Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 189 (2007) 1223–1240
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

# On the metamathematics of the $P$ vs. $NP$ question ☆

N.C.A. da Costa, F.A. Doria *, E. Bir

*Institute for Advanced Studies, University of São Paulo, Av. Prof. Luciano Gualberto, trav. J, 374, 05655-010 São Paulo, SP, Brazil*
*Academia Brasileira de Filosofia, R. do Riachuelo 303, 20230-011 Rio de Janeiro, RJ, Brazil*

**Abstract**

We review the work (from 1976 to 1996) by several researchers on the metamathematics of $P$ vs. $NP$. That work points towards the possibility that, given some strong consistent axiomatic system $S$ with a recursively enumerable set of theorems which includes arithmetic, for the $\Sigma_2$ sentence $[P = NP]$ that formalizes the $P = NP$ hypothesis, $S + [P = NP]$ is consistent. We consider the work of several authors like for instance [J. Hartmanis, J. Hopcroft, Independence results in computer science, SIGACT News 13 (1976); M. O'Donnell, A programming language theorem which is independent of Peano Arithmetic, in: Proceedings of the 11th Annual ACM Symposium on the Theory of Computation, 1979, pp. 176–188; R.A. DeMillo, R.J. Lipton, The consistency of $P = NP$ and related problems with fragments of number theory, in: Proceedings of the 12th Annual ACM Symposium on the Theory of Computing, 1980, pp. 45–57; D. Joseph, P. Young, Fast programs for initial segments and polynomial time computation in weak models of arithmetic, STOC Milwaukee 1981, 1981, pp. 55–61; W. Kowalczyk, A sufficient condition for the consistency of $P = NP$ with Peano Arithmetic, Fund. Inform. 5 (1982) 233–245]. We then relate all those results to the [N.C.A. da Costa, F.A. Doria, Consequences of an exotic formulation for $P = NP$, Appl. Math. Comput. 145 (2003) 655–665, also "Addendum" 172 (2006) 1364–1367] conditional consistency result for ZFC + $[P = NP]$ and elaborate on it.


*Keywords:* $P = NP$; Consistency; Independence; Peano arithmetic; Zermelo-Fraenkel set theory

## 1. Prologue

We believe that the existing evidence in the literature indicates that the following result will eventually be proved:

> For a reasonably strong consistent axiomatic system $S$ that includes arithmetic and has a recursively enumerable set of theorems

☆ N.C.A. da Costa acknowledges a grant by CNPq, Philosophy Section. N.C.A. da Costa and F.A. Doria are members of the Brazilian Academy of Philosophy. E. Bir collaborates with the Research Group on Foundations at the ABF.
* Corresponding author. Address: Academia Brasileira de Filosofia, R. do Riachuelo 303, 20230-011 Rio de Janeiro, RJ, Brazil.
*E-mail addresses:* ncacosta@usp.br (N.C.A. da Costa), fadoria@filosofia.org.br, fadoria@gmail.com (F.A. Doria), eric.bir@gmail.com (E. Bir).

0096-3003/$ - see front matter 
doi:10.1016/j.amc.2006.12.012

$S + [P = NP]$

is also a consistent theory.

In this paper we discuss and ponder the chief results in that direction.

### 1.1. The metamathematical approach

The goal of the metamathematical approach to some particular question is threefold – given a consistent axiomatic theory $S$ which is relevant to our problem, and given some formal sentence $X$, we can ask three not mutually exclusive questions:

- Does $S \vdash X$ or $S \nvdash X$?
- Is $S + X$ a consistent theory? Or, alternatively, is $S + \neg X$ consistent?
- Do we have independence, that is, do we have that both $S + X$ and $S + \neg X$ are consistent?

We review in the present paper several results which go back to the 1970s until the mid 1990s. They originate in metamathematical approaches to the $P$ vs. $NP$ question. When collectively seen those results are surprisingly homogeneous, and point in the same direction, namely:

- For a strong theory $S$ like Peano Arithmetic (PA) or perhaps even Zermelo–Fraenkel set theory plus the Axiom of Choice (ZFC), they suggest that $S + P = NP$ will eventually be proved a consistent theory. Equivalently, $S \nvdash P < NP$.
- If both $P < NP$ and $P = NP$ are independent of $S$, and if $S$ has a model $\mathbf{N}$ with standard arithmetic, then $\mathbf{N} \models P < NP$.

Remarkably, on the other side, no specific information is given about the consistency of $S + P < NP$; we have been so far left in the dark here.

The present paper discusses and comments on those results. We notice a remark by Kreisel on that respect, according to Takeuti et al. [39]:

> [Takeuti says:] I am now very much interested in proving $P < NP$.
> [Kreisel] says that he is interested in the subject only if $P = NP$.

In the papers that we quote here, it seems that we may get close to a next-best option: the consistency of some reasonable formalization of $P = NP$ with a strong axiomatic theory $S$.

### 1.2. A brief review of the P vs. NP question

We intend to make this paper as self-contained as possible. Therefore we now review the basic ideas on the matter.

#### 1.2.1. What is the P vs. NP question?

The $P$ vs. $NP$ question arises in very concrete situations, such as the traveling salesman problem, or in allocation problems [29]. However in order to deal with it we consider a more abstract example of the problem, the satisfiability question for Boolean expressions:

> Given a Boolean expression, find a set (if any) of truth-values for its propositional variables so that the whole expression is made true.

It is enough to consider Boolean expressions in conjunctive normal form (cnf), that is, conjunctions of disjunctions of propositional variables or of their negations [29].

The set of all Boolean expressions in cnf can be coded by the natural numbers $\omega$ in a 1–1 way; the set of *satisfiable* Boolean expressions in cnf, $S\text{AT} \subset \omega$ is a primitive recursive subset of $\omega$, and so it can be given a

primitive recursive coding which is 1–1 and onto $\omega$; it is also a strict subset, as there are unsatisfiable Boolean expressions in cnf, namely the totally false expressions, such as $p \wedge \neg p$.

An adequate statement for the $P = NP$ hypothesis is:

> Consider a Turing machine that inputs and outputs finite binary strings, and whose operation time, that is, the number of cycles over any given input $x$, is bounded by $|x|^a + a$, where $a$ is a positive integer and $|x|$ is the length of $x$ (the number of bits of $x$ as a binary word).
> Then there is one such machine that settles the whole of SAT.

*1.2.2. The formal sentences $[P = NP]$ and $[P < NP]$*

$P < NP$ is the negation of $P = NP$.

**Remark 1.1.** Some notation: we write $P < NP$ and $P = NP$ for intuitive versions of the corresponding hypotheses. $[P < NP]$ and $[P = NP]$ are the corresponding $\Pi_2$ and $\Sigma_2$ sentences that formalize them as in Definitions 1.2 and 1.3. We will later introduce the so-called exotic formalizations $[P = NP]^{\mathsf{F}}$ and $[P < NP]^{\mathsf{F}}$; see Section 8.

**Definition 1.2**

$$[P = NP] \leftrightarrow_{\mathrm{Def}} \exists m,\ a \in \omega\ \forall x \in \omega[(t_m(x) \leqslant |x|^a + a) \wedge R(x, m)].$$

$R(x, m)$ is a polynomial predicate, that is, one which can be checked by a polynomial Turing machine. It means, roughly, "machine $\{m\}$ outputs a satisfiable line of truth values for $x$," and can be easily constructed with the help of a "checking Turing machine" that sees whether the output $\{m\}(x)$ does in fact satisfy $x$. However we can make things a bit more general at this point and only ask that $R(x, m)$ be a poly (time-polynomial, computable in polynomial time over the length of the input) predicate. Then:

**Definition 1.3.** $[P < NP] \leftrightarrow_{\mathrm{Def}} \neg[P = NP]$.

To sum it up and to announce what lies behind the results we review here: $[P = NP]$ is a $\Sigma_2$ arithmetic sentence, and therefore $[P < NP]$ is $\Pi_2$. We have several straightforward well-known results about the provability of arithmetic $\Pi_2$ sentences in formal theories like Peano Arithmetic (PA) or Zermelo–Fraenkel set theory (with the Axiom of Choice, ZFC) which we are going to apply to the $P$ vs. $NP$ question.

*1.2.3. A research program*

Suppose that we have an arithmetic $\Pi_2$ sentence in ZFC, that is, Zermelo–Fraenkel Set theory (with the Axiom of Choice, to fully strengthen it with the usual mathematical tools)

$$\forall x \in \omega\ \exists y \in \omega P(x, y).$$

Here $\omega$ is the set of natural numbers, and $P(x, y)$ is a primitive recursive predicate. Define the function

$$\{f_P\}(x) = \min_y P(x, y).$$

Immediately

$$\mathrm{ZFC} \vdash \forall x \in \omega\ \exists y \in \omega P(x, y),$$

if and only if

$$\mathrm{ZFC} \vdash \forall x \in \omega\ \exists z \in \omega T(f_P, x, z),$$

where $f_P$ is the Gödel number of $\{f_P\}$ and $T$ is Kleene's predicate [23]. That is, ZFC proves $\forall x \in \omega\ \exists y \in \omega P(x, y)$, if and only if it also proves that $\{f_P\}$ is total.

