Is **P** = **PSPACE** for Infinite Time Turing Machines?

Joost Winter

1 Introduction

In the existing literature on infinite time Turing machines, which were originally defined in [HaLe], issues of time complexity have been widely considered. The question $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ for Infinite Time Turing Machines, and several variants on it, are treated in e.g. [Sc], [DeHaSc], and [HaWe].

Besides time complexity, we may also try to look at issues of space complexity in ITTMs. However, because an ITTM contains tapes of length ω, and all nontrivial ITTM computations will use the entire, ω-length tape, simply measuring the space complexity by counting the portion of the tape used by the computation is not an option. In [Lö], therefore, an alternate notion of space complexity is provided, that is based on looking at the levels of Gödel's constructible hierarchy where the snapshots of the computation can be found.

With this notion of space complexity of ITTMs, we can consider questions such of the type $\mathbf{P} \stackrel{?}{=} \mathbf{PSPACE}$, analogous to the existing work on the question $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ for ITTMs. In this paper, we will look at some of these questions, in a manner analogous to the earlier work on ITTM time complexity classes.

2 Definitions

For any infinite time Turing machine computation, we define the following operations:

Definition 2.1. (1) $q^T_\alpha(x)$ denotes the state a machine T is in at stage α, having started from the input x; (2) $h^T_\alpha(x)$ denotes the position of the head at stage α, having started from the input x; and (3) $c^T_{i,\alpha}(x)$, where $i \in \{1, 2, 3\}$, denotes the content of tape i at stage α, having started from the input x. Also, we will define another operation, $c^T_{i,n,\alpha}(x)$, which will be defined as 1 if $n \in c^T_{i,\alpha}(x)$, and 0 otherwise.

Following [Lö] and earlier articles such as [Sc], the ITTM space and time complexity classes are defined as follows:

Definition 2.2. If T is a machine that eventually reaches the halting state q_{f}, and α is the unique ordinal such that $q^T_\alpha(x) = q_{\mathrm{f}}$, then we say that $\mathbf{time}(x, T) = \alpha$.

Definition 2.3. For any function $f : \mathbb{R} \to \mathrm{Ord}$ and any ITTM T, we say that T is a **time** f **machine** if $\mathbf{time}(x, T)$ is defined for all x and, for all $x \in \mathbb{R}$, we have $\mathbf{time}(x,T) \leq f(x)$. For any ordinal ξ, we say that T is a **time ξ machine** if T is a time f machine for the constant function f with $f(x) = \xi$ for all x.

Definition 2.4. The family of all sets of reals that are decidable by a time f machine is denoted by $\mathbf{P}_f$. For any ordinal ξ, $\mathbf{P}_\xi$ denotes the family of all sets of reals that are decidable by a time η machine for some $\eta < \xi$. The class $\mathbf{P}$ is defined as the class $\mathbf{P}_{\omega^\omega}$.

Definition 2.5. We define: (1) $\ell^T_\alpha(x) := \min\{\eta : c^T_{i,\alpha} \in \mathbf{L}_\eta[x] \text{ for } i \in \{1,2,3\}\}$, and (2) $\mathbf{space}(x,T) := \sup\{\ell^T_\xi : \xi \leq \mathbf{time}(x,T)\}$

Definition 2.6. For any ITTM T, we say that T is a **space f machine** if, for all $x \in \mathbb{R}$, we have $\mathbf{space}(x,T) \leq f(x)$. For any ordinal ξ, we say that T is a **space ξ machine** if T is a space f machine for the constant function f such that $f(x) = \xi$ for all x.

Definition 2.7. For any function f, $\mathbf{PSPACE}_f$ denotes the class of all sets of reals decidable by a space f machine. For any ordinal ξ, $\mathbf{PSPACE}_\xi$ denotes the class of all sets of reals that are decidable by a space η machine for some $\eta < \xi$. **PSPACE** is defined as the class $\mathbf{PSPACE}_{\omega^\omega}$.

Furthermore, we define the weak halting problem h and its relativized versions as follows:

Definition 2.8. We let h denote the set $\{e : \phi_e(e) \downarrow 0\}$. Furthermore, for any α, we let h_α denote the set $\{e : \phi_e(e) \downarrow 0 \wedge \mathbf{time}(e,e) \leq \alpha\}$

In the remainder of this paper, we will readily use some of the more standard terminology and definitions in the ITTM literature, such as *clockable* ordinals and *writable* reals and ordinals. We also will refer to a number of basic or easily proven results, a few of which will be listed below:

Lemma 2.9 (Welch)**.** *For any real x, the supremum of x-clockable ordinals γ^x is equal to the supremum of x-writable ordinals λ^x.*

We also have the following result about the weak halting problem h, and its relativized variants h_α:

Proposition 2.10. *For every ordinal α, we have $h_\alpha \notin \mathbf{P}_{\alpha+1}$. Moreover, h is undecidable.*

Regarding the constructible hierarchy $\mathbf{L}$, we have the follwing:

Lemma 2.11. *If $\alpha > \omega$, and $A \subset \omega$, and $A \in \mathbf{L}_\alpha[w]$ for a certain set w, and for a certain set B, $|(A \backslash B) \cup (B \backslash A)| < \omega$, then $B \in \mathbf{L}_\alpha[w]$.*

