Every Polynomial-Time 1-Degree Collapses iff P = PSPACE[*]

Stephen A. Fenner
University of Southern Maine

Stuart A. Kurtz
University of Chicago

James S. Royer
Syracuse University

August 16, 1996

[*] A version of this paper is to be presented at the 1989 IEEE Foundations of Computer Science Conference.

Abstract

A set A is *m-reducible* (or Karp-reducible) to B iff there is a polynomial-time computable function f such that, for all x, $x \in A \iff f(x) \in B$. Two sets are:

- *1-equivalent* iff each is m-reducible to the other by one-one reductions;
- *p-invertible equivalent* iff each is m-reducible to the other by one-one, polynomial-time invertible reductions; and
- *p-isomorphic* iff there is an m-reduction from one set to the other that is one-one, onto, and polynomial-time invertible.

In this paper we show the following characterization.

Theorem *The following are equivalent:*

(a) P = PSPACE.

(b) Every two 1-equivalent sets are p-isomorphic.

(c) Every two p-invertible equivalent sets are p-isomorphic.

1. Overview

If A is m-reducible to B, we usually interpret this to mean that A is computationally no more difficult than B, since a procedure for computing B is easily converted into a procedure for computing A of comparable complexity. In fact, this interpretation is supported by much a weaker reduction: A polynomial-time Turing reducible to B suffices. Therefore the existence of an m-reduction from A to B implies a stronger relationship between A and B than the conventional interpretation suggests. This possibility of gaining additional information about the relationship between A and B gains interest from the frequency with which proofs of m-reducibility are obtained. Indeed, the reducibilities obtained in practice are usually stronger still: they are almost always honest,[1] usually length-increasing, and frequently one-one. We hope and expect to get additional *useful* information from the strength of these reducibilities. For example, it is known that the class of m-complete sets for deterministic exponential-time are pairwise one-one, length-increasing equivalent [Ber77].

In a seminal paper, Berman and Hartmanis [BH77] conjectured that the m-complete sets for NP are pairwise p-isomorphic, that is, that the complete m-degree[2] of NP collapses to a p-isomorphism type. It is easy to prove that there are m-equivalent sets that fail to be 1-equivalent, let alone p-isomorphic. Thus, the specific *location* of the Berman-Hartmanis conjecture is critical. However, if one considers strengthenings of m-reducibility, e.g., 1-reducibility and 1-honest-reducibility, then until a few years ago there were no known examples of degrees of these sorts of reducibilities that failed to collapse. The first important result in this area was Ko, Long, and Du's [KLD87]

[1] Suppose $f, h :: \omega \to \omega$. We say that f is *h-honest* if and only if, for all x, $h(|f(x)|) \geq |x|$, and we say that f is *honest* if and only if for some polynomial p, f is p-honest.

[2] Reducibilities relate the hardness of sets. Hence, an equivalence class of sets with respect to a reducibility relation consists of sets of the same "degree" of difficulty. We thus define a *degree* to be an equivalence class under a reducibility. So, we speak of m-degrees, 1-degrees, 1-honest-degrees and 1-li-degrees according to the reducibility intended. The term "degree" comes from Post's [Pos44]—the paper that founded modern recursion theory.

theorem that every 1-li-degree collapses if and only if (as seems unlikely) P = UP. In this paper, we show that the statements that (a) every 1-degree collapses and (b) every p-invertible degree collapses are both equivalent to (c) P = PSPACE. In retrospect, the most remarkable aspect of our results is the equivalence of (a) and (b) which we still find counterintuitive.

1.1. Related Work

Myhill [Myh55] showed

Myhill's Theorem *Every two recursively 1-equivalent sets are recursively isomorphic.*

This deep and fundamental result implies that recursive 1-equivalence is much tighter than might initially be expected: if two sets are so similar that they are recursively 1-equivalent, then they are recursively identical. There are a number of complexity theoretic version of Myhill's Theorem. Dowd [Dow82] has perhaps the strongest complexity theoretic, exact analog of Myhill's Theorem.

Dowd's Theorem *Every two strictly linear-space 1-equivalent*[3] *sets are strictly linear-space isomorphic.*

In the theory of polynomial-time reducibilities the closest known analog to Myhill's Theorem is due to Berman and Hartmanis.