**Remark 1.4.** From here on we will write just f and f instead of $f_P$ and $\{f_P\}$ for simplicity; $\{f_P\}$ will be called the *counterexample function* to the $\Sigma_2$ sentence $\exists x\ \forall y \neg P(x, y)$.

Notice that, if ZFC cannot prove $\{f\}$ to be total, then it cannot prove sentence $\forall x \in \omega\ \exists y \in \omega P(x, y)$. Moreover, if we suppose that ZFC has a model with a standard arithmetic part, and if $\forall x \in \omega\ \exists y \in \omega P(x, y)$ is true of that model, then by soundness $\forall x \in \omega\ \exists y \in \omega P(x, y)$ is independent of ZFC, since we suppose that ZFC cannot prove false arithmetic statements, that is, it does not prove $\neg\forall x \in \omega\ \exists y \in \omega P(x, y)$.

These ideas are simple and widely known, and go back to Kleene [22,23]. Kreisel [25] (see also [26]) related them (for arithmetic and some extensions) in a highly nontrivial way to Gentzen's consistency proof for Peano Arithmetic and to a hierarchy of strictly increasing PA – provably total recursive functions. Then Paris and Harrington applied those ideas – to a $\Pi_2$ sentence that cannot be proved in Peano Arithmetic (PA) and yet it is true of the standard integers – to give their famous example of a combinatorial sentence that cannot be derived in arithmetic [13,27,33]. Since it is unprovable in PA, and yet true of the standard model, then it is independent of PA.

#### *1.2.4. Fast-growing total recursive functions and provability of $\Pi_2$ arithmetic sentences in formal theories*

Usually the simplest way to show that a theory like ZFC cannot prove $\{f\}$ to be total is to show that $\{f\}$ dominates all ZFC-provably total recursive functions in the standard model for arithmetic (we suppose that ZFC has a model with standard arithmetic part). If ZFC cannot prove $\{f\}$ to be total, then it cannot prove the corresponding $\Pi_2$ sentence. That is to say, if $\{f\}$ associated to $[P < NP]$ grows too fast, then ZFC cannot prove $[P < NP]$, and therefore ZFC + $[P = NP]$ is consistent.

If we moreover determine that such an $\{f\}$ is total in the standard model for arithmetic, then $[P < NP]$ is true – and independent of ZFC. Conversely, if we manage to show that some ZFC-provably total recursive function dominates $\{f\}$, then ZFC proves $[P < NP]$.

### *1.3. A brief overview of the results presented here*

We believe that the results discussed here point towards the possibility that the formal sentences $[P = NP]$ and $[P < NP]$ will be proven independent of strong axiomatic systems, such as PA, Peano Arithmetic, or ZFC, Zermelo–Fraenkel Set Theory plus the Axiom of Choice. There are a few common traits worth emphasizing:

- These results purport to prove consistency metatheorems for $[P = NP]$ with respect to various formal theories. In general, when one considers reasonably strong theories like PA or beyond, some extra, not always intuitive, conditions are imposed in order for consistency to hold.
- The main tool used is almost always related to the fast-growing function approach, as sketched above.
- Already known results determine in passing that $[P = NP]$ holds of some nontrivial nonstandard model for arithmetic (or for one of its fragments), while if independence holds, $[P < NP]$ is true of the standard model for arithmetic.

Our review is a selective one, and covers a period from 1976 to 1996, plus the recent (2003) work by da Costa and Doria [5]. We then confront and compare those previous results to the two first authors' recent work [5].

**Remark 1.5.** We have tried to stress the underlying ideas at the expense of rigor. So, our approach in the present paper is an informal, intuitive one.

### *1.4. More notation and preliminary data*

**Remark 1.6.** We will note algorithms, Turing machines, etc as both $\{f\}$ where $f \in \omega$ belongs to a Gödel numbering system for Turing machines which is onto $\omega$, or as sans-serif letters f, g, A, B,...We have the correspondence $\mathsf{A} = \{e_{\mathsf{A}}\}$.

We will refer below to the BGS [1] set of time-polynomial Turing machines. A BGS machine is a pair $\langle\{e\}, |x|^a + a\rangle$ that denotes the coupling of Turing machine $\{e\}$ of Gödel number $e$ to a clock (again a Turing

machine) that strictly bounds the operation time of $\{e\}$ by $|x|^a + a$, $a$ a natural number. (If and when the clock interrupts the operation of $\{e\}$, we agree that the machine outputs some fixed word and shuts down.) It is easily seen that all BGS machines are polynomial, and that given some time-polynomial machine, there will be some element in the BGS set that simulates it.

We code elements in the BGS set by the couples $\langle e, a\rangle$, which can be made onto $\omega$ by the Cantor pairing function [34].

We will also use the exotic set of BGS machines, noted $\mathrm{BGS}^{\mathsf{F}}$; that set contains all pairs $\langle\{e\}, |x|^{\mathsf{F}(a)} + \mathsf{F}(a)\rangle$, where $\mathsf{F}$ is some fast-growing function.

Recall that given a $\Pi_2$ sentence $\forall x \in \omega\ \exists y \in \omega P(x, y)$, the associated *counterexample function* is the function of Gödel number $\{f_P\}$.

### 1.5. The metamathematical approach in a nutshell

To sum it up: we review here previous work on $P$ vs. $NP$ which focuses on the metamathematical approach as sketched above. As we have already noticed, their common traits of are:

- They usually consider some version of the function $\{f\}$ associated to an arithmetic $\Pi_2$ sentence that formalizes $[P < NP]$.
- They prove consistency results for $[P = NP]$, sometimes given some strong, nontrivial and nonintuitive conditions.

We will summarize the main results, and add some extra comments when we feel it is adequate in order to place everything within a reasonable conceptual framework.

## 2. Hartmanis and Hopcroft (1976)

The first results we give here show that many simple, ordinary questions in computer science turn out to be undecidable. The Hartmanis and Hopcroft 1976 paper [17] presents three undecidability results in computer science, two of them directly related to $P$ vs. $NP$. Actually the results exhibited by Hartmanis and Hopcroft seem to imply that they believe the $P$ vs. $NP$ question itself might turn out to be undecidable within some strong axiomatic framework.

Hartmanis and Hopcroft start from a formal theory which we note $S$ that:

- Includes set theory (more precisely, they ask that $S$ be of "sufficient power to prove the basic theorems of set theory").We add that $S$ or some adequate conservative extension of it must allow for predicate symbols $P, Q, \ldots$
- $S$ has a recursively enumerable set of theorems.
- Its theorems are "intuitively true". Since this would be a too strong and vague requirement for the whole of set theory with the axiom of choice (for instance, is the Banach–Tarski theorem intuitively true?), then we take this third requirement to be what the first two authors of the present paper have called the *arithmetically soundness* condition, that is, $S$ must have a model with standard arithmetic.

We will use these conditions later, so let us agree that:

**Definition 2.1.** A *Hartmanis–Hopcroft* (*HH*) *theory* is any axiomatic theory $S$ with a language that allows for predicate symbols, has a recursively enumerable set of theorems, includes (an interpretation of) Peano Arithmetic, and has a model with a standard arithmetic portion for its arithmetic segment.

**Remark 2.2.** The importance of HH-theories is: most results given here extend to any such theory, or, in other words, are sort of insensitive to the axiomatic framework where they are constructed, but for the fact that one requires arithmetic to formulate them.

Now let us summarize Hartmanis and Hopcroft 1976:

- The first undecidability result in [17] has to do with the BGS relativization result [1]. The BGS result says that there are recursive oracles $A, B, A \neq B$, so that one has (for the relativized versions) $P^A = NP^A$ and $P^B < NP^B$. Hartmanis and Hopcroft show that there is an oracle $C$ so that the assertion $P^C = NP^C$ is undecidable with respect to the axioms of $S$. Their proof is by a diagonal argument; we have used here an alternative, quite general argument.
- Then they show that there is an algorithm A (a Turing machine) of which it is true that for input $x$ it runs in time $x^2$, but so that the formal version of the sentence "$\mathsf{A}(x)$ runs in time $t_{\mathsf{A}} < 2^x$" is undecidable in $S$.
- The final result in the paper has to do with an undecidability result within $S$ for a class of languages. Again the same trick we used above can be applied to get it.

