$P^f \neq NP^f$ for almost all f

J. D. Hamkins*
Mathematics, Georgia State University, 30 Pryor Street, Atlanta, GA 30300 USA.
Mathematics, City University of New York, 365 Fifth Avenue, New York, NY 10016 USA.
http://jdh.hamkins.org

P. D. Welch†
Department of Mathematics, University of Bristol, Bristol BS8 1TW, England.
Mathematisches Institut, Beringstraße 6, 53115 Bonn, Germany.
http://www.maths.bris.ac.uk/~mapdw

October 21, 2002

Abstract

We discuss the question of Ralf-Dieter Schindler whether for infinite time Turing machines $P^f = NP^f$ can be true for any function f from the reals into ω_1. We show that "almost everywhere" the answer is negative.

1 Introduction

After establishing $P \neq NP$ for infinite time Turing machines, Ralf-Dieter Schindler in [5] introduced the more general question of whether $P^f = NP^f$ for these machines. The classes are defined as follows:

*The research of J. D. Hamkins is supported in part by grants from the NSF and the PSC-CUNY Research Foundation.

†P.D.Welch would like to thank the Deutsche Forschungsgemeinschaft and the University of Bonn for the support of this research.

Definition 1.1 *Let* $f : {}^{\omega}2 \longrightarrow \omega_1$ *and* $A \subseteq {}^{\omega}2$.

(A) We say that $A \in P^f$ *if there is an (infinite time) Turing machine computable function* φ_e *so that*
(i) A *is decidable by* φ_e*, that is* $x \in A$ *if and only if* $\varphi_e(x)\downarrow 1$*, and*
(ii) $\forall z \in {}^{\omega}2 \quad \varphi_e(z)\downarrow$ *in at most* $f(z)$ *many steps.*

(B) We say that $A \in NP^f$ *if there is a Turing machine computable function* φ_e *so that*
(i) $x \in A$ *if and only if there exists* $y \in {}^{\omega}2$ *so that* $\varphi_e(x \oplus y)\downarrow 1$*, and*
(ii) $\forall z = (x \oplus y) \in {}^{\omega}2 \quad \varphi_e(z)\downarrow$ *in at most* $f(x)$ *many steps.*

The function f plays the role here of the class of polynomials in the classical $P = NP$ question, bounding the length of the allowed computations. Because of this, one is primarily interested here in the functions f which are Turing invariant, in the sense that if x and y are Turing equivalent, then $f(x) = f(y)$. Indeed, since one might expect that a more complicated input should be allowed more time for computation, it is natural to restrict attention only to the functions f for which $x \leq_T y$ implies $f(x) \leq f(y)$. The main results of this article, however, do not happen to rely on these assumptions. Since the computations allow for infinite input, one might usually want to assume that $f(x)$ is infinite.

If the value of $f(x)$ is some constant α then the classes P^f lie strictly within the Borel hierarchy ([5] Lemma 2.7). If $f(x) = \omega_1^x$ then P^f coincides with hyperarithmetic (and so we are really still within the realms of Kleene recursion *e.g.* see [3]). When $f(x) > \omega_1^x$ for all x we then truly enter for the first time the world of sets that are essentially computed by infinite time Turing machines (see [2] for the basic concepts). [5] raises two questions concerning these classes for such functions dominating $f(x) = \omega_1^x$.

Note that unlike the basic notions of P and NP Schindler defined in §2 of [5], his class NP^f is not, in general, just the projection of P^f.

We wish to prove that for almost all functions f the classes P^f and NP^f are different. Given the extra information that the verifying witness y can encode, this is, or should be, unsurprising. The first point to note is that if the values f takes are sufficiently large, they will exceed the times needed by a machine to establish membership of any decidable set.

We recall a definition from [2]:

Definition 1.2 $\lambda^x =_{df} \sup\{\alpha \mid \exists e\, \varphi_e(x)\downarrow y \wedge y \in WO \wedge rk(y) = \alpha\}$.