Theorem 1 ([BH77]). *If two sets are m-equivalent as witnessed by reductions that are (a) one-one, (b) length-increasing, and (c) p-invertible, then the sets are p-isomorphic.*

The hypothesis that the reductions be one-one is clearly necessary, however, the length-increasing and the p-invertibility hypotheses seem quite

[3] We say that a function f is *strictly* DSPACE(t) *computable* if and only if f is computable by a deterministic TM that runs within an $\mathcal{O}(t(n))$ space bound on the work tapes *and* the input and output tapes.

strong, perhaps unnecessarily strong. An obvious question is whether either of these hypotheses can be weakened. Ko, Long, and Du showed that under the hypothesis that P $\neq$ UP (i.e., one-way functions exist [Ber77] [GS84] [GS88][Ko85]), the p-invertibility hypothesis is indeed necessary.

Theorem 2 ([KLD87]). *If* P $\neq$ UP, *then there are 1-li equivalent*[4] *sets that fail to be p-isomorphic.*[5]

This is a remarkable result. 1-li equivalence is a very strong equivalence, but this theorem says that under the reasonable hypothesis of P $\neq$ UP, 1-li degrees are distinct from p-isomorphism types. The theorem's P $\neq$ UP hypothesis is tight. A simple argument shows

Proposition 3. *If* P = UP, *then every two 1-li equivalent sets are p-isomorphic.*

Thus, Theorem 2 and Proposition 3 yield the following striking characterization.

Corollary 4. P = UP *iff every two 1-li equivalent sets are p-isomorphic.*

One of the reasons this corollary is so striking is that it gives a complexity characterization of a degree-theoretic property. Thus, this corollary essentially settles the question whether every 1-li degree collapses.

1.2. Our Results

We establish analogs of both Theorem 2 and Proposition 3 for 1-reductions and p-invertible reductions. We first consider our analogs of Theorem 2. We show

Theorem 5. *If* P $\neq$ PSPACE, *then there are 1-equivalent sets that fail to be honest m-equivalent.*

[4] That is, equivalent under one-one, length-increasing m-reductions.

[5] Moreover, there are such sets that are 2-tt complete for the class of deterministic exponential-time decidable sets.

Theorem 6. *If* P $\neq$ PSPACE, *then there are p-invertible equivalent sets that fail to be p-isomorphic.*[6]

Two sets that are p-invertible equivalent have exceedingly similar structure. It is very surprising (at least to us) that under as weak a hypothesis as P $\neq$ PSPACE, this very strong equivalence fails to imply p-isomorphism. Theorem 6 indicates that under the assumption that P $\neq$ PSPACE, the length-increasing hypothesis of Berman and Hartmanis's Theorem 1 is close to tight. [7]

To establish an analog of Proposition 3, we first show a version of Dowd's Theorem for strictly polynomial-space reductions.

Theorem 7. *Every two strictly polynomial-space 1-equivalent sets are strictly polynomial-space isomorphic.*

Now, using Theorem 7 it is straightforward to show

Theorem 8. *If* P = PSPACE, *then every two 1-equivalent sets are p-isomorphic.*

Therefore, by combining Theorems 5, 6, and 8 we obtain our main result:

Theorem 9. *The following are equivalent:*

(a) P = PSPACE.

(b) Every two 1-equivalent sets are p-isomorphic.

(c) Every two p-invertible equivalent sets are p-isomorphic.

[6] For both Theorems 5 and 6 the witnessing sets can be constructed to be 2-tt complete for exponential-time.

[7] Note that Theorem 6 does not rule out the possibility that "length-nondecreasing" can replace "length-increasing" in the hypothesis of Theorem 1. We suspect that under a stronger condition than P $\neq$ PSPACE , the length-increasing hypothesis of Theorem 1 is indeed necessary.

2. Technical Details

We say that f is *honestly-invertible* iff the function

$$\lambda x, n. \begin{cases} f^{-1}(x), & \text{if } f^{-1}(x) \text{ is defined and of length} \leq n; \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

is computable in time polynomial in $n + |x|$. For example,

$$\lambda x. \begin{cases} 2n, & \text{if } x \text{ is a power of 2 and } x = 2^n; \\ 2x+1, & \text{otherwise;} \end{cases}$$

is not p-invertible, but it is honestly-invertible. On the other hand, a one-way function is neither p-invertible nor honestly-invertible.

