# Constructivity and Computability in Historical and Philosophical Perspective

LOGIC, EPISTEMOLOGY, AND THE UNITY OF SCIENCE

VOLUME 34



*Logic, Epistemology, and the Unity of Science* aims to reconsider the question of the unity of science in light of recent developments in logic. At present, no single logical, semantical or methodological framework dominates the philosophy of science. However, the editors of this series believe that formal techniques like, for example, independence friendly logic, dialogical logics, multimodal logics, game theoretic semantics and linear logics, have the potential to cast new light on basic issues in the discussion of the unity of science.

This series provides a venue where philosophers and logicians can apply specific technical insights to fundamental philosophical problems. While the series is open to a wide variety of perspectives, including the study and analysis of argumentation and the critical discussion of the relationship between logic and the philosophy of science, the aim is to provide an integrated picture of the scientific enterprise in all its diversity.

More information about this series at http://www.springer.com/series/6936

Jacques Dubucs • Michel Bourdeau
Editors

# Constructivity and Computability in Historical and Philosophical Perspective

Springer

*Editors*
Jacques Dubucs
Michel Bourdeau
Institut d'histoire et de philosophie
des sciences et des techniques
(IHPST; CNRS-Paris1-ENS)
Paris, France

ISSN 2214-9775 ISSN 2214-9783 (electronic)
ISBN 978-94-017-9216-5 ISBN 978-94-017-9217-2 (eBook)
DOI 10.1007/978-94-017-9217-2
Springer Dordrecht Heidelberg New York London

Library of Congress Control Number: 2014947463



Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

This book covers the almost 80 years from Turing's seminal paper of 1936 to the present and focuses on two developments during that period.[1]

On the one hand, the development of computability theory and complexity theory called for generalizations, restrictions, and other modifications of Turing's original machine. On the other hand, it became clear that recursion theory cannot serve as a foundational analysis of the notion of computable function(al), which has to be accepted as primitive.

In different ways, both of these developments contributed to a unified view of logic. The first development helped bringing various phenomena together into a

[1]In her recent book *Logic and Philosophy of Mathematics in the Early Husserl* (Dordrecht: Springer 2010), Stefania Centrone has drawn attention to the fact that the first to call for a theoretical study of computability probably was Husserl. In his *Philosophy of Arithmetic* of 1891 (*Collected Works* vol. X, trl. D. Willard, Dordrecht: Springer 2003), he formulated the following 'general postulate of arithmetic': 'the symbolic formations that are different from the systematic numbers must, wherever they turn up, be reduced to the systematic numbers equivalent to them, as their normative forms. Accordingly there arises, as the first task of Arithmetic, to separate all conceivable symbolic modes of formation of numbers into their distinct types, and to discover for each type the methods that are reliable and as simple as possible for carrying out that reduction' (p. 277). He considered the arithmetical operations to be but the methods to carry out that reduction (p. 277), and understood computation, in arithmetic but also more generally, as 'any rule-governed mode of derivation of signs from signs within any algorithmic sign-system according to the "laws"—or better: the conventions—for combination, separation, and transformation peculiar to that system' (p. 273). He also raised the question of the computability of numbers that are defined by a system of equations (pp. 296–298). Husserl did not, however, attempt to develop such a theory of computability, and his suggestion seems not to have been picked up on by his contemporaries either. Moreover, it seems that none of those who came to play a role in the development of computability theory was aware of Husserl's suggestion. They had, of course, ample independent motivation. But it would be interesting to know, for example, whether Skolem knew this chapter by Husserl. Skolem visited Göttingen during the winter of 1915–1916; that was Husserl's last term as a professor there, before moving to Freiburg.

framework in which the original Turing machine turned out to be a special, and in some contexts ideal(ized), case. The second development dispelled a serious misunderstanding according to which recursion theory should ideally replace informal reflection on the notion of computable function. Instead, there is room and indeed need for both, depending on one's purpose.[2]

The two developments are discussed in seven chapters, as follows.

Göran Sundholm's opening chapter, *Constructive Recursive Functions, Church's Thesis, and Brouwer's Theory of the Creating Subject*, first discusses recursive versus constructive functions and, following Heyting, stresses that from a constructive point the former cannot replace the latter. The second half of the paper treats of the Kreisel-Myhill theory CS for Brouwer's Creating Subject, and its relation to BHK meaning-explanations and Kripke's Schema. Kripke's Schema is reformulated as a principle and shown to be classically valid. Assuming existence of a verification-object for this principle, a modification of a proof of conservativeness of Van Dalen's, is shown to give a relative BHK meaning explanation for the Kreisel-Myhill connective. The result offers an explanation of why Kripke's Schema can be used as a replacement of the Theory of Creating Subject when formulating Brouwerian counter-examples. It also shows that the Theory of Creating Subject is classically valid.