Equivalently (and the reader may take this as a definition):

Fact 1.3 *([8] Theorem 1.1) λ^x is the supremum of halting times of any Turing computable function on input x.*

Implicit in this latter result—when taken with the definition of decidable sets of reals [2]—(see the discussion in [6]) is the following characterisation of such sets.

Fact 1.4 *$A \in {}^{\omega}2$ is decidable if and only if there are Σ_1 formulae in the language of set theory $\varphi_0(v_0), \varphi_1(v_0)$ so that*

$$x \in A \iff L_{\lambda^x}[x] \models \varphi_0[x] \iff L_{\lambda^x}[x] \models \neg\varphi_1[x]$$

Clearly then:

Lemma 1.5 *If $\lambda^x \leq f(x)$ for all x, then P^f is the class of decidable sets.*

We recall a definition from [8]:

Definition 1.6 $\Sigma^x =_{df} \sup\{\ \alpha \mid$ *a code $x \in WO$ for α occurs on a tape of some computation $\varphi_e(x)$ at some time* $\}$

(Thus Σ^x is the supremum of ordinals with so-called *accidentally writable* (relative to x) codes, as defined in [2] p. 580.) It is shown in [7] that (i) Σ^x is the least σ so that $L_\sigma[x]$ has a proper Σ_2-elementary substructure: and in [8] that (ii) $L_{\Sigma^x}[x]$ is inadmissible.

Our first theorem is that any function f that dominates the function $g(x) = \Sigma^x$ separates the classes NP^f and P^f. This answers Question 2 of [5] "almost everywhere":

Theorem 1.7 *Let f satisfy $f(x) \geq \Sigma^x$ for all x. Then $NP^f \supsetneq P^f$.*

The next theorem answers Question 1 of [5]. Let f_0 be defined as $f_0(x) = \omega_1^x + \omega$.

Theorem 1.8 $NP^{f_0} \supsetneqq P^{f_0}$.

In [5] the classes P^{f_0} and NP^{f_0} are denoted P^{++} and NP^{++}, respectively, so we have shown $P^{++} \neq NP^{++}$, the question left open in [5]. As [5] points out, of course P^{f_0} contains all Σ^1_1 and Π^1_1 sets. It is not hard to see that if $A \in \mathrm{Diff}(< \omega_1^{ck}, \Sigma^1_1)$ the Hausdorff difference hierarchy for levels below the first non-recursive ordinal, then $A \in P^{f_0}$.

Variants of these methods will also show that many other functions whose range falls between these two ordinals will also separate these two classes. We have not attempted an exhaustive classification.

Suppose $\phi_e : {}^\omega 2 \longrightarrow \mathbb{N}$ is total. Then the length of the computation $\phi_e(x)$, ν_x say, defines for us a "clock". Namely let $f(x) = \nu_x$; suppose for convenience ν_x is always of limit length and at least ω^ω.

Theorem 1.9 *With f as above:* $NP^f \supsetneqq P^f$.

2 Preliminaries

We shall let ω_1^x stand for the first ordinal not recursive in x. Then $L_{\omega_1^x}[x]$ is an *admissible* set. We refer the reader to [1] for an account of admissible sets and their basic properties. We shall use the following notation for the machine configurations. Let the cells of the tape be enumerated $\langle C_i | i < \omega \rangle$ with the cell C_i having value $C_i(\xi)$ at time ξ. We assume that the first n blocks on the tape are enumerated by $\langle C_i | i < 3n \rangle$ with C_0, C_1, C_2 being the leftmost output,scratch, and input cells respectively. A *snapshot* of the tape at time γ is then a function $s \in {}^\omega 2$ coding these cell values, with $s(i) = C_i(\gamma)$ (possibly also allowing it to encode somewhere internal states, the location of the head and the instruction of the program about to be performed). A halting computation is then entirely given by the wellordered snapshot sequence of computations of the length of the computation. The machine is considered to be specified by a finite program, just as for ordinary Turing machines, although the head is allowed to read, and write to, triples of cells at any one stage. Thus a typical instruction might be of the form

