac.il/>) in 2001. An improved system, which uses its input DNA molecule as its sole source of energy, was reported in 2003 and was listed as the smallest biological computing device. In Shapiro's opinion:

It is clear that the road to realizing our vision is a long one; it may take decades before such a system operating inside the human body becomes reality. Nevertheless, only two years ago we predicted that it would take another 10 years to reach the point we have reached today.¹⁰¹

In the foreseeable future DNA computers will not be replacing the common old PC. Studying DNA computers may also lead us to a better understanding of a more complex computer – the human brain. Teuscher (2004) writes about its knowledge:

People have started thinking about the possibility that simulating the mind in silicon might be impossible – or at least impossible using today's methods. Should we first forget about computers and look closer at the gray stuff in the brain, since the actual knowledge of the brain is severely fragmented and many details are still not at all understood?

Despite these new ideas of computers it is generally believed today that the halting function is uncomputable on any model of computation. Next section will be devoted to a short presentation of an argument based on Chaitin's concept of randomness and the number Ω . In his opinion (Chaitin 2004):

... not only Hilbert was wrong, as Gödel and Turing showed. ... I want to summarize all of this. With Gödel it looks surprising that you have incompleteness, that no finite set of axioms can contain all of mathematical truth. With Turing incompleteness seems much more natural. But with my approach, when you look at program size, I would say that it looks inevitable. Wherever you turn, you smash up against a stone wall and incompleteness hits you in the face!

¹⁰¹ Further information and photos can be obtained online at: www.weizmann.ac.il/udi/PressRoom

6. The number Ω

The question concerning the notion of effectiveness and boundaries of effective calculability has to be formulated as follows: are there any non-calculable facts, even if the calculability is conceived as large as possible in any reasonable sense? The answer to this question is given by Chaitin.¹⁰²

In order to choose among possible theories, which have the same predictions and the data available cannot distinguish between them, Ockham's razor is a very useful tool. Ockham's razor is the principle proposed by William of Ockham in the fourteenth century:

Pluralitas non est ponenda sine neccesitate

which translates as

entities should not be multiplied unnecessarily.

Ockham's razor directs us to study the simplest of the theories in depth. The principle prefers the less complex and more universal theory. If we have some competing theories that essentially explain one and the same phenomenon, we must choose the simplest and the most universal theory possible.

Our intuitions concerning the complexity of a number (a sequence of 0 and 1) are well explained by the algorithmic information theory.¹⁰³ The complexity of a number is associated with the size of the smallest program that generates it. The algorithmic information content (algorithmic entropy) of a number n , $H(n)$ is the size of the shortest program which produces the number n .¹⁰⁴

A theory can be conceived as a program that has a data describing experiment as an input and the results of the experiment are seen as the output. In 1964 Solomonoff proposed to take the length of the shortest of such programs as the measure of the complexity of a scientific theory. The task of a theory is the compression of data. For Chaitin, this theory is

¹⁰² Gregory J. Chaitin is an Argentine-American mathematician and computer scientist. Beginning in the late 1960s, Chaitin made important contributions to algorithmic information theory. His work paralleled the earlier work of Kolmogorov in many respects. For more about Chaitin see his home page: <http://www.cs.auckland.ac.nz/CDMTCS/chaitin/>

¹⁰³ This theory was founded by Andrei Kolmogorov (1965) and Gregory Chaitin (1966, 1987, 1997).

¹⁰⁴ To remember a number of e.g., a telephone, we try to find the simplest way of doing it. Such a practice is satirized by Jaroslav Hasek: On his attempt to remember a secret number of a locomotive, the brave soldier Švejk put efforts to elaborate a highly complicated rule how to do it rather than try to remember it itself.

better because it is more compressed. Any information may be presented as a number and the number is a sequence of 0 and 1. Sequences of 0 and 1 can be generated by a program. The complexity of a sequence of 0 and 1 may be measured by the length of the shortest of such programs. A random number is such a number for which the length of the shortest of the programs that it generates is almost the same as the number, i.e. the length of its minimal program approaches the length of the number itself. If information (a number) is not random and we have to send it to someone using as few bits as possible, we could send the program to generate it. In case of random information it does not matter whether we send the information or the program: in any case the number of the sent bits is comparable. While this definition of randomness of number as length of minimal programs or compressibility seems promising, it suffers practically since determining the minimal program for an arbitrary number is equivalent to solving the halting problem and is thus uncomputable by Turing machines. It is clear that almost all numbers are random. But it is not possible to give an example of a random number. No one is able to prove that a given number is a random number. If a program P has generated the number there is a program P_1 that is no longer than P and such that P_1 generates a number n , than the program P would generate the number n . The random numbers are definable but they are not calculable.¹⁰⁵ In any case, the fact that we do not have a program generating such a number when it is no longer the same number is not a sufficient reason to say that this number is a random number. It is not excluded that in the future somebody will find such a program. It is only possible to prove that a given number is not a random number.

For recursive infinite strings, such as the binary expansion of π , the algorithmic information content is simply the size of the smallest program generating π . For a non-recursive string such as the binary expansion of τ (the Turing's constant), there is no finite program, so the algorithmic information content is infinite. The number π is globally compressible: there are finite programs to generate it. By definition there is not such a finite program that would be able to generate τ . τ is not globally compressible. Nevertheless, the number is compressible locally: there is a program which on the basis of information consisting of n digits allows to calculate initially 2^n digits of the number.

¹⁰⁵ Chaitin's considerations are based on Berry's paradox. Bertrand Russell (1908, p. 223) put it as follows:

The least integer not nameable in fewer than nineteen syllables' is itself a name consisting of eighteen syllables; hence the least integer not nameable in fewer than nineteen syllables can be named in eighteen syllables, which is a contradiction.

In 1987 Gregory Chaitin discovered the number Ω (the 'halting probability'):

$$\Omega = \sum_{P \text{ halts}} 2^{-|P|}$$

Ω represented by a sequence of 0 and 1 so that for any place the probability of occurrence of 0 is $\frac{1}{2}$ and the same is true about 1. It is not possible to point out a rule that governs the succession of the digits. Any program, no matter which, if its complexity is n , can generate maximally only n elements of the sequence. It means Ω is incompressible globally as well as locally. Chaitin defines an infinite sequences s as random if and only if the information content of the initial segment, s_n , of length n eventually becomes and remains greater than n . By this definition, Ω is random while τ is not.

The number Ω can be described in terms of exponential diophantine equations with a parameter n . Thus, the question of Ω is moved into the domain of arithmetic. Chaitin has shown that in no consistent formal system of k bits in size, the proof whether or not the diophantine equation with a parameter n has infinitely many solutions for more than k different values of n . In predicting the information about the patterns of the solutions to the equation, no formal system can do better than chance. The incompleteness (Gödel) and undecidability (Turing) of formal system have been filled up by randomness by Chaitin. To conceive the issue, let us take as an example the question of defining of the set of prime numbers. It seems that there is no formula which in a compression form (the program that generates the formula should be significantly shorter than the program that produces the numbers) could "contain" only prime numbers. The set of prime numbers is random. If so, by the nature of randomness, the fact that the information about the set is not compressible could not be proved. The knowledge about the set of prime numbers is empirical/statistical by the nature of the prime numbers (if their set is random).

From the fact of randomness one can conclude that no matter which is the notion of effective procedure there are facts that could not be compressed. Moreover, such facts dominate over the facts that could be compressed. There are such numbers that no computer – biological, quantum or any other – is able to produce by a program that is essentially shorter than that number. There will always be facts that could not be calculated by whatever computer. For any program there is a finite number t that is the most complicated number that can be generated by that program. For any phenomenon there is a number that characterizes its complexity. For any theory/program there is a phenomenon when the number that characterizes

its complexity is greater than the number that could be generated by that program.

Ω is maximally unrecognizable. Ω marks the current boundary of what mathematics can achieve. It means that there are infinite number of facts that could not be inferred from the axioms of arithmetics. Gödel believed that the human mind is not able to mechanize all its intuitions. In 1989 a physicist Roger Penrose in *The Emperor's New Mind*¹⁰⁶ (1989) based on quantum mechanics argued that the human mind is able to go beyond the mechanical reasoning but no machine is able to do it. Chaitin reveals limits to what we can know.

Einstein said:

I want to know God's thoughts [...] I am not interested in this phenomenon or that phenomenon [...] I want to know God's thoughts – the rest are mere details.

To realize that dream he thought about theory of everything. It was his attempt to extend general relativity and unite the known forces in the universe. It was a project that hopefully would unlock the mind of God. As Hilbert's program was canceled by Gödel, Church and Turing, Einstein's idea was rejected by Heisenberg, Bohr and Schrödinger. They created quantum mechanics. A core element to their new interpretation of the world was that at a fundamental level, everything was unpredictable. For Einstein's

God does not play dice,

they replied:

Einstein, stop telling God what to do with his dice.

Hilbert believed in complete, consistent and decidable mathematics and Einstein believed in a theory of everything that in an elegant mathematical form describes the universe with absolute accuracy and predictability. Both Hilbert and Einstein were wrong. Mathematics as well as the world has a random structure. God plays dice both in mathematics and in the physical world.

Leibniz's words:

Sans les mathématiques on ne pénètre point au fond de la philosophie. Sans la philosophie on ne pénètre point au fond des mathématiques. Sans les deux on ne pénètre au fond de rien.

¹⁰⁶ The title is understandable to anybody who has read Andersen's tales.

may be translated/paraphrased as follows:

Without information science [mathematics] we cannot penetrate deeply into philosophy. Without philosophy we cannot penetrate deeply into information science [mathematics]. Without both we cannot penetrate deeply into anything.

References

- (1965), in M. Davis, ed., 'The Undecidable: Basic Papers on Undecidable Propositions, Unsolvability Problems, and Computable Functions', Raven Press, Hewlett, N.Y.
- (1971), in K. Berka & L. Kreiser, eds, 'Logik-Texte. Kommentierte Auswahl zur Geschichte der modernen Logik', Akademie-Verlag, Berlin. Zweite durchgesehene Auflage, 1973.
- (1992), in D. C. Ince, ed., 'Collected Works of A. M. Turing: Mechanical Intelligence', North Holland, Amsterdam.
- (2004a), in C. Teuscher, ed., 'Alan Turing: Life and Legacy of a Great Thinker', Springer-Verlag, New York.
- (2004b), in M. Burgin & A. Klinger, eds, 'Theoretical Computer Science, Special Issue on: Super-recursive algorithms and hypercomputation', Vol. 317.
- Ackerman, W. (1940), 'Zur Widerspruchsfreiheit der Zahlentheorie', *Mathematische Annalen* **117**, 162–194.
- Adleman, L. M. (1994), 'Molecular computation of solutions to combinatorial problems', *Science* **266**, 1021–1024.
- Akl, S. G. (2005), 'The myth of universal computation', in R. Trobec, P. Zinterhof, M. Vajteršić & A. Uhl, eds, 'Parallel Numerics', Vol. Part 2, Systems and Simulation, University of Salzburg, Austria and Jožef Stefan Institute, Ljubljana, Slovenia, Salzburg, Ljubljana, pp. 21–236. Technical Report No. 2005-492, School of Computing, Queen's University, Kingston, Ontario, Canada. <http://www.cs.queensu.ca/Parallel/technical.html#Technical>; also available: <http://pluton.ijs.si/marjan/parnum05/>.
- Becker, O. (1927), 'Mathematische existenz. zur logik und ontologie mathematischer phänomene', *Jahrbuch für Philosophie und phänomenologische Forschung* **8**, 441–809.

- Behmann, H. (1921), 'Das Entscheidungsproblem der mathematischen Logik', *Jahresberichte der Deutschen Mathematiker-Vereinigung*, 2. Abteilung **30**, 47.
- Behmann, H. (1922), 'Beiträge zur Algebra der Logik, insbesondere zum Entscheidungsproblem', *Mathematische Annalen* **86**, 163–229.
- Bernays, P. & Schönfinkel, M. (1928), 'Zum Entscheidungsproblem der mathematischen Logik', *Mathematische Annalen* **99**, 342–372.
- Blumenthal, O. (1935), David Hilbert: Lebengeschichte, in '(Hilbert 1932–1935)', pp. 388–429.
- Boole, G. (1847), *The mathematical analysis of logic: Being an essay towards a calculus of deductive reasoning*, Macmillan, Barclay, and Macmillan, Cambridge/George Bell, London. Reprinted Basil Blackwell, Oxford, 1951.
- Boolos, G. S. & Jeffrey, R. C. (1980), *Computability and Logic*, Cambridge University Press, Cambridge.
- Bowie, G. L. (1973), 'An argument against Church's thesis', *The Journal of Philosophy* **70**, 66–76.
- Börger, E., Grädel, E. & Gurevich, Y. (1997), *The Classical Decision Problem*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin, Heidelberg. Textbook edition: 2001.
- Browder, F. E., ed. (1976), *Mathematical Developments Arising from Hilbert Problems*, Vol. 28 (two parts) of *Proceedings of Symposia in Pure Mathematics*, American Mathematical Society, Providence.
- Burgin, M. (1999), Super-recursive algorithms as a tool for high performance computing, in 'Proceedings of the High Performance Computing Symposium', San Diego, pp. 224–228.
- Carnap, R. (1950), *Logical Foundations of Probability*, University of Chicago Press, Chicago.
- Chaitin, G. J. (1966), 'On the length of programs for computing finite binary sequences', *Journal of the Association of Computing Machinery* **13**, 547–569.
- Chaitin, G. J. (1987), *Algorithmic Information Theory*, Cambridge University Press, Cambridge.
- Chaitin, G. J. (1997), An invitation to algorithmic information theory, in 'DMTCS'96 Proceedings', Springer-Verlag, Singapore, pp. 1–23.
- Chaitin, G. J. (2004), 'Leibniz, randomness & the halting probability'. <http://www.cs.auckland.ac.nz/CDMTCS/chaitin/turing.html>.

- Church, A. (1932), 'A set of postulates for the foundation of logic', *Annals of Mathematics* **33**, 346–366. second series.
- Church, A. (1935), 'An unsolvable problem of elementary number theory', *Bulletin of the American Mathematical Society* **41**, 332–333. Preliminary Report (abstract). Praca nadesłana 22.III.1935.
- Church, A. (1936a), 'A note on the Entscheidungsproblem', *Journal of Symbolic Logic* **1**, 40–41, 356–361 (received April 15, 1936). Correction *ibid.* (1936) 101–102; received August 13, 1936. Praca nadesłana 13.VIII.1936. Przedruk w: (1965, 110–115).
- Church, A. (1936b), 'An unsolvable problem of elementary number theory', *American Journal of Mathematics* **58**, 345–363. Presented to the American Mathematical Society, April 19, 1935; abstract in *B. Am. Math. S.*, vol(41), May 1935. Reprinted in (1965, s. 89–107).
- Church, A. (1937), 'Reviews of (Turing 1936–37) and (Post 1936)', *Journal of Symbolic Logic* **2**(1), 42–43.
- Cooper, S. B. (2005), 'Definability as hypercomputational effect'.
- Copeland, B. J. (2002a), 'Accelerating Turing machines', *Minds and Machines* **12**(2), 281–300.
- Copeland, B. J. (2002b), 'Hypercomputation', *Minds and Machines* **12**(4), 461–502.
- Copeland, B. J. (2005), The Church-Turing Thesis, in 'Stanford Encyclopedia of Philosophy', <http://plato.stanford.edu/entries/church-turing/>.
- Copeland, B. J. & Proudfoot, D. (1999), 'Alan Turing's forgotten ideas in computer science', *Scientific American* **280**.
- Copeland, B. J. & Proudfoot, D. (2005), 'Introduction to hypercomputation: Computing the uncomputable'.
- Copeland, J. B. (1997), 'The broad conception of computation', *American Behavioural Scientist* **40**, 690–716.
- Copeland, J. B. (1998), 'Super Turing-machines', *Complexity* **4**, 30–32.
- Corbeil, M. (1997), 'Mathematics and logic', <http://www.mcorbeil.com>.
- Davis, M. (1973), 'Hilbert's tenth problem is unsolvable', *Amer. Math. Monthly* **80**, 233–269.
- Davis, M. (1982), 'Why Gödel didn't have Church' Thesis', *Information and Control* **54**, 3–24.
- Davis, M. (1995), 'American logic in the 1920s', *Bulletin of Symbolic Logic* **1**, 273–278.

- Davis, M. (2000), *The Universal Computer: The Road from Leibniz to Turing*, W. W. Norton & Company, New York.
- Davis, M. (2004), The myth of hypercomputation, in C. Teuscher, ed., 'Alan Turing: Life and legacy of a great thinker', Springer-Verlag, Berlin, Heidelberg.
- Davis, M., ed. (1994), *Solvability, Provability, Definability: The Collected Works of Emil L. Post*, Birkhäuser, Boston.
- Davis, M. & Hersh, R. (1973), 'Hilbert's 10th problem', *Sci. Amer.* **229**, 84–91.
- Davis, M., Matijasevic, Y. & Robinson, J. (1976), Hilbert's tenth problem: Diophantine equations; positive aspects of negative solution, in 'Proceedings of Symposia in Pure Mathematics', Vol. 18, pp. 323–78.
- Dawson, J. W. (1997), *Logical dilemmas: the life and work of Kurt Gödel*, Wellesley, Mass.
- Deutsch, D. (1985), 'Quantum theory, the Church-Turing principle and the universal quantum computers', *Proceedings of the Royal Society of London* **97**, 97–117.
- Deutsch, D. (1997), *The Fabric of Reality: The Science of Parallel Universes and Its Implications*, Penguin, New York.
- Du Bois-Reymond, E. (1912), Über die grenzen des naturerkennens, in E. D. Bois-Reymond, ed., 'Reden von Emil Du Bois-Reymond', Vol. 1, Leipzig, pp. 441–473.
- Ewald, W., ed. (1996), *From Kant to Hilbert: A Source Book in the Foundation of Logic*, Vol. 2, Clarendon Press, Oxford.
- Feynman, R. (1982), 'Simulating physics with computers', *International Journal of Theoretical Physics* **21**, 467.
- Frege, G. (1879), *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*, Halle. Introduction dated: 18 XII 1878. Reprinted in (Frege 1993). Shortened version in (1971). English translation in (Heijenoort 1967, 1–82).
- Frege, G. (1884), *Die Grundlagen der Arithmetik. Eine logisch-mathematische Untersuchung über den Begriff der Zahl*, W. Koebner, Breslau. Unveränderter Neudruck: M. & H. Marcus, Breslau, 1934. English translation: (Frege 1980a).
- Frege, G. (1976), *Wissenschaftlicher Briefwechsel*, Felix Meiner, Hamburg.
- Frege, Nachgelassene Schriften und wissenschaftlicher Briefwechsel, Bd. 1.

- Frege, G. (1980a), *The Foundations of Arithmetic: A logico-mathematical enquiry into the concept of number*, Northwestern University Press, Evanston, Illinois. (Frege 1884), trans. by J. L. Austin.
- Frege, G. (1980b), *Philosophical and Mathematical Correspondence*, University of Chicago Press, Chicago. The complete Frege-Hilbert Correspondence. The relevant parts of the correspondence are also printed in Gottlob Frege, *On the Foundations of Geometry and Formal Theories of Arithmetic*, (Yale University Press, New Haven, 1971). The translator's introduction is also useful here. Good commentaries include Michael Dummett's 'Frege on the consistency of mathematical theories' in his *Frege and Other Philosophers* (OUP, Oxford 1991), and Michael Resnik's 'The Frege-Hilbert Controversy', *Philosophy and Phenomenological Research* **34**, 3 (March 1974), 386–403.
- Frege, G. (1993), *Begriffsschrift und andere Aufsätze*, Hildesheim.
- Gödel, K. (1931), 'Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I', *Monatshefte für Mathematik und Physik* **38**, 173–198. Received: 17. XI. 1930. Translated in: Solomon Feferman (ed.), "Kurt Gödel: Collected Works", Oxford University Press, volume 1 (1986) 144–195. Includes the German text and parallel English translation. English translations in (1965), pp. 5–38; (Heijenoort 1967). Online version (translation by B. Meltzer, 1962): <http://www.ddc.net/ygg/etext/godel/> PDF version (translation by M. Hirzel): <http://nago.cs.colorado.edu/hirzel/papers/canon00-godel.pdf>.
- Gödel, K. (1934), On undecidable propositions of formal mathematical systems. (mimeographed lecture notes, taken by S. C. Kleene and J. B. Rosser at the Institute for Advanced Study), Princeton; Reprinted in: (1965), pp. 39–74.
- Gödel, K. (1965), Postscriptum, in '(1965)'.
- Gödel, K. (1995), A proof of Cantor's continuum hypothesis from a highly plausible axiom about orders of growth, in S. Feferman, ed., 'Kurt Gödel: Collected Works', Vol. 3, Oxford University Press, New York.
- Gentzen, G. (1936), 'Die Widerspruchsfreiheit der reinen Zahlentheorie', *Mathematische Annalen* **112**, 493–565. English translation in: (Gentzen 1969), pp. 132–213.
- Gentzen, G. (1938), 'Neue Fassung des Widerspruchsfreiheitsbeweises für die reine Zahlentheorie', *Forschungen zur Logik und zur Grundlagen der exakten Wissenschaften, new series* **4**, 19–44.

- Gentzen, G. (1969), *The Collected Papers of Gerhard Gentzen*, North-Holland Publishing Company, Amsterdam.
- Gentzen, G. (1974), 'Der erste Widerspruchsfreiheitsbeweis für die klassische Zahlentheorie', *Archiv für mathematische Logik und Grundlagenforschungen* **16**, 97–112.
- Gödel, K. (1930a), 'Die Vollständigkeit der Axiome des logischen Funktionskalküls', *Monatshefte für Mathematik und Physik* **37**, 349–360. (Gödel's Dissertation, 1929) also with English transl. in *Collected Works*, S. Feferman et al., eds., vol. 1, Oxford University Press, Oxford, 1986, pp. 60–101.
- Gödel, K. (1930b), 'Nachtrag zu der diskussion zur grundlegung der mathematik am sonntag, dem 7. sept. 1930', *Erkenntnis* **2**, 149–151.
- Gödel, K. (1995), Undecidable diophantine propositions, in S. Feferman, ed., 'Collected Works', Vol. III, Oxford University Press, Oxford. Reprinted in *Collected Works*, vol. III.
- Goldstine, H. H. (1972), *The Computer from Pascal to von Neumann*, Princeton University Press, Princeton, New Jersey.
- Graham, R. L., Knuth, D. E. & Patashnik, O. (1989), *Concrete Mathematics*, Addison-Wesley, Reading MA.
- Gray, J. (2000a), *The Hilbert Challenge*, Oxford University Press, Oxford.
- Gray, J. (2000b), The Hilbert problems 1900–2000. <http://www.mathematik.uni-bielefeld.de/~kersten/hilbert/gray.html>.
- Gurevich, Y. (1995), 'The logic in computer science column', *Bulletin of European Assoc. for Theor. Computer Sci.* **1**, 129–135.
- Hahn et al., H. (1930), 'Diskussion zur Grundlegung der Mathematik am Sonntag, dem 7. Sept. 1930', *Erkenntnis* **2**, 135–149. Note both the Vorbemerkung of the editors on the page before the content and Gödel's supplement (Gödel 1995).
- Halsted, G. B. (1900), 'The international congress of mathematicians', *American Mathematical Monthly* **7**, 188–189.
- Heijenoort, J. v., ed. (1967), *From Frege to Gödel. A Source Book in Mathematical Logic 1879–1931*, 1 edn, Harvard University Press, Cambridge. 2nd ed., 1971.
- Herbrand, J. (1932), 'Sur la non-contradiction de l'arithmétique', *Journal für die reine und angewandte Mathematik* **166**, 1–8.

- Hilbert, D. (1897), 'Die Theorie der algebraischen Zahlkörper', *Jahresbericht der Deutschen Mathematiker-Vereinigung* **4**, 175–546. re-edited in (Hilbert 1932–1935), vol. 1, pp. 63–363; English trans. (Hilbert 1998).
- Hilbert, D. (1899), *Grundlagen der Geometrie*, Teubner-Verlag, Leipzig. 8th amended ed. by P. Bernays with supplements, Stuttgart, 1956–1987; 14th ed., with contributions by H. Kiechle et al., M. Toepell, ed., Stuttgart, 1999; the first English translation *The Foundations of Geometry* by E. J. Townsend, Open Court, Chicago, 1902; new translation by L. Unger, Open Court, La Salle, 1972. [The first German edition of the *Grundlagen* was a contribution to the Festschrift zur Enthüllung des Gauss-Weber Denkmals in Göttingen, pp. 1–92; later editions were published separately].
- Hilbert, D. (1900), 'Mathematische Probleme', *Nachrichten von der Königlich-Preussischen Gesellschaft der Wissenschaften zu Göttingen, Math.-Phys. Klasse* (3), 253–297. Lecture held at the Paris ICM 1900. Revised version in (Hilbert 1901b) and (Hilbert 1932–1935), pp. 290–329; English translation in (Hilbert 1901a); also in (Browder 1976), pp. 1–34; (Yandell 2002), pp. 324–389; (Gray 2000a), pp. 240–282; selected parts in (Ewald 1996), pp. 1096–1105; (Reid 1970), chap. 9; French translations by M. L. Laugel (with corrections and additions) in *Compte Rendu Deuxième Congrès International des Mathématiciens tenu à Paris du 6 au 12 août 1900*, E. Duporcq, ed., Gauthier-Villars, Paris, 1902, pp. 58–114; (fragments) *L'Enseignement Mathématique* 2 (1900) 349–354 and *Revue Générale des Sciences Pures et Appliquées* 12 (1901) 168–174; annotated Russian edition, *Problemy Gilberta*, P. S. Alexandrov, ed., Nauka, Moscow, 1969; its German edition *Die Hilbertschen Probleme*, Akademische Verlagsgesellschaft, Leipzig, 1976, 2nd ed. 1979, reprint 1998 (both the Russian and German editions give the revised version of (Hilbert 1932–1935)., David Joyce, Clark University, produced a list of Hilbert's problems and a web version of Hilbert's 1900 address in March 1997 – <http://aleph0.clarku.edu/djoyce/hilbert/problems.html>.
- Hilbert, D. (1901a), 'Mathematical problems', *Bulletin of the American Mathematical Society* **8**, 437–479. Lecture delivered before the International Congress of Mathematicians at Paris in 1900, translated by Mary Winton Newson. Reprinted in the *Bulletin* 37 (2000) 407–436.
- Hilbert, D. (1901b), 'Mathematische Probleme', *Archiv der Mathematik und Physik* **1**, 44–63; 213–237.