Let $\langle \varphi_i \rangle_{i \in N}$ be an acceptable numbering of the partial recursive functions [Rog67] based on a coding of deterministic, multi-tape Turing machines. By standard results in the literature there is a function

$$T \;=\; \lambda i, x, n. \begin{cases} \varphi_i(x), & \text{if Turing machine } i \text{ on input } x \text{ halts} \\ & \quad \text{within } n \text{ steps;} \\ 0, & \text{otherwise} \end{cases}$$

that is computable in $\mathcal{O}((|i| + |x| + n)^2)$ time. (T is essentially Kleene's T predicate.) For each i, let $\widetilde{\psi}_i = \lambda x. [\, T(j, x, (|x|+2)^{|k|}), \text{ where } i = \langle j, k \rangle \,]$. It follows that $\langle \widetilde{\psi}_i \rangle_{i \in N}$ is an enumeration of the polynomial-time computable functions such that $\lambda i, x. \widetilde{\psi}_i(x)$ is computable in $2^{\mathcal{O}(|i|+|x|)}$ time.

In this section we sketch the proofs of Theorems 5 and 7. One of the attractive features of these proofs this that they are naturally set in the common context of the Cantor-Bernstein Theorem.[8] The constructions for Myhill's and Dowd's Theorems, Theorem 1, and Theorem 7. are all effective variants of the standard construction for Cantor-Bernstein. The proofs of Ko, Long, and Du's Theorem and our Theorems 5 and 6 establish that certain plausible effective forms of Cantor-Bernstein fail.

Notation and Conventions[9] In the following ω denotes the set of natural numbers. We identify each number with its dyadic representation over

[8] This theorem states that if there is a one-one map from set A to set B and a one-one map from B to A, then there is a one-one correspondence between A and B.

[9] For any unexplained notation or terminology in the following, see [KMR88].

$\{\mathbf{0},\mathbf{1}\}$. Let $\langle\cdot,\cdot\rangle$ denote polynomial-time computable and invertible pairing function—the one in [Rog67] will do. Let ω' denote a disjoint copy of ω. For each $x \in \omega$, x' denotes the corresponding element of ω'. We assume the ordering $0 < 0' < 1 < 1' < 2 < 2' < \cdots$ on $(\omega \cup \omega')$.

Now, suppose that $A \subseteq \omega$, $B \subseteq \omega'$, $f{:}\,\omega \to \omega'$ recursively 1-reduces A to B, and $g{:}\,\omega' \to \omega$ recursively 1-reduces B to A. We introduce the directed graph $G = (\omega \cup \omega', E)$, where

$$E \;=\; \{\,(x, f(x)) : x \in \omega\,\} \;\cup\; \{\,(x', g(x')) : x' \in \omega'\,\}\,.$$

G is clearly bipartite. Since f and g are functions, every vertex of G has out-degree one. Since f and g are one-one, every vertex of G has in-degree of at most one.

The maximal connected components of G we call *f,g-chains* or simply *chains* when f and g are understood. A *root* of a chain C is a vertex in C with in-degree zero. Each chain C is a directed path and has one of four possible structures:

a. a finite cyclic path;

b. a two-way infinite path;

c. an infinite path with a root in ω; or

d. an infinite path with a root in ω'.

We say that a function $h{:}\,\omega \to \omega'$ *respects chains* iff for all x, x and $h(x)$ belong to the same chain. Since f and g recursively 1-reduce A to B and B to A, respectively, it follows that for any h that respects chains we have that, for all x, $x \in A \iff h(x) \in B$. We say that a function $h{:}\,\omega \to \omega'$ *crosses a chain* C iff for some x, an ω-vertex of C, $h(x)$ is not an ω'-vertex of C.