(q_i, j, j', X, q_k) where $j, j' \in {}^3 2$, interpreted to mean that in state i viewing cells C_{3l}, C_{3l}, C_{3l+2} with values $j(0), j(1), j(2)$ the machine moves to state q_k, changes the cell values to those of j' and moves one unit in the direction $X \in \{L, R\}$ (for Left and Right). The machine has however a special limit state q_L (and at limit times it is in this state viewing C_0, C_1, C_2). The machine may thus halt at a limit time if it contains a quadruple of the form (q_L, j, j', q_H). Note however that executing this last step of computation means that it changes the entries of C_0, C_1, C_2 to those given by j'. It is this feature that allows the classes P^{f_0} to be closed under complementation: an "accepting" entry of C_0 as 1 can be switched to a "rejecting" zero at the last moment. To be completely clear, if the machine executes the halt instruction at the limit stage ν, we reiterate that the final value is $C_i(\nu + 1) = j'(i)$, which might differ from the limit value $C_i(\nu) = j(i)$.

We shall use the following fact.

Fact 2.1 *[4] There is an index $e_0 \in \omega$ so that, uniformly, for any $x \in {}^\omega 2$ $\{e_0\}^x$ is an illfounded linear ordering of ω, recursive in x, with wellfounded part of order type ω_1^x.*

We use this index to give us a "canonical $\omega_1^x + \omega$-clock": an algorithm that halts in exactly $\omega_1^x + \omega$ steps. (The following argument is the "uniform in x" version of that of Theorem 3.2 of [2].) The algorithm does the following: it first computes the field of the relation $<_R = \{e_0\}^x$ and then proceeds, by picking the least element of the field, n say, in ω many steps to find the $<_R$-least element below n of the ordering. It then, in another ω many steps, proceeds to strike out all mention of this element from the field of $\{e_0\}^x$; it then picks an element n' of the field that is left and searches for the next $<_R$-least element below this n'; at limit stages below ω_1^x the procedure continues smoothly as the wellfounded part of $<_R$ has order type ω_1^x. However in the interval $(\omega_1^x, \omega_1^x + \omega)$ it searches in vain for a least element. It chooses some $\overline{n}$ in the field that is left. We may assume that each time it descends in $<_R$ it flashes a signal in C_2, by alternating in the next 3 stages the value of C_2 to be "0,1,0". After ω many stages, by the limsup rule of the machines, the value $C_2(\omega_1^x + \omega) = 1$, and moreover this is the first time this happens at a limit stage. We assume then the program has been written so as to immediately halt if this occurs.

An alternative $\omega_1^x + \omega$ clock is obtained by the algorithm that on input x simulates all computations on input x, looking for a stage at which none of the programs halt. Since ω_1^x is the least such stage, and it takes ω many steps to recognize that this situation has occurred, the algorithm can halt exactly at $\omega_1^x + \omega$.

Definition 2.2 $\zeta^x =_{df} \sup\{\, \alpha \mid \exists e \exists y \exists \gamma \forall \delta > \gamma\ y \in WO$ *lies on the output tape* $\varphi_e(x)$ *at time* $\delta \wedge rk(y) = \alpha\,\}$.

We shall appeal to the following fact:

Fact 2.3 *("(λ, ζ, Σ) Theorem")*

(i)([8] cf. *2.3.) For any computation of the form $\phi_e(x)$, the snapshot at time ζ, s_ζ, is exactly that at time Σ^x, s_Σ; they are both* settled *snapshots,* i.e. *they are destined to recur on a closed and unbounded class of ordinals;*

(ii) ([7] 2.1,2.3) if ξ is the least ξ satisfying $L_\xi[x] \prec_{\Sigma_2} L_{\Sigma^x}[x]$, then (a) $\xi = \zeta^x$ and (b) $(\lambda^x, \zeta^x, \Sigma^x)$ is the lexicographic least increasing triple (λ, ζ, Σ) satisfying $L_\lambda[x] \prec_{\Sigma_1} L_\zeta[x] \prec_{\Sigma_2} L_\Sigma[x]$.