3 P $\subseteq$ PSPACE for infinite time Turing machines

First of all, we want to be sure that $\mathbf{P}$ is always a subset of **PSPACE**. Due to the somewhat unusual definition of **PSPACE**, however, it is not directly evident that $\mathbf{time}(x,T) \geq \mathbf{space}(x,T)$ for all x, T. It will be shown that this indeed is the case, by showing that we can represent any ITTM-computation of length α, starting from a set w, in level $\mathbf{L}_\alpha[w]$ of the constructible hierarchy. We will first define what we mean with a 'representation' of an ITTM computation:

Definition 3.1. We use the notation $C_\alpha^T(w)$ for a representation of an ITTM computation of a machine T from w in α steps, of the form

$$C_\alpha^T(w) = \{(\beta, q_\beta^T(w), h_\beta^T(w), c_{1,\beta}^T(w), c_{2,\beta}^T(2), c_{3,\beta}^T(w)) : \beta \leq \alpha\}$$

where β is an ordinal representing the stage, and $q_\beta^T(w)$, $h_\beta^T(w)$, $c_{i,\beta}^T(w)$ are as they were defined earlier.

Lemma 3.2. *When we have a suitable representation of Turing machines as finite sets, the notion 'a one step ITTM computation by a machine T from the state $t = (q, h, c_1, c_2, c_3)$ results in a state $t' = (q', h', c_1', c_2', c_3')$ is representable by a formula of first order logic.*

Lemma 3.3. *The notion '$X = C_\alpha^T(w)$: X represents an ITTM computation from w in α steps' is representable in the language of set theory.*

Proof. Note that X represents an ITTM computation from w in α steps, if and only if all of the following five conditions hold: (1) Every element of X is of the form $(\beta, q, h, c_1, c_2, c_3)$ where β is an ordinal smaller than or equal to α; (2) If we have $(\beta, q_1, h_1, c_{1,1}, c_{2,1}, c_{3,1}) \in X$ and $(\beta, q_2, h_2, c_{1,2}, c_{2,2}, c_{3,2}) \in X$, then $q_1 = q_2$, $h_1 = h_2$, $c_{1,1} = c_{1,2}$, $c_{2,1} = c_{2,2}$ and $c_{3,1} = c_{3,2}$; (3) We have $(0, q_\mathrm{s}, h_0, c_{1,0}, c_{2,0}, c_{3,0})$ where q_s is the initial state of the Turing machine, $h_0 = 0$, $c_{1,0} = w$, $c_{2,0} = c_{3,0} = \emptyset$; (4) If β is a successor ordinal $\gamma + 1$, and $\beta < \alpha$, then there are elements $(\gamma, q_1, h_1, c_{1,1}, c_{2,1}, c_{3,1})$ and $(\beta, q_2, h_2, c_{1,2}, c_{2,2}, c_{3,2})$ such that a one step Turing computation from the snapshot $(q_1, h_1, c_{1,1}, c_{2,1}, c_{3,1})$ results in $(q_2, h_2, c_{1,2}, c_{2,2}, c_{3,2})$; (5) If β is a limit ordinal, then there is an element $\beta, q, h, c_1, c_2, c_3$ such that q is the limit state of the Turing machine, $h = 0$, and for each of c_i, we have that $x \in c_i$ if and only if for every ordinal $\gamma < \beta$ there is an ordinal δ greater than or equal to γ and smaller than β such that if $(\delta, q', h', c_1', c_2', c_3') \in X$, then $x \in c_i'$.

It should be clear that all of these notions are representable by a formula of first order logic in the language of set theory. □

Now we will turn to the main theorem. The final aim will be to show, that all computations starting from an input w of length α can be carried out while only having tape contents slightly more complicated than in $\mathbf{L}_\alpha[w]$. The addition of 'slightly' here only signifies that one needs a small finite fixed extra number of steps—the construction made here assumes gives a crude upper bound of 12, although the precise number might be lower than that. First, we need an additional auxiliary lemma:

Lemma 3.4. *If $\beta, q, h, c_1, c_2, c_3$ are all in $\mathbf{L}_\alpha[w]$, then $(\beta, q, h, c_1, c_2, c_3)$ is in $\mathbf{L}_{\alpha+10}[w]$.*

Proof. We assume that (c_2, c_3) is defined as $\{\{c_2\}, \{c_2, c_3\}\}$. If c_2 and c_3 are in $\mathbf{L}_\alpha$, then $\{c_2\}$ and $\{c_2, c_3\}$ are definable subsets of $\mathbf{L}_\alpha$, and are thus in $\mathbf{L}_{\alpha+1}$. Consequently, $\{\{c_2\}, \{c_2, c_3\}\}$ is a definable subset of $\mathbf{L}_{\alpha+1}$, and is hence in $\mathbf{L}_{\alpha+2}$.

