

Security of Computational Cryptography and Computational Lower Bounds

Tatsuaki Okamoto

NTT Laboratories, Nippon Telegraph and Telephone Corporation
1-1 Hikarino-oka, Yokosuka-shi, Kanagawa, 239-0847 Japan

We introduce new notions of asymptotic proofs, PT(polynomial-time)-extensions, PTM(polynomial-time Turing machine)- ω -consistency, etc. on formal theories of arithmetic including PA (Peano Arithmetic). An asymptotic proof is a set of infinitely many formal proofs, which is introduced to define and characterize a property, PTM- ω -consistency, of a formal theory. Informally speaking, PTM- ω -consistency is a *polynomial-time bounded* version (in asymptotic proofs) of ω -consistency, and characterized in two manners: (1) (in the light of the *extension of PTM to TM*) the resource *unbounded* version of PTM- ω -consistency is equivalent to ω -consistency, and (2) (in the light of *asymptotic proofs by PTM*) a PTM- ω -inconsistent theory includes an axiom that only a super-polynomial-time Turing machine can prove asymptotically over PA, under some assumptions.

We show that $P \neq NP$ (more generally, any super-polynomial-time lower bound in PSPACE) is unprovable in a PTM- ω -consistent theory T , where T is a consistent PT-extension of PA (although we do not show that $P \neq NP$ is unprovable in PA, since PA has not been proven to be PTM- ω -consistent). This result implies that to prove $P \neq NP$ by any technique requires a PTM- ω -inconsistent theory, which should include an axiom that only a super-polynomial-time machine can prove asymptotically over PA (or implies a super-polynomial-time computational upper bound) under some assumptions.

This result is a kind of generalization of the result of “Natural Proofs” by Razborov and Rudich [6], who showed that to prove “ $P \neq NP$ ” by a class of techniques called “Natural Proofs” implies a super-polynomial-time (e.g., sub-exponential-time) algorithm that can break a typical cryptographic primitive, a pseudo-random generator. Our result also implies that any relativizable proof of $P \neq NP$ requires the *resource unbounded version* of PTM- ω -inconsistent theory, ω -inconsistent theory, which suggests another negative result by Baker, Gill and Solovay [1] that no relativizable proof can prove “ $P \neq NP$ ” in PA, which is a ω -consistent theory. Therefore, our result gives a unified view to the existing two major negative results on proving $P \neq NP$, Natural Proofs and relativizable proofs, through the two manners of characterization of PTM- ω -consistency.

We also show that the PTM- ω -consistency of T cannot be proven in any PTM- ω -consistent theory S , where S is a consistent PT-extension of T . That is, to prove the independence of P vs NP from T by proving the PTM- ω -consistency of T requires a PTM- ω -inconsistent theory, or implies a super-polynomial-time computational upper bound under some assumptions. This seems to be related to the results of Ben-David and Halevi [2] and Kurz, O’Donnell and Royer [5], who showed that to prove the independence of P vs NP from PA using any currently known mathematical paradigm implies an extremely-close-to-polynomial-time (but still super-polynomial-time) algorithm that can solve NP-complete problems.

Based on this result, we show that *the security of any computational cryptographic scheme is unprovable* in the setting where adversaries and provers are modeled as polynomial-time Turing machines and only a PTM- ω -consistent theory is allowed to prove the security.

The following table (Table 1) is an overview of our results in the light of the relationship between our theory and classical theory in logic.

Table 1. The Relationship between Our Theory and Classical Theory in Logic

	Our Theory	Classical Theory
Computational Resources for a Prover	Polynomial-time bounded	Unbounded
Feasibility of a Proof System	PTM- ω -consistency	Consistency ω -consistency
Unprovable Statement in a Feasible Proof System T	$P \neq NP$ Super-poly-time lower bounds (by our main theorems)	Gödel sentence of T Consistency of T (by Gödel's 1st and 2nd incompleteness thms.)
Corresponding Theory of Computability (Feasible Computation)	Computational complexity (Polynomial-time computable: by Cook et.al.)	Recursion Theory (Recursive: by Church-Turing)

As shown in this table, our theory is a polynomial-time bounded version of the classical theory in logic. Previously, as for computability, the resource bounded version of recursion theory has been studied as computational complexity. As for provability, however, no theory has been developed as a resource bounded (or polynomial-time bounded) version of the classical theory in logic. Therefore, the relationship between our theory and the classical theory in logic is similar to that between polynomial-time computation and recursive computation, since the former is feasible computation in the computational complexity theory and the latter is feasible computation of the recursion theory.

Our work is not just a polynomial-time bounded version of the classical theory in logic, but the most important point of our work is that the statement, $P \neq NP$, can be clearly characterized in our framework as shown in the above table. In the classical theory some specific statements cannot be proven in a consistent or ω -consistent proof system (feasible proof system in the classical sense). It is natural that there similarly exist such unprovable statements in a feasible proof system in our theory. But, it could be an artificial and meaningless statement like a Gödel sentence. It is very important that one of the most important conjecture in mathematics, $P \neq NP$ (and a super-polynomial-time lower bound in PSPACE), is such an unprovable statement in our theory.

References

1. T.P. Baker, J. Gill and R. Solovay, Relativizations of the $P=?NP$ Questions, SIAM J.Comput., Vol.4, No.4, pp.431-442, 1975.
2. S. Ben-David and S. Halevi, On the Independence of P versus NP , Technion, TR 714, 1992.
3. J. Hartmanis and J. Hopcroft, Independence Results in Computer Science, SIGACT News, 8, 4, pp.13-24, 1976.
4. J. Hartmanis, Feasible Computations and Provable Complexity Problems, SIAM, 1978.
5. S. Kurz, M.J. O'Donnell and S. Royer, How to Prove Representation-Independent Independence Results, Information Processing Letters, 24, pp.5-10, 1987.
6. A.A. Razborov and S. Rudich, Natural Proofs, JCSS, Vol.55, No.1, pp.24-35, 1997.