The relation between computation and machine is as old as the abacus, but only with Turing's pioneering work to this relation became central to computability theory. Chapter 2, Jean Mosconi's *The Developments of the Concept of Machine Computability from 1936 to the 1960s*, deals with this aspect, which is crucial from a technological point of view. Mosconi explains how Turing's ideas were gradually adopted, developed and modified, leading to something much closer to the actual computer. The development can be divided into three stages. First, there is a strong contrast between a quick acceptance of the conceptual analysis of computation and a scarce use of the technical potentialities of Turing's contribution: While Gödel quickly saw the philosophical relevance of Turing's work (it made him overcome his objections to Church's Thesis), Post was perhaps the first to see its mathematical fruitfulness. The Turing Machine enters in a crucial way in Post's 1947 proof of the algorithmic unsolvability of Thue's problem. Then follows the rise of the theory of Turing Machines in the 1950s. Here the Turing Machine was not used to solve problems or gain philosophical insights, but was taken as a proper object of mathematical investigation. The main idea was to include the Turing Machine in the 'general and logical theory of automata' that Von Neumann suggested to develop in the late 1940s. From this point of view, it became possible to study a larger class of automata: not only finite ones, as it was initially the case with

[2] That leaves open the question whether the notions or recursive function and computable function nevertheless have the same extension. Church's Thesis asserts that they do; but within the intuitionistic theory of the creating subject, Kripke has constructed a computable function that is not recursive. This counterexample is discussed in the papers by Sundholm and Van Atten in the present volume.

Shannon or McCulloch's automata, but also infinite ones. Starting from the general structure of the original Turing Machine, various restrictions and generalizations were defined, leading to a hierarchy of automata, and to a better understanding of what Turing Machines can achieve and how. Finally, the progress of technology made it necessary to rework Turing's model so as to establish the link between computability theory and computer practice. Hao Wang's machine B (1957), which uses the notion of instruction instead of the notion of state, was the first attempt to bridge the gap, but a completely explicit model of a program and register machine was arrived at only 1963, by Stephenson and Sturgis.

With the development of computer science and the increasing demand for feasibility, computability theory gave birth to complexity theory. Chapters 3 and 4, by Serge Grigorieff and Marie Ferbus-Zanda, respectively, are dedicated to Kolmogorov complexity, which may be seen as an offspring of computability theory. Chapter 3, Grigorieff's *Information and randomness*, is the more technical one of the two. Kolmogorov introduced a radically new approach to the measurement of information. In the combinatorial approach, the information content of an object $x$ was defined as the length of the shortest binary word which 'encodes' $x$. Kolmogorov's idea was to bring in the resources of computability by defining the information content of $x$ as the length of the shortest program which computes $x$. The chapter gives the main concepts and results, together with their proofs; some related notions of complexity, like Levin monotone complexity or Schnorr process complexity are also presented. A new step was taken when Kolmogorov, and independently Chaitin, noticed that the algorithmic theory of information could also be used to give a definition of randomness, which was still lacking after the axiomatization of probability theory. The basic idea is that a word is random if it is incompressible, that is, if there is no shorter way to describe it. Martin-Löf has shown that the necessary condition was also a sufficient one.

Chapter 4, *Application to Classification Theory*, takes a more philosophical stance and shows the relevance of Kolmogorov complexity to computer science, where it has already found a number of useful applications. With the World Wide Web and its huge network of machines, the analysis of information processing has become even more challenging and the need for a classification even more urgent. Up to now the two main approaches have been classification by compression and the so-called Google classification. But they still lack a good formalisation. As they both use complexity, Ferbus-Zanda proposes to take Kolmogorov algorithmic information theory as the mathematical foundation of information classification we are looking for. The two main approaches can now be formulated in terms of two types of definition of mathematical objects, namely iterative definitions, based on set theoretical union, and inductive or recursive definitions, based on set theoretical intersection; they can also be seen as bottom-up and top-down versions of the same underlying theory, of which Ferbus-Zanda gives a short presentation. Furthermore, she shows how these two dual modes are also found in information systems, particularly the relational database model introduced by Codd in the 1970s.