We present here a general proof for these three results which is based on a version of Rice's theorem to fragments of set theory with the $\iota$ symbol (which we suppose to be available) and which stems from an idea first used in da Costa and Doria 1991 [4,7]. More precisely:

**Remark 2.3.** For consistent $S$, let $\mathrm{Consis}\,S$ be the usual formal sentence that asserts the consistency of $S$; $S \nvdash \mathrm{Consis}\,S$ and $S \nvdash \neg\mathrm{Consis}\,S$. Let $\xi$, $\zeta$ be terms in the language of $S$, so that for some predicate $P$ in the language of $S$, $S \vdash P(\xi)$ while $S \vdash \neg P(\zeta)$. Then

$$\lambda = \iota_x\{[\mathrm{Consis}\,S \wedge x = \xi] \vee [\neg\mathrm{Consis}\,S \wedge x = \zeta]\}.$$

$S \nvdash \lambda = \xi$ and $S \nvdash \lambda = \zeta$, but if $\mathbf{N} \models S$ and $\mathbf{N}$ has a standard arithmetic part, then $\mathbf{N} \models \lambda = \xi$. Moreover, $S \nvdash P(\lambda)$ and $S \nvdash \neg P(\lambda)$, while $\mathbf{N} \models P(\lambda)$.

- For the first result, put oracles $A, B$ as $\xi = A$ and $\zeta = B$. Then $C$

  $$C = \iota_x\{[\mathrm{Consis}\,S \wedge x = A] \vee [\neg\mathrm{Consis}\,S \wedge x = B]\}$$

  is proved to be a recursive oracle in $S$, but $S \nvdash C = A$ and $S \nvdash C = B$. So, $S \nvdash P^C = NP^C$ and $S \nvdash P^C < NP^C$.
- For the second result, if P is a polynomial Turing machine, and E is an exponential Turing machine, then

  $$\mathsf{M} = \iota_x\{[\mathrm{Consis}\,S \wedge x = \mathsf{P}] \vee [\neg\mathrm{Consis}\,S \wedge x = \mathsf{E}]\}$$

  is such that $S$ proves M to be a total Turing machine which has an exponential time bound which cannot be improved in $S$, but such that it is true of $\mathbf{N}$ that it is time-polynomial.

The third result (which we did not make explicit) can be given a similar proof. For a related construction see [7]. We may also use the term

$$\lambda = \iota_x[x = \xi \wedge \beta = 0] \vee [x = \zeta \wedge \beta = 1].$$

$\beta$ [7] is an algebraic expression which can be explicitly constructed and such that $S \nvdash \beta = 0$ and $S \nvdash \beta = 1$, while $\beta = 0$ holds of the standard model for arithmetic; see the reference for details. For related incompleteness phenomena in an axiomatization of Turing machine theory see [6], where it is shown that we can explicitly exhibit an infinite set of poly machines so that individually each machine in the set is polynomial, but such that we cannot prove (or disprove) that the whole set only includes poly machines.

**Remark 2.4.** We wish to stress that these results, which show that the running time of an algorithm can be undecidable in very strong HH-theories, should be a caveat for those that do not wish to recognize the import of metamathematical phenomena in computer science [6,9].

## 3. O'Donnell (1979), DeMillo and Lipton (1980), Sazonov (1980), Joseph and Young (1981)

O'Donnell's theoretical framework is Peano Arithmetic (PA); see [31]. His main result is a conditional theorem that strictly refines the following argument:

- PA proves $[P < NP]$ if and only if PA proves a formalized version of "there is a strict exponential lower-bound to the algorithms that correctly compute satisfiable values for all problems in SAT".
- Therefore PA does not prove $[P < NP]$ if and only if PA does not prove that there is one such lower bound.
- Follows two alternatives. Either PA proves $[P = NP]$, or PA neither proves $[P < NP]$ nor $[P = NP]$, that is, if $[P < NP]$ and $[P = NP]$ are independent of PA. (The independence conjecture is explored by O'Donnell.)

O'Donnell shows that $[P < NP]$ can be formalized as a $\Pi_2$ sentence (his formulation is slightly different from the one given at the beginning of this paper). As we said, O'Donnell presents a refinement to the independence alternative. Consider a recursive enumeration of all poly machines – like the enumeration in the Baker–Gill–Solovay [1] paper.

**Remark 3.1.** The *recursive counterexample function* $\mathsf{f}$ to $[P = NP]$ is the function that is given as follows: if $n$ is the Gödel number of a poly machine in the BGS enumeration, then:

$\mathsf{f}(n) =$ first $x \in$ SAT so that $\{n\}(x)$ is a nonsatisfying line of truth-values for $x$.

It is the first instance $x$ where the machine fails to compute a correct answer, $x \in$ SAT.

Then O'Donnell's result amounts to:

**Proposition 3.2.** *If both* $[P < NP]$ *and* $[P = NP]$ *are independent of* (*consistent*) PA, *then*:

1. $[P < NP]$ *holds true of the standard model for arithmetic.*
2. *There is an algorithm for the whole of* SAT *of which it is true that it runs in time* $\mathrm{O}(|x|)^{\mathsf{f}^{-1}(n)}$.
3. *It is true that* $\mathsf{f}$ *is not dominated by any PA-provably total function.*

We will prove the first and second O'Donnell result when we discuss Ben-David and Halevi [2], who prove them in a very intuitive way. (The third statement above is immediate.)

### 3.1. DeMillo and Lipton (1980)

The main result in the much-quoted, path-breaking DeMillo and Lipton 1980 [11,12] paper goes as follows: they consider several fragments of arithmetic, among which they select a theory $A$ which basically contains all arithmetical predicates computable in polynomial time (plus a few technical conditions). They then show that:

**Proposition 3.3.** *If* $A$ *is consistent, then so is* $A + [P = NP]$.

Very briefly, they construct a nonstandard model for arithmetic where a property equivalent to $[P = NP]$ holds. DeMillo and Lipton explore ideas akin to those in the O'Donnell and Kowalczyk [24] papers in Section 6 of [12], where they give a formal (even if somewhat opaque) characterization for $[P = NP]$, and then in Section 8, where after a long discussion of techniques to prove independence they refer to the main argument in the O'Donnell paper, the rate of growth of a Skolem function (which we have called the counterexample function in the present situation). The crucial fact is that the consistency result obtained only holds of some nonstandard model. There is a reason for that: as we will see below in the summary of Ben-David and Halevi 1991 [2], if $[P = NP]$ is true of the standard model for arithmetic, then it can be proved at least in a theory very similar to Peano Arithmetic. However if it is independent, then $[P = NP]$ will only hold of nonstandard arithmetic models. See Section 6.

We prove what may be seen as a strengthening of the DeMillo–Lipton result after we review Costa Doria [5].

### 3.2. Sazonov (1980)

The paper by Sazonov [35] contains flaws later corrected [36]. The corrected version has a result about as strong as DeMillo–Lipton, with a totally different technique. One of the interesting points is a $\Pi_1$ formulation for $P = NP$ as the one in Ben-David and Halevi 1991 [2].

### 3.3. Joseph and Young (1981)

We can directly quote from [20]:

> In this paper we study two alternative approaches for investigating whether *NP*-complete sets have fast algorithms. One is to ask whether there are long initial segments on which these sets are easily decidable by relatively short programs. The other approach is to ask whether there are weak fragments of arithmetic for which it is consistent to believe that $P = NP$. We show, perhaps surprisingly, that the two questions are equivalent: it is consistent to believe that $P = NP$ [holds] in certain models of weak arithmetic theories if and only if it is true (in the standard model of computation) that there are infinitely many initial segments on which satisfiability is polynomially decidable by programs that are much shorter than the length of the initial segment.

Again we can offer a paraphrasis for that result: if the counterexample function to $P = NP$ grows faster than any $S$-provably total recursive function then $S + P = NP$ is consistent. Here $S$ is a HH-theory (see Definition 2.1); so our paraphrasis is more general than the result stated by Joseph and Young. See below the reviews of Kowalczyk (1982) and da Costa and Doria (2003).

## 4. Kowalczyk (1982)

Kowalczyk's short 1982 paper [24] gives a sufficient condition for the consistency of Peano Arithmetic with $[P = NP]$ which mirrors Joseph and Young (it is interesting to notice that Kowalczyk is aware of the first paper by Joseph and Young [19], but not of the second [20], where their result is stated).

The Joseph–Young–Kowalczyk result anticipates the first two authors' main result in [5].

Kowalczyk uses indicator functions [21] to construct a function that grows faster than any PA-provably total recursive function – there is a simpler way to obtain the same function ([21, p. 52], [5, first paper, Definition 3.1]). Then Kowalczyk obtains his main result, which is thus paraphased:

> If "relatively small" Turing machines give correct truth-value lines for "relatively large" segments of SAT, then theory PA + $[P = NP]$ + [All true $\Pi_1$ sentences in PA] is consistent.