3. Isomorphisms

The constructions for Theorem 7 and Dowd's Theorem are space-bounded versions of the construction for Myhill's Theorem. Below we sketch a proof of Myhill's Theorem followed by a proof of our Theorem 7. First we note that for f and g as in the beginning of Section 2, the standard construction for the Cantor-Bernstein theorem defines

$$\pi = \lambda x . \begin{cases} g^{-1}(x), & \text{if } x\text{'s chain is } \omega' \text{ rooted;} \\ f(x), & \text{otherwise.} \end{cases} \tag{1}$$

A simple argument shows that π is one-one and onto. Moreover, since π respects chains, we have for each x that $x \in A \iff \pi(x) \in B$. A problem with this construction is that π may not be computable even though f and g are.

In order to prove that various NP-complete sets are p-isomorphic, Berman and Hartmanis [BH77] recycle the Cantor-Bernstein construction by finding conditions on f and g so that the function π defined in (1) is computable and invertible in polynomial time. They proved the following theorem.

Theorem 1 *If two sets are m-equivalent as witnessed by reductions that are (a) one-one, (b) length-increasing, and (c) p-invertible, then the sets are p-isomorphic.*

Proof Suppose that f and g satisfy hypotheses (a), (b), and (c). Let π be as in (1). So, π is an isomorphism between A and B. Fix a $z \in (\omega \cup \omega')$. Since f and g are length increasing, we have that each chain is rooted and that there are at most $|z|$ many vertices preceding z in its chain and all of these vertices are of length less than $|z|$. Since f and g are p-invertible, it follows that one can find the root of a vertex z's chain in polynomial (in $|z|$) time. Therefore, π is polynomial-time computable. □

Myhill [Myh55] showed that if f and g are recursive, then a recursive bijection π exists that respects chains, although now π is of necessity defined quite differently than in (1) above. Our proof of Theorem 7 is in the same vein, but in addition we must observe space bounds on the isomorphism we are building, and thus our construction is considerably more delicate.

Theorem 10 ([Myh55]). *Every two recursively 1-equivalent sets are recursively isomorphic.*

Proof Sketch The definition of π in (1) is based on a global analysis of the structure of chains. The construction for this theorem is more local in character. Given recursive f and g as above, we build in stages $\hat{\pi}$, a recursive isomorphism that respects chains. Initially, $\hat{\pi} = \emptyset$. During stage $2x$, if $\hat{\pi}(x)$ is not yet defined, then x's chain is traversed forward and $\hat{\pi}(x)$ is defined to be the first ω'-vertex encountered that is not yet in the range of $\hat{\pi}$. During stage $2x+1$, if $\hat{\pi}^{-1}(x')$ is not yet defined, then x''s chain is traversed forward and $\hat{\pi}^{-1}(x')$ is defined to be the first ω-vertex encountered that is not yet in the domain of $\hat{\pi}$. A straightforward argument shows that $\hat{\pi}$ is a recursive isomorphism between A and B. □ **Theorem 10**

Theorem 7 *Every two strictly polynomial-space 1-equivalent sets are strictly polynomial-space isomorphic.*

Proof Suppose f and g are 1–1 strictly polynomial-space computable functions. Below we describe the construction of $\tilde{\pi}$, a strictly polynomial-space computable isomorphism that respects f, g-chains. In the construction of Theorem 10 above, although the root of a given f, g-chain is inaccessible in general, one can traverse the chain forward an unlimited amount to find an unmatched vertex, obviating the need to search the chain backwards. In the construction below, our view of each f, g-chain is more myopic; at each stage we can only see (and match vertices in) a portion of the chain residing below a certain length bound. We cannot follow a chain forward indefinitely, so we must search backwards along the chain to ensure that each of its vertices get matched with a vertex of roughly the same length.

Let G be as in the beginning of Section 2. For each n, define:

$$\omega_n \;=\; \{\, x \in \omega : |x| \leq n \,\}. \qquad \omega'_n \;=\; \{\, x' \in \omega' : |x| \leq n \,\}.$$

For each n, let G_n be the subgraph of G induced by $(\omega_n \cup \omega'_n)$. The maximal connected components of G_n we call *n-chains.* The successive vertices of a path in G alternate between being in ω and ω'. Hence, a finite path P in G (such as an n-chain) has one of the following three possible structures.

Unbiased: The number of ω-vertices in P is the same as the number of ω'-vertices. In this case P is either cyclic or else has one of its ends in ω and the other in ω'.