Assuming that $(\beta, q, h, c_1, c_2, c_3)$ is defined as $(\beta, (q, (h, (c_1, (c_2, c_3)))))$, it easily follows by repeating the above procedure that $(\beta, q, h, c_1, c_2, c_3)$ is in $\mathbf{L}_{\alpha+10}$. □

Now we will turn to the main theorem, which will be proved using a kind of simultaneous induction:

Theorem 3.5. *The following propositions are true:*

1. *For any infinite ordinal α and any input set $w \subseteq \omega$, we have that $c^T_{1,\alpha}(w)$, $c^T_{2,\alpha}(w)$, $c^T_{3,\alpha}(w)$ are in $\mathbf{L}_{\alpha+1}[w]$. If α is a finite ordinal, however, we can only be sure that the sets are in $\mathbf{L}_{\alpha+2}[w]$.*

2. *If α is a successor ordinal above ω and is equal to $\beta + n$, where n is a natural number, then $c^T_{1,\alpha}(w)$, $c^T_{2,\alpha}(w)$, $c^T_{3,\alpha}(w)$ are in $\mathbf{L}_{\beta+1}[w]$. This is a strengthening of the above property.*

3. *For any ordinal $\alpha \geq \omega$ and any input set $w \subseteq \omega$, we have $(\alpha, q^T_\alpha(w), h^T_\alpha(w), c^T_{1,\alpha}(w), c^T_{2,\alpha}(w), c^T_{3,\alpha}(w)) \in \mathbf{L}_{\alpha+11}[w]$.*

4. *For any ordinal $\alpha \geq \omega$ and any input set $w \subseteq \omega$, we have $C^T_\alpha(w) \in \mathbf{L}_{\alpha+12}[w]$.*

Proof. The tactic here will be to first show that, for any α, if property (1) holds for ordinals smaller than or equal to α, and property (4) holds for all ordinals strictly smaller than α, then properties (2), (3), and (4) hold for α. Then, we will show that if properties (1), (2), (3), and (4) hold for all ordinals strictly smaller than α, then property (1) also holds for α.

- $1 \to 2$: We use here the fact that, assuming $\alpha > \omega$, and $\alpha = \beta + n$, for each i in $\{1, 2, 3\}$, $c^T_{i,\beta}(w)$ and $c^T_{i,\alpha}(w)$ can only differ by finitely many elements. So it follows from $c^T_{i,\beta}(w) \in \mathbf{L}_{\beta+1}[w]$ that $c^T_{i,\alpha}(w) \in \mathbf{L}_{\beta+1}[w]$.

- $1 \to 3$: We have that $\alpha \in \mathbf{L}_{\alpha+1}[w]$ for any w. Because $q^T_\alpha(w)$ and $h^T_\alpha(w)$ are finite sets, they are also in $\mathbf{L}_{\alpha+1}[w]$ by the assumption that α is infinite, and the desired result follows from the assumption of (1) and an application of Lemma 3.4.

- $1\ (+3) \to 4$ for successor ordinals: If α is a successor ordinal $\beta + 1$, then we have that x is in $C^T_\alpha(w)$ if and only if $x \in C^T_\beta(w)$ or if x is equal to $(\alpha, q^T_\alpha(w), h^T_\alpha(w), c^T_{1,\alpha}(w), c^T_{2,\alpha}(w), c^T_{3,\alpha}(w))$. Because by the inductive hypothesis $C^T_\beta(w)$ and hence, by transitivity, every element of it, is in $\mathbf{L}_{\beta+12}[w]$, or $\mathbf{L}_{\alpha+11}[w]$, and because by (3)—which was already proven from (1)—$(\alpha, q^T_\alpha(w), h^t_\alpha(w), c^T_{1,\alpha}(w), c^T_{2,\alpha}(w), c^T_{3,\alpha}(w))$ is in $\mathbf{L}_{\alpha+11}$, it follows that $C^T_\alpha(w)$ is a definable subset of $\mathbf{L}_{\alpha+11}[w]$, and hence an element of $\mathbf{L}_{\alpha+12}[w]$.

- $1\ (+3) \to 4$ for limit ordinals: If α is a limit ordinal, note that, for any $\beta < \alpha$, we have by the inductive hypothesis that $C^T_\beta(w) \in \mathbf{L}_{\beta+12}[w]$, and hence also that $C^T_\beta(w) \in \mathbf{L}_\alpha[w]$. Now consider the following set $C^T_{<\alpha}(w)$:

  $$\{X \in L_\alpha[w] : \exists\beta(\beta \in \alpha \wedge X \in C^T_\beta(w))\}$$

  Note that $\beta \in \alpha$ basically just says, 'β is an ordinal smaller than α'. This set is clearly a set definable from $L_\alpha[w]$, and it is the union of all computations from w in less than α steps. To obtain $C^T_\alpha(w)$, we only need to add $(\alpha, q^T_\alpha(w), h^t_\alpha(w), c^T_{1,\alpha}(w), c^T_{2,\alpha}(w), c^T_{3,\alpha}(w))$ to this set, which can be done easily like in the above case of successor ordinals.

- Property (1) for finite ordinals: note that, at stage n, because the head starts at 0 and can only move forward one step at a time, the head can only be at a location between 0 and n. Because of this, the only positions of $c^T_{i,n}$ that can have changed by stage n are the positions in the range $[0, n-1]$. Because we have $c^T_{0,0} = w$ and $c^T_{1,0} = c^T_{2,0} = \emptyset$, we also have $c^T_{i,0} \in L_1[w]$.

  Now, we can define $c^T_{i,n}$ as $(c^T_{i,0} \backslash \{j_1, \ldots, j_l\}) \cup \{k_1, \ldots, k_m\}$, which is a definable subset of $L_n[w]$ as a result of the fact that all the j and k are not larger than $n-1$ and are thus elements of $L_n[w]$.