We now clarify the meaning of the expressions between quotation marks, with a reformulation of Kowalczyk's result. Let $\mathsf{f}$ be the counterexample function (see Remark 3.1) defined on poly machines and let $\mathsf{g}$ be primitive recursive. Then:

**Proposition 4.1.** *If for each natural number* $n$, $\mathsf{g}(n)$ *are the Gödel numbers of poly machines so that* $\mathsf{f}(\mathsf{g}(n))$ *overtakes any* PA*-provably total recursive function* $\mathsf{h}$, *that is, for infinitely many* $n$, $\mathsf{f}(\mathsf{g}(n)) > \mathsf{h}(n)$, *any such* $\mathsf{h}$, *then* PA + $[P = NP]$ *is consistent.*

**Remark 4.2.** This also holds if we add to PA + $[P = NP]$ all true arithmetical $\Pi_1$ sentences.

**Proof of the proposition.** If $\mathsf{f}$ is total in the standard model for arithmetic, then the fact that no PA-provably total recursive function dominates it shows that $\mathsf{f}$ cannot be proved to be total in PA. Therefore one cannot prove the equivalent sentence, namely $[P < NP]$, as we cannot prove that the counterexample function is total in PA, given the hypothesis. Follows the desired consistency.

If it holds true of the standard model that $\mathsf{f}$ is not total, as we suppose that PA is sound, then PA $\vdash [P = NP]$; or we have independence. Again follows our result. □

**Remark 4.3.** We can immediately extend that result to any HH-theory, as in Definition 2.1. Notice that if the counterexample function grows faster than any PA-provably total function, then for infinitely many poly machines there will be "long stretches" of instances from SAT which are accepted by those machines.

Later when we consider Ben-David and Halevi we will prove that if independence holds, then $[P < NP]$ is true of the standard model for arithmetic. The two original conditions in the theorem proved by Kowalczyk are:

- "Relatively small" Turing machines translates as a bound on both the usual Gödel numbers for the machines and its operation time. We use here the primitive recursive bound given by the sequence $g(n)$, where these values are to be seen as BGS Gödel numbers for poly machines.
- A "relatively large" segment of SAT which is given correct truth-values by some machine $g(n)$ is here associated to a fast-growing f as the values $f(g(n))$ are supposed to grow beyond any PA-provably total recursive function.

## 5. A closer look at the O'Donnell–Joseph–Young–Kowalczyk conditions

We will now take a closer look at those conditions when they refer to the counterexample function:

### 5.1. The counterexample function as a fast growing function

The next construction amounts to an alternative, more explicit, way to obtain the da Costa–Doria result of 2003 by the direct construction of a segment of the counterexample function to the exotic $[P = NP]^{\mathsf{F}}$ that is shown to grow faster than any $S$-provably total function, where $S$ is the HH-theory that serves as our axiomatic background. The idea in the next proof goes as follows:

- Use the *s–m–n* theorem to obtain Gödel numbers for an infinite family of "quasi-trivial machines" – soon to be defined. The table for those Turing machines involves very large numbers, and the goal is to get a compact code for that value in each quasi-trivial machine so that their Gödel numbers are in a sequence $g(0), g(1), g(2), \ldots$, where g is primitive recursive. (See on what follows Remarks 1.4 and 1.6.)
- Then add the required clocks as in the BGS sequence of poly machines, and get the Gödel numbers for the pairs machine + clock. We can embed the sequence we obtain into the sequence of all Turing machines.
- Notice that the subsets of poly machines we are dealing with are (intuitive) recursive subsets of the set of all Turing machines. More precisely: if we formalize everything in some HH-theory $S$, then the formalized version of the sentence "the set of Gödel numbers for these quasi-trivial Turing machines is a recursive subset of the set of Gödel numbers for Turing machines" holds of the standard model for arithmetic in $S$, and vice versa. However $S$ may not be able to prove or disprove that assertion, that is to say, it will be formally independent of $S$ [6].
- We can thus define the counterexample functions over the desired set(s) of poly machines, and compare them to fast-growing total recursive functions over similar restrictions.

**Definition 5.1.** For $f, g : \omega \to \omega$,

$$f \text{ dominates } g \leftrightarrow_{\text{Def}} \exists y\, \forall x (x > y \to f(x) \geqslant g(x)).$$

We write $f \succ g$ for $f$ dominates $g$.

#### 5.1.1. Quasi-trivial machines

Recall that the operation time of a Turing machine is given as follows: if M stops over an input $x$, then the operation time over $x$,

$$t_{\mathsf{M}} = |x| + \text{number of cycles of the machine until it stops.}$$

**Example 5.2**

- *First trivial machine*: Note it $\mathsf{O} \cdot\cdot \mathsf{O}$ inputs $x$ and stops.

$$t_{\mathsf{O}} = |x| + \text{moves to halting state} + \text{stops.}$$

So, operation time of O has a linear bound.

- *Second trivial machine*: Call it $\mathsf{O}'$. It inputs $x$, always outputs 0 (zero) and stops. Again operation time of $\mathsf{O}'$ has a linear bound.
- *Quasi-trivial machines*: A *quasi-trivial machine* $\mathsf{Q}$ operates as follows: for $x \leqslant x_0$, $x_0$ a constant value, $\mathsf{Q} = \mathsf{R}$, $\mathsf{R}$ an arbitrary total machine. For $x > x_0$, $\mathsf{Q} = \mathsf{O}$ or $\mathsf{O}'$. This machine has also a linear bound.

**Remark 5.3.** Now let $\mathsf{H}$ be any fast-growing, superexponential total machine. Let $\mathsf{H}'$ be another such machine. From the following family of quasi-trivial Turing machines with subroutines $\mathsf{H}$ and $\mathsf{H}'$:

1. If $x \leqslant \mathsf{H}(n)$, $\mathsf{Q}^{\mathsf{H},\mathsf{H}',n}(x) = \mathsf{H}'(x)$.
2. If $x > \mathsf{H}(n)$, $\mathsf{Q}^{\mathsf{H},\mathsf{H}',n}(x) = 0$.

**Proposition 5.4.** *There is a family* $\mathsf{R}_{\mathsf{g}(n,|\mathsf{H}|)}(x) = \mathsf{Q}^{\mathsf{H},\mathsf{H}',n}(x)$, *where* $\mathsf{g}$ *is primitive recursive, and* $|\mathsf{H}|$ *denotes the Gödel number of* $\mathsf{H}$.

**Proof.** By the composition theorem and the *s–m–n* theorem. □

We first give a result for the counterexample function when defined over all Turing machines (with the extra condition that the counterexample function $= 0$ if $\mathsf{M}_m$ is not a poly machine). We have:

**Proposition 5.5.** *If* $N(n) = \mathsf{g}(n)$ *is the Gödel number of a quasi-trivial machine as in Remark 5.3, then* $f(N(n)) = k(n) + 1 = \mathsf{H}(n) + 1$.

**Proof.** Use the machines in Proposition 5.4. □

*5.1.2. Proof of non-domination*

Our goal here is to prove the following result:

**Proposition 5.6.** *For no total recursive function* $\mathsf{h}$ *does* $\mathsf{h} \succ f$.

**Proof.** Suppose that there is a total recursive function $\mathsf{h}$ such that $\mathsf{h} \succ f$. □

**Remark 5.7.** Given such a function $\mathsf{h}$, obtain another total recursive function $\mathsf{h}'$ which satisfies:

1. $\mathsf{h}'$ is strictly increasing.
2. For $n > n_0$, $\mathsf{h}'(n) > \mathsf{h}(\mathsf{g}(n))$.

**Lemma 5.8.** *Given a total recursive* $\mathsf{h}$, *there is a total recursive* $\mathsf{h}'$ *that satisfies the conditions in Remark 5.7.*

**Proof.** Given $\mathsf{h}$, obtain out of that total recursive function by the usual constructions a strictly increasing total recursive $\mathsf{h}^*$. Then if, for instance, $\mathsf{F}_\omega$ is Ackermann's function, $\mathsf{h}' = \mathsf{h}^* \circ \mathsf{F}_\omega$ will do. (The idea is that $\mathsf{F}_\omega$ dominates all primitive recursive functions, and therefore $\mathsf{h}^*$ composed with it dominates $\mathsf{g}(n)$.) □

We have that the Gödel numbers of the quasi-trivial machines $\mathsf{Q}$ are given by $\mathsf{g}(n)$. Choose adequate quasi-trivial machines, so that $f(\mathsf{g}(n)) = \mathsf{h}'(n) + 1$, from Proposition 5.5. From Remark 5.7 and Lemma 5.8 we conclude our argument. If we make explicit the computations, for $\mathsf{g}(n)$ (as the argument holds for any strictly increasing primitive recursive $\mathsf{g}$)

$$f(\mathsf{g}(n)) = \mathsf{h}'(n) + 1 = \mathsf{h}^*(\mathsf{F}_\omega(n)) + 1$$

and

$$\mathsf{h}^*(\mathsf{F}_\omega(n)) > \mathsf{h}^*(\mathsf{g}(n)).$$

For $N = \mathsf{g}(n)$,

$$f(N) > \mathsf{h}^*(N) \geqslant \mathsf{h}(N), \quad \text{all } N.$$

Therefore no such h can dominate $f$.