- Hilbert, D. (1905), *Logische Principien des mathematischen Denkens*. Vorlesung, Sommer-Semester 1905. Lecture notes by Ernst Hellinger.
- Hilbert, D. (1918), 'Axiomatisches Denken', *Mathematische Annalen* **78**, 405–415.
- Hilbert, D. (1922a), *Grundlagen der Mathematik*. Vorlesung, Winter-Semester 1921–22. Lecture notes by Helmut Kneser.
- Hilbert, D. (1922b), *Neubegründung der Mathematik: Erste Mitteilung*, Technical Report 1, *Abhandlungen aus dem Seminar der Hamburgischen Universität*. (Series of talks given at the University of Hamburg, July 25–27, 1921. Reprinted with notes by Bernays in (Hilbert 1932–1935), pp. 157–177. English translation in (Mancosu 1998), pp. 198–214 and (Ewald 1996), vol. 2, pp. 1115–1134.
- Hilbert, D. (1925), 'Über das Unendliche', *Mathematische Annalen* **95**, 161–190. Lecture given in Münster, 4 June 1925. Also in (Hilbert 1899), 7th ed., pp. 262–288, shortened version in *Jahresber. Deutsch. Math. Verein.* **36** (1927) 201–215; English translation (by S. Bauer-Mengelberg) in (Heijenoort 1967), pp. 367–392; 2nd ed., 1971; French translation (by A. Weil), *Sur l'infinie*, *Acta Math.* **48** (1926) 91–122.
- Hilbert, D. (1928), 'Die Grundlagen der Mathematik', *Abhandlungen aus dem Seminar der Hamburgischen Universität* **6**, 65–85. followed by *Diskussionsbemerkungen zu dem zweiten Hilbertschen Vortrag* by H. Weyl, pp. 86–88, and *Zusatz zu Hilberts Vortrag* by P. Bernays, pp. 89–95; shortened version in (Hilbert 1998), 7th ed., pp. 289–312; English translation (by S. Bauer-Mengelberg and D. Follesdal) in (Heijenoort 1967), pp. 464–479.
- Hilbert, D. (1929), *Probleme der Grundlegung der Mathematik*, in N. Zanchelli, ed., 'Atti del Congresso Internazionale dei Matematici. 3–10 September 1928', Bologna, pp. 135–141. also in *Math. Ann.* **102** (1930) 1–9, (Hilbert 1899), 7th ed., pp. 313–323, and (Hilbert 1971), pp. 9–19.
- Hilbert, D. (1930), 'Naturerkennen und Logik', *Naturwissenschaften* pp. 959–963. An address given by David Hilbert at the meeting of the Society of German Scientists and Physicians in Königsberg in fall 1930, on the occasion of presentation to Hilbert (upon his retirement) of an honorary citizenship of the town. English translations: (Hilbert 1932–1935), pp. 378–387; (Ewald 1996) vol. 2, pp. 1157–1165; selected parts in (Reid 1970), chap. 12; English translation of the corresponding radio

- broadcast in (Vinnikov 1999); see also: math.sfsu.edu/smith/Documents/HilbertRadio/HilbertRadio.pdf.
- Hilbert, D. (1931), 'Die Grundlegung der elementaren Zahlenlehre', *Mathematische Annalen* **104**, 485–94. Shortened version in (Hilbert 1932–1935), pp. 192–195. English translation in (Ewald 1996), pp. 1148–1157.
- Hilbert, D. (1932–1935), *Gesammelte Abhandlungen*, Vol. 3, 2nd ed., 1970 edn, Springer-Verlag, Berlin.
- Hilbert, D. (1971), *Gedenkband*, Springer, Berlin, Heidelberg, New York.
- Hilbert, D. (1992), *Natur und mathematisches Erkennen*, Birkhäuser, Basel. Vorlesungen, 1919–20.
- Hilbert, D. (1998), *The Theory of Algebraic Number Fields*, Springer Verlag, Berlin.
- Hilbert, D. (n.d.), *Mathematische Notizbücher*. 3 notebooks.
- Hilbert, D. & Ackerman, W. (1928), *Grundzüge der theoretischen Logik*, Springer, Berlin. 6th ed., 1972; English translation of the second edition *Principles of Mathematical Logic*, by L. M. Hammond et al., Chelsea, New York, 1950; Chinese translation Science Press, Beijing, 1950; Russian translation Nauka, Moscow, 1979.
- Hilbert, D. & Bernays, P. (1970), *Grundlagen der Mathematik*, Vol. 2, 2 edn, Springer-Verlag, Berlin.
- Hodges, A. (1983), *Alan Turing: The Enigma*, Simon and Schuster, New York. Update: <http://www.turing.org.uk/book/update/index.html>.
- Hodges, A. (1992), *Alan Turing: The Enigma*, Vintage, London, Melbourne, Johannesburg.
- Hofstadter, D. R. (1979), *Gödel, Escher, Bach: An Eternal Golden Braid*, Basic Books, New York.
- Hypercomputation* (2006), Wikipedia.
- Kalmár, L. (1959), An argument against the plausibility of Church's Thesis, in A. Heyting, ed., 'Constructivity in Mathematics. Proceedings of the Colloquium Held at Amsterdam 1957', North-Holland Publ. Co., Amsterdam.
- Kambartel, F. (1975), Frege und die axiomatische Methode. Zur Kritik mathematikhistorischer Legitimationsversuche der formalistischen Ideologie, in C. Thiel, ed., 'Frege und die moderne Grundlagenforschung', Hain, Meisenheim a. Glan, pp. 77–89.

- Kleene, S. C. (1935), 'A theory of positive integers in formal logic', *American Journal of Mathematics* **57**, 153–173, 219–244.
- Kleene, S. C. (1936), 'Lambda-definability and recursiveness', *Duke Mathematical Journal* **2**, 340–353.
- Kleene, S. C. (1943), 'Recursive predicates and quantifiers', *Transactions of the American Mathematical Society* **53**(1), 47–73. reprinted in Davis (1964), pp. 254–287.
- Kleene, S. C. (1952), *Introduction To Metamathematics*, North-Holland Publishing Co, Amsterdam and P. Noordhoff N. V., Groningen.
- Kleene, S. C. (1967), *Mathematical Logic*, Wiley, New York.
- Kneale, W. & Kneale, M. (1962), *The Development of Logic*, Clarendon Press, Oxford. repr. 1986.
- Künzel, W. & Heiko, C. (1991), *Die Ars Generalis Ultima des Raymundus Lullus. Studien zu einem geheimen Ursprung der Computertheorie*, 5 edn, Edition Olivia Künzel, Berlin.
- Kolmogorov, A. N. (1965), 'Three approaches to the quantitative definition of information', *Problems Information Transmission* **1**, 1–7.
- Krajewski, S. (2003), *Twierdzenie Gödla i jego interpretacje filozoficzne: od mechaniczmu do postmodernizmu*, Wydaw. Instytutu Filozofii i Socjologii PAN, Warszawa.
- Kreisel, G. (1970), Church's thesis: a kind of reducibility axiom for constructive mathematics, in K. A. Myhill J. & V. R.E., eds, 'Intuitionism and Proof Theory', North-Holland Publ. Co., Amsterdam, pp. 121–150.
- Kuratowski, K. (1930), 'Sur le probleme des courbes gauches en topologie', *Fund. Math.* **15**, 217–283.
- Kurzweil, R. (1992), *The Age of Spiritual Machines: When Computers Exceed Human Intelligence*, The MIT Press, Chicago.
- Leibniz, G. W. (1846–1863), *Dialogus*, in C. I. Gerhardt, ed., 'Die Philosophischen Schriften von G. W. Leibniz', Vol. 7, Halle. Reprint: Hildesheim 1960.
- Leibniz, G. W. (1890), *Philosophische Schriften*, Vol. VII, Berlin.
- Leibniz, G. W. (1923), *Dissertatio de arte combinatoria*, in 'Sämtliche Schriften und Briefe', Akademie Verlag. A VI 1, p. 163.
- Leśniewski, S. (1931), 'Über Definitionen in der sogenannten Theorie der Deduktion', *Comptes Rendus des Séances de la Société des Sciences et des Lettres de Varsovie. Class III* **24**, 289–309.

- Lewis, H. R. & Papadimitriou, C. H. (1981), *Elements of the Theory of Computation*, Prentice Hall, Englewood Cliffs, New Jersey.
- Li, K., Pan, Y. & Zheng, S.-Q., eds (1998), *Parallel Computing Using Optical Interconnections*, Kluwer Academic Publishers, Dordrecht.
- Lucas, J. R. (1961), 'Minds, Machines and Gödel', *Philosophy* **36**, 112–127. Reprinted in *The Modeling of Mind*, Kenneth M. Sayre and Frederick J. Crosson, eds., Notre Dame Press, 1963, pp. 269–270; and *Minds and Machines*, ed. Alan Ross Anderson, Prentice-Hall, 1954, pp. 43–59. On line version: <http://users.ox.ac.uk/~jrlucas/mmg.html>.
- Löwenheim, L. (1915), 'Über Möglichkeiten im Relativkalkül', *Math. Ann.* **76**, 447–479.
- Mancosu, P., ed. (1998), *From Brouwer to Hilbert. The Debate on the Foundations of Mathematics in the 1920s*, Oxford University Press, Oxford.
- Marciszewski, W. (2003), 'On mechanization of reasoning, decidability of logic, and uncomputable numbers', *Studies in Logic, Grammar and Rhetoric* **6**(19), 79–93.
- Marciszewski, W. (2005), 'Text commented: B. Jack Copeland's "The Church-Turing Thesis"'.
<http://www.mimuw.edu.pl/~mimuw/arciszewski/ctt.html>
- Marciszewski, W. & Murawski, R. (1995), *Mechanization of Reasoning in a Historical Perspective*, Edition Rodopi, Amsterdam-Atlanta, GA.
- Markov, A. A. (1955), *Teorya algorifmov*, Vol. 42, Trudy Matem. Inst. AN SSSR im. W. A. Steklova, Moscow. Przekład angielski: *The Theory of Algorithms*, Office of Technical Services, U.S. Department of Commerce, Washington, D.C., 1962.
- Martin, J. (2003), *Introduction to Languages and the Theory of Computation*, 3 edn, McGraw-Hill, New York.
- Matiyasevich, Y. V. (1970), 'Solution of the Tenth Problem of Hilbert', *Mat. Lapok* **21**, 83–87.
- Matiyasevich, Y. V. (1993), *Hilbert's Tenth Problem*, MIT Press, Cambridge, MA. <http://www.informatik.uni-stuttgart.de/ifi/ti/personen/Matiyasevich/H10Pbook/>.
- Mendelson, E. (1963), 'On some recent criticism of Church's Thesis', *Notre Dame Journal of Formal Logic* **4**, 201–205.
- Mendelson, E. (1990), 'Second thoughts about Church's thesis and mathematical proofs', *The Journal of Philosophy* **87**, 225–233.

- Milburn, G. J. (1997), *Schrödinger's Machines: The Quantum Technology Reshaping Everyday Life*, W. H. Freeman, New York. Foreword by Paul Davies.
- Minkowski, H. (1973), *Briefe an David Hilbert*, Springer-Verlag, Berlin. The 105 letters are preserved in the Niedersächsische Staats- und Universitätsbibliothek Göttingen, Handschriftenabteilung, Nachlass Hilbert, Cod. Ms. D. Hilbert 258.
- Müller, T. (2001), Arthur Priors Zeitlogik. Eine problemorientierte Darstellung, Phd, Albert-Ludwig-Universität zu Freiburg i. Br., Freiburg i. Br.
- Murawski, R. (1999), *Recursive Functions and Metamathematics. Problems of Completeness and Decidability, Gödel's Theorems*, Kluwer Academic Publishers, Dordrecht-Boston-London.
- Murawski, R. (2005), Kilka uwag metodologicznych na temat tezy Churcha, in K. Trzęsicki, ed., 'Ratione et studio', Białystok, pp. 235–249.
- Norton, J. D. (2000), 'Nature is the realisation of the simplest conceivable mathematical ideas': Einstein and the canon of mathematical simplicity', *Studies in the History of Modern Physics* **31**, 135–170.
- O'Connor, J.J. & Robertson, E.F. (2005), 'Ramon Llull'. <http://www-groups.dcs.st-and.ac.uk/history/Mathematicians/Llull.html>.
- Pearson, D. (1996), Programming a crystal-lattice computer, in M. B. T. Toffoli & J. Leao, eds, 'Proceedings of the Fourth Workshop on Physics and Computation, PhysComp96', New England Complex Systems Institute, pp. 270–274. Online in the InterJournal: http://www.interjournal.org/cgi-bin/manuscript_abstract.cgi?540078425.
- Peckhaus, V. (1994), Leibniz als Identifikationsfigur der britischen Logiker des 19. Jahrhunderts, in 'VI. Internationaler Leibniz-Kongreß. Vorträge. I. Teil, Hannover, 18.–22.7.1994', Gottfried-Wilhelm-Leibniz-Gesellschaft, Hannover, pp. 589–596.
- Peckhaus, V. (1997), *Logik, Mathesis universalis und allgemeine Wissenschaft. Leibniz und die Wiederentdeckung der formalen Logik im 19. Jahrhundert*, Akademie Verlag, Berlin.
- Peckhaus, V. (1999), '19th century logic between philosophy and mathematics', *The Bulletin of Symbolic Logic* **5**(4), 433–450.
- Penrose, R. (1989), *The Emperor's New Mind. Concerning Computers, Minds, and The Laws of Physics*, Oxford University Press, Oxford, New York, Melbourne.

- Penrose, R. (1994), *Shadows of the Mind. A Search for the Missing Science of Consciousness.*, Oxford University Press, Oxford. 2nd ed. 1996.
- Petrov, P. (2005), 'The Church-Turing Thesis as an immature form of the Zuse-Fredkin Thesis (More arguments in support of the "Universe as a Cellular Automaton" idea)', <http://digitalphysics.org/Publications/Petrov/Pet02a2/Pet02a2.htm>.
- Petrov, P. (August 30, 2003), Church-Turing Thesis is almost equivalent to Zuse-Fredkin Thesis (An argument in support of Zuse-Fredkin Thesis), in '3rd WSEAS International Conference on Systems Theory and Scientific Computation, Special Session on Cellular Automata and Applications, to be held on 15–17 November 2003 at Rhodes Island, Greece'. This article is a part of (Petrov n.d.a). Available online at: <http://digitalphysics.org/Publications/Petrov/Pet02a1/Pet02a1.htm>.
- Petrov, P. (n.d.a), 'Series of brief articles about digital physics (dp)'. In preparation. For the latest update, check out: <http://digitalphysics.org/Publications/Petrov/DPSeries/>.
- Petrov, P. (n.d.b), 'The Zuse-Fredkin Thesis is the ultimate form of the Church-Turing Thesis'. In preparation.
- Poincaré, H. (1914), *Wissenschaft und Methode*, B. G. Teubner, Leipzig/Berlin. Original edition: *Science et méthode*, Flammarion: Paris 1908.
- Porte, J. (1960), Quelques pseudo-paradoxes de la 'calculabilité effective', in '2^e Congrès Internationale de Cybernétique, Namur, 3–10 Septembre 1958, Actes', Namur, pp. 332–334.
- Post, E. (1936), 'Finite combinatory processes – Formulation I', *Journal of Symbolic Logic* **1**, 103–105. received Oct 7, 1936. Presented to the American Mathematical Society Jan 1937. Abstract in B. Am. Math. S., Nov 1936. Reprinted in (Davis 1994), pp. 289–291.
- Post, E. (1944), 'Recursively enumerable sets of positive integers and their decision problems', *Bulletin of the American Mathematical Society* **50**, 284–316. Reprinted in Davis (1994) (Davis 1994), pp. 305–337.
- Post, E. (1965), Absolutely unsolvable problems and relatively undecidable propositions – account of an anticipation, in M. Davies, ed., 'The Undecidable: Basic Papers on Undecidable Propositions, Unsolvability Problems, and Computable Functions', Raven Press, Hewlett, N.Y., pp. 340–433.
- Reid, C. (1970), *Hilbert*, Springer-Verlag, New York. reprinted Copernicus, New York, 1996.

- Resnik, M. D. (1974), 'The Frege-Hilbert controversy', *Philosophy and Phenomenological Research* **34**, 386–403.
- Resnik, M. D. (1980), *Frege and the Philosophy of Mathematics*, Cornell University Press, Ithaca.
- Rosen, R. (1998), *Essays on Life Itself*, Columbia University Press, Columbia.
- Róza, P. (1959), Rekursivität und konstruktivität, in H. A., ed., 'Constructivity in Mathematics. Proceedings of the Colloquium Held at Amsterdam 1957', North-Holland Publ. Co., Amsterdam, pp. 226–233.
- Russell, B. (1908), 'Mathematical logic is based on the theory of types', *Am. J. of Mathematics* **30**.
- Russell, B. & Whitehead, A. N. (1910, 1912 and 1913), *Principia Mathematica*, Vol. 1–3, Cambridge University Press, Cambridge. Second edition, 1925 (Vol. 1), 1927 (Vols 2, 3).
- Savage, J. E. (1998), *Models of Computation*, Addison-Wesley, Reading, Massachusetts.
- Schröder, E. (1895), *Vorlesungen über die Algebra der Logik (Exakte Logik)*, Vol. 3, part I, Leipzig. Part I: Algebra und Logik der Relative.
- Shanker, S. (1987), 'Wittgenstein versus Turing on the nature of Church's Thesis', *Notre Dame Journal of Formal Logic* **28**(4).
- Shies, C. & Mayer, B. (1999), 'Dynamical hierarchies of structure and control in chemical reaction networks', *Nanotechnology* **10**, 464–471.
- Shoenfield, J. R. (1991), *Recursion Theory*, Vol. 1 of *Lecture Notes In Logic*, Springer-Verlag, Heidelberg, New York.
- Shor, P. W. (1994), Algorithms for quantum computation: discrete logarithm and factoring, in S. Goldwasser, ed., 'Proc. 35th A. Symp. on the Foundations of Computer Science', IEEE, Los Alamitos, CA, p. 124.
- Siegelmann, H. T. (1995), 'Computation beyond the turing limit', *Science* **268**(5210), 545–548.
- Stern, N. (1981), *From ENIAC to UNIVAC: An Appraisal of the Eckert-Mauchly Computers*, Digital Press, Bedford, Massachusetts.
- Teuscher, C. (2004), Turing's connectionism, in C. Teuscher, ed., 'Alan Turing: Life and legacy of a great thinker', Springer-Verlag, Berlin, Heidelberg, pp. 499–529.
- The Clay Mathematics Institute (http://www.claymath.org/prize_problems/html), Millennium Prize Problems. Collège de France, Paris.

- Thiele, R. (2003), 'Hilbert's twenty-fourth problem', *American Mathematical Monthly* **110**, 1–24.
- Toepell, I. (1986), *Über die Entstehung von David Hilbert „Grundlagen der Geometrie“*, Vandenhoeck & Ruprecht, Göttingen.
- Trzęsicki, K. (2000), *Logika. Nauka i sztuka*, 2 edn, WSFiZ, Białystok.
- Turing, A. M. (1936–37), 'On computable numbers, with an application to the Entscheidungsproblem', *Proceedings of the London Mathematical Society* **42** (Series 2), 230–265. Received May 25, 1936; Appendix added August 28; read November 12, 1936; corrections Ibid. vol. 43(1937), pp. 544–546. Turing's paper appeared in Part 2 of vol. 42 which was issued in December 1936 (Reprint in M. Davis (ed.) 1965, pp. 116–151, (1965); corr. ibid. pp. 151–154). Online version: <http://www.abelard.org/turpap2/tp2-ie.asp>.
- Turing, A. M. (1939), Systems of logic based on ordinals, in 'Proceedings of the London Mathematical Society', Vol. 45 of 2, pp. 161–228. reprinted in M. Davis (1965) (1965), pp. 155–222; publication based on Ph.D. thesis of 1938 (supervised by Church).
- Turing, A. M. (1950), 'Computing machinery and intelligence', *Mind* **49**, 433–460. Available online: <http://cogprints.org/499/00/turing.html>. Reprinted in (1992).
- Turing, A. M. (1969), 'Intelligent Machinery'. National Physical Laboratory Report, 1948, in B. Meltzer & D. Michie, eds, 'Machine Intelligence 5', Edinburgh University Press, Edinburgh, pp. 3–23. Digital facsimile viewable at http://www.AlanTuring.net/intelligent_machinery.
- Vinnikov, V. (1999), 'We shall know: Hilbert's apology', *Math. Intelligencer* **21**(1), 42–46.
- von Neuman, J. (1981), First draft of a report on the edvac, in N. Stern, ed., 'From ENIAC to UNIVAC: An Appraisal of the Eckert-Mauchly Computers', Digital Press, Bedford, Massachusetts, pp. 177–246.
- Wang, H. (1996), *A logical Journey: from Gödel to Philosophy*, MIT Press, Cambridge, Mass.
- Weisstein, E. W. (2005), Cellular automaton, in 'MathWorld', A Wolfram Web Resource, <http://mathworld.wolfram.com/CellularAutomaton.html>.
- Weyl, H. (1944), 'David Hilbert and his mathematical work', *Bulletin of the American Mathematical Society* **50**, 612–654.
- Weyl, H. (1951), 'A half-century of mathematics', *American Mathematical Monthly* **58**, 523–553. See also (Weyl 1968), pp. 464–494.

- Weyl, H. (1968), *Gesammelte Abhandlungen*, Springer-Verlag, Berlin.
- Wittgenstein, L. (1980), *Remarks on the Philosophy of Psychology*, University of Chicago Press.
- Wittgenstein, L. (1994), *Remarks on the Foundations of Mathematics*, MIT Press.
- Wolfram, S. (2002), *A New Kind of Science*, Wolfram Media, Champaign, IL.
- Wood, D. (1987), *Theory of Computation*, Harper & Row, New York.
- Wright, R. (1988a), 'Did the universe just happen?', *Atlantic Monthly* pp. 29–44. Online version (in English): <http://digitalphysics.org/Publications/Wri88a/html/>.
- Wright, R. (1988b), *Three Scientists and Their Gods: Looking for Meaning in an Age of Information*, Times Books.
- Yandell, B. H. (2002), *The Honors Class. Hilbert's Problems and Their Solvers*, A. K. Peters, Natick, MA.
- Zach, R. (2001), *Hilbert's Finitism: Historical, Philosophical, and Metamathematical Perspectives*, PhD thesis, University of California, Berkeley, Berkeley. A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Logic and the Methodology of Science in the Graduate Division of the University of California, Berkeley.
- Zuse, K. (1936), *Verfahren zur selbsttätigen Durchführung von Rechnungen mit Hilfe von Rechenmaschinen*, Patentanmeldung Z 23 139/GMD Nr. 005/021.
- Zuse, K. (1967), 'Rechnender Raum', *Elektronische Datenverarbeitung* 8, 336–344. Available online at: <ftp://ftp.idsia.ch/pub/juergen/zuse67-scan.pdf>.
- Zuse, K. (1969), *Rechnender Raum*, in 'Schriften zur Datenverarbeitung', Vol. 1, Friedrich Vieweg & Sohn, Braunschweig. English translation: "Calculating Space", MIT Technical Translation AZT-70-164-GEMIT, MIT (Proj. MAC), Cambridge, Mass. 02139, Feb. 1970. Available online at: <ftp://ftp.idsia.ch/pub/juergen/zuserechnenderraum.pdf>.

Witold Marciszewski

University of Białystok

UNDECIDABILITY AND INTRACTABILITY IN SOCIAL SCIENCES

Motto 1: *There are actually lots of threads that led to computer technology, which come from mathematical logic and from philosophical questions about the limits and the power of mathematics.*

Greg Chaitin¹

Motto 2: *Computer simulations are extremely useful in the social sciences. It provides a laboratory in which qualitative ideas about social and economic interactions can be tested. This brings a new dimension to the social sciences where 'explanations' abound, but are rarely subject to much experimental testing.*

Richard J. Gaylor (University of Illinois),
Louis J. D'Andria (Wolfram Research, Inc.)²

Introduction

In scientific practice a computer has become so indispensable a tool as a sheet of paper, pen or laboratory instruments used to be before it was discovered³. Hence a question regarding a range of its possibilities occupying

¹ "A Century of Controversy over the Foundations of Mathematics" in: C. Claude and G. Paun, *Finite versus Infinite*, Springer-Verlag London 2000, pp. 75–100. Due to his original and influential ideas and results on range and limits of algorithmical methods, Chaitin, the IBM mathematician, has become a classic of both computer science and philosophy of science. In this respect, he is an eminent continuator of Gödel's and Turing's thought. He is much acknowledged for the discovery of the incalculable Omega number defined as a probability that a computer program will end after a random binary sequence has been entered. For a comprehensible study of the definition see Paul Davies, *The Mind of God*, Chapter 5, Part "Unknowable", pp. 128–134 (Simon and Schuster, New York, 1992).

² From the introduction to the book entitled *Simulating Society: A Mathematical Toolkit for Modeling Socioeconomic Behavior*, Springer Verlag, 1998. See www.telospub.com/catalog/FINANCEEON/SimSoc.html.

³ This paper develops ideas underlying the research project "Nierozstrzygalność i algorytmiczna niedostępność w naukach społecznych" [Undecidability and Intractability in Social Sciences], no. 2 H01A 030 25, supported by the Polish State Committee for Research and Development, in years 2003–2006. It is a revised version of the Polish paper

the minds of the philosophers of science and methodologists. In the first place, one has to consider astonishing results of the logical and mathematical research – like the fact that not every problem of arithmetic can be solved with the help of a computer algorithm. There is no algorithm to decide the following: is there a solving algorithm of any mathematical problem? And even if such algorithms do exist, their use often requires so huge time or space (memory) resources that practically one can hardly expect them to solve a problem. This is an example of algorithm intractability.

Is this limitation only an internal business of pure mathematics? Or perhaps it also refers to empirical sciences (social sciences being among them) which mathematics provides with algorithms to model the reality computationally?

A positive answer to the last of the above-asked questions is not given a priori. Here perhaps some amount of philosophical faith in good demon would not be absurd – the demon would have created the empirical world in such a way that there would be only relations represented by computable functions. What is more, such functions would always allow to grasp problems algorithmically.

What we already know about natural and social problems does not confirm that faith. However, an awareness of this fact does not easily reach researchers, especially those dealing with social sciences. Some of them seem to advocate a view that issues of undecidability remain in an exotic sanctuary of pure mathematics, being away from empirical facts. Computer and logical researches putting that view into question are relatively fresh. Hence a need to provide necessary information along with some reflection what is to be done in such a situation.

However, this is not the end of astonishing messages. Physicists claim that certain problems solved with the help of algorithms or software (on condition that a parallel development of hardware is equal) may remain unsolved due to the developmental limits on the part of physics. Miniaturization, constantly enlarging its computational power, is in the long run limited by material granularity, whereas the development of computers towards a bigger capacity will finally result into a slowdown in signal transmission, obviously unable to exceed the velocity of light.

entitled “Nierozstrzygalność i algorytmiczna niedostępność w naukach społecznych” (likewise the title of the project in question) which appeared in the Polish quarterly “Filozofia Nauki” [Philosophy of Science], 2004, vol. 12, no. 3–4, pp. 5–31.

On the other hand, physics along with some philosophical reflection bring good news, too. In Nature, especially in the brain, there are some computing processes that have not been proved to undergo the restrictions that the universal Turing machine, or in other words, a digital computer, is subjected to. There is a significant group of physicists who see the source of superiority of the human brain over algorithm in the fact that the brain belongs to the sphere governed by quantum mechanics. To explain its superiority, they highlight the ability of the brain (not to say “of the mind”) to recognize the correctness of Gödel’s statement, discover axioms and invent algorithms. Since none of these things can be gained from a mechanical procedure, their source must be hidden somewhere in organic nature (or in its surroundings penetrated by philosophy).

1. The state of the problem and the ways how to solve it

1.1. By the end of the 20th century the theory of computational complexity had appeared in computer studies. It continued the main thread of logic starting from the point marked by Gödel, Turing, Church, Post and Tarski (just to mention the main leading figures), who followed Hilbert’s thought. At that point logic surprised the world of science by the discovery of the issue of undecidability both in mathematics and in logic itself. After a short time there appeared a recognition of the fact that in the sphere of decidable problems there are some problems whose solutions can never be found even if we were to employ as many supercomputers as a number of electrons in the space, giving them as many years to count as the age of the world’s history. This practical insolvability of the problems that are “solvable” themselves has been called *intractability*, sometimes preceded by the adjective *computational* or *algorithmic*.

This term should not be understood as a lack of the proper algorithm to solve a problem but rather as an implication of the fact that a solution expected from such an algorithm is inaccessible (because of the reasons which will be discussed later – like a lack of time or memory resources). Analogically, when it is said that a problem is computationally intractable, it does not imply the impossibility of the computational process that would lead towards a solution (it is possible when a solution is a computable number). Rather, it implies the fact that apart from making calculation, the solution will not be reached. In this expression the adjective “algorithmic” has been used because the term “algorithm” is better known than the term “computing” in the technical sense of Turing [1936], which is different from colloquial. Colloquially, one can say,

“a jumper has calculated the distance well” even though this process does not require any number operations (as it is in case of Turing).

The sources of algorithmic intractability, tackled by the theory of computational complexity, will be discussed later. At this stage, all we need is to take note of it just to recognize the following methodological issues relating to social sciences.

- [1] Can algorithms, used to model and simulate social phenomena, have the complexity that would lead towards indecidability or intractability of the problem?
- [2] If so, are there any methods to transform the problem so that it is no longer intractable and at the same time its solution would be close enough to the solution of the original problem?
- [3] If so, what are these methods?
- [4] Are there statements or other elements of the theory whose acceptance is not justified by any of the algorithms applied in the given theory?