- Inductive case for successor ordinals larger than ω: If α is a successor ordinal $\beta + 1$, for each $i \in \{1, 2, 3\}$ $c^T_{i,\alpha}(w)$ and $c^T_{i,\beta}(w)$ can only differ by one element. Taking the largest limit ordinal γ below α, such that $\alpha = \gamma + n$, an appeal to Lemma 2.11 (and possibly to case (2) for β, if β itself is a successor ordinal) suffices to show that for each $i \in \{1, 2, 3\}$, $c^T_{i,\alpha}$ is in $\mathbf{L}_{\gamma+1}[w]$ and hence also in $\mathbf{L}_{\alpha+1}[w]$.

- Inductive case for limit ordinals: if α is a limit ordinal, we have, by inductive assumption, that for all $\beta < \alpha$, $C^T_\beta(w) \in \mathbf{L}_{\beta+12}[w]$, and hence, also that $C^T_\beta(w) \in \mathbf{L}_\alpha[w]$.

  To start, let us find a way of identifying ordinals smaller than α in $\mathbf{L}_\alpha[w]$. If $\alpha \notin \mathbf{L}_\alpha[w]$, the set of ordinals in $L_\alpha[w]$ is exactly the set of ordinals smaller than α: in this case we can identify any ordinal smaller than α in $\mathbf{L}_\alpha[w]$ with the property $\mathrm{Ord}(x)$. If somehow we do have $\alpha \in \mathbf{L}_\alpha[w]$, we can identify any ordinal smaller than α in $L_\alpha[w]$ with the property $x \in \alpha$. Depending on which case we are in, let us take the appropriate property.

  Now note that the definition of cells at limit values implies that $x \in c^T_{i,\alpha}(w)$ if and only if for every ordinal $\beta < \alpha$, there is an ordinal $\gamma < \alpha$ with $\gamma \geq \beta$, such that $x \in c^T_{i\gamma}(w)$. We can now define $c^T_{i,\alpha}(w)$ in the following way:

  $c^T_{i,\alpha}(w)$ is the set of all $n \in \omega$, such that for every ordinal $\beta < \alpha$ there is an ordinal $\gamma < \alpha$ with $\gamma \geq \beta$, such that n is an element of the $i+3$th argument of $C^T_\gamma(w)$ (i.e. $n \in c^T_{i,\gamma}(w)$).

Although it should be noted that we have proved property (3) and (4) only for infinite ordinals, this does not pose a problem, because for any finite n, we have that $C^T_n(w)$ will occur at at least some finite stage of Gödel's Constructible hierarchy. As a result, we will have $C^T_n(w) \in \mathbf{L}_\omega[w]$ for any finite n, and the inductive steps will still work for the ω-case. □

This gives us the following result about **P** and **PSPACE**:

Proposition 3.6. *For any* ITTM T *and any real* x, *if* $\mathbf{time}(x, T) > \omega$, *we have* $\mathbf{space}(x, T) \leq \mathbf{time}(x, T)$. *Also, if* $\mathbf{time}(x, T) \leq \omega$, *we have* $\mathbf{space}(x, T) \leq \omega$. *Hence, for any function* f *such that* $f(x) > \omega$ *for all* x, *every time* f *machine is a space* f *machine, and consequently we have* $\mathbf{P}_f \subseteq \mathbf{PSPACE}_f$.

Proof. Assume that $\alpha > \omega$ and $\alpha = \mathbf{time}(x, T)$. For any $\xi < \alpha$ with $\xi \geq \omega$, we have, by Theorem 3.5, $c^T_{i,\xi}(x) \in \mathbf{L}_{\xi+1}[x]$ (here we have $+1$ instead of $+2$ because $\xi \geq \omega$) for $i \in \{1, 2, 3\}$, and hence $c^T_{i,\xi}(x) \in \mathbf{L}_\alpha[x]$. For finite ξ, we have $c^T_{i,\xi}(x) \in \mathbf{L}_\omega[x]$ for $i \in \{1, 2, 3\}$, and again $c^T_{i,\xi}(x) \in \mathbf{L}_\alpha[x]$. Finally, we notice that α must be equal to $\beta+1$ for some β, as ITTMs cannot halt at limit ordinal stages. It is also immediate from the definition of ITTMs that $c^T_{i,\beta}(x)$ and $c^T_{i,\beta+1}(x)$ can only differ by one element at most. Because of this, it follows from $c^T_{i,\beta}(x) \in \mathbf{L}_\alpha[x]$ that $c^T_{i,\beta+1}(x) \in \mathbf{L}_\alpha[x]$ using Lemma 2.11 and the fact that $\alpha > \omega$. So at all stages of the computations, the content of the tape is inside $\mathbf{L}_\beta[x]$, so we have $\mathbf{space}(x,T) \leq \beta$, and hence $\mathbf{space}(x,T) \leq \mathbf{time}(x,T)$.

If $\alpha \leq \omega$ and $\alpha = \mathbf{time}(x,T)$, we have for all $\xi < \alpha$, by Theorem 3.5, $c^T_{i,\xi}(x) \in \mathbf{L}_\omega[x]$ for $i \in \{1,2,3\}$, and $\mathbf{space}(x,T) \leq \omega$ now follows directly. □

4 The question whether $\mathbf{P}_f = \mathbf{PSPACE}_f$

Now we have seen that, in all cases where the range of f contains only infinite ordinals, $\mathbf{P}_f$ is a subset of $\mathbf{PSPACE}_f$, one is inclined to wonder whether this inclusion can be shown to be proper. In this section, we will show that this, indeed, is the case at least for a number of functions f.