Chapter 5, *Proof Theoretical Semantics and Feasibility*, by Jean Fichot, returns to constructivity. According to one of the best known justifications of constructive reasoning, the meaning of logical constants is given by their introduction rules or, what amounts to the same, by what counts as a canonical proof of it. But canonical proofs are idealisations of the ones we normally use; depending on what types of knowledge one admits, proof theoretical semantics may therefore be liable to be invalidated by the non-feasibility of the canonical proofs. Fichot's contribution reviews two attempts to overcome this difficulty and to go one step further towards feasible canonical proofs. The first is the new recursion-theoretic characterisation of polytime functions given in 1997 by Bellantoni and Cook. Besides natural numbers, they use feasible numbers and succeed in giving a meaningful explanation in the style of Dummett for those numbers. Unfortunately, there remains a gap between feasible arithmetic and the feasible theory of proofs we are looking for. The second attempt is light affine logic, a system introduced by Asperti in 1998 and further studied by Baillot and Terrui. The main idea comes from light linear logic, where Girard added a new modal operator, '§', to the *of course* operator, '!', of linear logic. Just as '!' allows for contraction, '§' allows for weakening. Besides the formulas that express perennial propositions and can be contracted and reused as often as wanted, and the formulas that are simply true, we must now admit a third kind of formulas. But this splitting of '!' into two modal operators gives very useful tools for controlling the computational complexity of the cut elimination procedure. Jean Fichot explains how the justification for the logical rules of such a system can be given.

The sixth chapter, *Recursive Functions and Constructive Mathematics*, by Thierry Coquand, addresses one of the most fundamental questions concerning the relations between constructivity and computability: is the theory of recursive functions needed for a rigorous development of constructive mathematics? The answer is negative in both the theoretical and the practical sense. The argument proceeds in two steps. The first one shows how the success of recursion theory fostered a lack of sense for constructivity. As was noted early on by Heyting and Skolem, from a constructive point of view, the theory of recursive functions cannot give us a formal definition of the intuitive notion of computable function. Kleene's definition of $\mu$, for instance, uses existential quantification: if $R(x, y)$ is a recursive relation and if $(x)EyR(x, y)$ holds, then $\mu yR(x, y)$ is a recursive function of $x$. But we are left with a dilemma: if the quantifier is interpreted non-constructively, the relation between computability and constructivity is lost; if it is interpreted constructively, then the definition in fact presupposes some notion of a computable function. The second step in Coquand's argument begins with Bishop's *Foundations of Constructive Analysis* (1967). The book was a breakthrough: It taught us that it was not only theoretically possible but also practically more satisfactory to introduce functions in constructive mathematics without mentioning recursivity. Much current work in constructive mathematics strongly relies on Bishop's ideas. It is also noteworthy that one popular definition of constructive mathematics, according to which it is *mathematics developed using intuitionistic logic*, is independent of any notion of algorithm.

The final chapter, *Gödel and Intuitionism*, by Mark van Atten, starts with a brief survey of Gödel's personal contacts with Brouwer and Heyting. Some examples are discussed where intuitionistic ideas had a direct influence on Gödel's technical work. Then it is argued that the closest rapprochement of Gödel to intuitionism is seen in the development of the Dialectica Interpretation, during which he came to accept the notion of computable functional of finite type as primitive. It is shown that Gödel already thought of that possibility in the Princeton lectures on intuitionism of Spring 1941, and evidence is presented that he adopted it in the same year or the next, long before the publication of 1958. Draft material for the revision of the Dialectica paper is discussed in which Gödel describes the Dialectica Interpretation as being based on a new intuitionistic insight obtained by applying phenomenology, and also notes that relate the new notion of reductive proof to phenomenology. In an appendix, attention is drawn to notes from the archive according to which Gödel anticipated autonomous transfinite progressions when writing his incompleteness paper.

This book has grown out of the meeting 'Computability and Constructivity in Historical and Philosophical Perspective', which took place at the École normale supérieure in Paris, December 17–18, 2006. The organisers were Jacques Dubucs (IHPST), Michel Bourdeau (IHPST), Jean-Paul Delahaye (Université des Sciences et Technologies de Lille), and Gerhard Heinzmann (Université de Nancy II). The meeting was a Joint Session of the two divisions of the International Union of History and Philosophy of Science (IUHPS): the Division of Logic, Methodology and Philosophy of Science (DLMPS) and the Division of History of Science and Technology (DHST).

Paris, France — Jacques Dubucs
June 2013 — Michel Bourdeau

# Contents

**1 Constructive Recursive Functions, Church's Thesis, and Brouwer's Theory of the Creating Subject: Afterthoughts on a Parisian Joint Session** ........ 1
Göran Sundholm

**2 The Developments of the Concept of Machine Computability from 1936 to the 1960s** ........ 37
Jean Mosconi

**3 Kolmogorov Complexity in Perspective Part I: Information Theory and Randomness** ........ 57
Marie Ferbus-Zanda and Serge Grigorieff

**4 Kolmogorov Complexity in Perspective Part II: Classification, Information Processing and Duality** ........ 95
Marie Ferbus-Zanda

**5 Proof-Theoretic Semantics and Feasibility** ........ 135
Jean Fichot

**6 Recursive Functions and Constructive Mathematics** ........ 159
Thierry Coquand

**7 Gödel and Intuitionism** ........ 169
Mark van Atten