**Corollary 5.9.** *No total recursive function dominates f.*

*5.1.3. Exotic BGS$^{\mathsf{F}}$ machines*

Now let F be as in Remark 8.7. We consider exotic BGS$^{\mathsf{F}}$ machines, that is, poly machines coded by the pairs $\langle m, a\rangle$, which code Turing machines $\mathsf{M}_m$ with bounds $|x|^{\mathsf{F}(a)} + \mathsf{F}(a)$. Since the bounding clock is also a Turing machine, now coupled to $\mathsf{M}_m$, there is a primitive recursive map c so that

$$\langle \mathsf{M}_m, |x|^{\mathsf{F}(a)} + \mathsf{F}(a)\rangle \mapsto \mathsf{M}_{\mathsf{c}(m,a)},$$

where $\mathsf{M}_{\mathsf{c}(m,a)}$ is a poly machine within the sequence of all Turing machines. We similarly obtain a g as above, and follows:

**Proposition 5.10.** *Given the counterexample function* f *defined over the* BGS$^{\mathsf{F}}$ *– machines, for no* ZFC*-provable total recursive* h *does* $\mathsf{h} \succ \mathsf{f}$.

**Proof.** As in Proposition 5.6. □

*5.2. Do the conditions hold?*

The formal sentence $[P < NP]$ is formalized in such a way that it can be seen as defined only for the BGS set of machines. We have seen that the Kowalczyk conditions hold for the *exotic* BG$S^{\mathsf{F}}$ machines. Can we transpose them to the actual BGS set? (We deal with that below.)

**Remark 5.11.** Notice again that it is true of the standard model for arithmetic that the set of pairs $\langle \mathsf{M}_m, |x|^{\mathsf{F}(a)} + \mathsf{F}(a)\rangle$, $a \in \omega$ is a set of poly machines. However due to the fact that $S$ cannot prove that F is total (or not), $S$ cannot prove that set of Turing machines to be a set of poly machines [6].

## 6. Ben-David and Halevi (1991)

This never published preprint [2] accurately summarizes in detail some of the work we have just sketched. Their explicit intent is a negative one: to show that it is implausible that $[P = NP]$ is independent of PA, and so they do not refer to consistency results such as DeMillo and Lipton. But they stress several significant points, which we now present in a slightly reformulated way. We will argue here for ZFC, but our argument holds for any HH theory.

We first show:

**Proposition 6.1.** *If $[P = NP]$ holds true of the standard model for arithmetic, then there is a HH-theory with the same provably total recursive functions as* PA (*Peano Arithmetic*) *that proves* $[P = NP]$.

Proof will be given in a series of steps. First:

**Lemma 6.2**

$$[P = NP] \leftrightarrow \exists e,\ a\forall x\ \exists z \leqslant (|x|^a + a)[T(e,x,z) \wedge R(x, U(T(e,x,z))].$$

**Proof.** $T, U$ are Kleene's predicates. Proof is immediate, from Definition 1.2. □

**Remark 6.3.** Recall that a $\Pi_1$ sentence for $S$ is a sentence of the form:

1. $\forall x P(x)$, $P$ primitive recursive.
2. $\forall x\ \exists y \leqslant \mathsf{g}(x) P(x)$, where $P$ is primitive recursive, and g is $S$-provably total.

It is enough for our purposes to consider $\Pi_1$ sentences for PA. Clearly then

$$\forall x\ \exists z \leqslant (|x|^a + a)[T(e,x,z) \wedge R(x, U(T(e,x,z))]$$

is a $\Pi_1$ sentence for PA.

**Hypothesis 6.4.** We suppose that $\mathrm{ZFC} \vdash [P = NP]$.

We suppose that ZFC is *arithmetically sound*, that is, ZFC has a model with standard arithmetic; we note that model **N**. From the hypothesis and by arithmetic soundness:

**Corollary 6.5.** $\mathbf{N} \models [P = NP]$.

Moreover, for some integers $e, a$.

**Corollary 6.6.** $\mathbf{N} \models \forall x\ \exists z \leqslant (|x|^a + a)[T(e,x,z) \wedge R(x, U(T(e,x,z))]$.

Now let $\mathrm{PA}_1$ be the (nonrecursive) theory that consists of PA plus all true $\Pi_1$ sentences for PA. Then clearly

**Lemma 6.7.** *If there are constants $e, a$ so that*

$$\forall x\ \exists z \leqslant (|x|^a + a)[T(e,x,z) \wedge R(x, U(T(e,x,z))]$$

*holds true of the standard model for arithmetic, then*

$$\mathrm{PA}_1 \vdash [\forall x\ \exists z \leqslant (|x|^a + a)[T(e,x,z) \wedge R(x, U(T(e,x,z))]].$$

**Corollary 6.8.** *If there are constants $e, a$ so that*

$$\forall x\ \exists z \leqslant (|x|^a + a)[T(e,x,z) \wedge R(x, U(T(e,x,z))]$$

*holds true of the standard model for arithmetic, then*: $\mathrm{PA}_1 \vdash [P = NP]$.

**Proof.** For we have

$$[\forall x\ \exists z \leqslant (|x|^a + a)[T(e,x,z) \wedge R(x, U(T(e,x,z))]] \rightarrow [\exists e, a \forall x\ \exists z \leqslant (|x|^a + a)[T(e,x,z) \wedge R(x, U(T(e,x,z))]].$$

Then by detachment we get $[P = NP]$. □

**Corollary 6.9.** *Either* $\mathrm{PA} \vdash [P = NP]$ *or* (*trivially*) $\mathrm{PA} + [P = NP]$ *proves it. And moreover in this particular case*, $\mathbf{N} \models [P = NP]$.

Notice that given our hypothesis, $[P = NP]$ is equivalent to a $\Pi_1$ sentence, and from Kreisel's Lemma [2] the extended theory has the same provably total recursive functions as PA. By contraposition, from Corollary 6.8:

**Corollary 6.10.** *If* $\mathrm{PA}_1$ *does not prove* $[P = NP]$, *then it does not hold true of the standard model for arithmetic.*

Two simple equivalences show that the preceding is equivalent to:

**Corollary 6.11.** *If either* $\mathrm{PA}_1 \vdash [P < NP]$ *or* $[P < NP]$ *is independent of* $\mathrm{PA}_1$, *then* $[P < NP]$ *holds true of the standard model for arithmetic.*

Now since $\mathrm{PA}_1 \vdash [P < NP]$ implies that $[P < NP]$ is true of the standard model for arithmetic, then:

**Corollary 6.12.** *If* $[P < NP]$ *is independent of* $\mathrm{PA}_1$, *then* $[P < NP]$ *is true of the standard model for arithmetic.*

The preceding result was first obtained by O'Donnell in 1979 [31]. It is immediately extended to $S$.

### *6.1. The O'Donnell algorithm*

We can now describe the O'Donnell quasi-polynomial algorithm for SAT. Recall that f is the counterexample function to $[P = NP]$. Then:

- We consider the set of all Boolean expressions in cnf, including those that are unsatisfiable, or totally false. We give it the usual coding which is 1–1 and onto $\omega$.
- Consider the poly Turing machine $\mathsf{V}(x,s)$, where $\mathsf{V}(x,s)=1$ if and only if $s$ satisfies $x$, and $\mathsf{V}(x,s)=0$ if and only if $s$ does not satisfy $x$.
- Consider the enumeration of the BGS [1] machines, $\mathsf{P}_0,\mathsf{P}_1,\mathsf{P}_2,\ldots$
- Consider $x$, the binary code for a Boolean expression in cnf form.
- Alternatively check for $\mathsf{V}(x,0)$, $\mathsf{V}(x,1)$, ...up to – if it ever happens – some $s$ so that $\mathsf{V}(x,s)=1$; or,
- Input $x$ to $\mathsf{P}_0,\mathsf{P}_1,\mathsf{P}_2,\ldots$ up to the first $\mathsf{P}_j$ so that $\mathsf{P}_j(x)=s_j$ and $s_j$ satisfies $x$.Notice that $j=\mathsf{f}^{-1}(x)$.
- Now, if f is fast-growing, then as the operation time of $\mathsf{P}_j$ is bounded by $|x|^k+k$, we have that $k\leqslant j$, and therefore it grows as $\mathsf{O}(\mathsf{f}^{-1}(x))$. This will turn out to be a very slowly growing function.More precisely, it will have to be tested up to $j$, that is the operation time will be bound by $\mathsf{f}^{-1}(x)(|x|^{\mathsf{f}^{-1}(x)}+\mathsf{f}^{-1}(x))$.

Then either $x$ is unsatisfiable – and therefore one will have to test all possible $s$ – or, if satisfiable, operation time has the nearly polynomial time given above. This means that if independence holds, then we will have something that might be informally written as $P\approx NP$ – there are nearly polynomial algorithms.