4.1 $\mathbf{P}_\alpha \neq \mathbf{PSPACE}_\alpha$ for ordinals up to ω^2

For the first result, the idea will be to construct a function that can be shown to be $\mathbf{PSPACE}_{\omega+2}$, but that, at the same time, can also be shown to be *not* $\mathbf{P}_\alpha$ for any $\alpha < \omega^2$. This will be done using the notion of arithmetical reals, which are defined more-or-less analogously to arithmetic sets of reals.

First, we observe the following fact about the location of recursive sets in Gödel's constructible hierarchy:

Lemma 4.1. *For any recursive set $w \in \mathbb{R}$, we can write w on the tape in ω steps, and thus we have that $w \in \mathbf{L}_{\omega+1}[\emptyset]$, and thus also $w \in \mathbf{L}_{\omega+1}[x]$ for any $x \in \mathbb{R}$.*

Proof. Because w is recursive, there is a Turing machine T that halts on all inputs $n \in \mathbb{N}$, such that $\phi_T(n) = 1$ if and only if $n \in w$. Now consider an ITTM that, in turn, for each number n, simulates the machine T, and then writes the correct number to the nth cell of the output tape. Because every individual computation will terminate in a finite number of steps, this can be done in ω steps, and we can go directly from the limit state to the halting state. □

We will now consider the notion of arithmetical sets of natural numbers: a set $S \subseteq \mathbb{N}$ is arithmetical if and only if there is a formula ϕ of PA such that $\phi(x) \iff x \in N$.

Lemma 4.2. *The set $A := \{x \in \mathbb{R} : x$ is arithmetical$\}$, is not arithmetical.*

Proof. This is shown in Example 13.1.9 in [Co]. □

This gives us the following result:

Lemma 4.3. *For any ordinal $\alpha \leq \omega^2$, the set $A := \{x \in \mathbb{R} : x$ is arithmetical$\}$ is not in $\mathbf{P}_\alpha$.*

Proof. By Theorem 2.6 in [HaLe], the arithmetic sets are exactly the sets which can be decided in time $\omega \cdot n$ for some $n \in \mathbb{N}$. Because A is not arithmetical, it cannot be decided by any algorithm using a bounded finite number of limits, and hence, it is not in $\mathbf{P}_\alpha$ for any $\alpha \leq \omega^2$. □

It turns out, however, that this set A is in $\mathbf{PSPACE}_{\omega+2}$:

Theorem 4.4. *The set $A := \{x \in \mathbb{R} : x$ is arithmetical$\}$ is in $\mathbf{PSPACE}_{\omega+2}$.*

Proof. We will make use of an ITTM with three scratch tapes (see [Wi, Proposition 4.28] for a justification why this can be done without affecting the space complexity): the first is used to enumerate over all possible formulas, the second is used to enumerate over, and store the choices of the quantifiers occurring in each formula, and the third is used for the actual evaluation of the resulting, quantifier-free, formulas.

First, we can note that we can, without any problems, enumerate over all possible formulae that determine arithmetic sets, on one of the scratch tapes: every formula can be coded by a set which is finite, and hence in $\mathbf{PSPACE}_{\omega}$. Now, we may also assume that this enumeration only gives formulae in a normal form, with all quantifiers at the front. By enumerating over the formulas in such a way that they are enumerated over by increasing length, we can be sure that no infinite (and hence, no nonrecursive) set will appear on the tape before the very end of the computation; by switching all cells with content 0 to 1 and back during the enumeration process, we can furthermore ensure that the tape, at the very end, will be filled with 1s, and hence again contain a recursive set.

Given a formula ϕ, we can also determine for any x whether $\phi(x)$ holds, by only writing down finite sets on a scratch tape. That this is the case can be shown by induction: if ϕ is quantifier free, we can simply evaluate the formula; if ϕ has an existential quantifier at the front and is of the form $\exists x_1 \psi$, we can write down every possible value n for x_1 on the second scratch tape (right after any earlier quantifiers that may have been written there), coded by n 1s followed by a single 0, recursively evaluate $\psi[n/x_1]$ (which is possible by the inductive assumption) using the rest of the second, as well as the third scratch tape; succeed if this is successful, and fail otherwise; if ϕ has a universal quantifier at the front and is of the form $\forall x_1 \phi$, we again write down every possible value n for x_1 at the front of the scratch tape, recursively evaluate $\psi[n/x_1]$, and now fail if this fails, and continue otherwise.

At limit stages during this process, we will, after a fixed prefix of sequences of 1s followed by a single 0, end up with an infinite sequence of 1s on the second scratch tape: this is a set containing only finitely many 0s, and is, hence, clearly recursive. During the actual evaluation of formulas, on the third tape, we can switch every cell that is accessed and that has value 0 to 1 and back again: this way, we also easily prevent the accidental occurrence of nonrecursive sets on the third scratch tape.

Now, the strategy for the whole function will be:

1. enumerate over all possible formulae in the aforementioned normal form;
2. for each such formula, test for each x whether the formula holds for x if and only if the xth position on the input tape is a 1;
3. succeed if we have found such a formula, and fail otherwise.