The answer to the question [4] comes immediately both for deductive sciences (logic and mathematics) and for empirical ones, social sciences being among them. Certainly, such statements exist. In the deductive theory these are axioms and rules whereas in the empirical theory they are observational statements as well as meaning postulates in the Carnapian sense [1956], also called language postulates by Ajdukiewicz [1965]. Even though the answer to question [4] is obvious, the question needs to be asked for it reveals the following:

- [5] On which basis are statements of a particular theory accepted when none of the algorithms justifies their acceptance?

If we were to address questions [1], [2] and [3] to physicists, we would receive answers illustrated by an example list of the problems of physics – undecidable, intractable or having only approximate solutions. Physics has at its disposal a significant collection of such limitation results, dealing with the limitations of solvability.

Stephen Wolfram’s text entitled *Undecidability and Intractability in Theoretical Physics* [1985]⁴ makes a good starting point to consider the

⁴ Wolfram is widely known due to his works on cellular automata collected in the book [1994]. He is also known as the author of the software “Mathematica” meant for computing and programming in scientific research. His monumental book [2002], which states that cellular automata of a certain type constitute an adequate model of the physical world, was the scientific bestseller of the year.

problems in view. The author gives examples of undecidable and computationally inaccessible problems in physics employing the notion of *reducibility*, which also reveals the term *compressibility* in the Kolmogorov-Chaitain algorithmic information theory. A lack of this feature consists in the fact that an algorithm, simulating the process under examination, has to reproduce it step by step (explicit simulation), having no possibility to reduce it. Because of its length, unreducible computation is subjected to face a lack of time or memory resources or, in other words, it will prove to be intractable. Wolfram finds examples of unreducibility in certain processes taking place in cellular automata, electric circuits, nets of chemical reactions, etc. In this collection some problems are undecidable and other are intractable. In the end the author concludes that such situations are not exceptional. Instead, they are common⁵. The title of Wolfram’s article has served as a model for formulating the topic of this work. Since he has been largely quoted, Wolfram can make a widely accepted model of problems of certain type. The question, which in Wolfram’s article refers to physics, can as well be asked in relation to any science or a group of sciences. Therefore, let us ask this question in relation to social sciences.

1.2. To state how researchers into social phenomena and theorists of the issue of computational complexity relate to questions [1]–[5], let us consider them once again.

An awareness of the types of the employed algorithms’ complexity would be a rare thing to meet while searching for the answer to question [1] in sociological and economic literature, giving an account of the use of mathematical models. However, if computational complexity was to be found in the works of theorists, there would easily appear a pile of limitation results similar to the ones present in Wolfram’s work, on which the present paper has been based. In the light of these results the answer to [1] is positive.

Therefore, there appears a need to confront both trends: the one using algorithms of the empirical studies with the logical studies of these algorithms. Bearing in mind the fact that the problem of possible limitations of the computational power of algorithms is often ignored in some empirical

⁵ Wolfram mentions numerous examples of undecidability and intractability in physics: “Quantum and statistical mechanics involve sums of over possibly infinite sets of configurations in systems. To derive finite formulas one must use finite specifications for these sets. But it may be undecidable whether two finite specifications yield equivalent configurations. So, for example, it is undecidable whether two finitely specified four-manifolds or solutions to the Einstein equations are equivalent (under coordinate reparametrization).”

studies, this need becomes even more important. In this case, the methodologically important questions [2] and [3] have no chance to appear.

A procedure that does not take any account of algorithm limitations is only justified when it is known that a problem under examination is simple enough to be solved by the algorithm employed. However, something different is being observed. Ignorance of the questions regarding complexity of the problem takes place while considering highly difficult problems, with the highest level of complexity. Three examples of this type are given below (Ex 1 – Ex 3).

Ex 1 – Strong artificial intelligence (AI). It is a highly ambitious project for it aims at the entire (having no difference from the original) simulation of the most complicated product of Nature, that is, the human brain. The realization of this project would have a significant influence on social sciences, enabling them to gain knowledge how to create optimum social configurations with the help of simulation through artificial societies. Artificial societies (AS) are the societies where every member is represented by a subprogram defining its (artificial) mind as well as its interactions with the surroundings (this branch of computer science falls under the definition of “multi-agent simulation”). Not only is AS conditioned by the prior creation of the AI, but it is also a necessary condition of the advanced AI since the development of intelligence requires adequate social surroundings. Apart from this complexity, intensified by the feedback AI-AS, the research into AI does not offer any reports on limitation results. However, there are promises of an upcoming success.

Ex 2 – Central socialist planning supported by computers is a concept propagated by Oscar Lange in the polemics with the Austrian Economic School (von Mises, Hayek). According to that School, socialist planners were doomed to face a failure due to their disability to show the complexity of economic facts. At the beginning of the 60s Lange fought that blame back arguing that the creation of the computer made it possible what had seemed impossible before. Among a group of leftist economists the concept remains fertile till today. Due to the fact that economic phenomena are relatively easy to measure, it must be possible to state an order of magnitude when it comes to the size of input data. Also, there are proposals of some mathematical models of (for example) the market equilibrium (Pareto, Lange, etc). As a result, there are good reasons for estimating the complexity of the algorithms necessary for an effective socialist calculation. A search query must be held to find calculations of this type in the already existing literature. In case of its failure, one has to estimate the complexity of the problem on his own.

Ex 3 – Referring to the professional computer simulation (conducted in MIT), by the end of the 20th century a report of the Club of Rome foresaw a total worldwide economic and ecological failure. It is obvious that such a complicated enterprise cannot be achieved unless the model is simplified. There are

simplifications that do not deform the reality making the study easier and quicker, instead. On the other hand, there are simplifications that lead towards a picture that is completely different from the real world. A similar simplification of the report of the Club of Rome was the omission of the human creativeness factors as well as people’s ability to exert creativeness when being under threat (similarly, this factor must be omitted in the concept of central economic planning for it is impossible to plan discoveries). As a result, the report produced fictitious prognosis. A lack of knowledge anticipating what may happen in the mind of scientists, discoverers, reformers, etc. makes a considerable difficulty here. But even if such knowledge were to be revealed by Laplace’s demon, in the face of the infinite complexity of creative thinking one can hardly expect this process to be simulated by the algorithms within a digital computer’s reach.

What may the best reaction towards ignoring the real complexity in social studies be? A fatalistic reaction would depend on the assumption that the evaluation of the model complexity is impossible whereas the gap in the knowledge about the model would be recompensed by the faith that it could match the reality. In other words, it would be able to explain and foresee real processes. Perhaps it would be acceptable to agree with such a reaction if the evaluations of the model complexity in social sciences were beyond researchers’ possibilities. Fortunately, the truth is different. To prove it, it is enough to consider any widely used model of social phenomena and mention the studies of this model conducted within the theory of computational complexity.

1.3. Prisoner’s Dilemma, which is a standard model for a wide class of social interactions, makes a perfect case for this aim. The popularity of this model is confirmed by the fact that at the beginning of 2003 Google showed over 800 Internet sites offering links related to the topic. This example enables one to observe how an increase of the number of the output data (number of social actors or players, number of strategies, number of rounds to play) in some cases leads to undecidability or intractability.

The name “Prisoner’s Dilemma” is related to the story illustrating a problem of two prisoners. Being suspected of robbery, they face a dilemma (this is the plot of the dilemma stated in 1940 for the first time). To get evidence, an investigating magistrate offers each prisoner a deal so that they think they have been treated individually. The conditions of the deal are as follows. If two prisoners acknowledge their guilt, they will get a fifteen-year prison sentence. If they do not admit their fault so that it is impossible to prove the robbery, they will be sentenced only for three years

(for a different crime). If one of the prisoners acknowledges his fault, he will be released (as the one who has contributed to the investigation a lot) whereas the other will get a twenty-year prison sentence.

The question what pays off – to cooperate together loyally or to pursue one's selfish interest even at the partner's expense when it is not known what he is going to do makes the root of the dilemma. In other words, the unawareness of the partner's intentions is the heart of the problem. Different ways of behavior to choose from are called strategies of the game. They are defined respectively as *cooperative strategy* (in the example of the prisoners this is abstaining from burdening themselves with evidence) and *competitive strategy* (aiming at one's profit without taking into consideration a loss of the partner).

Undoubtedly, the prisoners' example perfectly highlights the drama of the dilemma. On the other hand, a situation when the members of the game can play it repeatedly is not illustrated well enough. This situation is important for the explanation of the evolutionary processes. Consisting of a few component games called rounds, such *iterated* games give its players a chance to learn by getting to know a partner better and possibly harmonizing their actions. Let us illustrate this general situation by a game with the following rules.

If A and B are willing to cooperate, each will get 3 gold coins;
 If A chooses cooperation with B while B chooses competitiveness – B gets 5 gold coins and A gets nothing;
 When both choose competitiveness, each gets one gold coin⁶.

In the one-round game, when uncertainty as to the other member's behavior is marked as 0,5 probability for each option, a competitive strategy would be more beneficial. If a game consists of many rounds, the case gets more complicated when, for example, each player may anticipate the other player's strategy. In the situation when each side gets the certainty about the partner's willingness to cooperate, each partner stands a chance of winning 3 gold coins whereas their choice to compete would result in only one-coin profit for each. Here we have a certain model of social evolution directed at the increase of cooperation.

A natural computational model in the research on decidability of the prisoner's problem is provided by cellular automata⁷.

⁶ This version of the game can be practised at: serendip.brynmawr.edu/playground/pd.html.

⁷ This is a summary of the material presented at: www.sunysb.edu/philosophy/faculty/pgrim/SPATIALP.HTM. This is Patrick Grim's article entitled "Undecidability in the

Among others, there has been considered an arrangement where a cell has eight direct neighbors as its partners. The strategy, being the state of the cell, can be (a) purely competitive, (b) purely cooperative, (c) mixed – "tit for tat" – when a cooperative move is followed by cooperative and a competitive one by competitive, and (d) – the strategy in which a player observes the strategies of his/her neighbor and chooses the one which has proved to be more beneficial. In the course of the game certain strategies appear more often and become predominant, which can be observed as a change of configuration on a two-dimensional panel where the evolution of automata is being played.

To some extent it is possible to anticipate the direction of this evolution. Consequently, there appears a question: is there an algorithm able to decide for each case whether *a given configuration of the strategies will gain a permanent predominance over the rest?* Grimm [1997 as well as the already quoted WWW text] has given evidence that this problem is undecidable in the classical Gödelian sense, which means that such an algorithm does not exist⁸.

A research on the connections between computational complexity of a game and its players' rationality makes another important trend. The theory of complexity should help the theory of games realize an empirical factor that cooperative strategies can be beneficial for both sides. Paradoxically, the theory of games itself implies that when the players do not change a strategy (which is known as the state of equilibrium), it is beneficial for both sides to stick to the competitive strategy. The consequence, which takes place in case of giving the players unlimited means to solve problems (the so-called unlimited rationality), stops to be obligatory, for example, in cases of limiting the storage capacity; memory can be measured by a num-

Spatialized Prisoner's Dilemma: Some Philosophical Implications". The author presents an interesting account of the problem emphasizing the significance of the Prisoner's Dilemma as a model of social processes: "This simple game-theoretic model seems to capture in miniature something of the tensions between individual acquisitiveness and the goals of collective cooperation. That is of course precisely why it has become a major focus of modeling within theoretical sociology, theoretical biology, and economics. [...] It is no simplification to say that our strongest and simplest models of the evolution of biological and sociological cooperation – and in that respect our strongest and simplest models of important aspects of ourselves as biological and social organisms – are written in terms of the Iterated Prisoner's Dilemma.

⁸ See Grim [WWW, op. cit]. "There is no general algorithm [...] which will in each case tell us whether or not a given configuration of Prisoner's Dilemma strategies embedded in a uniform background will result in progressive conquest. Despite the fact that it is one of the simplest models available for basic elements of biological and social interaction, the Spatialized Prisoner's Dilemma proves formally undecidable in the classical Gödelian sense."

ber of the states of finite automaton realizing a particular strategy in the game with n rounds. A. Neymann [1985], a pioneer of the concept of limited rationality, stated that when the limitation of memory of both players consists in its capacity at the interval $[n^{1/k}, n^k]$, where n is a number of rounds and $k > 1$, cooperation becomes the most beneficial strategy for both players. It has also been proved that cooperation dominates competitiveness in the games with an infinite number of rounds as well as in the games with a finite but unknown number of rounds (see Papadimitriou and Yannakakis [1991]).

These and other numerous results dealing with a particularly useful model of interaction – the Prisoner’s Dilemma – illustrate how the methodology of social sciences can benefit from the theory of computational complexity. However, there appears a problem similar to the one about its practical use in industry and economy. An initial discernment shows that research practice of social sciences has made little use of this theoretical background. This initial discernment should be put under a systematic verification and in case of its confirmation, a question regarding its reasons should be asked. Are these reasons objective, based on the fact that the results of the theory of complexity are too subtle or abstract for the real needs of the research practice? Or perhaps there are subjective reasons, resulting from the researchers’ disability to follow the progress of the theory of complexity? Indeed, these questions are worth asking in the research conducted within the methodology of science and the study of the progress of human knowledge.

1.4. The above-mentioned Grim’s research on decidability of one of the problems within Prisoner’s Dilemma (compare footnote 8) appears to be of great use when it comes to the question which was formulated earlier (point 1.1, question [4]), namely, are there statements or other elements of the theory whose acceptance is not based on any of the algorithms applied in the given theory? Indeed, a positive answer is obvious since in the empirical theory neither observation statements nor meaning postulates give evidence as to their derivation from the algorithm. However, results similar to those of Grim indicate statements capable of being axioms (at the stage of axiomating a given theory)

To specify this conclusion, let us refer to the clue that is hidden directly in Grim’s research. The question that, according to this research, is undecidable can be answered with a hypothesis of an intuitive character drawn from certain social phenomena. For instance, the fact that civil wars (well-situated within the scheme of Prisoner’s Dilemma) often reach a compromise after the two

sides “have lost blood” (after many rounds of the iterated game). It means that a cooperative strategy wins for both sides, which results in a certain state of equilibrium. Its approach, as it is indicated by Grim’s limitative result, cannot be concluded from the axioms of the theory of games supplying here the model required. Therefore, it is a clue that our hypothesis is likely to take a position in a group of axioms.

The question [4] produces another question, namely [5]: *on which basis are statements accepted in a specified theory when no algorithm supports it?*

The following circumstances may constitute such a basis: statements, accepted in a given theory without any proof, may have proof in another theory that is worth accepting (we deal with a certain algorithm in case when a proof has been formalized). This is an example of a current procedure in science. A supposition that some algorithms “exist” beyond theories, supplying certain theories with propositions worth accepting as theorems, is less obvious.

The admission of such a supposition (in this context) comes from the need to begin a dialogue with the theory called strong artificial intelligence. It helps to articulate the thought of the theory that the statements of the class represented by the Gödelian statement are also produced by some algorithm. Having formulated it in such a way, there appears a following question: to which theory does an algorithm belong if it does not belong (for example) to mathematics? This marks the next move in the discussion as, for example, a statement that the algorithm must be beyond any theory – let us say – functioning somewhere in Gödel’s brain. It would mean another statement that this brain has computational power equal to the power of the Turing machine, which would be another step forward prejudging on whom *onus probandi* rests.

Remaining central in the discussions about artificial intelligence, this problem has been outlined to be looked at from a methodological point of view. From this point it is important that we deal with the statements which in a given theory neither derive from any other statements nor belong to its axioms. What is more, they are not observational statements, registering only sensory perception of something taking place here and now. **A priori** statements (or propositions) would be a suitable term for them⁹.

⁹ If this category seems incomprehensible or even mystical, supporters of the “scientific sobriety” may prefer the algorithms that remain beyond any theory. They can be hardly rejected by those who represent the “only scientific” view that the brain is the Turing machine, producing both axioms and any other apriority according to the axioms unknown even by itself.

To justify this term, let us highlight the following. The Gödelian statement is a prototype of the statements deserving to be regarded as true although they are neither axioms of a given theory nor they are derived algorithmically from the axioms. The above-mentioned metatheoretical research shows that such statements also appear in those empirical theories (social theories being among them) that use mathematical models. This is a methodologically important category that has to be distinguished by a special term. Such statements deserve the name of potential axioms. However, apart from the aspect of their possible purpose, there is a need to hint at the aspect of their origin. The traditional expression "a priori", used in philosophy for years, reveals this genetic aspect.

A view on social theory as including a priori statements corresponds well with the view of one of the classics of methodology of social sciences – Ludwig von Mises. In the work entitled *Human Action* [1966] he characterized economy as a highly general science about human action that he called praxeology. From a methodological point of view it was described by him as follows:

"Praxeology is a theoretical and systematic science. [...] It aims at knowledge valid for all instances in which the conditions exactly correspond to those implied in its assumptions and inferences. [...] Its statements are, like those of logic and mathematics, a priori." [p. 32]

"The fact that man does not have the creative power to imagine categories [of thought, action, etc] at variance with the fundamental logical relations and with the principles of casualty and teleology enjoins upon us what may be called *methodological apriorism*." [p. 35, italics LvM.]

The table of content of the work *Human Action* illustrates which social regularities, according to von Mises, are experienced in such an a priori way. Typical topics of economy such as account, market, price, credit, work, the role of government, taxes, etc can be found there. Indeed, a view that economic laws of the above-mentioned things are as a priori as mathematical statements are sounds radical. However, it is possible to make it less radical by adopting the idea of W. V. Quine regarding the levels of apriority. Its highest level is to be revealed by logic and mathematics (hence a complementary element – empirical content – can only be found in them to a little extent). It is further divided in different rations. In this hierarchy economy could be placed relatively close to the top of apriority. Indeed, such principles as a principle of expected utility, laws of supply and demand, or an informative function of prices (F. Hayek's idea) are highly a priori.

These considerations encourage us to adopt von Mises's concept of **methodological apriorism**, which reveals the fact that numerous laws of social sciences behave like mathematical or logical axioms. Therefore **meaning postulates** – the term introduced in 1.1. – would be suitable for them. It includes axioms but refers to the class that is more capacious, holding the statements that share only axiom apriorism without their role of being first premises of the system.

Meaning postulates are not a product of the algorithm acting within the theory. It may be believed that some algorithm is produced beyond the theory, functioning in our brains to produce a priori principles. They can also be produced as a result of some non-algorithmic process. In any case, they are unlikely to become a pain in the neck for the researchers because of their excessive algorithmic complexity. A problem of the algorithmic procedure in relation to them does not appear at all.

How should such a situation be approached in social sciences? What is a cognitive value of these a priori principles, having a kind of privilege, for they are subjected neither to empirical verification nor to control as to their consistency with other statements of the system? This is a significant research problem. On the one hand, the recognition of such an important role of the a priori element disturbs a widely accepted empirical paradigm. On the other hand, a practice of mathematical modeling (when mathematical equations become statements of an empirical theory) introduces a significant element of apriorism. The necessity of meaning postulates as constituting the language of the theory makes another element of this kind.

This problem can be successfully attacked with the help of the following strategy. In social sciences the performance of such fertile and widely accepted models as the theory of games or cellular automata should be put under a methodological analysis. In the course of its application, a mathematical model, as expressing certain assumptions of the rationality (intelligence) of the actions, is being confronted with the observations that do not always confirm such rationality (often they lead towards its negation). What rule of preference should a researcher choose in this situation? Should he change the a priori assumptions or should he leave them and interpret the observation data in their light? There is no ready answer. Each case should be analyzed separately and a final verdict is bound to appear in the developmental course of science when a solution will be accepted or rejected after some time, remaining a mere exhibit in the records of the abandoned theories.

1.5. This passage is a kind of an erudite annex. Offering neither problems nor hypotheses, it only comments upon a few terms which were mentioned

above (the passage is dedicated to those who feel a need to have a closer look at them).

The concept of decidability is derived from logic where it appeared explicitly in the context of Hilbert's Program with the conviction that every well-formulated mathematical problem is decidable (see Hilbert and Ackermann [1928]). The acceptance of the hypothesis is defined as positive whereas its refutation as a negative solution of the problem of decidability (*Entscheidungsproblem*). In the original formulation of Hilbert and Ackermann [1928, p. 73], it is put as follows (italics by H and I).

Das Entscheidungsproblem ist gelöst, wenn man ein Verfahren kennt, das bei einem vorgelegten logischen Ausdruck durch endlich viele operationen die Entscheidung über die Allgemeinheit bzw. Erfüllbarkeit erlaubt.

Die Lösung des Entscheidungsproblem ist für die Theorie aller Gebiete, deren Sätze überhaupt einer logischen Entwickelbarkeit aus endlich vielen Axiomen fähig sind, von grundsätzlicher Wichtigkeit.

Apart from relativizing to particular axiomatics and a specified group of formal rules of inference, the concepts of procedure appearing here (Verfahren) has a scale as wide as it is in case of the terms of algorithm or computer program. Being a computational operation, every step of the algorithm execution conducted by a computer is at the same time an inference from arithmetical axioms, done according to the rules of logic (the implication form of these rules allows for their use as the rules of inference).

A positive solution to the problem of decidability for logic would allow for such an algorithm when the correctness of every step would be indicated by referring to a proper logical formula whose validity (*Allgemeinheit*) could always be demonstrated due to decidability of logic. Such an algorithm would ensure decidability of every axiomatized theory, if formalised by the laws of logic.

When Turing [1936] and Church [1936] demonstrated undecidability of logic or, in other words, proved the negative solution to the problem of decidability, there appeared reasons for asking more questions concerning algorithms, which gave birth to a new theory. Called **computational complexity**, this theory is considered to be a section of computer studies. The history of the problem shows that it is an area bordering computer studies and logic, remaining a point of interest for both of them.

The following terminological nuances deserve a special attention. The term "algorithmic" is being used interchangeably with "computational" in the contexts of the expressions "algorithmic intractability" and "computational intractability". A similar interchangeability has not been adopted

with "complexity". "Algorithmic complexity" has a meaning that is different from "computational complexity". The former relates to the measure of complexity (defined independently by Kolmogorow, Chaitin, and Solonoff) regarding a relation between the length of symbol sequence produced by the algorithm and the length of the algorithm in question (see Chaitain [2002]). The last relates to memory and time resources necessary to solve a given problem by the algorithm. The complexity is measured with the size of the resources required. (Hartmanis and Stearns [1965] were pioneers in this field).

Algorithmic tractability or intractability (as well as decidability or undecidability) is predicated of problems. The problem is **algorithmically tractable** when it is decidable, and it is not too computationally complex for a solving algorithm (program) to be conducted with space (memory) and time resources within the computer users' reach (definitely, this conception of reach is not precise but this imprecision is not really harmful). Section 3 will discuss how the complexity of the problem is connected with the limits of both space (memory) and time of the algorithm execution.

2. The idea of rationality and the notion of intelligence in social sciences

2.1. It was on purpose that in the title the word rationality was used in the context of the word "idea" while the word "intelligence" was linked with "notion". Out of these two words it is the word "idea" which reveals a bigger amount of normative or axiological charge (surely because of its nearness to the meaning of the word "ideal"), which makes the difference between these close meanings.

A person who is capable of solving his/her problems successfully and possibly with a little expenditure, being able to differentiate between the levels of problem's importance, is called intelligent. More or less the same applies to the definition of rationality. Therefore, the difference lies not in any radically different substance but rather in association, stress and context. This justifies a connection of the two topics, which should lead towards a mutual complement of the two spheres of our consideration. An example of the social problem, which is going to be presented in the second part of this Section, can be discussed in the category of rationality as well as in the category of intelligence.

The notion of rationality is inseparable from the standard one in social sciences of the game models. The aim of the game is to win. Therefore, it is

only natural to define the behavior that results in profit as being rational whereas the behavior that brings losses is defined as irrational. In this context a similar thought will be expressed when the word “rational” will be replaced by the word “intelligent”. However, apart from the interchangeability, this theory reveals its ability to enrich one content with another. The problems of the issues of artificial intelligence connect intelligence with computational power – one of the main topics of the theory of computational complexity. Hence, as it is seen from the texts mentioned in 1.3., the authors often refer to a limitation of computational power and, consequently, intelligence as to bounded rationality.

In this manner the two notions start converging into one notion, which also leads towards a tie-up between social sciences and the theory of intelligence. A few directions of this tie-up are worth mentioning. Among others, the following facts are concerned:

It was in the early 90s that an intensive process of the tie-up between social sciences and Artificial Intelligence (AI) started. The progress of artificial intelligence led towards the programs that enabled an interaction between artificial brains represented by appropriate programs. It was called **distributed AI**. The appearance of the net interactions (Internet, etc) made another step possible – at that stage an interaction between the programs functioning in different computers was within reach. The sides of such interactions were called “agents”, hence the appearance of the term “multi-agent models”. Due to those results, a new research direction appeared – **Artificial Society (AS)** – a continuation of AI in the direction of social sciences. As a result, the programs functioning as artificial brains are used in the computational models for computer simulations of social phenomena.

The theory of cellular automata (started by John von Neumann and Stanisław Ulam) is a rich source of computational models for (among others) the processes taking place in different societies. A cellular automaton is a collection of objects, situated in the space that is regularly divided into cells. The states of these objects change according to where and which objects appear in their close neighborhood. It delivers models of different social interactions such as gossip expansion or an appearance of isolated ethnic groups. Simple rules, specifying dependences, often lead towards highly complex and unforeseeable processes, which result in undecidable or algorithmically intractable problems.

The AI section, which constructs learning machines, delivers computational models meant not only for the observation of the evolution of the individual brains but also for the observation of the evolution of social struc-

tures. Its ability to adapt to the new conditions is a typical example, illustrating that it is capable of learning. Therefore, learning machines make another computational model for social simulations.

Finally, let us note the benefits brought by the evolution of notions to the interpretation of traditional sociological problems. Although rationality of social structures (eg. certain types of civilization) makes the main point for Max Weber, a classic of sociology, the term “intelligence” has not been adopted in that context. Recognition of these two guises of the same notion will enable us to use the theory of intelligence and its methods for modeling the already mentioned structures.

2.2. The notion of intelligence or rationality as stated in relation to some social structure has its historical exemplification that reveals a methodological role of the notions of decidability and algorithmic tractability. That was a famous dispute initiated by Ludvig von Mises in the 20s of the last century. It considered a possibility of calculation in socialist calculation (see von Mises [1966]). The dispute reached its climax in the 30s when Friedrich Hayek and Oskar Lange joined it. The same polemicists continued it after the War till the death of Lange (1965). Today Lange’s ideas used in the context of informatics are resumed by some authors (e.g. Cottrell and Cockshott [1993]).

Lange argued with Friedrich A. Hayek (1899–1992) about the most intelligent economic regulator. The question was whether it was free market or central planning. The appearance of computers made Lange believe in his final victory for he treated free market only as a computational instrument for calculation of the proper prices, namely such prices that could ensure the equilibrium between supply and demand. He did not deny the fact that market was performing that duty but according to him, the manner of that performance was too slow and full of mistakes. He believed that it would take a second for the Central Planning Commission computer to calculate perfectly something that would take a long time for the market to count (bearing in mind its slowness).

On the other hand, being aware of the fact that computational power of machines could not always meet economic complexity, Lange allowed for an auxiliary role of the market as an instrument of central economic control while controlling economy in short distances. Central planner’s absolute superiority, according to him, consisted in solving long-term problems of the economic growth. Because market could only currently regulate the economic equilibrium, it was not capable of marking any long-term goals of its development.

Hayek opposed those views on the ground of the considerations about information processing. His thought could be expressed in a short way with the help of the modern terminology of computational complexity. The thesis about the realizability of central planning with the use of computers implies that the algorithms used for this aim are quick enough not to wait for the calculation results for many years. The complexity becomes even more monstrous bearing in mind the fact that a central planner would need the complete data of the whole country regarding supply, demand, etc, in relation to every product so that on that basis he could calculate an optimum price. Next, if necessary, an hour by hour he would need to bring it up to date. On the other hand, a computational system, which is free market, radically limits this flood of data in two ways. Every member of the market game needs only data regarding the price of the product and only these products that remain within his activity. It is similar to the data processing that is parallel, dispersed and, at the same time, it enjoys the merits of analogue computation.