By [8] (Claim (ii) of 3.4) there are computations ϕ_{q_0} so that for any input x, (ζ^x, Σ^x) is the lexicographically (on $On \times On$) least pair of ordinals with repeating snapshots $(s_{\zeta^x}, s_{\Sigma^x})$: running a universal machine provides such. (In fact, any computation on input x which does not repeat before λ^x is such an example. In such a case it is the snapshot s_{ζ^x} that provides a parameter witnessing the inadmissability of $L_{\Sigma^x}[x]$ - *cf.* [8], 3.4) In general then, this pair of snapshots witnesses that the computation is either halted or in an infinite loop.

3 Separating the classes

Proof of Theorem 1.7. By our observation in the introduction, under these assumptions P^f is the class of decidable sets of reals. Let $H = \{\langle p, x\rangle \in \omega \times {}^\omega 2 \mid \phi_p(x)\!\downarrow\,\}$. Then H is the complete set coding the halting problem for sets of reals. H is undecidable, but the above arguments, together with the (λ, ζ, Σ) Theorem will show that $H \in NP^f$.

It will suffice to verify whether $\phi_e(x)\downarrow$ by the following method.

We consider informally a Turing machine computable algorithm P_n that effects the following:

P_n on input $e \frown x \oplus y$:
(i) First checks whether y codes an ω-model containing (an isomorphic copy of) x and:

$$\mathcal{A} = \langle \omega, y \rangle \models \text{“}KP \wedge V = L[x] \wedge \Sigma^x \text{ exists} \wedge \phi_e(x)\downarrow\ .\text{”}$$

By way of explanation: we intend that $\mathcal{A}$ thinks there is a least initial segment of its $L[x]$-hierarchy with a proper Σ_2 elementary substructure - this is the import of "Σ^x exists." This is an arithmetic condition on $x \oplus y$ and thus can be checked by P_n in $< \omega^\omega$ many steps. If this fails for y then $P_n(e \frown x \oplus y)\downarrow 0$ thus halting with a zero in the first cell C_0 of the output tape.

(ii) Otherwise a preliminary "1" is written to C_0 and then P_n proceeds to check if $WFP(\mathcal{A})$ contains the true Σ^x. However we first dispose of the part of the model containing all sets of $(L[x]\text{-rank})^{\mathcal{A}} \geq (\Sigma^x)^{\mathcal{A}}$. We simply eliminate all reference to these in y, thus in effect rewriting y as some new real $\overline{y}$. However this is again a simple operation, and can be done in, say, ω^2 steps (note that y has some integer n which denotes the $(\Sigma^x)^{\mathcal{A}}$ so it is a trivial matter to do this). The process then proceeds to check for the wellfoundedness of the ordinals of this new ω-model $\overline{\mathcal{A}} = \langle \omega, \overline{y} \rangle$ determined by the initial segment $(\Sigma^x)^{\mathcal{A}}$ in the usual way by erasing integers from the field of its ordinals.

If the model $\overline{\mathcal{A}}$ is wellfounded then this process takes the true Σ^x many steps, (note that $(\Sigma^x)^{\mathcal{A}} \cong \Sigma^x$ as there is n denoting $(\Sigma^x)^{\mathcal{A}}$ and the property of an ordinal being Σ^x is absolute), and furthermore the model $\overline{\mathcal{A}}$ is correct about $\phi_e(x)\downarrow$. Using the trick of keeping track of when the least (in some standard ordering of $\omega \times \omega$) pair is erased we may realise also that the field of the ordering of $On^{\overline{\mathcal{A}}}$ has become empty (*cf.* the proof of Theorem 3.1 in [2]). If so it can halt exactly at the Σ^x'th step with the required 1 in C_0.

If the model $\overline{\mathcal{A}}$ is illfounded (and hence $(\Sigma^x)^{\mathcal{A}}$ is in the illfounded part of the original $On^{\mathcal{A}}$), then in fact $\Sigma^x \not\subseteq WFP(\overline{\mathcal{A}}) = WFP(\mathcal{A})$: this is because (a) we cannot have $\Sigma^x \in WFP(\overline{\mathcal{A}})$ (as otherwise $\mathcal{A}$ would recognize it as Σ^x); (b) but neither can $WFP(\overline{\mathcal{A}}) = \Sigma^x$ (as $L_{\Sigma^x}[x]$ is inadmissible, and this would contradict the Truncation Lemma ([1]).