**Remark 6.13.** We can easily extend the chief constructions above to any HH-theory $S$.

## 7. Maté (1990), Sureson (1996), plus a note on categories

The work of Maté [30] and of Sureson [38] both present characterizations for the consistency of some fragment of arithmetic $S$ – up to PA – with $P=NP$. Their theorems are of the form, "if [some metamathematical condition] $C$ holds, then $S+[P=NP]$ is consistent". However $C$ is given as a complex statement, not obviously correct or interpretable in intuitive concepts.

### 7.1. Can we approach it from the viewpoint of category theory?

A parallel approach was inaugurated as soon as 1980 by Huwig [18], using a categorical perspective. To our knowledge, this is the first attempt to touch $P$ vs. $NP$ with such tools. Its motivation is essentially raised from metamathematical considerations:

- Instead of exploring $P$ vs. $NP$ inside SET (the equivalent of ZFC in the categorical universe), the author chooses other categories in order to observe how the conjecture behaves in these new contexts.
- For this purpose, Huwig selects CIS (the category of commutative, idempotent monoids).
- Despite being weaker than ZF, CIS does allows all the PA axioms, and all the primitive recursive functions; that goes beyond the system used by DeMillo.
- After playing with the categorical machinery Huwig shows that $P_{cis}=NP_{cis}$.
- This result, it should be noted, is only a nontrivial example of general properties coming from the first-order theories employed (symmetrical monoidal closed category, equipped with NNO (Natural Number Object), finite limits, finite colimits and a generator.
- Thereafter, the author concludes: first-order theories of the type employed are too weak to solve $P$ vs. $NP$.
- Nonetheless the machinery employed opens a new toolbox of promising research directions.

## 8. Da Costa and Doria (2003)

Da Costa and Doria first considered the Paris–Harrington-like argument about the fast-growing counterexample function condition for the nonprovability of $[P<NP]$ in 1995, and set out to try prove a hypothesis equivalent to that in O'Donnell–Joseph–Young–Kowalczyk, namely that the counterexample function to $[P=NP]$ grows too fast to be proved total in, say, PA, and perhaps in even stronger theories like ZFC. Their idea was to compress Gödel numbers by expressing them with the help of fast growing functions, e.g. if we have the Gödel number $\{e_{\mathsf{F}}\}$ for a fast-growing function $\mathsf{F}$, if $\mathsf{Q}(q,x)$ is a family of Turing machines parametrized by

$q \in \omega$, and if $q$ is large, then the Gödel number of each member of the sequence $\mathsf{Q}(q,x)$ will be correspondingly large (the Gödel numbers will be primitive recursive on $q$). However if we put $\mathsf{Q}(\mathsf{F}(n),x)$, the Gödel number of this function composition will grow as $\mathsf{g}(e_{\mathsf{F}}, e_{\mathsf{Q}}, n)$, where the $e$'s are constants and g is primitive recursive. (This follows from the *s–m–n* theorem [34]; cf. Proposition 5.4.) The sequence $\mathsf{Q}(\mathsf{F}(n),x)$ does not include all the elements of the sequence $\mathsf{Q}(q,x)$, but if we are just interested in the way some function defined on the Q's grows, it is enough to consider an infinite set of selected points instead of the whole set of values.

This idea developed into what the two first authors of this paper have called "exotic" formalization for $P < NP$: a $\Pi_2$ sentence $[P < NP]^{\mathsf{F}}$ that naïvely translates as $P < NP$, is in fact equivalent to $[P < NP]$ in the standard model for arithmetic [13], but cannot have that equivalence established within HH-theories $S$, for adequate fast-growing functions $\mathsf{F}$ [5,7], for that equivalence is independent of such theories.

We summarize here our previous results [5] and give naïve, informal proofs for them; the rigorous proofs appear in the reference. Our axiomatic framework $S$ is taken to be ZFC, in order to ensure enough "elbow room" for our arguments.

In what follows we write $S$ for ZFC. However our argument holds for any HH-theory.

The exotic formalization for $P < NP$ is naïvely the same as the standard formalization, but cannot (in general) be proved equivalent to the latter within even strong systems such as ZFC. Let $t_m(x)$ be the (primitive recursive function that gives the) operation time of $\{m\}$ over an input $x$ of length $|x|$. Suppose that $\{e_{\mathsf{f}}\} = \mathsf{f}$ is total recursive and strictly increasing. The naïve version for the exotic formalization is

$$[P = NP]^{\mathsf{f}} \leftrightarrow \exists m \in \omega,\ a\forall x \in \omega[(t_m(x) \leqslant |x|^{\mathsf{f}(a)} + \mathsf{f}(a)) \wedge R(x,m)].$$

However as we will soon see, there is no reason why we should ask that f be total; on the contrary, there will be interesting situations where such a function may be partial and yet provide a reasonable exotic formalization for $P < NP$. We will have to modify it accordingly.

So, for the next definitions and results let f be in general a (possibly partial) recursive function which is strictly increasing over its domain, and let $e_{\mathsf{f}}$ be the Gödel number of an algorithm that computes f. Let $p(\langle e_{\mathsf{f}}, b, c\rangle, x_1, x_2, \ldots, x_k)$ be [10,37] an universal Diophantine polynomial with parameters $e_{\mathsf{f}}, b, c$; that polynomial has integer roots if and only if $\{e_f\}(b) = c$. We may if needed suppose that polynomial to be $\geqslant 0$. We omit the "$\in \omega$" in the quantifiers, since they all refer to natural numbers.

**Definition 8.1**

$$M_{\mathsf{f}}(x,y) \leftrightarrow_{\mathrm{Def}} \exists x_1, \ldots, x_k[p(\langle e_{\mathsf{f}}, x, y\rangle, x_1, \ldots, x_k) = 0].$$

Actually $M_{\mathsf{f}}(x,y)$ stands for $M_{e_{\mathsf{f}}}(x,y)$, or better,

$$M(e_{\mathsf{f}}, x, y)$$

as dependence is on the Gödel number $e_{\mathsf{f}}$.

**Definition 8.2.** $\neg Q(m,a,x) \leftrightarrow_{\mathrm{Def}} [(t_m(x) \leqslant |x|^a + a) \rightarrow \neg R(x,m)]$.

**Proposition 8.3.** $[P < NP] \leftrightarrow \forall m,\ a\exists x \neg Q(m,a,x)$.

**Definition 8.4.** $\neg Q_{\mathsf{f}}(m,a,x) \leftrightarrow_{\mathrm{Def}} \exists a'[M_{\mathsf{f}}(a,a') \wedge \neg Q(m,a',x)]$.

We will sometimes write $\neg Q(m,\mathsf{f}(a),x)$ for $\neg Q_{\mathsf{f}}(m,a,x)$, whenever f is total.

**Definition 8.5** (*Exotic formalization*).

$$[P < NP]^{\mathsf{f}} \leftrightarrow_{\mathrm{Def}} \forall m, a\exists x \neg Q_{\mathsf{f}}(m,a,x).$$

Notice that this is also a $\Pi_2$ arithmetic sentence.

**Definition 8.6.** $[P = NP]^{\mathsf{f}} \leftrightarrow_{\mathrm{Def}} \neg[P < NP]^{\mathsf{f}}$.

We use here a well-known recursive function that is diagonalized over all $S$-provably total recursive functions. We note it $\mathsf{F}$, or sometimes $\mathsf{F}_S$. See [5,21], pp. 51–52 on it.

**Remark 8.7.** For each $n$, $\mathsf{F}(n) = \max_{k \leqslant n}(\{e\}(k)) + 1$, that is the sup of those $\{e\}(k)$ such that:

1. $k \leqslant n$.
2. $\lceil \mathrm{Pr}_S(\lceil \forall x\ \exists z T(e,x,z) \rceil) \rceil \leqslant n$.

$\mathrm{Pr}_S(\lceil \xi \rceil)$ means, there is a proof of $\xi$ in $S$, where $\lceil \xi \rceil$ means: the Gödel number of $\xi$. So $\lceil \mathrm{Pr}_S(\lceil \xi \rceil) \rceil$ means: "the Gödel number of sentence 'there is a proof of $\xi$ in $S$'". Condition 2 above translates as: there is a proof of $[\{e\}$ is total$]$ in $S$ whose Gödel number is $\leqslant n$.

**Proposition 8.8.** *We can explicitly compute a Gödel number $e_{\mathsf{F}}$ so that $\{e_{\mathsf{F}}\} = \mathsf{F}$.*

**Proposition 8.9.** *If $S$ is consistent then*

$$S \nvdash \forall m \exists n [\{e_{\mathsf{F}}\}(m) = n].$$

Notice that $[P < NP]^{\mathsf{F}} \leftrightarrow \forall m,\ a \exists x \neg Q_{\mathsf{F}}(m,a,x)$.