Hayek justified those statements intuitively. Today a notional apparatus of the theory of complexity allows for their precise confirmation. One of the procedures could be as follows. Since there are active advocates of Lange's view who operate the informatics notions, they are expected to offer a proof that economic problems of central planning are solved with the help of algorithms working at polynomial rather than at exponential time and that there exists sufficient memory resources which are also measured polynomially, etc. On the other hand, the analysis of the behavior of the market members should indicate whether the problems they solve can be modeled with the help of some decidable theory and if so, whether they are more algorithmically tractable than the problems faced by a central planner.

The dispute cannot be treated only as a historical tale. Nowadays it gains a new interest for two reasons. Equally important, they come from different directions. Due to the growing wave of socialistic tendencies, which has been observed worldwide, a political reason can be indicated. These tendencies can be observed in the impetuous antiglobal movement or in some standards of "political correctness". Hence, there arises a need of a possibly most precise comparative analysis of both socialist economy and free market. According to Lange and Hayek, the theory of computational complexity in its present state of advancement makes an ideal tool. Even if it was not for a practical demand, the advanced state of the tools encourages us to test them in such an interesting theoretical field. Being theoretical and mutually reinforced by the practical one, it makes the second reason to continue the dispute.

3. Addressing complexity

3.1. There are many ways to deal with the complexity of natural, mental and social processes: to simplify problems, to reinforce computational means, to go beyond the Turing machine, and to create an interaction between intuition and algorithm. The information civilization consists in a growing ability to deal with complexity in these various ways. A struggle against complexity takes place at least on two fronts. The first front is operated by the theory of deterministic chaos (of the unstable dynamic systems) whereas the second one is dominated by the theory of complexity – the topic of our considerations.

First, it is recognized what problems are within algorithm's reach. Undecidable problems as well as those potentially decidable but depending on inaccessible time and space resources remain out of reach. Time is a number of steps necessary to solve a problem. Space is a memory capacity that could be insufficient when confronted with a huge number of input data (other resources may be involved – for example, a number of sharing processors – but the former two are considered most often). Differentiation between two time categories – polynomial and exponential – makes a demarcation line dividing the sphere of what is algorithmically impossible from what remains within reach.

Polynomial time may be exemplified by function n^3 whereas function 2^n (where n is a number of input data) exemplifies **exponential time**. Let the polynomial $7n^3 + 5n^2 + 27$ (with n of input data) define a maximum number of steps. To evaluate the complexity of a polynomial algorithm, it is enough to consider its highest exponent omitting the coefficient as a negligible quantity (as 7 in $7n^3$). This distinguished exponent defines the order of the algorithm complexity. It is said that a given algorithm requires, for example, time $O(n^3)$ (notation with O indicates the limitation to the order of magnitude with the omission of the negligible quantities). A sorting algorithm with the order of complexity of $O = n \log n$ (hence it is less than $O(n^2)$) is an example of the algorithm working at polynomial time.

The problem of satisfiability of the formula of the propositional calculus (abbreviated as SAT) belongs to the class of the problems demanding exponential time. Having a given formula of the propositional calculus in, for example, the conjunctive normal form (that is, the conjunction of alternatives), it is necessary to identify whether there exists such a configuration of logical values assigned to the variables, which makes the formula true. Suppose, the formula has 300 variables. In the least favorable case (when,

for example, only one assignment makes the formula true and it is found in the end) the solution will take 2^{300} steps.

The traveling salesman problem makes another example of an unimaginably big request for time that, being factorial, is even greater than exponential. Having data of the whereabouts of n towns, his task is to visit all of them using the shortest route and visiting each town only once. Let him give 20 towns to visit (a starting place does not count). Then a number of routes is $20!$ since this is the number of possible arrangements in the set of 20 elements. An algorithm, which consists of all its possible arrangements, summing the length of the sections of each combination and a final recognition of the smallest sum, has not been found yet. In the light of the fact that

$$20! = 2\ 432\ 902\ 008\ 176\ 640\ 000$$

it is possible to imagine what algorithmic intractability depends on. If a computer is capable of checking a million combinations during a second, it will take 77.000 years to check all of them. Suppose we add a few more towns and the calculation would last longer than the age of the universe. This is an example of the algorithm that uses the so-called brute force; that is to say, it is based on the mechanical realization of all possibilities. There is no quicker algorithm for this problem, capable of giving an equally certain and precise result. However, in case of our acceptance of approximate results, the time of solving the problem of the traveling salesman can be shortened to a large extent.

3.2. Usually, the class of the problems solved by polynomial algorithms is marked with the letter P. This concise notation simplifies a consideration that has become the source of imposing results in the theory of complexity. Having marked another class of problems by the abbreviation NP, there arises a fundamental question – are these classes equal: $P=NP$?

This consideration has its roots in the observation that there are polynomial algorithms that answer the questions regarding the decision (the answer “yes” or “no”) by giving certificate then and only then when the answer “yes” is true. The question whether a given formula of propositional logic is satisfiable (if it is really satisfiable) will be answered positively, for example, by a polynomial algorithm. Likewise it is in case of the traveling salesman when the question is: is the length of a given route not bigger than such-and-such number?

To have a closer look at the example of decidability concerning the property of being prime, let me make use of the following recollection. Once Andrzej

Mostowski, a widely known researcher of the problem of decidability, gave me a funny and simple task to solve. One day, while arranging a meeting, he asked me about my address. When he heard that the number of the flat was 917, he was quick to ask whether it was a prime number. Surprised, I thought that he should have known the answer (although I did not). Now I am sure that the question had a didactic character – Professor Mostowski was testing the way I would cope with it. Although on a late reflection, I would cope with it in a typically non-deterministic way by giving, for a start, the first answer that comes to my mind after the elimination of those that are certainly erroneous. After calculating that $9 + 1 + 7$ cannot be divided by 3, I eliminate even numbers and number 3. Then I eliminate 5 for 917 ends neither with 0 nor with 5. So I approach 7, ready to consider the following candidates (11, 13, etc). A strategy has been chosen but I am ready to leave it and look for another one any time it fails. Therefore, I am using a simple polynomial algorithm where a number of steps (execution time) has a linear dependence on the length of the number sequence indicating the divided number. I am lucky – the first attempt gives the result without a remainder of the division – 131. Consequently, the solution is as follows – 917 is *not* a prime number. In this manner, it is possible to guess any big numbers – it is a matter of calculating ability. For instance, it would take a short time for a particularly talented person to guess that 226107 could be divided by 777 without a remainder (the result is 291).

This kind of polynomial algorithms is described as non-deterministic because they aim at the verification of the statements whose acquisition is not determined by any procedure (they should not be mistaken for probability algorithms). “Non” gives the letter N in the abbreviation NP. If the problem does not fall into the NP class, even a solution limited to such a confirmation may appear to be highly difficult. An assumption that the right solution is always within reach is a fiction, impossible to be realized by any machine. On the other hand, this fiction is very useful for it allows for the already mentioned question ($P=NP$?).

The class P is included in NP in the sense that if there is a polynomial algorithm to solve a problem in a general way, it will also serve in all those cases that are reduced to the question: whether such-and-such solution is correct? On the other hand, the thesis that P includes NP has not been proved. For if such a thesis held, the fact that a certificate problem in case of the traveling salesman is polynomial would result in the conclusion that the traveling salesman problem is polynomial in general or, in other words, it belongs to P. A hypothesis that the answer to the question $NP = P$? is negative is widely accepted.

The NP class contains a subset of problems which are NP-complete. The problem is said to be **NP-complete** when it belongs to NP and has the

following property: if there is a polynomial algorithm for some NP-complete problem, there is a polynomial algorithm for each problem in NP. It results in the conclusion that if a polynomial algorithm could solve at least one problem of this class, the same would apply to the rest (bearing in mind their mutual conversion). Therefore, the equality $P = NP$ would hold. Algorithm conversions of this type take place at polynomial time, which makes it tractable.

To summarize, relations between the classes of complexity under consideration are as follows. Both P and the class of NP-complete are proper subsets of the NP class. They are also thought to be disjointed.

A problem of satisfiability remains in the NP-complete class. It was the first to be recognized as having that quality. It also has become the measure of tractability for the remaining points of the NP class; if it could be solved polynomially, it would apply to the whole NP class. What is more, apart from the traveling salesman problem, the NP-complete class also encloses many problems from different fields – graph theory, operational research, cryptography, theory of games, and theory of social choice.

3.3. The disability to solve NP problems results in the search for their approximate solutions. Stratification of this class is conducted according to the complexity level, which is more complicated than the above-mentioned basic distinctions. The NP-complete theory has been developed in the direction of various issues of approximation. For this aim, approximation algorithms are being created. A below-given quotation, taken from the study [Impag-WWW, p. 2], shows that this research must be based on a subtler theory.

“Define SNP to be the class of properties expressible by a series of second order existential quantifiers, followed by a series of first order universal quantifiers, followed by a basic formula (a boolean combination of input and quantified relations applied to the quantified element variables). This class is considered for studying approximability of optimization problems”. The authors refer to Papadimitriou and Yannakakis [1991].

Another example how to deal with planning complexity involves coding a plan in the propositional calculus. After assigning logical values to variables, it is finally translated into the original planning problem. This method is described by Ernst et al [1997] at the beginning of his paper:

“Recent work by Kautz et al. [1992] provides tantalizing evidence that large, classical planning problems may be efficiently solved by translating them into

propositional satisfiability problems and translating the resulting truth assignments back into plans for original problems.”

The same paper shows (it has also been stated experimentally) that by using the above-mentioned method it is possible to reduce a number of variables by half whereas the formula length can be reduced by 80%. Consequently, planning problems, reconstructed as a result of their decoding from the simplified formulae, are made considerably easier.

The above examples fall within a general strategy of approximation and simplification, where the following directions may appear:

When, under a given mathematical model, a problem is too complicated, we simplify the model in question. However, care should be taken to make sure it still remains an approximation of the reality that is close enough not to ruin the accuracy of predictions.

If we do not decide to simplify a model, we have to be satisfied with what is only an approximate solution. This approach is successfully represented by genetic algorithms that is, imitating the process of the Darwin evolution in a specified population (e.g. of mathematical formulae or programs) with its laws of natural selection, heredity, struggle for survival (individuals who do not fulfill the criteria die) and chance mutation (which is reinforced when leading towards the criteria fulfillment). For instance, genetic algorithms cope well with the traveling salesman problem.

We look for an accurate solution but without being certain of getting it. In this case, there appears a necessity to accept a quite high level of the solution accuracy, which requires certain methods of the accuracy estimation.

Limitations of the results connected with these methods do not have to do much cognitive harm. In science, just as in everyday life, we are forced to simplify and approximate. Why should it be different in the sphere of the research that operates with algorithms? It is superior to the traditional research methods because it allows to estimate both precision diversion and their cognitive consequences.

3.4. In the process of dealing with complexity there exists a reverse direction, opposite to that advocating a deliberate agreement with the restrictions. As a starting point it also recognizes restrictions but only one of them is being considered. The fundamental limitation which has to be acknowledged consists in incompleteness of arithmetic as well as undecidability of logic. It is connected with the qualification of the range of algorithm power. In this respect, the orthodox position in computer science amounts to the

Church-Turing thesis. That is, to the claim that (roughly speaking) any device capable of algorithmic solving problems equals the computational power of the Universal Turing Machine (UTM).

Such a clear statement of limitations shows a field on which it would be possible to aim at getting superiority over UTM. To wit, there arises the following question whether the same problems, solved by UTM during too long a period of time (and therefore intractable), could be solved in a considerably quicker manner? The answer is positive. Some computational systems have appeared, due to which it is possible to cope with the complexity of different problems better. Here is their exemplifying overview.

Parallel computing takes place when a certain set of processors executes one task that is divided among them. **Distributed computing** appears when a computational process is divided among the computers remaining in the net; the data is being shared among them. Although each case involves different elements of the set (in the first case these are processors of the same computer whereas in the second one these are independent computers), a certain analogy takes place between the systems, which found its way in the title of the electronic magazine entitled *Journal of Parallel and Distributed Computing*. Both systems accelerate computing processes noticeably. Distributed computing deserves a special research aiming at its verification as a free market model, imitating the aspect which Friedrich Hayek called distributed (or local) economic knowledge as differentiated from centralized knowledge required by central planning system.

Interactive computing consists in the interaction between the system and the environment, as a result of which the system learns. The fact that there is no need to equip the system with highly complex algorithms, ready to face different circumstances, makes the essence of the obtained improvement. Instead, it is equipped with the program, controlling its learning on the basis of the information obtained from the environment, which makes the strategy incomparably more economic. A self-steering missile is an example of such a system. It behaves respectively to the obtained observation. The ability to react according to the environment requires a set to be equipped with suitable devices (input, output devices).

Cellular automata (CA) (mentioned earlier in the section regarding Prisoner's Dilemma) are called so because they consist of simple objects located in the cells, reminding of a pattern of the chessboard. Each object has a certain number of possible states (e.g. alive or dead; or white, black or other color, etc). Because these objects change their states, they undergo some evolution. It is done according to the imposed rules that condition

a choice of such-and-such state, depending on the situation in the environment (e.g. an object disappears after it has been squeezed by the presence of other individuals around). Self-reproduction is one of the processes taking place in the objects. A construction of automata, capable of self-reproduction with the help of the materials located in their environment, was a primary intention of von Neumann while his works on the construction of CA. Apart from the simplicity of the rules, underlying the CA behavior, it can often become unforeseeable. It allows them to be used to simulate unstable systems (chaotic), particularly studied by Stephen Wolfram. It has also been confirmed that CA has the power of UTM (the same range of the solvable problems). However, it is accompanied by an incomparably bigger efficiency.

Neural networks are highly simplified, physical (hardware) or logical (software) imitations of the nervous system. Their ability to learn gives them a basic advantage over UTM. Another difference is the fact that their actions remain only partly digital and partly analogue (which imitates analogue chemical conditions of the body, e.g. performance of the neuron transmitter). When it comes to the comparison with the UTM computational power, an orthodox view maintains that it is not bigger. On the other hand, the speed and ability to solve problems (to compute) are bigger. However, there is a group of non-conformists who firmly believe in the superiority of the networks, or in other words, in their ability to solve the problems which cannot be solved by UTM.

3.5. Together with this controversy let us consider the last point of the attacks on complexity. It should also be pointed out that a couple of other types of data processing (including quantum computation) will be omitted for the sake of a shortened character of the author's reasoning. The question whether an attack, frontal enough to go beyond a possibility limit of the Turing machine, can appear makes the last point to tackle. Going beyond a possibility limit has become known as **hypercomputation**.

The problem of hypercomputation ramifies into two issues. Although not under this name, one of them followed Gödel's thesis as regards the undecidability of arithmetic, and after the appearance of the undecidability results in logic in 1936. The question was whether the human brain was superior over the Turing machine. Gödel used to give a positive answer to that question whereas Turing (starting from 1950) used to answer it negatively. Apart from the Gödelian version of the negative answer, where the brain is comprehended independently from physics, there is a physical version, developed by Penrose [1989, 1994].

Penrose's view has been made radical in the ideas and projects of the researchers among whom Jack Copeland is acknowledged as taking a remarkable research and writing initiative. The notion of hypercomputation functions within that group. While Penrose defends the hypothesis that in Nature there are systems capable of solving the problems unsolvable by the Turing machine (meaning brains), hypercomputationalists go further, maintaining a possibility of imitating Nature. According to them, this could be done by the construction of technical devices that are superior over the Turing machine in terms of computational power. It also implies their advantage over equivalent systems, such as digital computers.

Paradoxically, hypercomputation makes attempts to refer to Turing (his surname, accompanied by his photo, was taken as a name of the hypercomputationalists' Internet site). This reference deserves a special attention for it allows to discuss an important and, at the same time, mysterious notion of oracle introduced by Turing in his work [1938]¹⁰. Because the work was his Doctor's thesis, supervised by Alonzo Church, it revealed two great names of the discoverers of the problem of computability. That is why historians of logic take a great interest in it.

Since the issue has been a subject of interpretative arguments, it cannot be presented without giving some specific interpretation (as long as it is not restricted only to quotations). The author of the present paper shares the view of A. Hodges, the author of the famous biography of Turing [1993], who returned to the issue in his lecture [2002] while his polemics with Copeland. Turing [1938] had made an attempt to formalize the notion of intuition understood as a factor that enables one to recognize the truth of the Gödelian statement. Without going deeper in the overview of the attempt's success and results, it is enough to note that Turing had introduced the notion of oracle as something that could compute incomputable functions. It is necessary to add here that he did not treat it as a hypothesis regarding the existence of such a real object (which, according to Copeland, he did) but rather as a fiction helpful for theoretical considerations. In this light, there is no basis to proclaim Turing a patron of superhypercomputationalists. What is more, there is no need to "reconcile" Turing [1938] with Turing [1950].

¹⁰ A caution against ambiguity of the term "oracle" present in literature must be taken. Apart from the original meaning, referring to Turing [1938], another meaning has appeared. It appears in the NP problem context, where a metaphor of the result guessing has also been called "oracle" (meaning something which is capable of guessing things hidden for others). "Oracle" as used in the second context has nothing to do with uncomputable numbers.

The success of the hypercomputationalist project depends on the truthfulness of the philosophical view stating that continuous or non-discrete processes (which include analogue computing) do not just appear as real but they really exist. Such "discrete" physicists as Ed Fredkin and Frank Tipler oppose this view. However, the project does not have to be defended on the basis of mere philosophical premises. It is the philosophy of continuity that can gain an experimental confirmation provided an analogue device (if proved to be predominant over the Turing machine in terms of its computational power) is constructed. The Havy Siegelman system [1999] described by its author in the book entitled *Neural Networks and Analog Computing: Beyond the Turing Limit* claims to achieve that aim. A summary of these ideas, included in the lecture (Siegelman [2001]), presents a perfectly described alternative as regards the disability to cross the Turing limit. The author puts it like this:

"[...] The theory of computational complexity requires the assumption of discrete computation and does not allow for other types of computational paradigms.

We consider a basic neural network model: finite number of neurons, recurrent interconnection, continuous activation function, and real numbers as weight. This model is considered "analog" for both the use of real numbers as weights that makes the phase space continuous, and the continuity of its activation function. In computational terms, this type of continuous flow (due to its activation function) is definitely a restriction in our model. There are no discrete flow commands such as "If z is greater than 0, compute one thing, otherwise continue in another computation path."

We show that the network [...] can compute anything that a digital machine does. Furthermore, due to its real-valued weights, it is inherently richer than the digital computational model and can compute functions that no digital computer can. The network, although transcending the Turing model, is still sensitive to resource constraints and still computes only a small subclass of all possible functions. It thus constitutes a well-defined super-Turing computational model."

As it is seen from the above-quoted extract, the computational power of the Siegelman network seems to be located somewhere between UTM and oracle in the Turingian sense [1938]. The question whether the network could be defined well enough as a theoretical object and whether, being a physical device, it will experimentally prove its superiority over UTM, remains open (a discussion of this topic taking place in the field allows one to note that).

3.6. Before the above-mentioned argument has been settled, we may rely on the knowledge about people as individuals as confronted with the knowledge about computers. Two human dispositions deserve a special attention here – our ability to accept axioms and the ability to ask questions. The use of these two dispositions along with our commission to perform digital tasks by machines (when they are better than people) will advance the research results to the maximum. Figuratively, if these two dispositions were to fall under the function performed by oracle in the Turingian sense, then an optimum research strategy would consist in a positive feedback between oracle and the machine.

When it comes to the ability to find axioms and recognize their truth, it is necessary to highlight that, apart from few marginal references, this issue is notoriously being neglected in discussions about AI. This fact seems to indicate that the most eager defenders of the claim to reduce human brains to UTM at this stage cannot imagine such an algorithmic procedure that would lead towards, for instance, the discovery of the axiom of choice¹¹. This issue is also omitted by the advocates of strong AI although they seem to be burdened with onus probandi that every axiom must be produced by the Turing machine. Though it is possible to imagine a production of axioms by genetic algorithms through the application of a consistency requirement as a probabilistic evolution criterion. In the long run, axioms are expected to be true, while the recognition of truth is a cognitive domain of the human brain.

A construction of the machine capable of making questions would mean another exam passed positively by AI. Among researchers, this issue is covered by silence as it is with the case of arriving at axioms. The silence is much too strange bearing in mind the fact that in the light of the popularity of the Turing thesis, many researchers should be expected to come up with its development when the nearness of the machine and brain is not

¹¹ A certain detail from the famous letter of Gödel to von Neumann written in 1956 gives evidence how the attention of logic and the theory of complexity concentrates on the problems of axiom demonstration and not on the search for axioms, which was precursory in relation to the theory of computational complexity and “P=NP?”. Gödel asks von Neumann what he thinks about a certain algorithm which, according to Gödel, can fully help a mathematician (as quoted from the text translated from German [Buss-WWW]): “It would obviously mean that in spite of the undecidability of Entscheidungsproblem, the mental work of a mathematician concerning Yes-No questions could be completely² replaced by a machine.” Footnote 2 after *completely* goes (italics – mine): except for the setting up of axioms. The fact that the matter deserves only a footnote in the letter allows us to think that the impossibility of the axiom acquisition on the basis of algorithms remained beyond discussion both for Gödel as well as (in Gödel’s opinion) for the addressee. A complete text accompanied by a historical commentary is given by Hartmanis [1989].

measured only (as it was proposed by Turing) by the similarity of the answers to given questions but also as to justification, originality and penetration of the questions being asked.

As long as such a machine does not exist, equality between artificial intelligence and natural one remains beyond the dream frontiers. To give answers it is enough to have a database put into memory and a program of their logical processing. To ask questions it is necessary, as it was noted by Ch. S. Peirce, to possess an ability to get irritated or annoyed. Irritation appears due to one’s lack of knowledge, ambiguity of notions or some contradictions. Moreover (let us develop Peirce’s thought) there are situations in which questions derive from curiosity. Either is a state of the mind that is not separable from a certain emotional state. As far as it is known now, this state can be achieved only by a living protein and not by pieces of silicon or other material carriers of artificial intelligence (von Neumann [1958] claimed that a nervous system has a logic of its own, other than that of digital machine; see also Marciszewski’s commentary [1996b]).

The above-given remarks are to reinforce the claim that an optimum research strategy of the future will not consist in successive replacing people by computers but in the parallel increase of their potentiality. Computing abilities of a computer increase together with the creation of quicker algorithms and perfection of equipment. It will result in the growth of knowledge that, in turn, will increase the rise of new questions and new axioms. This is bound to bring new research tasks for humans and computers, and so on, infinitely – provided that both the living intelligence and the mechanical intelligence are to exist in infinitum.

Translated by *Renata Botwina*

References

- Ajdukiewicz, Kazimierz, *Pragmatic Logic* (trans. Olgierd Wojtasiewicz). Reidel, Dordrecht, PWN, Warszawa 1974.
- [Buss-WWW] Buss, R. Samuel, “On Gödel’s Theorem on the Lengths of Proofs II: Lower Bounds for Recognizing Symbol Provability”. Displayed at: math.ucsd.edu/sbuss/ResearchWeb/publ.ps
- Carnap, Rudolf, *Meaning and Necessity*. University of Chicago Press, 1958.
- Chaitin, Gregory J., *The Limits of Mathematics, The Unknowable, Exploring Randomness, Conversations with a Mathematician*. Springer-Verlag, Berlin etc. 1998, 1999, 2001, 2002.

- Church, Alonzo, "An unsolvable problem of elementary number theory". *Am. J. Math.* 58, 345–363, 1936.
- Church, Alonzo, "A note on the Entscheidungsproblem". *J. of Symbolic Logic* 1, 40–41 and 101–102, 1936.
- Copeland, Jack and Proudfoot, Diane, "Introduction to Hypercomputation: Computing the Uncomputable". Copyright by the authors, July 2000. www.alanturing.net/turing_archive/pages/Reference
- Copeland, Jack, "Turing's O-Machines of 1938", May 2000. www.alanturing.net/turing_archive/pages/Reference%20Articles/hypercomputation/
- Cottrell, Allin and W. Paul Cockshott. "Calculation, Complexity and Planning: The Socialist Calculation Debate once again". www.dcs.gla.ac.uk/wpc/reports/. Compare the same authors "Socialist Planning after the Collapse of the Soviet Union", *Revue European des Sciences Sociales*, 167–186. Publisher: Academic Press, 1993.
- Davies, Paul, *The Mind of God* (Chapter 5, Part "Unknowable", 128–134). Simon and Schuster, New York 1992.
- Grim, Patrick, "The Undecidability of the Spatialized Prisoner's Dilemma". *Theory and Decision* 42, 53–80, 1997. See also: www.sunysb.edu/philosophy/faculty/pgrim/SPATIALP
- Ernst, Michael D.; Millstein Todd D.; Weld, Daniel, "Automatic SAT-Compilation of Planning Problems". *Proc. the 15th Intern. Joint Confer. on Art. Intell.*, Nagoya, Japan, August 23–29, 1169–1176, 1997. pag.lcs.mit.edu/mernst/pubs/satcompile-ijcai97-abstract.html
- Gödel, Kurt, "Die Vollständigkeit der Axiome des logischen Funktionenkalküls". *Monatshefte für Mathematik und Physik* 37, 349–360, 1930.
- Gödel, Kurt, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme – I". *Monatshefte für Mathematik und Physik* 38, 173–198, 1931.
- Gödel, Kurt, "Über die Länge von Beweisen". *Ergebnisse eines mathematischen Kolloquiums*, Heft 7, Franz Deuticke, Leipzig und Wien 1936.
- Hartmanis J. and Stearns R., "On the computational complexity of algorithms". *Transactions of the AMS* 117, 285–306, 1965.
- Hartmanis J., "Gödel, von Neumann and the P=NP? problem". *Bulletin of the European Association for Theoretical Computer Science* 38, 101–107, 1989.
- Heller, Michał, *Wszechświat u schyłku stulecia* [The Universe in the Close of the Century]. Znak, Kraków, 1994.

- Hilbert D. und Ackermann W., *Grundzüge der theoretischen Logik*. Springer, Berlin 1928.
- Hodges, Andrew, "What would Alan Turing have done after 1954?". Lecture at the Turing Day, Lausanne, 2002. www.turing.org.uk/philosophy/lausanne2.html
- Impag-WWW] Impagliazzo, Russell; Paturi, Ramamohan, and Zane, Francis, "Which Problems have Strongly Exponential Complexity?" (Research supported by a NFI grant). cm.bell-labs.com/cm/ms/who/francis/papers/focs98-subexp.pdf
- Kautz H. and Selman B., "Planning as Satisfiability". *Proc. 10th Eur. Conf. on AI*, 359–363, Wiley, Vienna 1992.
- Marciszewski, Witold and Roman Murawski, *Mechanization of Reasoning in a Historical Perspective*. Editions Rodopi, Amsterdam 1995.
- Marciszewski, Witold, "Leibniz's Two Legacies. Their Implications for Knowledge Engineering". *Knowledge Organization* vol. 23, no. 2, 1996a.
- Marciszewski, Witold, "Hätte Leibniz von Neumanns logischen Physikalismus geteilt?". *Beiträge zur Geschichte der Sprachwissenschaft*, 6.2, 1996b.
- Neyman, A., "Bounded Complexity Justifies Cooperation in the Finitely Repeated Prisoners' Dilemma". *Economic Letters* 19, pp. 227–229, 1985.
- Papadimitriou C. and M. Yannakakis, "Optimization, Approximation and Complexity Classes". *Journal of Computer and System Sciences* 43, 425–440, 1991.
- Penrose, Roger, *The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics*. Oxford Univ. Press, Oxford etc. 1989.
- Penrose, Roger, *Shadows of the Mind*. Oxford University Press, New York 1994.
- Siegelmann, Hava T., *Neural Networks and Analog Computation: Beyond the Turing Limit*. 1999.
- Siegelman, Hava T., "Computing Beyond Digital Computers by Neural Models", a lecture at the conference "Hypercomputation Workshop", 2000, University College, London. www.alanturing.net/turing_archive/conference/hyper/abstracts.html
- [Sirridney-WWW] <http://www.sirrodney.nu/public/english/Misc>
- Tarski, Alfred; Mostowski, Andrzej; Robinson, Raphael M., *Undecidable Theories*. North Holland, Amsterdam, 1968.

