Annals of Pure and Applied Logic 167 (2016) 868–878

Contents lists available at ScienceDirect

Annals of Pure and Applied Logic

www.elsevier.com/locate/apal

Symbioses between mathematical logic and computer science

Andreas Blass

Mathematics Department, University of Michigan, Ann Arbor, MI 48109-1043, USA

ARTICLE INFO

Article history:
Available online 13 May 2016

MSC:
primary 03B70
secondary 03D70, 03C80, 68Q19

Keywords:
Polynomial-time computation
Fixed points
Capturing complexity classes
Existential fixed-point logic
Choiceless polynomial time
Undecidability

ABSTRACT

This is a survey of some of the many interactions between mathematical logic and computer science. The general theme is that mathematical logic provides tools for understanding and unifying topics in computer science, while computer science provides new ways of looking at logical issues and underlines the importance of areas of logic that might otherwise have been neglected.

1. Introduction

Computer science has strong connections with numerous aspects of mathematical logic, but those aspects are sometimes different from those traditionally studied for pure mathematical purposes. This paper is a survey of some of those interactions. Its main themes are (1) logic's clarification of computational concepts and (2) computation's infusion of new concepts and questions into logic.

Section 2 mentions some of the earliest interactions. They involve what would now be called computability theory or recursion theory, rather than computer science, as they do not involve any resource limitations. Nevertheless, I consider them worth pointing out, to emphasize the contribution of mathematical logic to a broad topic arising in diverse fields of mathematics.

Section 3 concerns first-order logic, especially the notion of structure at the basis of the semantics of first-order logic, pointing out that this same notion of structure is well-suited for many roles in computer science. I also describe here a computational view of the expressive power of first-order logic.

Section 4 concerns logical systems obtained by adding to first-order logic the central non-first-order concept involved in computation, namely the notion of iteration. The logical incarnation of iteration is

E-mail address: ablass@umich.edu.

http://dx.doi.org/10.1016/j.apal.2014.04.018
0168-0072/

given by fixed-point operators, which are involved in the logical characterizations of various complexity classes. Fixed-point operators play a role in all the subsequent sections of this paper.

Section 5 begins with a very general conjecture of Yuri Gurevich that polynomial-time computation, on general finite structures, cannot be exactly captured by any logic (in a very broad sense of "logic"). I include a description of a logic, "choiceless polynomial time," that comes surprisingly close to capturing PTime, and I discuss the possibility that it might actually be a counterexample to Gurevich's conjecture.

Section 6 is devoted to a logic of a rather different sort, obtained from first-order logic by not only adding a fixed-point operator but also removing part of the machinery of first-order logic, namely the universal quantifier. This existential fixed-point logic turns out to have strong connections to computation as well as a certain mathematical elegance.

Fixed-point logics do not admit a complete deductive system in the usual sense, but it is not difficult to write down a reasonable system of axioms and rules governing the behavior of a fixed-point operator. The question then arises whether the unavoidable incompleteness of such a system affects only complicated, Gödel-style sentences or whether very simple truths might be unprovable. In Section 7, I describe the system that I have in mind and conclude with an open question about a simple truth that might be unprovable.

2. Prehistory

In the early part of the twentieth century, several branches of mathematics had encountered fundamental questions about algorithmic issues. Examples include the following.

Number theory: Given a polynomial equation in many variables, with integer coefficients, determine whether there is an integer solution. (Hilbert's tenth problem [29])

Topology: Given two topological manifolds, in the form of finite triangulations, determine whether they are homeomorphic.

Group theory: Given a group, by means of (finitely many) generators and relations, and given two words in the generators, determine whether these words represent the same element of the group. (The Word Problem [24])

Logic: Given a first-order sentence, determine whether it is logically valid. (The Decision Problem)

In each of these cases, the problem was originally to find an algorithm. For example, Hilbert's statement of his tenth problem is, after specifying the equations under discussion, "One is to give a procedure by which it can be determined by a finite number of operations whether the equation is solvable in rational integers."[1] Similarly, Dehn's statement of the word problem is, "One is to give a method to decide, with a finite number of steps, whether [a given] element is equal to the identity or not."[2] And Hilbert and Ackermann [30] stated the decision problem as, "The decision problem is solved if one knows a procedure which, given a logical expression, allows one to decide by finitely many operations its validity resp. its satisfiability."[3]

Logicians provided a completely new way of viewing such problems. They replaced "Find an algorithm ..." with "Is there an algorithm ...?" That is, they introduced the notion that there might not exist an algorithm and (crucially) that such a result might be provable. This means that, instead of the then prevalent notion of algorithm, namely "I'll know it when I see it," one needs a precise mathematical notion of algorithm, or at least of algorithmic computability. Such definitions were provided independently by Church [16] using the λ-calculus and by Turing [38] using what are now known as Turing machines. Undecidability

[1] [M]an soll ein Verfahren angeben, nach welchem sich mittelst einer endlichen Anzahl von Operationen entscheiden läßt, ob die Gleichung in ganzen rationalen Zahlen lösbar ist.

[2] Man soll eine Methode angeben, um mit einer endlichen Anzahl von Schritten zu entscheiden, ob dieses Element der Identität gleich ist oder nicht.

[3] Das Entscheidungsproblem ist gelöst wenn man ein Verfahren kennt, das bei einem vorgelegten logischen Ausdruck durch endlich viele Operationen die Entscheidung über die Allgemeingültigkeit bzw. Erfüllbarkeit erlaubt.