Annals of Pure and Applied Logic 164 (2013) 1224–1235

Contents lists available at SciVerse ScienceDirect

Annals of Pure and Applied Logic

www.elsevier.com/locate/apal

Consistency, optimality, and incompleteness

Yijia Chen [a], Jörg Flum [b,*], Moritz Müller [c]

[a] *Shanghai Jiaotong University, China*
[b] *Albert-Ludwigs-Universität Freiburg, Germany*
[c] *Kurt Gödel Research Center, University of Vienna, Austria*

ARTICLE INFO

Article history:
Received 24 October 2011
Received in revised form 13 June 2013
Accepted 13 June 2013
Available online 2 July 2013

MSC:
03B30
03E35
03F30
03F40

Keywords:
Consistency
Optimal algorithms
First-order arithmetic
Gödel's Second Incompleteness Theorem

ABSTRACT

Assume that the problem P_0 is not solvable in polynomial time. Let T be a first-order theory containing a sufficiently rich part of true arithmetic. We characterize $T \cup \{Con_T\}$ as the minimal extension of T proving for some algorithm that it decides P_0 as fast as any algorithm $\mathbb{B}$ with the property that T proves that $\mathbb{B}$ decides P_0. Here, Con_T claims the consistency of T. As a byproduct, we obtain a version of Gödel's Second Incompleteness Theorem. Moreover, we characterize problems with an optimal algorithm in terms of arithmetical theories.

1. Introduction

By Gödel's Second Incompleteness Theorem a consistent, computably enumerable and sufficiently strong first-order theory T cannot prove its own consistency Con_T. In other words, $T \cup \{Con_T\}$ is a proper extension of T.

In Bounded Arithmetic one studies the complexity of proofs in terms of the computational complexity of the concepts involved in the proofs (see e.g. [1, Introduction]). Stronger theories allow reasoning with more complicated concepts. For example, a computational problem may be solvable by an algorithm whose proof of correctness needs tools not available in the given theory; moreover, stronger theories may know of faster algorithms solving the problem. When discussing these issues with the authors, Sy-David Friedman asked

* Corresponding author.
E-mail addresses: yijia.chen@cs.sjtu.edu.cn (Y. Chen), joerg.flum@math.uni-freiburg.de (J. Flum), moritz.mueller@univie.ac.at (M. Müller).

0168-0072/$ – see front matter
http://dx.doi.org/10.1016/j.apal.2013.06.009

whether $T \cup \{Con_T\}$ can be characterized in this context as a minimal extension of T. We could prove the following result (all terms will be defined in the paper).

Theorem 1. *Let P_0 be a decidable problem which is not decidable in polynomial time. Then there is a finite true arithmetical theory T_0 and a computable function F assigning to every computably enumerable theory T with $T \supseteq T_0$ an algorithm $F(T)$ such that* (a) *and* (b) *hold.*

(a) *T_0 proves that $F(T)$ is as fast as any algorithm T-provably deciding P_0.*
(b) *For every theory T^* with $T^* \supseteq T$ the following are equivalent:*
  (i) *T^* proves Con_T.*
  (ii) *The algorithm $F(T)$ T^*-provably decides P_0.*
  (iii) *There is an algorithm such that T^* proves that it decides P_0 and that it is as fast as any algorithm T-provably deciding P_0.*

Hence, by merely knowing the extension T of T_0 we are able to compute the algorithm $F(T)$, which is, provably in T_0, as fast as any algorithm T-provably deciding P_0; however, in order to prove that $F(T)$ decides P_0 we need the full strength of $T \cup \{Con_T\}$. In this sense, $T \cup \{Con_T\}$ is a minimal extension of T.

It is known [7] that there are problems P_0 such that one can effectively assign to every algorithm $\mathbb{A}$ deciding P_0 a further algorithm $\mathbb{B}$ deciding P_0 such that $\mathbb{A}$ is not as fast as $\mathbb{B}$. Based on this fact, from our considerations yielding a proof of Theorem 1 we obtain a version of Gödel's Second Incompleteness Theorem.

The content of the different sections is the following. In Section 3, by a standard diagonalization technique we derive a result showing for every computably enumerable set D of algorithms the existence of an algorithm that on every input behaves as some algorithm in D and that is as fast as every algorithm in D (see Lemma 2). In Theorem 7 of Section 4 we characterize problems with an optimal algorithm in terms of arithmetical theories. Section 5 contains a proof of Theorem 1. Finally, we derive the Second Incompleteness Theorem in Section 6.

Many papers in computational complexity, older and recent ones, address the question whether hard problems have *optimal* or *almost optimal* algorithms. Although Levin [5] observed that there exists an optimal algorithm that finds a satisfying assignment for every satisfiable propositional formula, it is not known whether the class of satisfiable propositional formulas or the class of tautologies have an almost optimal algorithm.

Krajíček and Pudlák [4] showed for the latter class that an almost optimal algorithm exists if and only if "there exists a finitely axiomatized fragment T of the true arithmetic such that, for every finitely axiomatized consistent theory S, there exists a deterministic Turing machine $\mathbb{M}$ and a polynomial p such that for any given n, in time $\leqslant p(n)$ the machine $\mathbb{M}$ constructs a proof in T of $Con_S(\underline{n})$." Here $Con_S(\underline{n})$ claims that no contradiction can be derived from S by proofs of lengths at most n.

Hartmanis [2] and Hutter [3] considered 'provable' algorithms, where 'provable' refers to a computably enumerable, more or less specified true theory T. Hartmanis compares the class of problems decidable within a given time bound with the class of problems T-provably decidable within this time bound and he studies time hierarchy theorems in this context. Hutter constructs an algorithm "which is the fastest and the shortest" deciding a given problem. As Hutter says, van Emde Boas pointed out to him that it is not provable that his algorithm decides the given problem and that his proof is a "meta-proof which cannot be formalized within the considered proof system" and he adds that "a formal proof of its correctness would prove the consistency of the proof system, which is impossible by Gödel's Second Incompleteness Theorem."

Unlike these papers we do not assume in Theorem 1 that T is a true theory.