- Turing, Alan, "On Computable Numbers, with an Application to the Entscheidungsproblem". *Proc. of the London Math. Society*, Series 2, 42, pp. 230–265, 1936.
- Turing, Alan, "Systems of Logic Based on Ordinals". *Proc. of the London Math. Society*, Series 2, 45, pp. 161–228, 1939. Quoted as Turing [1938].
- Turing, Alan, "Intelligent machinery: Report, National Physics Laboratory". In: B. Meltzer and D. Michie (eds), *Machine Intelligence 5*, Edinburgh Univ. Press, Edinburgh 1969.
- Turing, Alan, "Computing Machinery and Intelligence". *Mind* 59, 433–460, 1950.
- von Mises, Ludwig, *Human Action. A Treatise on Economics*. Contemporary Books, Chicago 1966.
- von Neumann, John, *The Computer and the Brain*. Yale Univ. Press, New Haven 1958.
- Wolfram, Stephen, "Undecidability and Intractability in Theoretical Physics". *Physical Review Letters* 54, 735–738, 1985. Displayed at: www.stephenwolfram.com/publications/articles/physics/85-undecidability/index.html
- Wolfram, Stephen, *Cellular Automata and Complexity*. Addison-Wesley, 1994.
- Wolfram, Stephen, *A New Kind of Science*. Wolfram Media, Inc., 2002.
- Witold Marciszewski
University of Białystok
and Białystok School of Public Administration
Section of the Methodology of Social Sciences
Home page: www.calculemus.org/witmar-pol.html

Anna Gomolińska*
University of Białystok

ROUGH INFORMATION GRANULES IN SOCIAL AGENT SYSTEM MODELLING**

Abstract. The aim of the paper is to present rough granular methodology as a promising and serviceable tool in modelling of both social agents, interacting under uncertain and incomplete information, as well as systems of such agents. In this article, we focus upon information granules and fundamentals of the rough set theory. An agent's architecture in terms of rough information granules, discussed in part in [23], and other special issues concerning social agent system modelling will be considered elsewhere.

1. Introduction

Modelling and analysis of social agent systems have been attracted many researchers not only in social sciences but also in scientific disciplines and research fields to which investigation of social phenomena was primarily considered as irrelevant, to mention computer science. The fundamental notion focusing the attention of a vast number of researchers is *agent*. Apart from traditional interests in studying individual or collective human agents, institutions, companies, and the nature, there is a rapidly growing interest in investigation of populations of animals (e.g., ants, bees, or birds), artificial agents like robots or computer programs, and various kinds of collective agents (e.g., teams consisted of human agents and intelligent computer programs). In general, an agent is an *autonomous* object able to act on behalf of others or for itself.

A number of types of agents can be observed, and it would be too hard – or even hardly possible – to give a complete typology here. As regarding agents' attitude to activity, *pro-active* (or *goal-oriented*) and *reac-*

* This work is dedicated to Ewa Roszkowska with whom the author collaborated on the Burns generalized game theory. Many thanks go to Andrzej Skowron for valuable discussions on modelling of multiagent systems from the rough-set standpoint.

** The research reported in this paper is a part of the project entitled *Undecidability and Algorithmic Intractability in the Social Sciences*, supported by the Polish Ministry of Science, grant no. 2H01A03025.

tive agents can be distinguished. Pro-active agents are goal-driven, whereas the causes for activity of reactive agents are events. We can also say about goal-oriented agents that they act according to *consequentialist* (or *instrumental-rationality*) modality as they mainly or solely pay attention to the realization of their goals and to achieving the best outcomes. To the contrary, *normatively-oriented* agents pay attention to or judge on the basis of norms and qualities of actions. There are also other action modalities like routine (habitual) modality, symbolic communication and rituals, play, and emotionally-driven acting [7, 9, 12]. *Social* agents, i.e., agents which interact with others can be *cooperative* or not. However, by *socially-oriented* agents we mean agents taking into account interests of others, i.e., agents which are not pure egoists. By a *cognitive* agent, we understand an agent with cognitive faculties like understanding, reason, judgment, perception, learning, adaptation, and so on. Such agents can have *beliefs* and *knowledge*. Where reason or adaptation are, for example, of particular interest, we can speak of *reasoning* agents or *adaptive* agents, respectively. Agents relying mainly on reason in their acting are called *rational*. By an *intelligent* agent we mean an agent granted or having developed some form of intelligence, so intelligent agents are cognitive as well. Modelling of *emotionally-embedded* agents and *intention-driven* agents are also of interest in the present agent technology [13, 14, 15]. From the standpoint of social sciences, studying of social systems of human agents is, of course, of the primary concern. On the other hand, computer scientists are mainly interested in systems of artificial agents or in mixed systems constituted both of human and artificial agents.

Real-life social systems and advanced systems of artificial or half-artificial agents are complex systems being subject to frequent changes and modifications. Usually, agents have to act and interact under incomplete, uncertain, and vague information about the situation of interaction, other participants, and even themselves. Also, agents have limited abilities (i) to identify the situation of interaction, the involvement of particular agents in this situation, the roles to be played, and the goals to be achieved, (ii) to predict the behaviour of others and the results of interaction, (iii) to make proper decisions, (iv) to perform the actions determined, etc., so the issue of modelling of such systems seems to be highly complicated or even intractable. As a matter of fact, traditional tools of mathematics, logic, and computer science turned out to be insufficient and often inadequate for the purpose of modelling of complex systems of agents working under uncertain and incomplete information. Soft computing methods like, e.g., rough sets, fuzzy sets, and neural networks are designed to deal just with imprecise, vague, and incomplete information. Usage of these or similar methods seems

to be a reasonable and proper thing to do. In this paper, we propose a rough granular approach, where methods and tools of granular computing, based on recent achievements of rough set theory, are used.

The notion of a *rough set* was introduced by Pawlak in the early 80's of the 20th century as a result of research on approximate classification of objects in *information systems* [39, 40, 41, 42, 44]. The theory and the methodology of rough sets, worked out by Pawlak and other researchers, are suitable for dealing with vague and incomplete information, and are applicable to a vast number of problems like data and concept analysis, classification and, in particular, decision making, generation of classification and association rules, synthesis and analysis of complex objects, ontology formation, analysis of conflicts, reasoning under uncertainty, and many others. Rough set methods are based on Boolean reasoning, and the typical example of a rough set problem to which Boolean reasoning can be employed is computation of minimal reducts of the set of attributes of an information system. This problem is known to be NP-hard [58]. Since this one and many other problems faced in knowledge engineering are computationally hard, people search for heuristics which are computationally much more attractive although they provide us with suboptimal and approximate solutions only.

In general, we aim at modelling of social, cognitive, and communicating agents, either natural or artificial. From our perspective, agents and their systems are complex objects which can be represented by rough information granules. In this article, we just present information granules as a promising tool for agent system modelling. Some attempts to use information granules in social agent system modelling have been already made, see, e.g., research articles on representation of human agents – either individual or collective ones – and on explanation of social interactions of such agents in terms of rule complexes [7, 9, 10, 11, 12, 20, 22, 23]. As regarding the agent architecture, an agent may be viewed as a complex structure built of rough information granules, and similarly for a multiagent system. Rules of some knowledge representation language play an important role here as, on one hand, they can be used as labels for information granules and, on the other hand, they are themselves objects to form information granules. An agent's knowledge and belief bases, value and norm systems, judgment system, classification procedures, and (inter)action modules can be modelled as information granules consisting of rules. Schemes of interaction and systems of agents can also be viewed as such granules. However, representation of agents and their systems in terms of granules is only one issue in our agenda. The list of important questions is much longer, for instance, how

agents reason, judge, and understand in an approximate way, how they follow rules, determine (inter)actions, and interact, how agents communicate with and understand one another, as well as how they derive procedures for collective classification of objects (and in particular, decision making). Formation of rough ontologies of concepts is another interesting research problem [37, 64]. It is already known that using sensory data as the only source of information is insufficient for learning complex concepts, so agents are assumed to have some domain knowledge as well. Examples of sources of domain knowledge are teachers, socio-cultural traditions, codes, handbooks, various know-how books, and web pages.

Throughout the paper, the power set of a set X , the cardinality of X , and the Cartesian product $X \times X$ will be denoted by $\wp X$, $\#X$, and X^2 , respectively. As usual, \circ will denote composition of relations. The propositional connectives of conjunction and disjunction will take the precedence of implication and double implication. A family \mathcal{X} of non-empty subsets of X is a covering of X if $\bigcup \mathcal{X} = X$. A covering \mathcal{X} is called a partition of X in case for any $Y, Z \in \mathcal{X}$, $Y \cap Z \neq \emptyset$ implies $Y = Z$.

Given a set X and a binary relation $\rho \subseteq X^2$, the *image* and the *co-image* of a set $Y \subseteq X$, $\rho^{\rightarrow} Y$ and $\rho^{\leftarrow} Y$, respectively, are defined along the standard lines. In particular, for any $x \in X$,

$$\rho^{\rightarrow}\{x\} = \{y \in X \mid (x, y) \in \rho\} \ \& \ \rho^{\leftarrow}\{x\} = \{y \in X \mid (y, x) \in \rho\}. \quad (1)$$

Every *reflexive* relation ρ on X (i.e., every $\rho \subseteq X^2$ such that $\forall x \in X. (x, x) \in \rho$) determines a covering of X , viz., $\{\rho^{\leftarrow}\{x\} \mid x \in X\}$. In the context of approximation spaces, reflexive relations are called *similarity* relations as well. Furthermore, ρ is *symmetric* if $\forall x, y \in X. ((x, y) \in \rho \Rightarrow (y, x) \in \rho)$. It is worth mentioning that ρ is symmetric if and only if ρ is equal to its converse relation ρ^{-1} . Relations being both reflexive and symmetric are referred to as *tolerance* relations. Obviously, every tolerance relation is also a similarity relation. In the sequel, ρ is *transitive* if $\forall x, y, z \in X. ((x, y) \in \rho \ \& \ (y, z) \in \rho \Rightarrow (x, z) \in \rho)$. Transitive tolerance relations are known as *equivalence* relations. In this case, $\rho^{\leftarrow}\{x\}$ is called the *equivalence class* of x and denoted by $[x]_{\rho}$ (or, simply, $[x]$ if ρ is understood)¹. The covering of X determined by ρ , $\{[x]_{\rho} \mid x \in X\}$, is the partition of X . Moreover, there is a one-to-one correspondence between the set of all equivalence relations on X and the set of all partitions of X . Within the framework of rough sets and Pawlak's information systems, equivalence relations on the universe of objects are understood as *indiscernibility* relations.

¹ Note that $\rho^{\leftarrow}\{x\} = \rho^{\rightarrow}\{x\}$ by symmetry of ρ .

The rest of the paper is organized as follows. In Sect. 2, a general notion of an information granule is introduced. Granules in Pawlak's information systems and neighbourhoods of objects are discussed in Sect. 3. Formulas and rules of a description language as labels for information granules are presented in Sect. 4. Next, rule complexes and, in particular, rough classifiers are recalled in Sect. 5. Section 6 contains remarks on collective agents and, in particular, economic clusters. In the sequel, the classical Pawlak rough set model is described in Sect. 7. Section 8 is devoted to two of several extensions of this model. The last section contains a brief summary.

2. Information Granules

The notion of an *information granule* (or simply, *granule*) was introduced by Zadeh in the context of fuzzy set analysis of complex systems [77], and elaborated in a series of research articles [52, 53, 61, 62, 63, 65, 66, 72, 78, 80]. According to Zadeh's proposal, an *information granule* is a clump of objects of some class, drawn together on the basis of indiscernibility, similarity, or functionality. In fact, the term 'indiscernibility' may be dropped since the property of being indiscernible is a particular, "perfect" case of being similar.

The definition of a granule leaves a lot of freedom as regarding the interpretation. By way of example, a set of objects X is often viewed as an information granule if all members of this set are similar to a given distinguished object, say u . Calling X an information granule is justified since all elements of X function as 'objects similar to u '. From another standpoint, X is treated as a granule formed on the basis of similarity, yet similarity is understood as similarity with respect to u . Also, a granule may consist of objects of varied sorts like rules and sets of rules. *Rule complexes* used to model agents as well as their systems and interactions in *generalized game theory*, proposed and developed by Burns and his collaborators [7, 9, 10, 11, 12, 20, 22, 23], are examples of such heterogeneous information granules. The inner structure of granules does not matter, either. From the mathematical point of view, granules may be ordinary sets of objects, ordered sets of objects, sequences of objects, tables of objects, and so on. However, not every set of objects is treated as a granule. What really matters is the reason (viz., similarity or functionality) for drawing objects together.

The idea of computing with granules of objects instead of single objects, also attributed to Zadeh [77, 78, 79, 80], has found a number of fol-

lowers (see, e.g., [22, 31, 35, 52, 53, 75, 81]). Recently, *granular computing* is a dynamically developing and expanding research area which makes use of various methods and tools, e.g. those offered by rough set theory, fuzzy set theory, or neural networks. Methods of granular computing not only serve the purpose of modelling of complex objects like agents and systems of agents but they are also suitable for specification and solving of a variety of research problems related to functioning of such objects. *Computing with words* (CW for short), appearing in the literature in connection with granular computing, is a step in performing computations on granules. Words are simply linguistic labels for information granules, for example, a formula may serve as a label for the set (i.e., granule) of all objects satisfying the formula. In computing with words, we manipulate with labels of granules instead of numbers, and – as pointed out by Zadeh [79] –

“[t]here are two major imperatives for computing with words. First, computing with words is a necessity when the available information is too imprecise to justify the use of numbers. And second, when there is a tolerance for imprecision which can be exploited to achieve tractability, robustness, low solution costs and better rapport with reality. Exploitation of the tolerance for imprecision is an issue of central importance in CW.”

Objects to be modelled by granules are creatures or productions which undergo various changes and modifications, and therefore transformation of granules is an important issue in granular computing. Such questions are addressed as how given granules are related to one another (inclusion, closeness, similarity), formation of new granules from elementary objects and/or granules already obtained (aggregation, decomposition), expansion, contraction, and revision of granules. As a result of aggregation, a new granule is obtained from given granules. To the contrary, decomposition of a granule leads to a collection of granules derived from the primary granule. The last three forms of transformation are named by us after well-known operations of belief change [3, 18]. Briefly speaking, expansion of a granule X consists in extending it with an object or a granule of objects as opposite to contraction of X , where some objects or granules are removed from X . Revision of X by an elementary or compound object is a composition of contraction with expansion. In particular, revision can take the form of replacement, where an object being a member or a part of X is replaced by another object. Operations on granules are constrained by conditions specifying numerical or symbolic values of variables. So, for instance, a constraint may say that the result of an aggregation of granules should be a granule close to the degree at least t to a given exemplary granule X , where t is a real number

from the unit interval, or that the resulting granule should differ from X as little as possible.

The definition of the notion of an information granule covers a broad class of cases. Examples of information granules are sets of similar rows in Pawlak information systems or, in other terms, sets of similar objects of the system, sets of neighbourhoods of objects, concepts in approximation spaces and approximations of these concepts, pairs of sets of objects satisfying premises and conclusion(s) of a rule, respectively, sets of rules like algorithms and classifiers, rule complexes representing agents and games [20, 22, 23], computational grids (i.e., collections of distributed resources which can be used to execute large-scale computations), coalitions and teams of cooperating agents, and economic clusters [6, 56, 57]. In the following sections, we give more details about some cases of information granules just mentioned.

3. Granules in Pawlak's Information Systems

The notion of an *information system* (IS for short) was introduced by Pawlak [39, 42]. In a recent formulation, an information system is a pair $\mathcal{A} = (U, A)$ of finite non-empty sets of objects and attributes, respectively. The set of all objects considered, U , is the universe of \mathcal{A} . Henceforth, elements of U (resp., A) will be denoted by u (resp., a), possibly with sub/superscripts. Each attribute a is viewed as a mapping $a : U \mapsto V_a$ which assigns to every object u , a value $a(u) \in V_a$. Let $V = \bigcup\{V_a \mid a \in A\}$. Any pair (a, v) , where $a \in A$ and $v \in V_a$ is called a *descriptor*. From a different standpoint, every object u may be viewed as a mapping $u : A \mapsto V$ such that $u(a) \in V_a$, i.e., $U \subseteq \prod\{V_a \mid a \in A\}$. From the mathematical point of view, every object u is, thus, a set of descriptors $u = \{(a, a(u)) \mid a \in A\}$. Therefore, granulation of objects on the basis of information contained in \mathcal{A} and formation of granules from sets of descriptors of \mathcal{A} are equivalent.

As well-known, every set of attributes $B \subseteq A$ induces an equivalence relation of *B-indiscernibility* of objects, ind_B , defined for any objects $u, u' \in U$ by

$$(u, u') \in \text{ind}_B \Leftrightarrow \forall a \in B. a(u) = a(u'). \quad (2)$$

In other words, $(u, u') \in \text{ind}_B$ if and only if for every $a \in B$, $(a, u(a)) = (a, u'(a))$, i.e., if and only if $u|_B = u'|_B$. The equivalence class of u , $[u]_B$, consisting of all objects which cannot be discerned from u by means of attributes of B , is a *B-elementary* information granule.

Mappings on U , assigning information granules to objects of U , may be called *granulation mappings*. Granulation mappings of the form $\Gamma : U \mapsto \wp U$, where for every $u \in U$, $u \in \Gamma u$, are referred to as *uncertainty mappings* [60]. $[\cdot]_B$ which assigns to every object u , its equivalence class $[u]_B$ is an example of an uncertainty mapping. Now, we recall a pretty universal recipe for construction of uncertainty mappings described, e.g., in [45]. To this end, suppose that for every attribute $a \in B \subseteq A$, a metric and a threshold function, $\delta_a, f_a : (V_a)^2 \mapsto [0, +\infty)$, respectively, are given. Define a family of uncertainty mappings $\Gamma_a : U \mapsto \wp U$, parameterized by attributes of B , as follows, for any objects u, u' :

$$u' \in \Gamma_a u \Leftrightarrow \delta_a(a(u), a(u')) \leq f_a(a(u), a(u')) \quad (3)$$

In words, u' is a member of the granule assigned to u by the mapping Γ_a if and only if the distance between the values of a at u, u' , given by δ_a , does not exceed the threshold value $f_a(a(u), a(u'))$. Next, a global uncertainty mapping, Γ , may be defined as the intersection of the mappings Γ_a , i.e.,

$$\Gamma u \stackrel{\text{def}}{=} \bigcap \{ \Gamma_a u \mid a \in B \}. \quad (4)$$

Each Γ_a induces a tolerance relation ϱ_{Γ_a} , given by

$$(u', u) \in \varrho_{\Gamma_a} \stackrel{\text{def}}{\Leftrightarrow} u' \in \Gamma_a u, \quad (5)$$

which is not an equivalence relation in general. Therefore, the relation ϱ_{Γ} , defined by

$$(u', u) \in \varrho_{\Gamma} \stackrel{\text{def}}{\Leftrightarrow} u' \in \Gamma u, \quad (6)$$

is merely a tolerance relation as well.

A *decision system* is an IS of the form $\mathcal{A} = (U, C \cup D)$, where C, D are non-empty disjoint sets of *condition* and *decision attributes*, respectively. Clearly, \mathcal{A} is the result of merging of ISs $\mathcal{C} = (U, C)$ and $\mathcal{D} = (U, D)$. The data table corresponding to \mathcal{A} is referred to as a *decision table* [39, 42, 43]. In most cases, D consists of only one decision attribute, say d . Then, we often write $\mathcal{A} = (U, C, d)$ instead of $\mathcal{A} = (U, C \cup \{d\})$.

As the set of descriptors $\{(a, a(u)) \mid a \in A\}$ represents the information accessible to an agent about an object u , information systems contain the available information about a collection of objects. From the standpoint of functionality, these information systems as well as their representations in the form of data tables serve as granules comprising information about classes of objects.

Example 3.1. By way of illustration, let us consider a decision system $\mathcal{A} = (U, A)$, where U consists of 8 objects (scholarship applications), denoted by $0, \dots, 7$, and A consists of 6 condition attributes a_0, \dots, a_5 (4 numerical and 2 nominal) and two decision attributes d_0, d_1 (1 numerical and 1 nominal). Every object is specified in terms of such attributes as: studies (a_0), year of studies (a_1), mean result (a_2), income of the family per month per person (a_3), age (a_4), and place of residence (a_5). The range of a_0 is $V_{a_0} = \{m, ch, ph\}$, where m stands for 'mathematics', ch - 'chemistry', ph - 'philosophy'; $V_{a_1} = \{1, 2, 3\}$; $V_{a_2} \subseteq [3.0, 5.0]$; $V_{a_3} \subseteq [800, 1200]$; $V_{a_4} = \{19, \dots, 22\}$; and $V_{a_5} = \{c, t, v\}$, where c stands for 'city', t - 'town', and v - 'village'. Furthermore, the decision attribute d_0 specifies whether or not the application is accepted and the form of a scholarship awarded. Its range is $V_{d_0} = \{s0, s1, s2, s3\}$, where $s0$ represents the negative decision, $s1$ is for 'social scholarship', $s2$ - 'research scholarship', and $s3$ - 'both social and reseach scholarship'. The second decision attribute, d_1 , specifies the amount of money awarded per month, and $V_{d_1} \subseteq [0, 500]$. System \mathcal{A} is presented in Table 1.

Table 1

The decision system \mathcal{A}

| u | a_0 | a_1 | a_2 | a_3 | a_4 | a_5 | d_0 | d_1 |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | m | 2 | 3.45 | 950 | 20 | c | s1 | 200 |
| 1 | m | 1 | 4.10 | 820 | 19 | v | s3 | 400 |
| 2 | ph | 1 | 4.30 | 850 | 19 | v | s3 | 420 |
| 3 | ch | 2 | 3.70 | 1130 | 20 | c | s0 | 0 |
| 4 | ph | 3 | 4.55 | 1050 | 22 | t | s2 | 300 |
| 5 | m | 2 | 3.75 | 980 | 20 | v | s1 | 180 |
| 6 | ch | 3 | 4.40 | 1200 | 21 | t | s2 | 250 |
| 7 | ch | 2 | 3.95 | 850 | 21 | c | s3 | 380 |

With each a_i , we can associate an uncertainty mapping Γ_{a_i} , and similarly for the decision attributes. For example, let

$$u' \in \Gamma_{a_i} u \stackrel{\text{def}}{\Leftrightarrow} \begin{cases} a_i(u') = a_i(u) & \text{if } i = 0, 1, 4, 5, \\ |a_i(u') - a_i(u)| \leq 0.25 & \text{if } i = 2, \\ |a_i(u') - a_i(u)| \leq 75 & \text{if } i = 3. \end{cases} \quad (7)$$

Γ_{a_i} generate partitions of U for $i = 0, 1, 4, 5$, viz., $\Gamma_{a_0}^{\rightarrow} U = \{\{0, 1, 5\}, \{2, 4\}, \{3, 6, 7\}\}$, $\Gamma_{a_1}^{\rightarrow} U = \{\{1, 2\}, \{0, 3, 5, 7\}, \{4, 6\}\}$, $\Gamma_{a_4}^{\rightarrow} U = \{\{1, 2\}, \{0, 3, 5\},$

$\{6, 7\}, \{4\}$, and $\Gamma_{a_5}^{-1}U = \{\{0, 3, 7\}, \{4, 6\}, \{1, 2, 5\}\}$. Thus, e.g. in the case of Γ_{a_0} , the elementary information granules are the equivalence classes $\{0, 1, 5\}, \{2, 4\}$, and $\{3, 6, 7\}$. Values of the uncertainty mappings Γ_{a_2} and Γ_{a_3} are given in Table 2. These mappings generate coverings of U which, nevertheless, are not partitions. Mappings Γ_{a_i} can be aggregated in many ways. For instance, we can derive the following uncertainty mappings Γ, Γ' (see Table 2):

$$\Gamma u \stackrel{\text{def}}{=} \Gamma_{a_0}u \cap \Gamma_{a_2}u \ \& \ \Gamma' u \stackrel{\text{def}}{=} \Gamma_{a_2}u \cap \Gamma_{a_5}u \quad (8)$$

Observe that $\Gamma^{-1}U$ is a partition of U , whereas $\Gamma'^{-1}U$ is merely a covering.

Table 2

Values of $\Gamma_{a_2}, \Gamma_{a_3}, \Gamma$, and Γ'

| u | $\Gamma_{a_2}u$ | $\Gamma_{a_3}u$ | Γu | $\Gamma' u$ |
|-----|------------------|-----------------|------------|---------------|
| 0 | $\{0, 3\}$ | $\{0, 5\}$ | $\{0\}$ | $\{0, 3\}$ |
| 1 | $\{1, 2, 7\}$ | $\{1, 2, 7\}$ | $\{1\}$ | $\{1, 2\}$ |
| 2 | $\{1, 2, 4, 6\}$ | $\{1, 2, 7\}$ | $\{2, 4\}$ | $\{1, 2\}$ |
| 3 | $\{0, 3, 5, 7\}$ | $\{3, 6\}$ | $\{3, 7\}$ | $\{0, 3, 7\}$ |
| 4 | $\{2, 4, 6\}$ | $\{4, 5\}$ | $\{2, 4\}$ | $\{4, 6\}$ |
| 5 | $\{3, 5, 7\}$ | $\{0, 4, 5\}$ | $\{5\}$ | $\{5\}$ |
| 6 | $\{2, 4, 6\}$ | $\{3, 6\}$ | $\{6\}$ | $\{4, 6\}$ |
| 7 | $\{1, 3, 5, 7\}$ | $\{1, 2, 7\}$ | $\{3, 7\}$ | $\{3, 7\}$ |

3.1. Neighbourhoods of Objects

As mentioned in the preceding section, an uncertainty mapping Γ on U assigns to every object u , a granule of information Γu . One can easily see that the corresponding binary relation ϱ_Γ , defined like in (6), is a similarity relation. Objects similar to a given object may be viewed as close to that object. As a consequence, Γu may be treated as a neighbourhood of u . Obviously, a number of various uncertainty mappings on U may be taken into account at the same time, say $\Gamma_0, \dots, \Gamma_m$. These mappings may be aggregated like we did before (cf. (4)) or they may be considered separately. In the latter case, with every object u , there is associated a global information granule $\Gamma u = (\Gamma_0 u, \dots, \Gamma_m u)$ being, in fact, a sequence of neighbourhoods of u .

Example 3.2. In connection with the previous example, note that the following global granules $\Gamma''0 = (\{0, 1, 5\}, \{0, 3, 5, 7\}, \{0, 3\}, \{0, 5\}, \{0, 3, 5\}, \{0, 3, 7\})$ and $\Gamma''1 = (\{0, 1, 5\}, \{1, 2\}, \{1, 2, 7\}, \{1, 2, 7\}, \{1, 2\}, \{1, 2, 5\})$, being sequences of neighbourhoods generated by Γ_{a_i} ($i = 0, \dots, 5$, respectively), can be associated with objects 0, 1.

There arises a question how this happens that so many uncertainty mappings can come into play. First, different subsets of attributes B may lead to different uncertainty mappings. Secondly, starting with an uncertainty mapping Γ_0 and taking into account (6), we can derive a sequence of uncertainty mappings Γ_i ($i = 0, 1, \dots$) such that for any objects u, u' :

$$u' \in \Gamma_i u \stackrel{\text{def}}{\Leftrightarrow} (u', u) \in \underbrace{\varrho_{\Gamma_0} \circ \dots \circ \varrho_{\Gamma_0}}_{i+1} \quad (9)$$

In particular, $u' \in \Gamma_1 u$ if and only if $(u', u) \in \varrho_{\Gamma_0} \circ \varrho_{\Gamma_0}$, i.e., if and only if there is u'' such that $(u', u'') \in \varrho_{\Gamma_0}$ and $(u'', u) \in \varrho_{\Gamma_0}$, i.e., if and only if there is u'' such that $u' \in \Gamma_0 u''$ and $u'' \in \Gamma_0 u$. ϱ_{Γ_0} is reflexive, so is an arbitrary composition of ϱ_{Γ_0} with itself. Therefore, $\Gamma_0 u, \Gamma_1 u, \Gamma_2 u, \dots$ may be understood as the granules of the nearest neighbours, the 1-step neighbours, the 2-step neighbours, and so on. It can happen that for some number i , the relation induced by Γ_i is an equivalence relation². Since any further iteration does not lead to new uncertainty mappings, the procedure may be stopped.