This can, as a result of some of the precautions, be done while writing out only recursive sets. □

This gives us the following result on $\mathbf{P}$ and $\mathbf{PSPACE}$:

Theorem 4.5. *For any α such that $\alpha \geq \omega + 2$ and $\alpha \leq \omega^2$, we have that* $\mathbf{P}_\alpha \subsetneq \mathbf{PSPACE}_\alpha$. *Also,* $\mathbf{PSPACE}_{\omega+2}$ *contains elements that are not in* $\mathbf{P}_\alpha$.

Proof. This follows directly from Theorem 4.3 and Theorem 4.4. □

4.2 The case of recursive ordinals

We will now show that the inequality $\mathbf{P}_\alpha \neq \mathbf{PSPACE}_\alpha$ holds for much wider range of ordinals. The strategy in showing this will be to show, that, for all recursive ordinals, $h_\alpha \in \mathbf{PSPACE}_{\alpha+1}$ which, together with the already known fact that $h_\alpha \notin \mathbf{P}_{\alpha+1}$, gives the desired result.

From Proposition 2.10, we know that for all α, $h_\alpha \notin \mathbf{P}_{\alpha+1}$. It turns out, however, that for all recursive ordinals α, we do have that $h_\alpha \in \mathbf{PSPACE}_{\alpha+1}$:

Proposition 4.6. *For any recursive ordinal α, such that $\alpha \geq \omega + 1$, we have* $h_\alpha \in \mathbf{PSPACE}_{\alpha+1}$.

Proof. We can compute h_α in the following way, by making use of an ITTM with several scratch tapes: to start, we will check whether the input corresponds to a natural number; if it does not, then we output 0, and if it does, we continue. Then, we will look if this natural number corresponds to a coding of an ITTM_0—again, we output 0 if this is not the case, and if it is, we continue. Note that the computation so far can be performed by writing only finite sets on the scratch tapes.

Now the real work can begin. First we write down the ordinal α, on the first scratch tape. Because α is recursive, we can do this in ω steps, so we are sure that the content of the first scratch tape, at all times, will be inside $\mathbf{L}_{\omega+1}[x]$ and, because $\alpha \geq \omega + 1$, also inside $\mathbf{L}_\alpha[x]$.

Once this is done, we will check for all ordinals β smaller than α (which can be easily found simply by restricting the ordinal written on the tape to all the elements smaller than a certain element, without affecting the space complexity in any way), whether the eth ITTM_0 has reached the halt state by stage β. If it turns out that this is the case, we look at the output: if the output is 0, then we will finish the computation by writing 1 on the output tape, and otherwise we finish by writing 0 on the output tape. If we reach stage β during the simulation without having halted, we go on checking with the next ordinal. Finally, if we have exhausted all ordinals $\beta < \alpha$, we again finish by writing 0 on the output tape.

In the case where α is a limit ordinal, we are now done, because we know that no machine can halt at any limit ordinal stage. In the case where α is a successor ordinal $\eta + 1$, however, we will additionally check whether the computation has finished at α itself.

For any $\beta < \alpha$, we know that $c^T_{i,\beta}[x] \in \mathbf{L}_\alpha[x]$ for $i \in \{1, 2, 3\}$. Furthermore, in the case where α is a successor ordinal $\eta + 1$, we know that $c^T_{i,\eta}[x]$ and $c^T_{i,\alpha}[x]$ can only differ by one element at most, and because α and η are known to be infinite, we obtain $\mathbf{L}_\alpha[x]$ from $\mathbf{L}_\eta[x]$ using Lemma 2.11.

It follows that this computation never writes a set on any of the tapes that is not in $\mathbf{L}_\alpha[x]$. This proves that $h_\alpha \in \mathbf{PSPACE}_{\alpha+1}$ indeed holds. □

Hence we have:

Theorem 4.7. *For every recursive successor ordinal $\alpha \geq \omega+1$, $\mathbf{P}_\alpha \subsetneq \mathbf{PSPACE}_\alpha$ holds.*

Unfortunately, however, this process cannot be easily extended to work for limit ordinals: given a limit ordinal α, we can still compute h_α by only writing out information of space complexity less than α on tape, but it seems hard, if not impossible, to do this bounded by a specific ordinal β below α.

4.3 The case of clockable ordinals

It is, however, possible, to extend the above process to many writable successor ordinals. The strategy here is essentially the same as in the case of recursive ordinals: we know that h_α cannot be in $\mathbf{P}_{\alpha+1}$, and then we show that $h_\alpha \in \mathbf{PSPACE}_{\alpha+1}$. In the case of clockable ordinals, we can make use of the following theorem due to Philip Welch (Lemma 15 in [DeHaSc]):

Theorem 4.8. *If α is a clockable ordinal, then every ordinal up to the next admissible beyond α is writable in time $\alpha + \omega$.*

Besides this, we already know that $\alpha \in \mathbf{L}_{\alpha+1}[0]$. However, we will also have $\alpha \cup \{\{\{\{\emptyset\}\}, \eta\} : \eta \leq \beta\} \in \mathbf{L}_{\alpha+1}[0]$ for any $\beta \leq \omega$. Because these sets $\{\{\{\emptyset\}\}, \eta\}$ are not ordinals, it is immediate that α and $\{\{\{\{\emptyset\}\}, \eta\} : \eta \leq \beta\}$ are always disjoint. Thus, we can consider these sets $\alpha \cup \{\{\{\{\emptyset\}\}, \eta\} : \eta \leq \beta\}$ as alternative representations for the ordinals up to $\alpha+\omega$, that are within $\mathbf{L}_{\alpha+1}[0]$. This way, using this 'alternative' representation of the ordinals between α and $\alpha+\omega$, instead of the regular ones, in combination with the fact that the snapshots $c^T_{i,\alpha+n}[x]$ can only differ from $c^T_{i,\alpha}[x]$ by finitely many elements, we can represent computations of length $\alpha+\omega$ from x within $\mathbf{L}_{\alpha+2}[x]$ using a construction similar to that in section 5.1.