**Remark 8.10.** Functions such as $\mathsf{F}$ are prominent in the study of transfinite progressions of theories [3,14,16]. The idea of such functions goes back to Kleene [22,23] and was first thoroughly explored by Kreisel [25].

**Lemma 8.11.** *If $I \subseteq \omega$ is infinite and $0 \in I$, then*

$$S \vdash \{[\forall m \forall a \in I \exists x \neg Q(m,a,x)] \rightarrow [\forall m \forall a \in \omega\ \exists x \neg Q(m,a,x)]\}.$$

The meaning of this result is: as long as we have an infinite succession of ever larger bounds that make the Turing machines polynomial, our standard definitions hold. The size of the intermediate gaps between each pair of bounds does not matter. However notice that there is in general no equivalence in $S$ between $[P < NP]$ and $[P < NP]^{\mathsf{F}}$ (see [5]):

**Proposition 8.12.** $S \vdash [P < NP]^{\mathsf{F}} \leftrightarrow \{[\mathsf{F}\ \textit{is total}] \wedge [P < NP]\}$.

**Sketch of proof.** For the formal, computational proof see [5]. It is easy to see that

$$S + [P < NP] + [\mathsf{F} \text{ is total}] \vdash [P < NP]^{\mathsf{F}}.$$

For the converse, the fact that the exotic counterexample function $\mathsf{f}_{\mathsf{F}}$ (see below after Lemma 8.13) is total implies that $[P < NP]$ and $[\mathsf{F}$ is total$]$ hold at the same time (we will require Lemma 8.11 here). □

We quote a result that follows from the above due to its importance:

**Lemma 8.13.** $S \vdash [P < NP]^{\mathsf{F}} \rightarrow [\mathsf{F}\ \textit{is total}.]$.

**Remark 8.14.** The formal proof is in [13]. The following argument clarifies the meaning of the lemma and gives an informal proof for it. Let

$$\mathsf{f}_{\mathsf{F}}(\langle m,a \rangle) = \min_x [\neg Q(m, \mathsf{F}(a), x)],$$

where we can look at $\mathsf{F}$ as a (partial) recursive function. (The brackets $\langle \ldots, \ldots \rangle$ note the usual 1–1 pairing function.) Now if $\mathsf{f}_{\mathsf{F}}$ is total, then $\mathsf{F}(a)$ has to be defined for all values of the argument $a$, that is, $\mathsf{F}$ must be total. The function $\mathsf{f}_{\mathsf{F}}$ is the so-called *exotic counterexample function* to $[P = NP]^{\mathsf{F}}$. We can similarly define a *standard counterexample function*

$$\mathsf{f}(\langle m,a \rangle) = \min_x [\neg Q(m,a,x)].$$

As $S \vdash [\mathsf{F}$ is total$] \leftrightarrow [S$ is $\Sigma_1$-sound$]$ (see [3] on that equivalence), and also as $S \vdash [S$ is $\Sigma_1$-sound$] \rightarrow \mathrm{Consis}(S)$, $S \vdash [\mathsf{F}$ is total$] \rightarrow \mathrm{Consis}(S)$. Then

**Lemma 8.15.** $S \vdash [P < NP]^{\mathsf{F}} \rightarrow \mathrm{Consis}\,(S)$.

**Proposition 8.16.** *If $S$ is consistent, then $S$ does not prove $[P < NP]^{\mathsf{F}}$.*

**Proof.** $S \vdash [[P < NP]^{\mathsf{F}} \rightarrow (\mathsf{F} \text{ is total})]$ (Lemma 8.13). So, $S$ cannot prove $[P < NP]^{\mathsf{F}}$. □

**Corollary 8.17.** *$[P = NP]^{\mathsf{F}}$ is consistent with $S$:*

If $\mathbf{N} \models S$ and makes it arithmetically sound, that is, $\mathbf{N}$ has a standard arithmetic part for the arithmetic in $S$:

**Proposition 8.18.** $\mathbf{N} \models S + [P < NP] \leftrightarrow [P < NP]^{\mathsf{F}}$.

**Proof.** $\mathsf{F}$ is total in the standard model for arithmetic. □

**Proposition 8.19.** *$[P < NP] \leftrightarrow [P < NP]^{\mathsf{F}}$ is independent of $S$.*

**Proof.** $S$ does not prove that equivalence due to Proposition 8.12, since otherwise it would prove [$\mathsf{F}$ is total]. On the other hand, consistency of

$$S + [P < NP] \leftrightarrow [P < NP]^{\mathsf{F}}$$

follows from the fact that it trivially holds in the standard model for arithmetic. □

**Remark 8.20.** Notice that if the sentence $\neg$[$\mathsf{F}$ is total] holds in some model for our theory $S$, the fact that we have in that model $\exists x(\mathsf{F}(0) = x)$, $\exists x(\mathsf{F}(1) = x)$, $\exists x(\mathsf{F}(2) = x), \ldots$, together with $\neg$[$\mathsf{F}$ is total] shows that: if $\neg$[$\mathsf{F}$ is total] holds, then $S + \neg$[$\mathsf{F}$ is total] is $\omega$-inconsistent. More precisely: if some theory $S'$ proves $\neg$[$\mathsf{F}$ is total], then it is $\omega$-inconsistent. So, if some theory $S'$ is *$\omega$-consistent*, then [$\mathsf{F}$ is total] is consistent with it.

Thus our main result:

**Proposition 8.21.** *If $S + [P = NP]^{\mathsf{F}}$ is $\omega$-consistent, then $S + [P = NP]$ is consistent.*

(In the authors' view, the version given in the 2003 paper [5] for the condition that implies the desired consistency was more obscure: if $S + [P = NP]^{\mathsf{F}} + [\mathsf{F} \text{ is total}]$ is consistent, then $S + [P = NP]$ is consistent. The condition used in [5] is implied by the $\omega$-consistency hypothesis above.)

**Remark 8.22.** Now we may ask: does consistent theory $S + [P < NP] \leftrightarrow [P < NP]^{\mathsf{F}}$ prove $[P < NP]^{\mathsf{F}}$? If it does not, we are done.

Let us mention that Okamoto and Kashima recently claimed a controversial but similar result that depends on a stricter $\omega$-consistency-like condition [32]; for an analysis and criticism see [28].

### *8.1. On theory $S + [P < NP] \leftrightarrow [P < NP]^{\mathsf{F}}$*

The present discussion is informal, and includes some steps that must yet be carefully considered. However we think that it is convincing enough to justify its presentation here as a speculative approach.

Suppose that predicate $K(m, a, x)$ means "poly Turing machine of Gödel number $m$ bounded by polynomial $|x|^a + a$ on the length of input $|x|$ outputs a satisfying line of truth-values for input $x$". Suppose also that $K$ can be made primitive recursive. We take our poly machines in the BGS family [1]. Now:

- We claim that theory

  $$S^* = S + \forall m, a, a'[\exists x \neg K(m, a, x) \leftrightarrow \exists x' \neg K(m, \mathsf{F}(a'), x')]$$

  is consistent and holds of $\mathbf{N} \models S$, a model with standard arithmetic part. (The naïve interpretation of sentence:

  $$\forall m, a, a'[\exists x \neg K(m, a, x) \leftrightarrow \exists x' \neg K(m, \mathsf{F}(a'), x')]$$

  in the standard model for arithmetic is quite straightforward; roughly it means that poly machine coded by $m$ with bound $|x|^a + a$ fails to output a satisfying line for input $x$ if and only if the same machine with bound

$|x|^{\mathsf{F}(a')} + \mathsf{F}(a')$ again fails to output a satisfying line given $x$ as input. *The preceding discussion is not intended as a proof of our claim.* We believe that a rigorous, formal proof must be given here instead of the "waving hands" argument we are presently offering.)

- Then $S^* \vdash [P < NP] \leftrightarrow [P < NP]^{\mathsf{F}}$.
- Now: suppose that $S \vdash [P < NP]$.
- Then the first $x$ where BGS machine $m$ fails to output a satisfying line of truth-values is bounded by a total provably recursive function g (as we prove the $\Pi_2$ sentence $[P < NP]$ in $S$, which means that the counter-example function f is provably total recursive in $S$).
- Therefore, we get that $S^*$ proves

$$\forall m, a, a'[\exists x < \mathsf{g}(m, a, a')\neg K(m, a, x) \leftrightarrow \exists x' < \mathsf{g}(m, a, a')\neg K(m, \mathsf{F}(a'), x')],$$

for some provably total recursive g. This is a $\Pi_1$ sentence.