Example 3.3. Continuing Example 3.1, it is worth noting that $\varrho_{\Gamma_{a_3}} \circ \varrho_{\Gamma_{a_3}}$ is an equivalence relation, so the process of granule production may be stopped at $i = 1$. The corresponding partition of U is equal to $\{\{0, 4, 5\}, \{1, 2, 7\}, \{3, 6\}\}$. For example, the granule of the 1-step neighbours of the object 0 equals to $\{0, 4, 5\}$.

4. Formulas and Rules as Labels of Granules

Adapting slightly Zadeh's idea of computing with words, formulas and rules may be viewed as linguistic labels representing information granules. Computation on granules is performed with help of the labels attached to these granules.

In information systems, the *descriptor language* is a formal language commonly used to express and reason about properties of objects and con-

² This always holds for finite U and ϱ_{Γ_0} being a tolerance relation.

cepts [39, 42, 45]. We briefly present this language starting with an information system $\mathcal{A} = (U, A)$ as earlier. For the sake of simplicity, elements of $A \cup V$ are identified with their names playing the role of constant symbols. Moreover, these symbols are the only terms. Commas and the round parentheses are auxiliary symbols. As primitive propositional connectives we may take \wedge (conjunction) and \neg (negation), whereas the remaining connectives of disjunction, material implication, and double implication, \vee , \rightarrow , and \leftrightarrow , respectively, can be defined by means of \wedge , \neg along the classical lines. Descriptors, viz., pairs of the form (a, v) , where $a \in A$ and $v \in V_a$, are atomic formulas. Compound formulas are formed from the atomic ones as usual. Formulas are denoted by α, β , possibly with sub/superscripts. The relation of (crisp) satisfiability of formulas for objects, \models , is understood in line with the classical Tarskian approach, viz., for any formulas $(a, v), \alpha, \beta$ and any object u ,

$$\begin{aligned} u \models (a, v) &\stackrel{\text{def}}{\Leftrightarrow} a(u) = v, \\ u \models \alpha \wedge \beta &\stackrel{\text{def}}{\Leftrightarrow} u \models \alpha \ \& \ u \models \beta, \\ u \models \neg \alpha &\stackrel{\text{def}}{\Leftrightarrow} u \not\models \alpha. \end{aligned} \quad (10)$$

With every formula α , there is associated an information granule $\|\alpha\|$ of objects satisfying this formula, called the (crisp) *meaning* of α . In other words, α is a label for the granule consisting of objects satisfying α . Thus,

$$\begin{aligned} \|(a, v)\| &= \{u \mid a(u) = v\}, \\ \|\alpha \wedge \beta\| &= \|\alpha\| \cap \|\beta\|, \\ \|\neg \alpha\| &= U - \|\alpha\|. \end{aligned} \quad (11)$$

In the classical case, the satisfiability of formulas is extended to the satisfiability of sets of formulas³ in such a way that for any set of formulas X and any object u ,

$$u \models X \stackrel{\text{def}}{\Leftrightarrow} \forall \alpha \in X. u \models \alpha. \quad (12)$$

Hence, the information granule associated with X , called the (crisp) *meaning* of X , is the set

$$\|X\| = \bigcap \{\|\alpha\| \mid \alpha \in X\}. \quad (13)$$

In [25], we propose and discuss various rough forms of satisfiability of formulas and sets of formulas for objects. For instance, given an uncertainty mapping $\Gamma : U \mapsto \wp U$ and a threshold value $t \in [0, 1]$, a formula α is said to

³ We use the same symbol \models for simplicity.

be satisfied for u to the degree at least t , written $u \models_t \alpha$, if and only if the degree of inclusion of the granule Γu in the crisp meaning of α , measured by a rough inclusion function (see Sect. 7), is not less than t .

Example 4.1. Consider a formula $\alpha = ((a_1, 2) \wedge (a_3, 850)) \vee (a_5, v)$ of the descriptor language for the decision system \mathcal{A} from Example 3.1 which says that the year of studies of a student is 2 and the income is 850, or a student comes from a village. Then, $\|\alpha\| = (\|(a_1, 2)\| \cap \|(a_3, 850)\|) \cup \|(a_5, v)\| = (\{0, 3, 5, 7\} \cap \{2, 7\}) \cup \{1, 2, 5\} = \{1, 2, 5, 7\}$. Let us assume that for any object u_i ($i = 0, \dots, 7$) and $t \in [0, 1]$, $u_i \models_t \alpha$ if and only if $\#(\Gamma u_i \cap \|\alpha\|) / \#\Gamma u_i \geq t$. Hence, if $t > 1/2$, then $u_i \models_t \alpha$ for $i = 1, 5$; if $0 < t \leq 1/2$, then $u_i \models_t \alpha$ for $i = 1, 2, 3, 4, 5, 7$; and, finally, $u_i \models_0 \alpha$ for every i .

A rule is a pair $r = (P_r, C_r)$ of finite sets of formulas P_r and C_r of premises and conclusions of r , respectively. By assumption, the set of conclusions is non-empty. Rules will be denoted by r with sub/superscripts if needed. Sets P_r and C_r are information granules built of formulas on the basis of functionality. On the other hand, ' P_r ' and ' C_r ' are labels for the granules consisting of all objects of U which satisfy the sets P_r, C_r , respectively. In data mining and knowledge discovery (KDD), induction of *classification* (in particular, *decision*) rules as well as *association* rules is of particular interest [1, 2, 4, 28, 29, 30, 32, 34, 36, 69, 71]. For the lack of space, we give a few details about decision rules only. Such rules are typically extracted from decision systems with one decision attribute. The set of premises of a decision rule r consists of descriptors with different attribute symbols⁴ and the set of conclusions of r contains a single descriptor of the form (d, v) only, where d is the decision attribute and $v \in V_d$.

In general, premises and conclusions of a rule may be understood differently, viz., application of rules may be treated as a game between two agents. One agent checks whether or not a rule is applicable to an object and communicates the result to the second agent. Then, the latter agent applies the rule to the object. Therefore, let us consider two forms of satisfiability of sets of formulas for objects, \models_1, \models_2 , and the corresponding forms of meaning $\|\cdot\|_1, \|\cdot\|_2$. The *meaning* of a rule r is defined as the pair $\|r\|_{1,2} = (\|P_r\|_1, \|C_r\|_2)$ of information granules $\|P_r\|_1, \|C_r\|_2$, so r is a label for an information granule being an ordered pair of granules. Furthermore, we can say that r is *applicable* to an object u if and only if $u \models_1 P_r$, i.e., $u \in \|P_r\|_1$. More generally, r is *applicable* if and only if there is u such

⁴ A conjunction of such descriptors is called a *template*.

that $u \models_1 P_r$, i.e., $\|P_r\|_1 \neq \emptyset$. More about rules and their rough applicability can be found in [24, 26].

Example 4.2. From the decision system considered in Example 3.1, one can derive the following exemplary rules: $r_1 = (\{(a_1, 1), (a_5, v)\}, \{(d_0, s_3)\})$, $r_2 = (\{(a_4, 19), (a_5, v)\}, \{(d_0, s_3)\})$, and $r_3 = (\{(a_1, 1) \vee (a_4, 19), (a_5, v)\}, \{(d_0, s_3)\})$. Clearly, the latter rule is obtained from the first two rules, being decision rules, by combination of premises $(a_1, 1), (a_4, 19)$. By way of example, r_2 informally says that if a student is 19 and comes from a village, then he/she will be awarded a social and scientific scholarship. Note that $\|(a_1, 1)\| = \|(a_4, 19)\| = \|(a_1, 1) \vee (a_4, 19)\| = \{1, 2\}$, $\|(a_5, v)\| = \{1, 2, 5\}$, and $\|(d_0, s_3)\| = \{1, 2, 7\}$. Moreover, $\|P_{r_1}\| = \|P_{r_2}\| = \|P_{r_3}\| = \{1, 2\}$ and $\|C_{r_1}\| = \|C_{r_2}\| = \|C_{r_3}\| = \{1, 2, 7\}$. Hence, each and every rule r_i ($i = 1, 2, 3$) is applicable in the crisp sense to objects 1, 2, and it is valid in \mathcal{A} since $\|P_{r_i}\| \subseteq \|C_{r_i}\|$.

5. Rule Complexes

Informally speaking, *rule complexes* are sets having a nested structure and built of rules over some language [20, 22]. They are examples of complexes of objects, where rules are taken as objects. It should be emphasized that neither complexes of objects nor even rule complexes are information granules by definition. However, they may be viewed as such granules if the objects forming them are clustered on the basis of some kind of similarity or functionality. For instance, rules building an agent's value complex are drawn together for they all represent or at least are related to the agent's values and norms. Another example of a rule complex, being an information granule, is a *rough classifier* briefly described in Section 5.1.

Let us overview the notion of a *complex* of objects⁵. A program with embedded procedures is both a prototypical rule complex as well as an example of an information granule formed on the basis of functionality. In the definition below, empty complexes of objects are excluded for the sake of convenience only. Given a non-empty set of objects U , the class of all complexes of objects upon U (or, simply, complexes of objects if U is understood), $\mathcal{C}(U)$, is the least class of sets which contains U and is closed under the following formation rules: (CPL1) every non-empty subset of a complex of objects is a complex of objects; (CPL2) every non-empty set of

complexes of objects is a complex of objects; and (CPL3) the set-theoretical union of any non-empty set of complexes of objects is a complex of objects. A characteristic feature of complexes of objects is the possibility of multiple occurrences of objects in the same complex of objects although every such complex is actually a set.

In the domain of complexes of objects, the usual set-theoretical notions of an element and a subset of a set can be generalized to the notions of a *generalized element* and a *subcomplex* of a complex of objects. First, an object or a complex of objects x is a generalized element (or *g-element*) of a complex of objects X , $x \in_g X$, if x occurs in X , i.e.,

$$x \in_g X \stackrel{\text{def}}{\iff} x \in X \vee \exists n \in \mathbb{N}. \exists X_0, \dots, X_n. x \in X_0 \in \dots \in X_n \in X. \quad (14)$$

Given $t \in [0, 1]$ and a family of *rough membership functions*⁶ $\mu_X : U \mapsto [0, 1]$, where X is any complex of objects upon U , a rough version of g-membership can be obtained. We can say that an object u is a *generalized element* (or *g-element*) to the degree t of a complex of objects X , $\mu_X^g(u) = t$, if $\mu_X(u) = t$ or there exists a complex of objects Y being a g-element of X that $\mu_Y(u) = t$, i.e.,

$$\mu_X^g(u) = \max\{\mu_Y(u) \mid Y = X \vee Y \in_g X\}. \quad (15)$$

In the sequel, a complex of objects X is referred to as a subcomplex of a complex of objects Y , $X \sqsubseteq Y$, if $X = Y$ or X can be obtained from Y by removing some g-elements of Y , copies of the empty set which may appear after removing g-elements, and useless parentheses. For instance, let $Y = \{\{\{\{u, u'\}, u\}, u''\}, u'\}$ be the original complex of objects. By removing u, u' from $\{u, u'\}$ as well as u'' , we obtain $Y' = \{\{\{\emptyset, u\}\}, u'\}$ which, nevertheless, is not a complex of objects. At the next step, we remove \emptyset to arrive at $Y'' = \{\{\{u\}\}, u'\}$, already being a complex of objects and a subcomplex of Y . For simplicity, we may decide to remove the external parentheses in the expression ' $\{\{\{u\}\}$ ' to obtain $X = \{\{u\}, u'\}$ being another subcomplex of Y .

Every element is also a g-element and every non-empty subset is a subcomplex of a complex of objects, yet the converse may not hold in general. Both being a g-element and being a subcomplex are transitive. Note also that every complex of objects being a g-element of a complex of objects X is a subcomplex of X .

⁵ In [22], the term 'complex of points' was used.

⁶ The notion of a rough membership function goes back to Pawlak and Skowron [46]. More details can be found in Sect. 8.1.

Example 5.1. Consider the universe U from Example 3.1, granulated by Γ . Sets $X_1 = \{1, 2\}$, $X_2 = \{1, 3, 4\}$, $X_3 = \{2, X_2\}$, and $X_4 = \{1, 2, X_1, X_3\}$ are exemplary complexes of objects upon U . In the case of X_4 , its elements are 1, 2, X_1 , and X_3 . Apart from the elements just mentioned, the g -elements of X_4 are 3, 4, and X_2 . Examples of subcomplexes of X_4 are its non-empty subsets, $\{1, \{2\}, \{3, 4\}\}$, and $\{1, 2, \{2, \{4\}\}\}$. Assuming that for any object u and a set of objects X , $\mu_X(u) = \#(\Gamma u \cap X) / \#\Gamma u$, we obtain

$$\mu_{X_4}^g(u_i) = \begin{cases} 0 & \text{if } i = 0, 5, 6, \\ 0.5 & \text{if } i = 2, 3, 4, 7, \\ 1 & \text{if } i = 1. \end{cases} \quad (16)$$

5.1. Rough Classifiers

A decision system contains information about classification of a given set of objects U into some, pairwise disjoint classes C_0, \dots, C_m ($m \in \mathbb{N}$) of objects. The classification task consists in mapping any object examined, regardless it belongs to U or not, to exactly one of these classes. Where $m = 1$, the classification resolves into decision making whether or not an object belongs to C_0 (or, equivalently, C_1). Rough set techniques and other soft computing technology provide a number of algorithmic tools to extract decision rules from decision systems for the purpose of classification of unseen objects [4, 5, 28, 29, 30, 36, 69]. Mining classification rules of a satisfactory quality from a decision system $\mathcal{A} = (U, A, d)$ resolves itself into computation of suitable *reducts* of A . Informally speaking, a reduct of A is a set of attributes $B \subset A$ such that the classification of objects under the set of attributes B is almost the same as in the case of the primary set of attributes A . Most of the methods offered by the rough set theory to determine exact reducts⁷ are based on computation of prime implicants of a Boolean function what is NP-hard [58]. Therefore, various heuristics are proposed to compute approximate reducts which are suboptimal, yet obtained in a shorter time.

Having obtained a set of decision rules is not sufficient for the purpose of classification. We need to have at our disposal methods for judging applicability of rules, for carrying these rules into effect, and for solving conflicts and inconsistencies which possibly can occur. In this way, we arrive at the notion of a *rough classifier* for a concept or a set of concepts. Any such classifier is a set of decision rules together with meta-rules (methods) for

rough judgment of applicability, rough rule following, and resolving conflicts among rules. Clearly, a rough classifier is a granule of information based on functionality.

6. Collective Agents

Examples of collective agents are coalitions, teamworks, nations, states, institutions, companies, associations, school classes, economic clusters, and swarms. A collective agent is a collection of individual agents and/or other collective agents together with interconnections among the members as well as interaction rules and procedures, plans, values and norms, judgmental procedures, objectives and goals, intentions, commitments, beliefs and knowledge, consciousness, action modalities, etc. which are shared by the group members⁸. A collective agent is more than a system of agents just because of the assumption of sharing. A distinction is made about being shared and being common. In the former case, members of a collective agent need not to reflect on what they share, and they may not be conscious that others share or not share something. On the other hand, being a common value, rule, belief, or anything is a very robust feature. In this case, members of a collective agent not only share the thing but also they are conscious of their sharing, they are conscious that others are conscious of their sharing, and so on [8, 13, 14, 15, 17].

In our terms, collective agents are compound information granules consisting of two sorts of objects: (i) individual or other collective agents and (ii) systems of shared rules, values, beliefs, interaction procedures, and so on. Rule complex is just a mathematical tool which seems to be suitable for modelling and analyzing collective agents.

6.1. Economic Clusters

Among collective agents one can distinguish *economic clusters*. The notion of an *economic cluster*, going back to Porter [56, 57], has gained much interest since its introduction in the 90's of the 20th century (see, e.g., [6], other web pages, and articles on economic clusters). It is a vague concept related to a social and economic phenomenon that in spite of globalization of market and excellent opportunities for making businesses via internet,

⁸ Clearly, what is shared among the members of a collective agent – and to which extent – depends on the agent and the situation of interaction, e.g., members of the group may share a goal or some interaction rules only.

⁷ That is, reducts which precisely preserve the primary classification.

geographical location is still fundamental to competition of companies. According to Porter who considers a variety of economic clusters,

“[c]lusters are geographic concentrations of interconnected companies and institutions in a particular field.”

Obviously, economic clusters are not merely associations of similar objects like companies in some industry sector. They comprize heterogenous objects like governmental or private institutions (including universities and research centres), various associations, and a number of companies⁹. The world well-known clusters are the California wine cluster, Silicon Valley, Hollywood, the finance cluster on Wall Street, the Massachusetts biotechnology cluster, the Italian leather fashion cluster, the Dutch transportation and flower clusters, and the cluster of built-in kitchens and appliances in Germany. Other typical examples of clusters are tourism clusters functioning successfully in a large number of countries. There is a conviction that cluster formation is an essential ingredient of economic development in the 21st century. From our perspective, economic clusters are information granules based on functionality. In an economic cluster, objects (i.e., institutions and companies) are interdependent and interconnected by (in)formal links and are located in a relatively short distance from one another. Both competition and cooperation occur. Companies in a cluster aim both at their own economic outcomes as well as the development of the region. Last but not least, objects draw a substantial productive advantage from their nearness and interconnections.

7. Pawlak's Rough Approximation of Sets of Objects

The foundations of the theory of rough sets were laid by Pawlak in the early 80's of the 20th century [40, 41, 42, 44]. The starting point was the notion of an information system recalled in Sect. 3. The classical *Pawlak rough set model* is based on an equivalence relation of indiscernibility of objects.

In this section, we concisely present Pawlak's rough set model based on an arbitrary equivalence relation ϱ on a non-empty set of objects U considered as the universe of discourse. Objects are considered as equivalent if they cannot be discerned from one another. As earlier, the equivalence

⁹ Clusters can comprize hundreds of companies.

class $[u]$ of an object u is an elementary information granule. Any subset of U is viewed as a *concept*. Concepts being set-theoretical unions of elementary granules are referred to as *definable*. Even if a concept is not definable in this sense, it can be approximated from the inside and the outside by a pair of concepts which are already definable. In this way, we arrive at the famous notions of the *lower* and *upper rough approximations* of a concept. For any concept $X \subseteq U$, the lower and upper rough approximations of X , $\text{low}^\cup X$ and $\text{upp}^\cup X$, respectively, are defined by

$$\text{low}^\cup X = \bigcup\{[u] \mid [u] \subseteq X\} \ \& \ \text{upp}^\cup X = \bigcup\{[u] \mid [u] \cap X \neq \emptyset\}. \quad (17)$$

$\text{low}^\cup X$, viewed as the *positive region* of X , is the largest definable concept included in X . In other words, $\text{low}^\cup X$ consists of all objects u which certainly belong to X since all objects indiscernible from u are members of X . On the other hand, $\text{upp}^\cup X$ is the least definable concept containing X or, from another standpoint, $\text{upp}^\cup X$ is the set of all objects u which possibly belong to X as at least one object indiscernible from u is a member of X . The complement of $\text{upp}^\cup X$ may be viewed as the *negative region* of X . The difference

$$\text{bnd}^\cup X \stackrel{\text{def}}{=} \text{upp}^\cup X - \text{low}^\cup X \quad (18)$$

is called the *boundary region* of X . A concept X is *exact* if its boundary region is empty; otherwise X is referred to as *rough*. It turns out that a concept is definable if and only if it is exact. Moreover, the pair (U, ϱ) is called a *rough approximation space*. It is worth noting that the lower and upper rough approximations of a concept may also be given by

$$\text{low}X = \{u \mid [u] \subseteq X\} \ \& \ \text{upp}X = \{u \mid [u] \cap X \neq \emptyset\}, \quad (19)$$

respectively, whereas the boundary region of X may be defined as $\text{bnd}X = \text{upp}X - \text{low}X$. Indeed, one can easily see that

$$\begin{aligned} \bigcup\{[u] \mid [u] \subseteq X\} &= \{u \mid [u] \subseteq X\}, \\ \bigcup\{[u] \mid [u] \cap X \neq \emptyset\} &= \{u \mid [u] \cap X \neq \emptyset\}. \end{aligned} \quad (20)$$

Let us recall fundamental properties of the lower and upper rough approximations.

Proposition 7.1. *For any concepts X, Y , it holds:*

- (a) $\text{low}X \subseteq X \subseteq \text{upp}X$
- (b) $\text{low}\emptyset = \text{upp}\emptyset = \emptyset \ \& \ \text{low}U = \text{upp}U = U$

- (c) $X \subseteq Y \Rightarrow \text{low}X \subseteq \text{low}Y \ \& \ \text{upp}X \subseteq \text{upp}Y$
 (d) $\text{low}(X \cup Y) \supseteq \text{low}X \cup \text{low}Y \ \& \ \text{low}(X \cap Y) = \text{low}X \cap \text{low}Y$
 (e) $\text{upp}(X \cup Y) = \text{upp}X \cup \text{upp}Y \ \& \ \text{upp}(X \cap Y) \subseteq \text{upp}X \cap \text{upp}Y$
 (f) $\text{upp}X = U - \text{low}(U - X)$
 (g) $\text{low}(\text{low}X) = \text{upp}(\text{low}X) = \text{low}X \ \& \ \text{upp}(\text{upp}X) = \text{low}(\text{upp}X) = \text{upp}X$

Some comments can be handy. According to (a), the lower rough approximation of a concept is included in that concept, and every concept is included in its upper rough approximation. In virtue of (b), both the empty set and the whole universe are exact (and definable) concepts. Due to (c), the operations of lower and upper rough approximations are monotone. (d) says that the lower rough approximation of the union of two concepts contains the lower rough approximations of both concepts, whereas the lower rough approximation of the intersection of two concepts equals to the intersection of the lower rough approximations of those concepts. (e) is an analogous property for the upper rough approximation, viz., the upper rough approximation of the union of two concepts coincides with the union of the upper rough approximations of these concepts, and the upper rough approximation of the intersection of two concepts is included in the intersection of the upper rough approximations of the concepts. By (f), the operations of lower and upper rough approximations are dual. Finally, (g) says that the lower and upper rough approximations of a concept are already exact (and also definable) concepts.

Example 7.2. (Continuation of Example 3.1.) The uncertainty mapping Γ induces the partition $P = \{\{0\}, \{1\}, \{2, 4\}, \{3, 7\}, \{5\}, \{6\}\}$ of U . Starting with P , one can compute the lower and upper rough approximations of the concept $X = \{1, 2, 7\}$ as the sets $\{1\}$ and $\{1, 2, 3, 4, 7\}$, respectively. Since the boundary region of X is equal to $\{2, 3, 4, 7\}$, X is rough and undefinable in terms of information granules constituting P . Examples of exact and definable sets are $\{1\}$ and $\{1, 2, 3, 4, 7\}$.

8. Two Generalizations of Pawlak's Approach

In this section, we recall two important generalizations of the Pawlak rough set model. First, Ziarko [82, 83] introduced a model known as the *variable-precision rough set model* (VPRS-model for short) and based on an indiscernibility relation, where concepts are approximated in terms of

variable-precision positive and negative regions instead of lower and upper rough approximations. Next, Skowron and Stepaniuk [59, 60] presented a rough set model based on a tolerance relation¹⁰, where concepts are approximated by means of rough lower and upper approximations, yet defined differently than in the classical case. In Skowron–Stepaniuk's framework, *rough inclusion functions* (RIFs for short) play an important role and the fundamental notion is that of a *parameterized approximation space*. The classical Pawlak's model was extended and refined in several other directions as well. For instance, approximation spaces based on arbitrary non-empty binary relations are investigated in [19]. The reader interested in other developments is referred, e.g., to [21, 27, 47, 48, 51, 55, 67, 68, 69, 70, 71, 73, 74, 76].

8.1. Rough Inclusion Functions and Rough Membership Functions

Broadly speaking, a rough inclusion function (RIF) upon U is a mapping $\kappa : (\wp U)^2 \mapsto [0, 1]$ which measures the degree of inclusion of a concept in a concept of U . Polkowski and Skowron have axiomatically characterized RIFs within *rough mereology*, a formal theory of the notion of being-part-to-degree [49, 50, 52, 53, 54]. The most known example of a RIF is the *standard* one, κ^ℓ , defined for finite universes. The idea behind this notion goes back to Łukasiewicz [33]. For any finite concepts X, Y , $\kappa^\ell(X, Y) = \#(X \cap Y) / \#X$ if X is non-empty, and $\kappa^\ell(\emptyset, Y) = 1$. Clearly, the definition may be extended to the case of infinite second arguments.

In our approach, every RIF κ is assumed to fulfil rif_1 and rif_2 given below:

$$\text{rif}_1(\kappa) \stackrel{\text{def}}{\Leftrightarrow} \forall X, Y. (\kappa(X, Y) = 1 \Leftrightarrow X \subseteq Y)$$

$$\text{rif}_2(\kappa) \stackrel{\text{def}}{\Leftrightarrow} \forall X, Y, Z. (Y \subseteq Z \Rightarrow \kappa(X, Y) \leq \kappa(X, Z))$$

The above conditions are in accordance with the axioms of rough mereology. One can easily see that κ^ℓ satisfies both of them. Other exemplary properties which might also be postulated are

$$\text{rif}_3(\kappa) \stackrel{\text{def}}{\Leftrightarrow} \forall X \neq \emptyset. \kappa(X, \emptyset) = 0,$$

$$\text{rif}_4(\kappa) \stackrel{\text{def}}{\Leftrightarrow} \forall X \neq \emptyset. \forall Y. (\kappa(X, Y) = 0 \Leftrightarrow X \cap Y = \emptyset),$$

$$\text{rif}_5(\kappa) \stackrel{\text{def}}{\Leftrightarrow} \forall X \neq \emptyset. \forall Y. \kappa(X, Y) + \kappa(X, U - Y) = 1.$$

Like in the fuzzy set theory, one can measure the degree of membership of an object in a concept of U . Rough membership functions just serve the pur-

¹⁰ This assumption was relaxed to the case of a similarity relation later on.

pose. In this way, the crisp notion of membership in a set is extended to the case of reasoning under vague information within the rough set theory [46]. Given a concept X in a rough approximation space (U, ϱ) , where U is finite and ϱ is an indiscernibility relation, the *rough X -membership function* is defined in [46] as a mapping $\mu_X : U \mapsto [0, 1]$ such that for any $u \in U$,

$$\mu_X(u) = \#([u] \cap X) / \#[u], \quad (21)$$

i.e., $\mu_X(u) = \kappa^\ell([u], X)$. This definition can be extended to the case of an arbitrary universe of objects U , any RIF κ upon U , and any uncertainty mapping $\Gamma : U \mapsto \wp U$. Namely, for any concept X , the rough X -membership function, $\mu_X : U \mapsto [0, 1]$, may be defined as follows, for any object u :

$$\mu_X(u) = \kappa(\Gamma u, X) \quad (22)$$

82. The VPRS Model

Ziarko refined Pawlak's rough set model by introducing degrees of precision of approximation [82, 83]. In Ziarko's framework, concepts are approximated in terms of the s -negative regions and the t -positive regions of concepts, where the significance threshold parameters s, t satisfy $0 \leq s < t \leq 1$. The original VPRS model is based on an equivalence relation, say ϱ , of indiscernibility of objects. For every target concept $X \subseteq U$, the existence of a prior probability value $\Pr(X)$ is assumed. The value can be estimated from a finite data sample $U' \subseteq U$ by $\Pr(X) = \kappa^\ell(U', X)$. $\Pr(X)$ is understood as the probability of the event that an object belongs to X . With every equivalence class $[u]$, viewed as an elementary information granule as earlier, there is associated a conditional probability value $\Pr(X|[u])$, understood as the probability that an object u' belongs to X provided that u' is indiscernible from u . For finite equivalence classes, this probability is estimated by $\Pr(X|[u]) = \kappa^\ell([u], X)$. Instead of lower and upper rough approximations, variable-precision negative and positive regions of concepts are considered. In detail, the *s -negative region*, the *t -positive region*, and the *(s, t) -boundary region* of a concept X , written $\text{neg}_s^\cup X$, $\text{pos}_t^\cup X$, and $\text{bnd}_{s,t}^\cup X$, respectively, are defined by

$$\begin{aligned} \text{neg}_s^\cup X &= \bigcup \{ [u] \mid \Pr(X|[u]) \leq s \}, \\ \text{pos}_t^\cup X &= \bigcup \{ [u] \mid \Pr(X|[u]) \geq t \}, \\ \text{bnd}_{s,t}^\cup X &= \bigcup \{ [u] \mid s < \Pr(X|[u]) < t \}. \end{aligned} \quad (23)$$

In words, the s -negative (resp., t -positive) region of X is the union of all granules $[u]$ that the conditional probability $\Pr(X|[u])$ does not exceed (resp., is not less than) a given threshold value s (resp., t). The (s, t) -boundary region of X is the union of all remaining equivalence classes. For finite U , the above equalities take the following forms:

$$\begin{aligned} \text{neg}_s^\cup X &= \bigcup \{ [u] \mid \kappa^\ell([u], X) \leq s \} \\ \text{pos}_t^\cup X &= \bigcup \{ [u] \mid \kappa^\ell([u], X) \geq t \} \\ \text{bnd}_{s,t}^\cup X &= \bigcup \{ [u] \mid s < \kappa^\ell([u], X) < t \} \end{aligned} \quad (24)$$

As in the case of Pawlak's lower and upper rough approximations, the variable-precision negative, positive, and boundary regions can be defined by

$$\begin{aligned} \text{neg}_s X &= \{ u \mid \kappa^\ell([u], X) \leq s \}, \\ \text{pos}_t X &= \{ u \mid \kappa^\ell([u], X) \geq t \}, \\ \text{bnd}_{s,t} X &= \{ u \mid s < \kappa^\ell([u], X) < t \}, \end{aligned} \quad (25)$$

respectively, since $\text{neg}_s^\cup X = \text{neg}_s X$, $\text{pos}_t^\cup X = \text{pos}_t X$, and $\text{bnd}_{s,t}^\cup X = \text{bnd}_{s,t} X$. Moreover, if $s = 0$ and $t = 1$, then $\text{neg}_s X = U - \text{upp}X$ and $\text{pos}_t X = \text{low}X$, i.e., the s -negative region and the t -positive region become the negative region and the positive region, respectively.