ω-biased: The number of ω-vertices in P is one more than the number of ω'-vertices. In this case P's root and tail vertices are in ω.

ω'-biased: The number of ω-vertices in P is one less than the number of ω'-vertices. In this case P's root and tail vertices are in ω'.

We say a partial function $h\colon \omega_n \to \omega'_n$ *respects n-chains* if and only if, for each $x \in \mathrm{domain}(h)$, $h(x)$ is in the same n-chain as x.

Our construction of $\tilde{\pi}$ will be in stages. For each n, $\tilde{\pi}_n\colon \omega_n \to \omega'_n$ will be the part of $\tilde{\pi}$ defined as of the end of stage n. ($\tilde{\pi}_{-1} = \emptyset$.) Each $\tilde{\pi}_n$ will be an n-chain respecting, 1–1 partial map between ω_n and ω'_n. We call the elements of $(\mathrm{domain}(\tilde{\pi}_n) \cup \mathrm{range}(\tilde{\pi}_n))$ the vertices *matched as of stage n*. Note that in order to be 1–1 *and* respect n-chains, it must be the case that biased n-chains (which have an odd number of elements) end up with at least one vertex that is unmatched as of stage n. In our construction of the $\tilde{\pi}_n$, we will maintain the following invariant, for each n:

(2) For each n-chain C, every vertex of C is matched as of stage n, *except* if C is ω-biased (respectively, ω'-biased) in which case exactly one ω-vertex (respectively, ω'-vertex) is unmatched.

Note that the invariant implies that if C is a biased n-chain, then the vertices of C matched as of stage n form two unbiased paths (either of which could be null) on either side of C's unmatched vertex and if C is a unbiased n-chain, then all of the vertices of C are matched as of stage n and, hence, form an unbiased path.

Assume $\tilde{\pi}_{n-1}$ is as required. We consider how to define $\tilde{\pi}_n$ on the ω_n-vertices of an n-chain C. First, let $\{\, z_1, z_2, \ldots, z_k \,\}$ be the set of length n vertices of C together with the vertices of C unmatched as of stage $n-1$. (There may in fact be several vertices of C unmatched as of stage $n-1$, since C may contain several biased $(n-1)$-chains.) Moreover, let $z_1, z_2, \ldots, z_k$ be in the (path) order in which they occur in C. (If C is cyclic, choose z_1 to be the lexigraphically least possible ω-vertex from among the z_i's. Note that in this case there are an equal number of unmatched ω- and ω'-vertices in C as G is bipartite.) It follows from our discussion of the invariant that the set of vertices of C that were *matched* as of stage $n-1$ form a series of disjoint, unbiased subpaths of C. Hence, the elements of the sequence $z_1, z_2, \dots, z_k$ must alternate between being in ω and ω' and this sequence has the same bias (i.e., unbiased, or ω-, or ω'-biased) as C. So, for each x, an ω_n-vertex of C, define

$$(3) \qquad \tilde{\pi}_n(x) = \begin{cases} \tilde{\pi}_{n-1}(x), & \text{if (i) } x \text{ is matched as of stage } n-1; \\ z_{2i-1}, & \text{if (ii) } x = z_{2i}; \\ z_{2i}, & \text{if (iii) } x = z_{2i-1} \text{ and } 2i \leq k; \\ \text{undefined}, & \text{(iv) otherwise.} \end{cases}$$

Note that clause (ii) applies to the z_i's of C if and only if C is ω'-rooted, and clause (iii) applies otherwise. Thus, clauses (ii) and (iii) of equation (3) parallel (1). If C is unbiased, then k is even; hence, all of C's vertices are matched as of stage n. If C is ω-biased (respectively, ω'-biased), all of C's vertices are matched as of stage n except z_k which is in ω (respectively, ω'). It follows then that $\tilde{\pi}_n$ is 1–1, respects n-chains, and satisfies the invariant (2).