Hence any instance of illfoundedness in $(On)^{\overline{\mathcal{A}}}$ will be detected before the true Σ^x many steps have been taken. This leaves time to change the contents of C_0 to a zero, and halt - here before Σ^x many steps have been taken.

In each case then $P_n(e \frown x \oplus y)$ halts in no more than Σ^x many steps with the correct output.

Q.E.D.(Theorem 1.7)

The algorithm above can be made more time efficient, so that confirming instances of the decision problem are settled more quickly. This modified algorithm can be made to actually follow the naive idea that to determine whether $\phi_e(x)\downarrow$, one should simply simulate the computation $\phi_e(x)$ to see if it halts, and somehow end simulations that have gone on too long. The point is that the model-checking method of the previous argument, where one checks whether $\overline{\mathcal{A}}$ is well-founded, is essentially a nondeterministic clock for Σ^x, in the sense that it halts at time Σ^x for certain witnesses y, and before Σ^x for all other witnesses. Our modified algorithm, therefore, is simply to run such a clock alongside the computation of $\phi_e(x)$, and accept the input if the computation halts before the clock runs out. Since in the worst case the clock runs to time Σ^x, this algorithm nondeterministically decides whether $\phi_e(x)\downarrow$ in time Σ^x. But the point is that affirmative instances are decided much earlier, in time before λ^x, because this is when the halting computations actually halt.

The theorem can be improved by ignoring the bold-face context of the situation:

Theorem 3.1 *Suppose that $f(p) \geq \Sigma$ for every finite p and $f(x) \geq \omega$ for all other reals x. Then $NP^f \supsetneq P^f$.*

Proof: The idea is that the (weak) halting problem $h = \{p \mid \varphi_p(0)\downarrow\}$ will be in NP^f but not in P^f. It clearly is not in P^f, since it is not decidable. But one can see it is in NP^f by the following algorithm: on input $x \oplus y$, first check whether x is a finite p or not. If not, then halt and reject the input. Otherwise, carry out the algorithm of Theorem 1.7 on the input $p \frown 0 \oplus y$. With suitable choice of y, this will decide whether $(p, 0) \in H$, which is equivalent to $p \in h$, in at most $\Sigma^0 = \Sigma$ many steps, as desired. **Q.E.D.**

Proof of Theorem 1.9 Let $H_f = \{\langle p, x\rangle \in \omega \times {}^{\omega}2 \mid \phi_p(x)\downarrow \text{ in } \leq \nu_x \text{ steps }\}$

(1) $H_f \notin P^f$.

Proof: Let r be the partial function defined as follows:

$$r(y) = \begin{cases} 0 & \text{if } y = \langle p, x\rangle \wedge y \notin H_f \\ \uparrow & \text{otherwise} \end{cases}$$

If "$y \in H_f$" were decidable by an algorithm that always halted in at most ν_y steps then r would also be computable by an algorithm P_m, that if it converged, would do so in ν_y steps. (We could obtain a program for r by simply changing the behaviour of that of the former algorithm by switching at the very last limit step where it halted on a 1, into some non-halting loop.) Let $\phi_m : {}^{\omega}2 \longrightarrow \omega$ be this latter function. Let c_0 be the constant zero function. However then

$$\begin{aligned} \langle m, c_0\rangle \in H_f &\iff \phi_m(\langle m, c_0\rangle)\downarrow \text{ in } \leq \nu_x \text{ steps} \\ &\iff r(\langle m, c_0\rangle) = 0 \iff \langle m, c_0\rangle \notin H_f \end{aligned}$$

a contradiction. **Q.E.D.(1)**

(2) $H_f \in NP^f$.