The notions of variable-precision negative and positive regions of a concept can be generalized to the case of an arbitrary RIF κ and an arbitrary similarity relation ϱ . After necessary modifications of (24) and (25), we can obtain, for example,

$$\begin{aligned} \text{neg}_s^\cup X &= \bigcup \{ \varrho^-\{u\} \mid \kappa(\varrho^-\{u\}, X) \leq s \}, \\ \text{pos}_t^\cup X &= \bigcup \{ \varrho^-\{u\} \mid \kappa(\varrho^-\{u\}, X) \geq t \}, \\ \text{bnd}_{s,t}^\cup X &= \bigcup \{ \varrho^-\{u\} \mid s < \kappa(\varrho^-\{u\}, X) < t \}, \\ \text{neg}_s X &= \{ u \mid \kappa(\varrho^-\{u\}, X) \leq s \}, \\ \text{pos}_t X &= \{ u \mid \kappa(\varrho^-\{u\}, X) \geq t \}, \\ \text{bnd}_{s,t} X &= \{ u \mid s < \kappa(\varrho^-\{u\}, X) < t \}. \end{aligned} \quad (26)$$

Unlike in the indiscernibility-based case, $\text{neg}_s^\cup X$ and $\text{neg}_s X$ need not to coincide, and analogously for the remaining pairs of concepts.

Example 8.1. (Continuation of Example 3.1.) Suppose that the granulation of U is induced by the uncertainty mapping Γ_{a_2} , and the RIF considered is standard. In Table 3, we give the t -positive and t -negative regions of the concept $X = \{1, 2, 7\}$ for various values of t . As regarding the variable-precision

boundary regions, we obtain, e.g., $\text{bnd}_{s,t}X = \{4, 5, 6\}$ for $0 < s \leq 1/4$ and $1/3 < t \leq 1/2$.

Table 3

The t -positive and t -negative regions of X

| t | $\text{pos}_t X$ | $\text{neg}_t X$ |
|--------------|------------------------|----------------------|
| $\{0\}$ | U | $\{0\}$ |
| $(0, 1/4]$ | $\{1, \dots, 7\}$ | $\{0, 3\}$ |
| $(1/4, 1/3]$ | $\{1, 2, 4, 5, 6, 7\}$ | $\{0, 3, 4, 5, 6\}$ |
| $(1/3, 1/2]$ | $\{1, 2, 7\}$ | $\{0, 2, \dots, 7\}$ |
| $(1/2, 1]$ | $\{1\}$ | U |

8.3. Parameterized Approximation Spaces

Skowron and Stepaniuk generalized Pawlak's rough set model to the case of similarity-based approximation of concepts. The notion of a parameterized approximation space, introduced in [59, 60] and elaborated in a series of articles, influenced the further research on the theory and the applications of approximation spaces. Let U be a non-empty set of objects, \mathcal{S} be a list of tuning parameters to obtain a satisfactory quality of approximation, $\Gamma_{\mathcal{S}} : U \mapsto \wp U$ be a mapping called an *uncertainty* mapping and assigning to every object $u \in U$, a set of objects in some sense similar to u , and $\kappa_{\mathcal{S}}$ be a RIF upon U . It is assumed that $\forall u.u \in \Gamma_{\mathcal{S}}u$, and indiscernibility is treated as a special case of similarity. Elementary granules of information are of the form $\Gamma_{\mathcal{S}}u$, where u is an object of U .

A parameterized approximation space is a triple $\mathcal{M}_{\mathcal{S}} = (U, \Gamma_{\mathcal{S}}, \kappa_{\mathcal{S}})$, where any concept $X \subseteq U$ can be approximated by its *lower* and *upper approximations*, $\text{low}^{\mathcal{S}}X$ and $\text{upp}^{\mathcal{S}}X$, respectively, such that

$$\text{low}^{\mathcal{S}}X \stackrel{\text{def}}{=} \{u \mid \kappa_{\mathcal{S}}(\Gamma_{\mathcal{S}}u, X) = 1\} \ \& \ \text{upp}^{\mathcal{S}}X \stackrel{\text{def}}{=} \{u \mid \kappa_{\mathcal{S}}(\Gamma_{\mathcal{S}}u, X) > 0\}. \quad (27)$$

According to this definition, $\text{low}^{\mathcal{S}}X$ consists of all objects u that their elementary granules are included in X to the degree 1. On the other hand, $\text{upp}^{\mathcal{S}}X$ is the set of all objects that their elementary granules are included in X to some positive degree. The *boundary region* of X , $\text{bnd}^{\mathcal{S}}X$, can be defined as the set

$$\text{bnd}^{\mathcal{S}}X = \{u \mid 0 < \kappa_{\mathcal{S}}(\Gamma_{\mathcal{S}}u, X) < 1\}. \quad (28)$$

For simplicity, we shall omit \mathcal{S} unless necessary.

In parameterized approximation spaces, concepts can also be approximated in a number of other ways, e.g. in line with Pawlak's rough approximation. By way of example, the lower and upper approximations of X may be given by

$$\text{low}X \stackrel{\text{def}}{=} \{u \mid \Gamma u \subseteq X\} \ \& \ \text{upp}X \stackrel{\text{def}}{=} \{u \mid \Gamma u \cap X \neq \emptyset\}, \quad (29)$$

respectively, whereas the boundary region of X may be defined as $\text{bnd}X = \text{upp}X - \text{low}X$. By $\text{rif}_1(\kappa)$, $\text{low}^{\mathcal{S}}X = \text{low}X$, yet $\text{upp}^{\mathcal{S}}X = \text{upp}X$ needs not to hold¹¹. When κ is standard and $\Gamma \dashv U$ is a partition of U , the lower and upper approximations defined by (27) coincide with Pawlak's lower and upper rough approximations, respectively.

Every uncertainty mapping Γ induces a reflexive relation ϱ_{Γ} on U such that $(u', u) \in \varrho_{\Gamma} \stackrel{\text{def}}{\Leftrightarrow} u' \in \Gamma u$ (cf. (6)), and vice versa, starting with a reflexive relation ϱ on U understood as a relation of similarity of objects, an uncertainty mapping $\Gamma_{\varrho} : U \mapsto \wp U$, defined by $u' \in \Gamma_{\varrho}u \stackrel{\text{def}}{\Leftrightarrow} (u', u) \in \varrho$, can be derived¹². Therefore, we can think of a *similarity-based rough approximation space* as a structure $\mathcal{M} = (U, \Gamma, \kappa)$ above or $\mathcal{N} = (U, \varrho, \kappa)$, where ϱ is a reflexive relation on U .

Example 8.2. Consider an approximation space $\mathcal{M} = (U, \Gamma', \kappa)$, where U and Γ' are as in Example 3.1, and κ is a RIF such that for any non-empty concept X and any concept Y , $\kappa(X, Y) = 0$ if and only if $\text{upp}X \cap \text{upp}Y = \emptyset$. Hence, $X \cap Y \neq \emptyset$ implies $\kappa(X, Y) > 0$, but the converse may not hold. Indeed, $\{3\}, \{1, 2, 7\}$ are disjoint, whereas $\text{upp}\{3\} \cap \text{upp}\{1, 2, 7\} = \{0, 3, 7\} \cap \{1, 2, 3, 7\} = \{3, 7\}$, i.e., $\kappa(\{3\}, \{1, 2, 7\}) > 0$. Furthermore, for any concept X , $\text{upp}X \subseteq \text{upp}^{\mathcal{S}}X$. However, the both forms of approximation are different. Namely, $\text{upp}\{1, 2, 7\} = \{u \mid \Gamma'u \cap \{1, 2, 7\} \neq \emptyset\} = \{1, 2, 3, 7\}$, whereas $\text{upp}^{\mathcal{S}}\{1, 2, 7\} = \{u \mid \kappa(\Gamma'u, \{1, 2, 7\}) > 0\} = \{u \mid \text{upp}(\Gamma'u) \cap \{1, 2, 3, 7\} \neq \emptyset\} = \{0, 1, 2, 3, 7\}$.

9. Summary

In this paper, we discussed the possibility of using the methodologies of rough sets and granular computing to build a rule-based model of a multi-agent system. The main assumption of our approach is that the universe

¹¹ Among others, the equality holds true if κ is standard.

¹² As a matter of fact, ϱ induces two different uncertainty mappings unless ϱ is symmetric. Apart from Γ_{ϱ} , we obtain Γ_{ϱ}^* such that $u' \in \Gamma_{\varrho}^*u \stackrel{\text{def}}{\Leftrightarrow} (u, u') \in \varrho$.

of all objects considered is granulated into information granules, i.e. clusters of objects formed on the basis of similarity or functionality. According to our idea, both agents and their systems, being complex objects of some kind, may be viewed as information granules. Apart from a brief overview of various aspects regarding the modelling of agent systems, we presented several examples of information granules which can be useful in such modelling. Furthermore, the classical Pawlak model of rough sets and its two extensions were recalled. Various detailed questions concerning social agent system modelling, mentioned in Introduction, will be elaborated in the future work.

References

- [1] R. Agrawal, T. Imieliński and A. Swami. Mining association rules between sets of items in large databases. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 207–216, Washington, D.C., 1993.
- [2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. 20th Conf. on Very Large Databases*, pages 487–499, Santiago, Chile, 1994.
- [3] C. E. Alchourrón, P. Gärdenfors and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symbolic Logic*, 50:510–530, 1985.
- [4] J. Bazan. A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables. In [51], volume 1, pages 321–365. 1998.
- [5] J. Bazan and A. Skowron. Classifiers based on approximate reasoning schemes. In [16], pages 191–202. 2005.
- [6] R. Breault. *The evolution of structured clusters*. 2000. <http://www.ptb-magazine.com/July00/ptb700.clusters2.html>.
- [7] T. R. Burns, J. C. Caldas and E. Roszkowska. Generalized game theory's contribution to multi-agent modelling. In [16], pages 363–384. 2005.
- [8] T. R. Burns and E. Engdahl. The social construction of consciousness: Collective consciousness and its socio-cultural foundations; Individual selves, self-awareness, and reflectivity, parts I and II. *J. Consciousness Studies*, 5(1–2):67–85, 1998.

- [9] T. R. Burns and A. Gomolińska. The theory of socially embedded games: The mathematics of social relationships, rule complexes, and action modalities. *Quality and Quantity Int. J. Methodology*, 34(4):379–406, 2000.
- [10] T. R. Burns and A. Gomolińska. Socio-cognitive mechanisms of belief change: Applications of generalized game theory to belief revision, social fabrication, and self-fulfilling prophesy. *Cognitive Systems Research*, 2(1):39–54, 2001.
- [11] T. R. Burns, A. Gomolińska and L. D. Meeker. The theory of socially embedded games: Applications and extensions to open and closed games. *Quality and Quantity Int. J. Methodology*, 35(1):1–32, 2001.
- [12] T. R. Burns and E. Roszkowska. Generalized game theory: Assumptions, principles, and elaborations in social theory. *Studies in Logic, Grammar and Rhetoric*, 8(21):7–40, 2005.
- [13] B. Dunin-Kępcicz and R. Verbrugge. Collective intentions. *Fundamenta Informaticae*, 51(3):271–295, 2002.
- [14] B. Dunin-Kępcicz and R. Verbrugge. Evolution of collective commitments during teamwork. *Fundamenta Informaticae*, 56(3):329–371, 2003.
- [15] B. Dunin-Kępcicz and R. Verbrugge. A tuning machine for cooperative problem solving. *Fundamenta Informaticae*, 63(2–3):283–307, 2004.
- [16] B. Dunin-Kępcicz et al., editor. *Monitoring, Security, and Rescue Techniques in Multiagent Systems*. Springer, Berlin, 2005.
- [17] R. Fagin. *Reasoning About Knowledge*. The MIT Press, Cambridge, MA, 1995.
- [18] P. Gärdenfors, editor. *Belief Revision*. Cambridge University Press, 1992.
- [19] A. Gomolińska. A comparison of Pawlak's and Skowron-Stepaniuk's approximation of concepts. To appear in LNCS *Transactions on Rough Sets*.
- [20] A. Gomolińska. Rule complexes for representing social actors and interactions. *Studies in Logic, Grammar and Rhetoric*, 3(16):95–108, 1999.
- [21] A. Gomolińska. Variable-precision compatibility spaces. *Electronical Notices in Theoretical Computer Science*, 82(4):1–12, 2003. <http://www.elsevier.nl/locate/entcs/volume82.html>.

- [22] A. Gomolińska. Fundamental mathematical notions of the theory of socially embedded games: A granular computing perspective. In [38], pages 411–434. 2004.
- [23] A. Gomolińska. Modelowanie systemów wieloagentowych w terminach granul informacyjnych. In K. Trzęsicki, editor, *Ratione et Studio*, pages 127–145. University of Białystok, Białystok, 2005.
- [24] A. Gomolińska. Rough validity, confidence, and coverage of rules in approximation spaces. *Transactions on Rough Sets III: Journal Subline, Lecture Notes in Computer Science*, 3400:57–81, 2005.
- [25] A. Gomolińska. Satisfiability and meaning of formulas and sets of formulas in approximation spaces. *Fundamenta Informaticae*, 67(1–3):77–92, 2005.
- [26] A. Gomolińska. Towards rough applicability of rules. In [16], pages 203–214. 2005.
- [27] S. Greco, B. Matarazzo and R. Słowiński. Fuzzy similarity relation as a basis for rough approximations. *Lecture Notes in Artificial Intelligence*, 1424:283–289, 1998.
- [28] J. W. Grzymała-Busse. Classification of unseen examples under uncertainty. *Fundamenta Informaticae*, 30(3–4):255–267, 1997.
- [29] J. W. Grzymała-Busse. LERS – A data mining system. In O. Maimon and L. Rokach, editors, *The Data Mining and Knowledge Discovery Handbook*, pages 1347–1351. Springer, Heidelberg, 2005.
- [30] J. W. Grzymała-Busse. Rule induction. In O. Maimon and L. Rokach, editors, *The Data Mining and Knowledge Discovery Handbook*, pages 255–267. Springer, Heidelberg, 2005.
- [31] M. Inuiguchi, S. Hirano and S. Tsumoto, editors. *Rough Set Theory and Granular Computing*, volume 125 of *Studies in Fuzziness and Soft Computing*. Springer, Berlin Heidelberg, 2003.
- [32] M. Kryszkiewicz. Fast discovery of representative association rules. *Lecture Notes in Artificial Intelligence*, 1424:214–221, 1998.
- [33] J. Łukasiewicz. Die logischen Grundlagen der Wahrscheinlichkeitsrechnung. In L. Borkowski, editor, *Jan Łukasiewicz – Selected Works*, pages 16–63. North Holland, Polish Scientific Publ., Amsterdam London, Warsaw, 1970. First published in Cracow in 1913.
- [34] H. S. Nguyen and S. H. Nguyen. Rough sets and association rule generation. *Fundamenta Informaticae*, 40(4):383–405, 1999.

- [35] H. S. Nguyen, A. Skowron and J. Stepaniuk. Granular computing: A rough set approach. *Int. J. Comput. Intelligence*, 17(3):514–544, 2001.
- [36] H. S. Nguyen and D. Ślęzak. Approximate reducts and association rules – correspondence and complexity results. *Lecture Notes in Artificial Intelligence*, 1711:137–145, 1999.
- [37] S. H. Nguyen et al. Layered learning for concept synthesis. *Transactions on Rough Sets I: Journal Subline, Lecture Notes in Computer Science*, 3100:187–208, 2004.
- [38] S. K. Pal, L. Polkowski and A. Skowron, editors. *Rough-Neural Computing: Techniques for Computing with Words*. Springer, Berlin Heidelberg, 2004.
- [39] Z. Pawlak. Information systems – Theoretical foundations. *Information Systems*, 6(3):205–218, 1981.
- [40] Z. Pawlak. *Rough Sets, Basic Notions*. Number 431 in ICS PAS Report. Warsaw, 1981.
- [41] Z. Pawlak. Rough Sets. *Int. J. Computer and Information Sciences*, 11:341–356, 1982.
- [42] Z. Pawlak. *Systemy informacyjne. Podstawy teoretyczne*. Wydawnictwo Naukowo-Techniczne, Warsaw, 1983.
- [43] Z. Pawlak. Rough sets and decision tables. *Lecture Notes in Computer Science*, 208:186–196, 1985.
- [44] Z. Pawlak. *Rough Sets. Theoretical Aspects of Reasoning About Data*. Kluwer, Dordrecht, 1991.
- [45] Z. Pawlak, L. Polkowski and A. Skowron. Rough sets and rough logic: A KDD perspective. In [55], pages 583–646. 2001.
- [46] Z. Pawlak and A. Skowron. Rough membership functions. In M. Fedrizzi, J. Kacprzyk and R. R. Yager, editors, *Advances in the Dempster-Shafer Theory of Evidence*, pages 251–271. John Wiley & Sons, New York, 1994.
- [47] J. F. Peters. Approximation space for intelligent system design patterns. *Engineering Applications of Artificial Intelligence*, 17(4):1–8, 2004.
- [48] L. Polkowski. *Rough Sets: Mathematical Foundations*. Physica-Verlag, Heidelberg, 2002.
- [49] L. Polkowski and A. Skowron. Rough mereology. *Lecture Notes in Artificial Intelligence*, 869:85–94, 1994.

- [50] L. Polkowski and A. Skowron. Rough mereology: A new paradigm for approximate reasoning. *Int. J. Approximated Reasoning*, 15(4):333–365, 1996.
- [51] L. Polkowski and A. Skowron, editors. *Rough Sets in Knowledge Discovery*, volume 1–2. Physica-Verlag, Heidelberg, 1998.
- [52] L. Polkowski and A. Skowron. Towards adaptive calculus of granules. In [81], volume 1, pages 201–228. 1999.
- [53] L. Polkowski and A. Skowron. Rough mereological calculi of granules: A rough set approach to computation. *Int. J. Comput. Intelligence*, 17(3):472–492, 2001.
- [54] L. Polkowski and A. Skowron. Rough mereology in information systems. A case study: Qualitative spatial reasoning. In [55], pages 89–135. 2001.
- [55] L. Polkowski, S. Tsumoto and T. Y. Lin, editors. *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems*. Physica-Verlag, Heidelberg, 2001.
- [56] M. E. Porter. Clusters and the new economics of competition. *Harvard Business Review*, 76(6):77–90, 1998.
- [57] M. E. Porter. *On Competition*. Harvard Business School Press, Boston, MA, 1998.
- [58] A. Skowron and C. M. Rauszer. The discernibility matrices and functions in information systems. In R. Słowiński, editor, *Intelligent Decision Support – Handbook of Applications and Advances of the Rough Sets Theory*, pages 331–362. Kluwer, Dordrecht, 1992.
- [59] A. Skowron and J. Stepaniuk. Generalized approximation spaces. In *Proc. 3rd Int. Workshop on Rough Sets and Soft Computing, San Jose, USA, November 1994*, pages 156–163, 1994.
- [60] A. Skowron and J. Stepaniuk. Tolerance approximation spaces. *Fundamenta Informaticae*, 27(2–3):245–253, 1996.
- [61] A. Skowron and J. Stepaniuk. Information granules in distributed environment. *Lecture Notes in Artificial Intelligence*, 1711:357–365, 1999.
- [62] A. Skowron and J. Stepaniuk. Towards discovery of information granules. *Lecture Notes in Artificial Intelligence*, 1704:542–547, 1999.
- [63] A. Skowron and J. Stepaniuk. Information granules and rough-neural computing. In [38], pages 43–84. 2004.

- [64] A. Skowron and J. Stepaniuk. Ontological framework for approximation. *Lecture Notes in Artificial Intelligence*, 3641:718–727, 2005.
- [65] A. Skowron, J. Stepaniuk and S. Tsumoto. Information granules for spatial reasoning. *Bull. Int. Rough Set Society*, 3(4):147–154, 1999.
- [66] A. Skowron, R. Swiniarski and P. Synak. Approximation spaces and information granulation. *Transactions on Rough Sets III: Journal Subline, Lecture Notes in Computer Science*, 3400:175–189, 2005.
- [67] R. Słowiński and D. Vanderpooten. Similarity relation as a basis for rough approximations. In P. P. Wang, editor, *Advances in Machine Intelligence and Soft Computing*, volume 4, pages 17–33. Duke University Press, 1997. Also in *ICS Research Report 53/95*, Warsaw University of Technology, Warsaw, 1995.
- [68] R. Słowiński and D. Vanderpooten. A generalized definition of rough approximations based on similarity. *IEEE Transactions on Data and Knowledge Engineering*, 12:331–336, 2000.
- [69] J. Stefanowski, *Algorytmy indukcji reguł decyzyjnych w odkrywaniu wiedzy*, volume 361 of *Rozprawy*. Wydawnictwo Politechniki Poznańskiej, Poznań, 2001.
- [70] J. Stefanowski and A. Tsoukiàs. Incomplete Information Tables and Rough Classification. *Int. J. Comput. Intelligence*, 17(3):545–566, 2001.
- [71] J. Stepaniuk. Knowledge discovery by application of rough set models. In [55], pages 137–233. 2001.
- [72] J. Stepaniuk. Tolerance information granules. In [16], pages 305–316. 2005.
- [73] S. Tsumoto et al., editor. *Proc. 4th Int. Conf. Rough Sets and Current Trends in Computing (RSCTC'2004), Uppsala, Sweden, June 2004*, volume 3066 of *Lecture Notes in Artificial Intelligence*. Springer, Berlin Heidelberg, 2004.
- [74] S. K. M. Wong, L. S. Wang and Y. Y. Yao. On modelling uncertainty with interval structures. *Int. J. Computational Intelligence*, 11:406–426, 1995.
- [75] Y. Y. Yao. Granular computing. In *Proc. of CRSSC'2004*, pages 1–5, 2004, ISSN 1002-137X.
- [76] Y. Y. Yao, S. K. M. Wong and T. Y. Lin. A review of rough set models. In T. Y. Lin and N. Cercone, editors, *Rough Sets and Data Mining: Analysis of Imprecise Data*, pages 47–75. Kluwer, Boston London Dordrecht, 1997.

- [77] L. A. Zadeh. Outline of a new approach to the analysis of complex system and decision processes. *IEEE Trans. on Systems, Man, and Cybernetics*, 3:28–44, 1973.
- [78] L. A. Zadeh. Fuzzy sets and information granularity. In M. Gupta, R. Ragade and R. Yager, editors, *Advances in Fuzzy Set Theory and Applications*, pages 3–18. North-Holland, Amsterdam, 1979.
- [79] L. A. Zadeh. Fuzzy logic = computing with words. *IEEE Trans. on Fuzzy Systems*, 4(2):103–111, 1996.
- [80] L. A. Zadeh. Toward a theory of fuzzy information granulation and its certainty in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, 90:111–127, 1997.
- [81] L. A. Zadeh and J. Kacprzyk, editors. *Computing with Words in Information/Intelligent Systems*. Physica-Verlag, Heidelberg, 1999.
- [82] W. Ziarko. Variable precision rough set model. *J. Computer and System Sciences*, 46(1):39–59, 1993.
- [83] W. Ziarko. Probabilistic decision tables in the variable precision rough set model. *Int. J. Comput. Intelligence*, 17(3):593–603, 2001.

Anna Gomolińska
University of Białystok
Department of Mathematics
Akademicka 2, 15-267 Białystok
anna.gom@math.uwb.edu.pl

APPENDIX

This Appendix offers a chronicle of the **Polish Association for Logic and Philosophy of Science**. This is to introduce the institution whose auspices this journal will enjoy from now on. Moreover, it contains Curricula Vitae of the contributors to this volume.

Anna Gomolińska
Witold Marciszewski
Roman Matuszewski
Roman Murawski
Kazimierz Trzęsicki

Izabela Bondecka-Krzykowska
Adam Mickiewicz University

HISTORY OF POLISH ASSOCIATION FOR LOGIC AND PHILOSOPHY OF SCIENCE

Polish Association for Logic and Philosophy of Science (Polskie Towarzystwo Logiki i Filozofii Nauki) was established in 1994. It is a continuation of Polish Logical Association (Polskie Towarzystwo Logiczne) founded on the initiative of professors Jan Łukasiewicz and Alfred Tarski on 22nd April 1936. The establishment of the latter was connected with the great development of logic in Poland in the interwar period. The aim of the association was – according to the statutes – “to practise and propagate logic and methodology of science, their history, didactics and applications”.

After the second world war the association has not been reactivated. Between 1980 and 1981 professor Ryszard Wójcicki undertook efforts to establish Polish Association of Logic. Unfortunately Polish authorities refused the registration. During the 9th International Congress of Logic, Methodology and Philosophy of Science held in 1991 in Uppsala (Sweden) a meeting of Polish participants took place. During this meeting professors Jan Woleński and Ryszard Wójcicki put forward a proposal to renew the endeavours to register an association named Polish Association for Logic and Philosophy of Science. The idea has been accepted and in 1994 the association and its statutes have been registered. The first president became professor Wojciech Buszkowski (1993–1996). Next presidents were: professor Ewa Orłowska (1997–1999), professor Jan Woleński (2000–2003), professor Grażyna Mirkowska (2003–2005). Since January 2006 the president is professor Roman Murawski. The association has nowadays about 210 members (among them also scientists from abroad).

The aims of the Association are – according to the statutes – “to support scientific studies, popularization and teaching of the following disciplines: logic, applications of logic, especially applications in the foundations of mathematics, computer science and linguistics, and philosophy of science.” They are realized by organizing scientific meetings and conferences, acqu-

iring and propagating information about scientific researches in the field of logic, its applications and the philosophy of science, by editorial, library and teaching activity, by cooperating with national and international institutions and organizations as well as by representing the scientific circle of logicians and philosophers of science.

Association is working all over Poland. Sections and local branches can be formed. The sources of funds of the association include membership fees, donations, subventions and incomes from economic activities.

The Association has an interdisciplinary character – this is the consequence of the existence of many fields in the contemporary logic. Its members are first of all philosophers, mathematicians and computer scientists working in logic, its applications and in the philosophy of science. It should be stressed that it is opened for everyone who is interested in logic in the broad sense including semantics, formal logic and the methodology of science.