Suppose q is a monotone increasing polynomial such that both f and g are strictly DSPACE($q(n)$) computable. Thus, for all z,

$$(4) \quad q(|z|) \geq |z|,\ |f(z)|,\ |g(z)|, \text{ space used to compute } f(z) \text{ and } g(z).$$

Lemma 11. *For each $z \in (\omega \cup \omega')$, z is matched as of stage $q(|z|)$.*

Proof Let $n = |z|$ and let C be z's n-chain. If C is cyclic, then, by the invariant (2), z is matched as of stage n and we are done. So, suppose C is acyclic. Let t be the tail of C and let $\hat{z}$ be t's successor in G. Since z is followed by $\hat{z}$, a length $|\hat{z}|$ vertex, in z's $|\hat{z}|$-chain, it follows by the construction that z is matched as of stage $|\hat{z}|$. Now, by (4) we have that

$|\hat{z}| \leq q(|t|)$. Since $|t| \leq |z|$ and since q is monotone increasing, we thus have $|\hat{z}| \leq q(|t|) \leq q(|z|)$. □

Lemma 12. *Both $\lambda n, x \in \omega_n . \tilde{\pi}_n(x)$ and $\lambda n, y \in \omega'_n . \tilde{\pi}_n^{-1}(y)$ are computable in $\mathcal{O}(n \cdot q(n))$ space.*

Proof Sketch To compute $\tilde{\pi}_n(x)$ using (3), one needs to

- compute $\tilde{\pi}_{n-1}(x)$,
- if it is defined, output the result,
- if not, then x is one of the z_i's for x's n-chain, in which case one needs to find: (a) the root (if any) of x's n-chain, (b) z_k, and, if x's chain is ω'-rooted, (c.i) the z_i immediately preceeding x in the list of z_i's, and if x's chain is not ω'-rooted and $x \neq z_k$, (c.ii) the z_i immediately following x. (If x's chain is not ω'-rooted and $x = z_k$, then $\tilde{\pi}_{n-1}(x)$ is undefined.)

All of this can be accomplished in the course of a constant number (independent of x) traversals of x's n-chain, making recursive calls to $\tilde{\pi}_{n-1}$ along the way to determine whether various $z \in (\omega_{n-1} \cup \omega'_{n-1})$ were matched as of stage $n-1$. Since f and g are 1–1 strictly polynomial-space computable functions, it is clear that traversing an n-chain can be done in $\mathcal{O}(q(n))$ space. It is also clear that in using (3) to compute $\tilde{\pi}_n(x)$, the depth of recursions is no more than n. Thus, it follows that $\tilde{\pi}_n(x)$ can be computed within the required space bound. The argument for $\tilde{\pi}_n^{-1}$ follows by symmetry. □

Define $\tilde{\pi} = \cup_{n \in \omega} \tilde{\pi}_n$. Since each $\tilde{\pi}_n$ extends $\tilde{\pi}_{n-1}$, $\tilde{\pi}$ is well defined. Since each $\tilde{\pi}_n$ is 1–1 and respects n-chains, $\tilde{\pi}$ is also 1–1 and respects chains. By Lemma 11, $\tilde{\pi}$ is total and onto. By (3) and Lemma 11 we also have that, for all $x \in \omega$, $|\tilde{\pi}(x)| \leq q(|x|)$ and $|x| \leq q(|\tilde{\pi}(x)|)$. Finally, by Lemmas 11 and 12, we have that $\tilde{\pi}$ and $\tilde{\pi}^{-1}$ are both polynomial-space computable.

□ **Theorem 7**

Theorem 13 (Dowd's Theorem). *Every two strictly linear-space 1-equivalent sets are strictly linear-space isomorphic.*

Proof Sketch Below we give a finer analysis of the space complexity of the construction of the previous proof and conclude the present theorem as a consequence of this analysis.

In the proof of Lemma 12 we gave a sketch of how to compute $\widetilde{\pi}_n(x)$. In that sketch we used recursive calls to $\widetilde{\pi}_{n-1}$ to determine whether a vertex in $(\omega_{n-1} \cup \omega'_{n-1})$ was matched as of stage $n-1$. Below we show how to perform this test without the recursive calls.

The vertex of a biased n-chain C that is unmatched as of stage n we call the *unmatched vertex of C*. We give a purely graph theoretic characterization of which vertex of a biased n-chain is its unmatched vertex.

Lemma 14. *Suppose that C is a biased n-chain, that t is C's tail, and that n' is the* largest *number $\leq n$ such that either (i) $|t| = n'$ or else (ii) t's $(n'-1)$-chain is unbiased.*

Then, in case (i), t is the unmatched vertex of C, and, in case (ii), the unmatched vertex of C is the (length n') predecessor of the root of t's $(n'-1)$-chain.