- Recall Kreisel's Lemma [2]: if $S$ has a recursively enumerable set of theorems, contains arithmetic and has a model with standard arithmetic, then $S$ and $S + \{\text{all the true } \Pi_1 \text{ arithmetic sentences}\}$ have the same provably total recursive functions.
- Now, we apply Kreisel's Lemma to that $\Pi_1$ sentence in $S^*$ [2,15]. We then conclude that $S$ and $S^*$ have the same provably total recursive functions, as these theories only differ by a true $\Pi_1$ sentence.
- Now: we know that [5] $S \vdash [P < NP]^{\mathsf{F}} \rightarrow [\mathsf{F} \text{ is total}]$.
- Since $S^*$ proves the equivalence $[P < NP] \leftrightarrow [P < NP]^{\mathsf{F}}$, $S^* \vdash [P < NP] \rightarrow [\mathsf{F} \text{ is total}]$, from the hypothesis that $S \vdash [P < NP]$ we get that $S^* \vdash [\mathsf{F} \text{ is total}]$. A contradiction, since $S^*$ cannot prove $[\mathsf{F} \text{ is total}]$.
- Thus $S \nvdash [P < NP]$.

(We stress the informality of this argument.) See also [8,13].

## 9. A final remark

All those results point in the direction of the consistency of some strong theory $S$ with $P = NP$. However it is surprising that we have no similar (or corresponding) results for $P < NP$. What can we make out of that?

The theorems collected here provide a very compelling reason to expect that sooner or later a proof of the consistency of $P = NP$ (formalized as a $\Sigma_2$ sentence) with some powerful axiomatic system will be provided. This is the natural outcome of our exposition here, if we expect formal mathematical entities to behave in a sensible, intuitive, way. Specifically, our result (Proposition 8.21) implies that consistency – and its hypothesis has a simple paraphrasis, namely, "if Turing machines in theory $\mathrm{ZFC} + [P = NP]^{\mathsf{F}}$ behave as they do in the 'real world'..."; see [5], Addendum.

On the other hand, if the $\Pi_2$ sentence $[P < NP]$ is proved by ZFC, consistent theory $\mathrm{ZFC} + [P = NP]^{\mathsf{F}}$ will only have models where Turing machines have a very weird, counterintuitive, behavior. Of course that might in fact be the case, but if it holds, it will be the indication of a deep cleavage between formalized theories and the intuitive kind of mathematics practiced by most professional mathematicians, an intuitive construct the formal axiomatizations are supposed (and intended) to mirror.

Naturally, if formalization and intuition go hand-in-hand, then our discussion means that no theory like $S$ can prove $[P < NP]$.

Would that mean that the sentence $[P = NP]$ and therefore $P = NP$ is true in our everyday, "real" world of integers and sums and products? We do not think so. We believe that independence holds. And if independence of $[P < NP]$ and of $[P = NP]$ holds with respect with some strong, arithmetically sound theory $S$, then $[P < NP]$ is true of the standard integers. That is to say, $[P < NP]$ will be proved in a theory like Peano Arithmetic plus some reasonable infinitary rule like Shoenfield's recursive $\omega$-rule.

That is to say, $P < NP$ will – quite naturally – be seen as true in the real world of computers, as expected.

## Acknowledgements

The authors wish to thank the Institute for Advanced Studies at the University of São Paulo, as well as its Director Professor J. Steiner for support of this ongoing research project, as well as the Academia Brasileira de

Filosofia and its chairman Professor J. R. Moderno. Portions of this work were done during the COBERA March 2005 Workshop at Galway, Ireland; we wish to thank Professor Vela Velupillai for the stimulating and fruitful environment he so kindly sponsored at that meeting. We also wish to acknowledge detailed help and criticisms by Professor M. Guillaume, whom we heartily thank.

And finally F.A.D. wishes to thank Professor C.A. Cosenza (Federal Univ. at Rio de Janeiro) for his invitation to join his research program team.

## References

[1] T. Baker, J. Gill, R. Solovay, Relativizations of the $P = ?NP$ question, SIAM J. Comput. 4 (1975) 431–442.
[2] S. Ben-David, S. Halevi, On the independence of $P$ vs. $NP$, Technical Report # 699, Technion, 1991.
[3] L. Beklemishev, Provability and reflection, Lecture Notes for ESSLLI'97, 1997.
[4] N.C.A. da Costa, F.A. Doria, Undecidability and incompleteness in classical mechanics, Int. J. Theor. Phys. 30 (1991) 1041–1073.
[5] N.C.A. da Costa, F.A. Doria, Consequences of an exotic formulation for $P = NP$, Appl. Math. Comput. 145 (2003) 655–665, also "Addendum" 172 (2006) 1364–1367.
[6] N.C.A. da Costa, F.A. Doria, On set theory as a foundation for computer science, Bull. Sec. Logic, Lodz 33 (2004) 33–40.
[7] N.C.A. da Costa, F.A. Doria, Computing the future, in: K. Vela Velupillai (Ed.), Computability, Complexity and Constructivity in Economic Analysis, Blackwell, 2005.
[8] N.C.A. da Costa, F.A. Doria, Strength of theory ZFC + $[P < NP] \leftrightarrow [P < NP]^{\mathsf{F}}$, in press.
[9] N.C.A. da Costa, F.A. Doria, Metamathematics of Science, in press.
[10] M. Davis, Hilbert's tenth problem is unsolvable, in: Computability and Unsolvability, Dover, 1982.
[11] R.A. DeMillo, R.J. Lipton, Some connections between computational complexity theory and mathematical logic, in: Proceedings of the 12th Annual ACM Symposium on the Theory of Computing, 1979, pp. 153–159.
[12] R.A. DeMillo, R.J. Lipton, The consistency of $P = NP$ and related problems with fragments of number theory, in: Proceedings of the 12th Annual ACM Symposium on the Theory of Computing, 1980, pp. 45–57.
[13] F.A. Doria, Informal vs. formal mathematics, Synthèse, in press.
[14] S. Feferman, Transfinite recursive progressions of axiomatic theories, J. Symb. Logic 27 (1962) 259–316.
[15] S. Fortune, D. Leivant, M. O'Donnell, The expressiveness of simple and second-order type structures, J. ACM 38 (1983) 151–185.
[16] T. Franzen, Transfinite progressions: a second look at completeness, Bull. Symb. Logic 10 (2004) 367–389.
[17] J. Hartmanis, J. Hopcroft, Independence results in computer science, SIGACT News 13 (1976).
[18] H. Huwig, A definition of the $P = NP$ problem in categories, Lect. Notes Comput. Sci. 117 (1981) 146–153.
[19] D. Joseph, P. Young, Independence results in computer science? in: Proceedings of the 12th Annual ACM Symposium on the Theory of Computing, 1980, 58–69.
[20] D. Joseph, P. Young, Fast programs for initial segments and polynomial time computation in weak models of arithmetic, STOC Milwaukee 1981, 1981, pp. 55–61.
[21] R. Kaye, Models of Peano Arithmetic, Clarendon Press, 1991.
[22] S.C. Kleene, General recursive functions of natural numbers, Math. Ann. 112 (1936) 727.
[23] S.C. Kleene, Mathematical Logic, Wiley, 1967.
[24] W. Kowalczyk, A sufficient condition for the consistency of $P = NP$ with Peano Arithmetic, Fund. Inform. 5 (1982) 233–245.
[25] G. Kreisel, On the interpretation of non-finitist proofs, I, J. Symb. Logic 16 (1951) 241, II, 17 (1952) 43.
[26] G. Kreisel, On the concepts of completeness and interpretation of formal systems, Fund. Math. 39 (1952) 103–127.
[27] K. Kunen, A Ramsey theorem in Boyer–Moore logic, J. Automated Reasoning 15 (1995) 217.
[28] S. Laur, Overview of recent claims about $P \neq NP$, preprint, 2005.
[29] M. Machtey, P. Young, An Introduction to the General Theory of Algorithms, North-Holland, 1979.
[30] A. Maté, Nondeterministic polynomial-time computations and models of arithmetic, J. ACM 37 (1990) 175–193.
[31] M. O'Donnell, A programming language theorem which is independent of Peano Arithmetic, in: Proceedings of the 11th Annual ACM Symposium on the Theory of Computation, 1979, pp. 176–188.
[32] T. Okamoto, R. Kashima, Resource bounded unprovability of computational lower bounds, preprint (original version 2003; revised 2005).
[33] J. Paris, L. Harrington, A mathematical incompleteness in Peano Arithmetic, in: J. Barwise (Ed.), Handbook of Mathematical Logic, Springer, 1989.
[34] H. Rogers Jr., Theory of Recursive Functions and of Effective Computability, reprint, MIT Press, 1992.
[35] V.Yu. Sazonov, A logical approach to the problem $P = NP$? Lect. Notes Comput. Sci. 88 (1980) 562–575.
[36] V.Yu. Sazonov, e-mail to the authors, 2005.
[37] C. Smorýnski, Logical Number Theory, I, Springer, 1991.
[38] C. Sureson, $P$, $NP$, Co–NP, and weak systems of arithmetic, Theor. Comput. Sci. 154 (1996) 145–163.
[39] G. Takeuti, Kreisel and I, in: P.G. Odifreddi (Ed.), Kreiseliana, A.K. Peters, 1996.