The Association was the co-organizer of several Polish-German workshops in logic (in the period 1995–2000), of 11th International Congress of Logic, Methodology and Philosophy of Science held in Cracow (1999) and of Alfred Tarski Centenary Conference that took place in Warsaw (2001). During those conferences special sessions were organized by the Association. Association together with *Studia Logica* organized international conferences held under the name “Trends in Logic”. They took place in Roskilde (Denmark, 2004), Maðralin (Poland, 2004), Warsaw and Ruciane Nida (2005). The latter was dedicated to the memory of Professors Helena Rasiowa, Cecylia Rauszer and Andrzej Mostowski. The fourth conference “Trends in Logic” will take place in Toruń in 2006.

Two journals are published under the auspices of the Association: *Studia Logica* and *Studies in Logic, Grammar and Rhetoric*.

Polish Association for Logic and Philosophy of Science is the member of International Union of the History and Philosophy of Science – Division of Logic, Methodology and Philosophy of Science.

ANNA GOMOLIŃSKA

Anna Gomolińska received Ph.D. in mathematics from Warsaw University in 1993. Her doctoral thesis, written under supervision of Cecilia M. Rauszer, was entitled “Logical Methods of Knowledge Representation under Incomplete Information”. Anna Gomolińska works as a university teacher in the Department of Mathematics of Białystok University. She spent the spring semester of 1994 as a visiting scholar in the Department of Philosophy of Uppsala University, and she was also a research fellow at SCASSS, Uppsala in 1995 and in the Institute of Computer Science (IASI) of Italian CNR, Rome in 2002. In years 1996–2001, she collaborated with Tom R. Burns from Uppsala University on the research project concerning mathematical foundations of socially-embedded games. Since 2001 she has been a member of the research group led by Andrzej Skowron from Warsaw University. Apart from Białystok and Warsaw, she gave invited seminar talks at several universities and research institutes in Europe, viz., in Uppsala, Trondheim, Milan, Dresden, Potsdam, and Rome. Among her fields of interests are logical aspects of computer science and artificial intelligence, modal logics, game theory, multi-agent systems, and last but not least, rough sets. Anna Gomolińska is a member of the Polish Association of Logic and Philosophy of Science (the secretary of the society in 2003–2005) and the Polish Mathematical Society. She is an author or a co-author of 40 research articles or chapters in books, and the references to some of them are given below.

References

- Gomolińska, A. (2006). Possible rough ingredients of concepts in approximation spaces, *Fundamenta Informaticae* 72(1-3):139-154.
- Gomolińska, A. (2005). Satisfiability and meaning of formulas and sets of formulas in approximation spaces, *Fundamenta Informaticae* 67(1-3): 77-92.

- Gomolińska, A. (2004). Fundamental mathematical notions of the theory of socially embedded games: A granular computing perspective, in: Pal, S. K., Polkowski, L. T., & Skowron, A. (eds.), *Rough-Neural Computing: Techniques for Computing with Words*, Springer-Verlag: Berlin Heidelberg, pp. 411-434.
- Gomolińska, A. (2002). Derivability of rules from rule complexes, *Logic and Logical Philosophy* 10:21-44.
- Burns, T. R. & Gomolińska, A. (2001). Socio-cognitive mechanisms of belief change: Applications of generalized game theory to belief revision, social fabrication, and self-fulfilling prophesy, *Cognitive Systems Research* 2(1):39-54.
- Burns, T. R. & Gomolińska, A. (2000). The theory of socially embedded games: The mathematics of social relationships, rule complexes, and action modalities, *Quality and Quantity: Int. J. Methodology* 34(4):379-406.
- Gomolińska, A. (1999). Rule complexes for representing social actors and interactions, *Studies in Logic, Grammar and Rhetoric* 3(16):95-108.
- Gomolińska, A. (1998). Credibility of information for modelling belief state and its change, *Fundamenta Informaticae* 34:33-51.
- Gomolińska, A. (1998). On the logic of acceptance and rejection, *Studia Logica* 60:233-251.
- Gomolińska, A. (1997). A nonmonotonic modal formalization of the logic of acceptance and rejection, *Studia Logica* 58:113-127.

WITOLD MARCISZEWSKI

Born on 18 November 1930 in Warsaw, Poland.

Education and degrees. 1948-1953: theology at Warsaw University. – 1953-1956. Philosophy at the Catholic University of Lublin: M.Ph. 1956, thesis on Duns Scotus' theory of meaning. – Ph.D. 1959, thesis on the Aristotelian a priori as compared with modern conceptions of analytic sentences; Promotor: Stanisław Kamiński.

In 1971 the degree of Habilitated Doctor obtained at Warsaw University, Faculty of Social Sciences, Institut of Philosophy, in Humanities; specialization: Logic. Thesis: *Foundations of the Logical Theory of Beliefs*. – Docent (a post similar to that of Associated Professor) at Warsaw University, Białystok Branch (1972-1979). – Extraordinary Professor 1979, Ordinary Professor 1987 (academic titles awarded by the President of Poland, granting high academic rights).

Positions. Assistant (1956-59) and lecturer (1959-62) on Logic, Methodology of Sciences, and History of Philosophy at Catholic University of Lublin. – Researcher in the Project *Methodology of Empirical Sciences* run by Kazimierz Ajdukiewicz in Polish Academy of Sciences, Logic Section. – Lecturer (in Polish 'adjunkt', 1963-72) on logic at Warsaw University, Logic Section run by Janina Kotarbińska.

Head of the Section of Logic, Methodology and Philosophy of Science at the University of Białystok (formerly Białystok Branch of Warsaw University), 1975-2004; since 1996 the unit bears the title: *Chair of Logic, Informatics and Philosophy of Science*. Retired in 2004.

In June 1981 elected Rector of the Białystok Branch of Warsaw University as a candidate supported by the Independent Trade Union 'Solidarity'; won the contest with the candidate of the local Communist Party. In February 1982 removed from this post by Ministry of Education on demand of local (Białystok) military rulers.

In 2000-04: courses on Methodology of Social Sciences and Formal Logic

in *Collegium Civitas* – a private higher school of social sciences in Warsaw.

Since 2001: Professor of Higher School of Public Administration in Białystok; there teaches formal logic, methodology of sciences, Artificial Intelligence, politico-economic philosophy.

Running research projects

— 1975-80: a grant from the Documentation Centre of the Polish Academy of Sciences made it possible to handle some problems of text processing. This resulted in three doctoral dissertations.

— 1986-90: the coordinating of the research carried out in ten Polish universities (supported by the Ministry of Science and Education) on *Logical Systems and Algorithms for Computer-Aided Reasoning*. The research speeded up the development of the system MIZAR for automated proof-checking and mathematical databases, now classed among leading systems of this kind in the world. Moreover, it resulted in the book by W. Marciszewski and R. Murawski *Mechanization of Reasoning in a Historical Perspective*, Amsterdam 1995.

— 1992-94: supported by the Polish state Committee for R&D research in *Electronic Information System for Philosophy in Poland*. The project resulted in a book on Internet and establishing the domain www.calculemus.org including data bank, called *Logbank* (presently not continued, as obsolete).

— 1997-99: supported by the Polish state Committee for R&D *Research on Natural and Artificial Intelligence Carried out through Automation of Reasoning*.

— 2003-06: supported by the Polish Ministry of Science research in *Undecidability and Intractability in Social Sciences* (a part of results is included in this volume).

Other academic organizational activities in Poland

— 1975: establishing Section of Logic at the University of Białystok, later transformed into Chair of Logic, Informatics and Philosophy of Science.

— 1977-1980: organizing and running (as the Editor) the team of authors to publish *Dictionary of Logic as Applied in the Study of Language. Concepts, Methods, Theories*, Nijhoff, The Hague, 1981.

— 1980: establishing the journal *Studies in Logic, Grammar and Rhetoric*.

— Organizing and chairing Sections of Logic at Polish Philosophical Congresses in 1978 and 1987.

— Organizing and chairing six Polish workshops on issues of computability, especially with applications to social sciences.

— Since 1995 establishing and running, as Editor and Webmaster, the Web site www.calculemus.org/ including e-journals, conference materials, academic courses (schedules, lectures, exercises), etc.

Some International Activities

Lectures

— 1974-75: Visiting Professor of Logic at Pädagogische Hochschule in Halle, Germany.

— 1984: invited lectures on philosophy of logic at universities of Amsterdam, Utrecht, Leiden, Groningen.

— 1987: Visiting Professor at Salzburg University, Austria. Lectures on Methodology of Sciences and on Leibniz's philosophy and logic.

— 1991: invited lectures on Categorical Grammar at the University of Trento (Italy).

Conferences organized (Organizer and Co-Chairman)

— 1983: Warsaw, Poland: "The Foundations of Statements in Mathematics and in Philosophy"

— 1991, Oxford: the "British-Polish Logico-Philosophical Symposium"

Membership in academic societies

— Leibniz Gesellschaft

— Warsaw Scientific Society

(real member, former Chairman of Humanities Division and of Logic Committee)

— Polish Philosophical Society

— Polish Semiotic Society

— Polish Association for Logic and Philosophy of Science (Board Member)

— Committee for Philosophy of Polish Academy of Sciences (Board Member)

— Foundation for Informatics, Logic and Mathematics (President)

Books in English

Editor of *Studies in Formal Linguistics*, a special issue of *Studia Logica*, vol. 37, no. 1, 1978, pp. 134.

Editor of and contributor (a half of size) to *Dictionary of Logic as Applied in the Study of Language: Concepts, Methods, Theories*. Nijhoff, The Hague etc., 1981, pp. 436.

Coeditor (with W. Buszkowski and J. van Benthem) of and contributor to (two chapters) "Categorical Grammar". Benjamins, Amsterdam etc. 1988, pp. 365.

- Logic from a Rhetorical Point of View*. Walter de Gruyter, Berlin etc. 1994, pp. 312.
- Mechanization of Reasoning in a Historical Perspective* (co-author R. Murawski). Rodopi, Amsterdam etc. 1995, pp. 267.

Books in Polish

- Sztuka dyskusowania* [The Art of Discussion]. Iskry 1969, 1971; Aleph 1994, 1996.
- (Editor and coauthor): *Mała encyklopedia logiki* [A Concise Encyclopedia of Logic]. Ossolineum 1970, 1988.
- Podstawy logicznej teorii przekonań* [Foundations of the Logical Theory of Beliefs]. PWN 1972.
- Zarys logiki dla bibliotekoznawców. Cz. 2: Wybrane zagadnienia metodologii nauk i logicznej teorii języka* [An Outline of Logic for Students of Librarianship. Part Two: Selected Issues of the Methodology of Science and Logical Theory of Language]. Warsaw University Publishers 1972.
- Poradnik dla niezdecydowanych* [A Guide to Undecided (fundamentals of logic of decision)]. Ossolineum 1974.
- Rola dyskusji w nauczaniu i wychowaniu* [The Role of Discussion in Education]. Książka i Wiedza 1976.
- Metody analizy tekstu naukowego* [Methods of Analyzing Scholarly Texts]. PWN 1977, 1981.
- (Editor and coauthor) *Logika formalna: zarys encyklopedyczny z zastosowaniem do informatyki i lingwistyki* [Formal Logic. An Encyclopedic Outline with Applications to Informatics and Linguistics]. PWN 1987.
- Logika z retorycznego punktu widzenia* [Logic from a Rhetorical Point of View]. Polish Semiotic Society 1991.
- Sztuka rozumowania w świetle logiki* [The Art of Reasoning in the Light of Logic]. Aleph 1994.
- Tajniki Internetu* [Secrets of Internet]. Aleph 1994.
- Sztuczna Inteligencja* [Artificial Intelligence]. Znak 1998.

Main philosophical and scientific interests

Since the early student times interested in performances of the human mind as a cognitive system enjoying (at least potentially) the virtue of rationality, and in the role of intersubjective formalized reasoning for developing rationality. At the same time, interested in the idea of Infinite Mind as the source of rationality of the universe. Initially, the former was by him entertained within the paradigm of logical empiricism, while the latter within

the paradigm of neothomism which he attempted (rather hopelessly) to reconcile with logical empiricism.

This attempt and the both said approaches were given by him up in the early sixties. Logical empiricism has been replaced with something like logical apriorism (in Leibniz's and Georg Cantor's vein) owing to a research of his own (within a project run by Kazimierz Ajdukiewicz) concerning a priori elements in the use of ostensive definitions. Neothomism, recognized by him as too alien to contemporary science, has been superseded with a vivid interest in the role played in cosmic evolution by the calculability factor as facing the challenge of complexity (in a broad sense of calculation, like in Leibniz's saying about calculating as the creative factor of the universe).

Since the seventies – interested in the role of computers in developing science, especially with respect to formalization of reasoning. Since the eighties this has been reinforced by studies on Leibniz. All that was combined with reflexion on similarities and differences between the computer and the human mind, accompanied by a conviction about mind's abilities that cannot be matched by algorithmic procedures. Since the nineties there appeared an interest in social structures as information-processing systems, this phenomenon being taken as a basis for politico-economic philosophy.

Main articles of faith. Appreciating the key philosophical position of the notions of complexity and computability, also with regard to social development, one should believe that the two following ideas, being the source of computability, are a priori given the human mind to deal with complexity: (i) the set-theoretical *axiom of comprehension*, or axiom of abstraction, which says that there exist sets defined by our concepts (if properly formed), and (ii) the *well-ordering theorem* (equivalent to the axiom of choice), which states that *every set can be well-ordered*. Both form the source of conceptual creativity unavailable to machines. Such a creativity flourishes even at a most primitive level, as enlarging a set of concepts by means of ostensive definitions, up to the heights of abstraction in mathematics and philosophy.

Hobbies: talking with wife on everything, forest and mountain wanderings, visiting nice cafes, perceiving old architecture, reading political and economic stories, contemplating Polish romantic poetry.

Mottos

On establishing sets and, thereby, cardinalities:

Omnia in numero dispoisisti. Book of Wisdom, 11: 21.

On ordering the universe via ordering sets:

Sapientis est ordinare. Aristotle (Meth. 982a) in a scholastic garb.

ROMAN MATUSZEWSKI

Scientific fields: computer science, logic

Scientific research

- computer aided formalisation of mathematics
- machine translation of mathematical texts into English
- mathematical knowledge management

Education

- m.sc, eng. 1975, Warsaw Technical University
- PhD, 2000, Shinshu University, Nagano, Japan

Employment

- 1975–1976, Warsaw Technical University
- 1976–1997, Warsaw University (Białystok),
Institute of Mathematics,
Department of Logic, Informatics and Philosophy of Science
- 1997, University of Białystok
Computer Laboratory of the Filological Faculty (chair)

Scientific experience

- 1996, 2000, 2004, Shinshu State University, Nagano, Japan
- 1984–1992, Universite Catholique de Louvain, Belgium

Last publications

- Computer Checked Mathematical Texts Presented in Natural Language*, International Congress of Mathematicians, ICM 2006, Madryt, Philippe le Hodey Foundation – supporting scientific research in Poland, proceedings of the conference *Metody ilościowe w badaniach ekonomicznych*, SGGW, 2005,
Mizar: the first 30 years, “Mechanized Mathematics and Its Applications”, Vol. 4, Nr. 1, 2005, with Piotr Rudnicki,

- Mizar texts automatically translated into English*, Proceedings of the Conference “Systems for integrated computation and deduction”, CADE 99, Trento, 1999,
- Automatic Translation of Machine-Checked Mathematical Texts into English*, Proceedings of the 11-th International Congress of Logic, Methodology and Philosophy of Science, Cracow, 1999,
- Journal of Formalized Mathematics – a collection of mechanically checked articles, Proceedings of International Congress of Mathematicians, ICM 98, Berlin, 1998,
- Projections in n-Dimensional Euclidean Space to Each Coordinates*, “Formalized Mathematics”, Vol. 6, Nr. 4, 1997 (with Yatsuka Nakamura),
- Mizar MSE – a computer aided course for elementary logic of quantifiers*, Proceedings of International Congress of Mathematicians, ICM 94, Zurich, 1994.

Last scientific grants

- member of the European Union 5FP Project: *MKMNET – Mathematical Knowledge Management*, 2002–2003 (IST-2001-37057),
- member of the European Union 5FP Project: *TRIAL SOLUTION – Tools for Learning Systems*, 2001–2002 (IST-2001-35447),
- member of the European Union 5FP Project: *CALCULEMUS – Systems for Integrated Computation and Deduction*, 2000–2004 (HPRN-CT-2000-00102),
- member of the European Union 5FP Project: *TYPES – Computer Assisted Reasoning Based on Type Theory*, 1999–2003 (IST-1999-29001),
- member of the Polish State Scientific Research Committee Project: *Temporal representation of the knowledge and their implementation in the medical systems*, 2006–2009, (3 T11F 011 30),
- member of the Polish State Scientific Research Committee Project: *Undecidability and Algorithmic Intractability in the Social Sciences*, 2003–2006, (2 H01A 030 25),
- member of the Polish State Scientific Research Committee Project: *New techniques for the computer aided formalization of mathematics*, 2003–2005, (4 T11C 039 24),
- co-principal investigator in the Polish State Scientific Research Committee Project: *Research on the natural and artificial intelligence with use of automated reasoning*, 1997–1999 (KBN 8 T11C 018 12),
- member of the Office of Naval Research (USA) Project: *Further Development of Journal Formalized Mathematics*, 1997–1998 (N00014-97-1-0777),
- co-principal investigator of the Office of Naval Research (USA) Pro-

- ject: *Automated Hyperlinking in an Electronic Mathematical Proof-Checked Journal*, 1995–1996 (N00014-95-1-1336),
- principal investigator in the Joint Project of the Office of Naval Research (USA) and the Polish State Scientific Research Committee: *The QED Workshop – Computer Oriented Formalization of Mathematics*, 1995, under auspices of the Polish Prime Minister (OGT 42/95, N00014-95-M-0072).

Other

- Conference Chair in the Mathematical Knowledge Management Conference, MKM2004, Bialowieza, <http://mizar.org/MKM2004/>
- Programme Committee member and reviewer in the workshop *Proof Transformation and Presentation, and Proof Complexities*, International Joint Conference on Automated Reasoning, Siena, 2001,
- elected full member *International Academy of Sciences of Nature and Society*, Moscow, from 1997,
- member *Association for Automated Reasoning*, from 1994, (<http://www-unix.mcs.anl.gov/AAR/names.html>),
- scientific secretary of the scientific project of European Union, DG XII, Brussels, 1987–1988,
- silver merit cross of the Republic of Poland for scientific activity in 2004.
- bronze merit cross of the Republic of Poland for scientific activity in 1981.

Editor

- *QED Workshop II*, Report (1995), <ftp://ftp.mcs.anl.gov/pub/qed/workshop95/qed95rep.ps>
- *Formalized Mathematics*, ISSN 1426–2630 (since 1990).

Recommended Reading

- History of the Mizar Project, <http://mizar.org/people/romat/MatRud-2005.pdf>
- The QED Manifesto, <ftp://ftp.mcs.anl.gov/pub/qed/manifesto>
- The Journal of Formalized Mathematics, <http://fm.mizar.org>

ROMAN MURAWSKI

Born on 15th July 1949 in Poznań, Poland.

Education

Studied mathematics at Adam Mickiewicz University in Poznań (1967–1972). 1972 M.Sc. (master thesis in mathematical logic). 1977–1979 postgraduate studies at the Faculty of Mathematics, Computer Science and Mechanics of the University of Warsaw. 1979 Ph.D. in mathematics (mathematical logic and foundations of mathematics). 1975–1979 studied at the Pontifical Theological Faculty in Poznań where in 1980 he was awarded the degree of master of theology. 1982–1985 postgraduate studies of philosophy at the Pontifical Theological Faculty in Poznań. 1985 he obtained the degree *licentiatus in sacra theologia*. In 1992 he was awarded the degree of *doctor habilitus* in mathematics (specialization: mathematical logic) at the Faculty of Mathematics and Physics of Adam Mickiewicz University in Poznań. In 2001 the president of Poland awarded him the title of professor of philosophy (specialization: philosophy of mathematics).

Positions

1972–1974 junior assistant and 1974–1979 assistant lecturer in the Institute of Mathematics of Adam Mickiewicz University in Poznań. 1979–1993 – assistant professor (adjunct) there. 1993–2004 associate professor, since 2004 – full professor. Since 1996 he is a chairman of the Department of Mathematical Logic at the Faculty of Mathematics and Computer Science of Adam Mickiewicz University in Poznań. Since 2005 he is a vice-dean for grants and international cooperation of this faculty.

Activity in scientific societies

1993–2006 deputy chairman of the Scientific Council of Polish Association for Logic and Philosophy of Science. Since 2006 president of the Association. Member of the Commission for the History of Mathematics of Polish Mathematical Society and of the Committee of History of Science and

Technology of Polish Academy of Sciences. Member of National Committee for Collaboration with International Union of History and Philosophy of Science.

Member of: Polish Mathematical Society, Polish Philosophical Society, American Mathematical Society, Deutsche Vereinigung für Mathematische Logik und Grundlagenforschungen der Exakten Wissenschaften, Commission of Logic of Warsaw Scientific Society, Kurt Gödel Society, European Foundation for Logic, Language and Information, Association for Symbolic Logic, Societas Humboldtiana Polonorum.

International activity

As a fellow of Alexander von Humboldt-Stiftung he visited Institute of Mathematics of the University of Heidelberg (1980–1982) and Institute of Philosophy of the University in Erlangen (1993–1994) (Germany). In 1992 and 1993 he was a fellow of the University of Amsterdam, Faculty of Mathematics and Computer Science (Holland) and in 1993 an associate fellow of Merton College in Oxford (Great Britain). Since 1996 he regularly visits as a fellow of DAAD the University of Hannover (Germany). In 2003 and 2004 he was a fellow of Vrije Universiteit Brussel (Belgium) and visited there Centre for Logic and Philosophy of Science.

He gave invited lectures at universities in Amsterdam, Heidelberg, Prague, Beograd, London, Cambridge, Oxford, Manchester, Leeds, Hannover, Erlangen, Neubrandenburg, Greifswald, Brussel and in ETH in Zurich.

Invited lectures at conferences

- 1985 – History of logic, Cracow (Poland),
- 1993 – Colloquium Logicum, Münster (Germany),
- 1995 – Cultural History of Mathematics, Berlin (Germany),
- 1995 – 6th Congress of Polish Philosophy, Toruń (Poland),
- 1996 – Limits of Science, Cracow (Poland),
- 1997 – 11th Summer School of the History of Mathematics Kołobrzeg (Poland),
- 1997 – Alfred Tarski and Vienna Circle, Vienna (Austria),
- 1997 – 3rd Congress of the Gesellschaft für Analytische Philosophie, München (Germany),
- 1997 – Mutual Influences Between Logic and Computer Science, Białystok (Poland),
- 1998 – Logic and Intelligent Systems, Zakopane (Poland),
- 1999 – Applications of Logic in Philosophy and Foundations of Mathematics, Karpacz (Poland),

- 1999 – 11th International Congress of Logic, Methodology and Philosophy of Science, Cracow (Poland),
- 2000 – Tarski vs. Hilbert. Semantical Consequences and Formal Derivability, Zakopane (Poland),
- 2001 – Logic and Logical Philosophy, Dresden (Germany),
- 2001 – Tarski Centenary Conference, Warsaw (Poland),
- 2002 – Congress of Polish Mathematical Society, Łódź (Poland),
- 2003 – Von Neumann and Hilbert's School, Zakopane (Poland),
- 2004 – La Reyunement de la Philosophie Polonaise au XX Sciecle. L'Heritage Philosophique de Kazimierz Twardowski, Paris (France),
- 2005 – 5th European Congress for Analytic Philosophy, Lisbon (Portugal)
- 2006 – Trends in Logic IV, Toruń (Poland).

Editorial activity

Member of editorial boards of the following journals: *Roczniki Polskiego Towarzystwa Matematycznego, Seria II: Wiadomości Matematyczne* (Poland; since 1990), *The Review of Modern Logic* (USA; since 1994), *Studies in Logic, Grammar and Rhetoric* (Poland; since 2002), *Diametros* (Poland, since 2004) as well as of the following book series: *Modern Logic* (USA; since 1994), *Advanced Studies in Mathematics and Logic* (Italy; since 2004). Secretary of the editorial board of the book series *Advanced Topics in Mathematics* (Poland; since 1996).

Awards

- Awards of Ministry of Higher Education in 1982, 1987 and 1996.
- Award of Polish Mathematical Society in 1997.
- In 2006 he awarded a scholar grant by Foundation for Polish Science.

Researches in mathematical logic and foundations of mathematics, philosophy of mathematics, history of logic and mathematics.

Publications

Published 11 books, 120 papers, 36 reviews and 19 abstracts.

Books

- Filozofia matematyki. Antologia tekstów klasycznych* [Philosophy of Mathematics. Anthology of Classic Texts], Wydawnictwo Naukowe UAM, Poznań 1986, 1994, 2003
- Rozwój symboliki logicznej* [Development of Logical Symbolism], Wydawnictwo Naukowe UAM, Poznań 1988

- Funkcje rekurencyjne i elementy metamatematyki. Problemy zupełności, rozstrzygalności, twierdzenia Gödla* [Recursive Functions and Elements of Metamathematics. Problems of Completeness, Decidability, Gödel's Theorems], Wydawnictwo Naukowe UAM, Poznań 1990, 1991, 2000
- Mechanization of Reasoning in a Historical Perspective*, Edition Rodopi, Amsterdam/Atlanta, GA, 1995 (co-author: W. Marciszewski)
- Filozofia matematyki. Zarys dziejów* [Philosophy of Mathematics. An Outline of History], Wydawnictwo Naukowe PWN, Warszawa 1995, 2001
- Euphony and Logos*, Edition Rodopi, Amsterdam–Atlanta, GA, 1997 (co-editor: J. Pogonowski)
- Recursive Functions and Metamathematics. Problems of Completeness and decidability, Gödel's Theorems*, Kluwer Academic Publishers, Dordrecht/Boston/London 1999
- Zählen. Grundlage der elementaren Arithmetik*, Verlag Franzbecker, Hildesheim–Berlin 2001 (co-author: Th. Bedürftig)
- Współczesna filozofia matematyki* [Modern Philosophy of Mathematics], Wydawnictwo Naukowe PWN, Warszawa 2002
- Wstęp do teorii mnogości* [Introduction to Set Theory], Wydawnictwo Naukowe UAM, Poznań 2005, 2006 (co-author: K. Świrydowicz)
- Podstawy logiki i teorii mnogości* [Elements of Logic and Set Theory], Wydawnictwo Naukowe UAM, Poznań 2006 (co-author: K. Świrydowicz)

Address

Adam Mickiewicz University
 Faculty of Mathematics and Computer Science
 ul. Umultowska 87
 61–614 Poznań
 Poland
 e-mail: rmur@amu.edu.pl
 home pages: <http://www.logika.amu.edu.pl>, <http://main.amu.edu.pl/~rmur>



0229816/22

KAZIMIERZ TRZĘSICKI

Kazimierz Trzęsicki, professor of the University in Białystok, Chair of Logic, Informatics and Philosophy of Science. He lectures on logic (foundation of logic and temporal logic), semiotics, methodology of science and computer ethics. He is the author of handbooks: *Logika. Nauka i sztuka* and *Elementy logiki i teorii mnogości*. Two his monographs were published in Biblioteka Myśli Semiotycznej: *Elementy logiki dla humanistów* and *Logika nieformalna*. The monograph *Logika i teoria mnogości. Ujęcie systematyczno-historyczne* was published by EXIT in the series: Problemy współczesnej nauki. Teoria i zastosowania. Informatyka. Among recently published papers are: „Medical informatics ethics and its subject”, *Annales Academiae Medicae Białostocensis*, vol. 50, 2005, supp. 2, pp. 20–22; „A. N. Priora związki ze szkołą lwowsko-warszawską” [in:] *Ratione et Studio*, Białystok 2005, pp. 269–288; „Medical informatics ethics – subject and major issues”, *Parerga*, Warszawa 2005, pp. 5–19; „Logical and methodological assumptions of Ajdukiewicz's and Kripke-Putnam's views of meaning” [in:] J. J. Jadacki and J. Paśniczek (eds.), *The Lvov–Warsaw School – The New Generation* (Poznań Studies in the Philosophy of the Sciences and the Humanities, vol. 89), pp. 315–323. Amsterdam/New York, NY: Rodopi, 2006; „Leibnizjańskie inspiracje informatyki”, *Filozofia Nauki* 2006, „Wkład polskich logików w światową informatykę”, *Filozofia Nauki*, 2006.