Proof: We use the ideas from the proof of Theorem 1.7. We devise an algorithm P_n to verify (2). P_n on input $p \frown x \oplus y$:

(i) First checks whether y codes an ω-model containing (an isomorphic copy of) x and:

$$\mathcal{A} = \langle \omega, y\rangle \models \text{“}KP \wedge V = L[x] \wedge \phi_p(x)\downarrow \text{ in } \leq \phi_e(x) = \nu_x \text{ steps.”}$$

This is again an arithmetic condition on $x \oplus y$ and thus can be checked by P_n in $< \omega^\omega$ many steps. If this fails for y then $P_n(p \frown x \oplus y)\downarrow 0$.

(ii) Otherwise a preliminary "1" is written to C_0 and then P_n proceeds to check if $WFP(\mathcal{A})$ contains the true ν_x. As before we dispose of the part of the model containing all sets of $(L[x]\text{-rank})^{\mathcal{A}} \geq (\nu^x)^{\mathcal{A}}$. The process then proceeds to check for the wellfoundedness of this new initial segment model $\overline{\mathcal{A}}$ up to $On^{\overline{\mathcal{A}}} \cong (\nu_x)^{\mathcal{A}}$. We use that [2] (Theorem 8.8) shows ν_x is not an x-admissible ordinal.

If the model is wellfounded then this process takes the true ν_x many steps, thus it can halt exactly at the ν_x'th step with the required 1 in C_0.

Arguing as before using the cited inadmissibility of $L_{\nu_x}[x]$, the wellfounded part of $\mathcal{A}$ cannot have rank exactly the true ν_x; hence we are justified in testing only the initial segment of the ordinals of $\overline{\mathcal{A}}$ determined by $(\nu_x)^{\mathcal{A}}$. Then any instance of illfoundedness will be encountered strictly before ν_x many steps have been taken.

In each case then $P_n(p \frown x \oplus y)$ halts in no more than ν_x many steps with the correct output.

Q.E.D.(Theorem 1.7)

As a final comment some of the above discussion may lead one to considering the class of sets A such that $x \in A$ and $x \notin A$ can each be verified quickly, that is, such that there are two programs, such that $x \in A$ if and only if there is a witness y such that $x \oplus y$ is accepted by the first program, and $x \notin A$ if and only if there is a witness y such that $x \oplus y$ is accepted by the second program, and both programs halt on any input $x \oplus y$ in time before $f(x)$ if they halt at all. That is, both $x \in A$ and $x \notin A$ can be verified quickly, with the correct choice of verifying witnesses, but there is no insistence that the programs compute quickly (or even halt at all) when given irrelevant witnesses verifying nothing. B. Löwe has pointed out that such a class of sets corresponds to a notion of $NPTIME^f \cap co\text{-}NPTIME^f$, but we have made no investigation of such concepts.

References

[1] J. Barwise, *Admissible sets and structures*, Perspectives in Mathematical Logic, Springer Verlag, 1975.

[2] J.D.Hamkins and A.Lewis, *Infinite time turing machines*, Journal of Symbolic Logic **65** (2000), no. 2, 567–604.

[3] K.Hrbacek and S.Simpson, *On kleene degrees of analytic sets*, Proceedings of the Kleene Symposium (J.Barwise, H.J.Keisler, and K.Kunen, eds.), Studies in Logic, North-Holland, 1980, pp. 347–352.

[4] R.O.Gandy, *On a proof of Mostowski's conjecture*, Bull. de l'Academie Polonaise des Sciences, série mathématique, astronomique, et physique **8** (1960), 571–575.

[5] R.-D. Schindler, *$P \neq NP$ for infinite time Turing machines*, Monatsheft für Mathematik (to appear).

[6] P. D. Welch, *Post's and other problems in higher type supertasks*, Complexity in Mathematics and Computer Science (B. Löwe W. Malzkorn T. Räsch, ed.), Foundations of the Formal Sciences, vol. III, 200?

[7] ———, *Eventually infinite time Turing degrees: infinite time decidable reals*, Journal for Symbolic Logic **65** (2000), no. 3, 1193–1203.

[8] ———, *The length of infinite time Turing machine computations*, Bulletin of the London Mathematical Society **32** (2000), 129–136.