This gives us that, for clockable α, every ordinal up to the next admissible ordinal beyond α can be written on the tape with the content of the tape inside $\mathbf{L}_{\alpha+2}[0]$ at all stages. This gives us the following theorem:

Theorem 4.9. *If β is a clockable ordinal, and α is a successor ordinal between $\beta + 3$ and the next admissible after β, then we have $\mathbf{P}_\alpha \subsetneq \mathbf{PSPACE}_\alpha$.*

Proof. This goes largely analogous to the case of Proposition 4.6, with the major difference that we can now write α on the tape while staying inside $\mathbf{L}_{\beta+2}[0]$. As a result from this, if $\alpha = \eta + 1$, we get from $\eta \geq \beta + 2$ that $h_\eta \in \mathbf{PSPACE}_{\eta+1}$, whereas $h_\eta \notin \mathbf{P}_{\eta+1}$, giving the desired result. □

4.4 The case of suitable functions f

We can extend the above results from ordinals to suitable functions as defined in [DeHaSc]. There, a suitable function is defined as follows:

Definition 4.10. A function or a function-like operation f from $\mathbb{R}$ to the ordinals is called *suitable* whenever, for all reals x and y, $x \leq_{\mathrm{T}} y$ implies $f(x) \leq f(y)$, and if we have, for all x, $f(x) \geq \omega + 1$. The symbol $\leq_{\mathrm{T}}$ here stands for ordinary Turing reducibility as defined in e.g. [Co].

It turns out that some important functions and operations are in fact suitable:

Proposition 4.11. *For any ordinal $\alpha > \omega + 1$, the constant function $f_0(x) = \alpha$ is suitable. Furthermore, the functions λ^x, and ζ^x, and Σ^x are all suitable.*

Proof. For constant functions, we have $f_0(x) = f_0(y)$, and hence $f_0(x) \leq f_0(y)$ in all cases, so also in the specific case where $x \leq_{\mathrm{T}} y$. Hence, all constant functions are suitable.

For the function λ^x, assume that $x \leq_{\mathrm{T}} y$, and assume that $\alpha < \lambda^x$, or, in other words, that α is a x-writable ordinal. We can now write α from y, by first computing x from y, and then going on with the computation that writes α from x. Hence, $\alpha < \lambda^y$, and $\lambda^x \leq \lambda^y$ follows directly. The cases of the functions ζ^x and Σ^x go very similarly. □

We have the following results about suitable functions:

Theorem 4.12. *For any suitable function f and any set A of natural numbers,*

(i) $A \in \mathbf{P}_f$ *if and only if* $A \in \mathbf{P}_{f(0)+1}$

(ii) $A \in \mathbf{PSPACE}_f$ *if and only if* $A \in \mathbf{PSPACE}_{f(0)+1}$

Proof. (i) was originally proven in [DeHaSc, Theorem 26]; here we will provide a slightly modified version of the proof. Because for any natural number n, we have $0 =_{\mathrm{T}} n$, we get $f(0) = f(n)$ by the assumption of suitability; also, we have $0 =_{\mathrm{T}} \mathbb{N}$, so $f(0) = f(\mathbb{N})$; moreover, for any real number x, we have $0 \leq_T x$, and hence $f(0) \leq f(x)$. Now consider the constant function g such that $g(x) = f(0)$ for all x. As a direct result of the definition, we obtain $\mathbf{P}_{f(0)+1} = \mathbf{P}_g$. We also have $g(x) \leq f(x)$ for all x, so the result $\mathbf{P}_g \subseteq \mathbf{P}_f$ is immediate.

For the converse, assume that $A \in \mathbf{P}_f$, and that T is a time f machine deciding A. We now construct a time g machine T', which performs the same computation as T, while, during the first ω steps of the computation, simultaneously checking if the input actually codes a natural number or the entire set $\mathbb{N}$. Because a natural number n, when considered as a real, is equal to the set $\{0, \dots, n-1\}$, and any such set corresponds to a natural number n, it follows that a real x does *not* correspond to a natural number or the complete set $\mathbb{N}$ if and only if the string 01 occurs in it. We now ensure that T', during the first ω steps, searches for the string 01, and halts whenever this string is encountered, while simultaneously simulating T. After we first reach the limit state, we know that the input did not contain the string 01, and we continue the original computation of T. If no 01 is encountered, the input x must be either a natural number n, or the complete set of natural numbers $\mathbb{N}$, and it will finish within time $f(x) = f(0)$. If a 01 is encountered in the input x, it is encountered within the first ω steps, and hence $f(x) < \omega < f(0)$. So T' is a time $f(0)$-machine deciding A, and hence, $A \in \mathbf{P}_{f(0)+1}$.

(ii) can be proven similarly. It again follows directly that $\mathbf{PSPACE}_{f(0)+1} \subseteq \mathbf{PSPACE}_f$. The converse now is a bit simpler. If T is a space f machine deciding A, consider the following machine T': on input x, we first check, without making any modifications (and thus, while staying within $\mathbf{L}_0[x]$), whether x is a natural number. We output 0 if it does not, and if it does, we continue the computation of which we now know that $\mathbf{space}(x, T) < f(x) = f(0)$. Because at the start of this computation, the tape is still unchanged, and the algorithm performed after the check if x is a natural number, is identical, we also obtain $\mathbf{space}(x, T) < f(0)$. It is now clear that $\mathbf{space}(x, T') < f(0)$ for all x, so T' is a space $f(0)$-machine deciding A, so $A \in \mathbf{PSPACE}_{f(0)+1}$. □

Now, because for any α, the set h_α always consists of only natural numbers, we can directly extend the earlier results about ordinals to results about suitable functions f:

Theorem 4.13. *If f is a suitable function, and $f(0)$ is either recursive, or a clockable ordinal, such that there is an ordinal β such that $f(0)$ is between $\beta+2$ and the next admissible ordinal after β, then we have $\mathbf{P}_f \subsetneq \mathbf{PSPACE}_f$.*

Proof. On one hand, we have $h_{f(0)+1} \notin \mathbf{P}_{f(0)+2}$ from Proposition 2.10, which, by Theorem 4.12 gives us $h_{f(0)+1} \notin \mathbf{P}_{f+1}$. On the other hand, we have $h_{f(0)+1} \in \mathbf{PSPACE}_{f(0)+2}$ from either Proposition 4.6 or Proposition 4.9, which gives us $h_{f(0)+1} \in \mathbf{PSPACE}_{f+1}$. Hence we have $\mathbf{P}_f \neq \mathbf{PSPACE}_f$, and the result follows. □

4.5 However, $\mathbf{P}_f = \mathbf{PSPACE}_f$ for 'almost all' f

So far, we have shown that, for certain classes of 'low' functions f and ordinals α, there is a strict inclusion $\mathbf{P}_f \subset \mathbf{PSPACE}_f$. This brings us to wonder we can also find functions and ordinals where this strict inclusion does not hold, and instead we have an equality $\mathbf{P}_f = \mathbf{PSPACE}_f$.

It is easy to see that we will have this equality, at least for very high, non-countable, ordinals and functions: if we have $\alpha > \omega_1$, then we must have $\mathbf{P}_\alpha = \mathbf{Dec}$, and because we also have $\mathbf{PSPACE}_\alpha \subseteq \mathbf{Dec}$ and $\mathbf{P}_\alpha \subseteq \mathbf{PSPACE}_\alpha$, we indeed obtain $\mathbf{P}_\alpha = \mathbf{PSPACE}_\alpha$. However, we can generalize this towards a wider range of functions:

Proposition 4.14. *If f satisfies $f(x) \geq \lambda^x$ for all x, then we have $\mathbf{P}_f = \mathbf{PSPACE}_f$.*

Proof. We clearly have $\mathbf{P}_f = \mathbf{Dec}$ because f is, on all x, larger than the supremum of halting times of ITTM computable functions on input x, as a result of the fact that $\gamma^x = \lambda^x$ for all x. Also, we have $\mathbf{P}_f \subseteq \mathbf{PSPACE}_f$, as well as $\mathbf{PSPACE}_f \subseteq \mathbf{Dec}$. Hence, we have $\mathbf{P}_f = \mathbf{Dec} = \mathbf{PSPACE}_f$. □

As this class of functions f is a superset of the class of functions f—called 'almost all f' there—for which it is shown, in [HaWe], that $\mathbf{P}_f \neq \mathbf{NP}_f$, we can, with a little wink, indeed say that we have $\mathbf{P}_f = \mathbf{PSPACE}_f$ for 'almost all' f.

References

[Co] S. Barry Cooper, *Computability Theory*, Chapman & Hall/CRC, 2004

[DeHaSc] Vinay Deolalikar, Joel David Hamkins, Ralf-Dieter Schindler, $\mathbf{P} \neq \mathbf{NP} \cap \text{co}-\mathbf{NP}$ *for infinite time Turing machines*, Journal of Logic and Computation 15 (2005), pp. 577-592

[HaLe] Joel David Hamkins and Andy Lewis: *infinite time Turing machines*, Journal Of Symbolic Logic 65 (2000), pp. 567-604

[HaWe] Joel David Hamkins and Philip D. Welch, $\mathbf{P}_f \neq \mathbf{NP}_f$ *for almost all f*, Mathematical Logic Quarterly, 49(2003) 5, pp. 536-540

[Lö] Benedikt Löwe: *Space bounds for infinitary computation*, in Arnold Beckmann, Ulrich Berger, Benedikt Löwe, John V. Tucker (eds.), *Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, CiE 2006, Swansea, UK, July 2006, Proceedings*, Springer-Verlag, Berlin [Lecture Notes in Computer Science 3988] (2006), pp. 319-329

[Sc] Ralf Schindler, $\mathbf{P} \neq \mathbf{NP}$ *for infinite time Turing machines*, Monatshefte der Mathematik 139 (2003), pp. 335-340

[Wi] Joost Winter: *Space Complexity in infinite time Turing Machines*, ILLC Scientific Publications, MoL-2007-14, 2007