Proof Let z be the vertex that the lemma claims is the unmatched vertex of C. For $\hat{n} = n', \ldots, n,$ let $C_{\hat{n}}$ denote z's $\hat{n}$-chain. Note that for $\hat{n} = n', \ldots, n,$ $C_{\hat{n}}$ must be biased because otherwise n' would not be the largest number $\leq n$ such that (i) or (ii) holds. Since z is of length n' and followed by a unbiased $(n'-1)$-chain (which is null in case (i)) and since $C_{n'}$ is biased, it is clear that z is the unmatched vertex of $C_{n'}$. By an easy induction we have that, for $\hat{n} = n'+1, \ldots, n,$ z is the last vertex in $C_{\hat{n}}$ which is unmatched as of stage $\hat{n}-1$ and z is followed in $C_{\hat{n}}$ by an unbiased $(\hat{n}-1)$-chain. Therefore, for $\hat{n} = n'+1, \ldots, n,$ z is is the unmatched vertex of $C_{\hat{n}}$. □

Using the characterization above, it is relatively simple to concoct a procedure for testing the predictate

$$\lambda n, z \in (\omega_n \cup \omega'_n).[\, z \text{ is matched as of stage } n\,]$$

that runs in $\mathcal{O}(q(n))$ space. Thus, in our sketch of how to compute $\tilde{\pi}_n(x)$, we can replace all the recursive calls to $\tilde{\pi}_{n-1}$ used to test matching with this $\mathcal{O}(q(n))$-space procedure. So, exclusive of the cost of the recursive call to compute $\tilde{\pi}_{n-1}(x)$ under clause (i) of (3), it follows that the computation of $\tilde{\pi}_n(x)$ can be done within $\mathcal{O}(q(n))$-space. However, the recursion to compute $\tilde{\pi}_{n-1}(x)$ is a tail recursion and so it does not require a stack to carry out. Therefore, it follows that

Lemma 15. *Both $\lambda n, x \in \omega_n . \tilde{\pi}_n(x)$ and $\lambda n, y \in \omega'_n . \tilde{\pi}_n^{-1}(x)$ are computable in $\mathcal{O}(q(n))$ space.*

By Lemma 11 we have that $\tilde{\pi} = \lambda x . \tilde{\pi}_{q(|x|)}(x)$ and $\tilde{\pi}^{-1} = \lambda x . \tilde{\pi}^{-1}_{q(|x|)}(x)$. Hence, by Lemma 15,

Corollary 16. *Both $\tilde{\pi}$ and $\tilde{\pi}^{-1}$ are computable in $\mathcal{O}(q(q(|x|)))$ space.*

If f and g are 1–1 strictly linear-space computable functions, then we can choose q to be a linear polynomial, and, hence, $q \circ q$ is linear too. Therefore, by Corollary 16, the theorem follows. □ **Theorem 13**

We return to the question of p-isomorphism by investigating conditions on the 1-reductions that make 1-equivalent sets p-isomorphic. Unlike Berman and Hartmanis's Theorem 1, which focuses on the reductions themselves, we look closer at the structure of the chains formed by the 1-reductions. In doing so, we obtain results stronger than Theorem 1.

We say that f and g have *polynomial-time constructible n-chains* if and only if there is a procedure such that, given n and $z \in (\omega_n \cup \omega'_n)$, constructs z's entire n-chain in time polynomial in n.

Theorem 17. *Suppose two sets are (polynomial-time) 1-equivalent as witnessed by reductions f and g which have polynomial-time constructible n-chains. Then, the two sets are p-isomorphic.*

On the surface this looks like a much stronger result than Theorem 1. Is isn't however. If f and g are such that there are no cyclic f,g-chains, then one can show that the hypotheses of Theorem 1 are equivalent to those of Theorem 17. We can use the construction for Theorem 7 to obtain a strictly stronger result than Theorems 1 and 17. In order to state this result we introduce the following terminology.

We say that f and g's n-chains have *polynomial-time uniform extremities* if and only if there is a procedure which, given n and $z \in (\omega_n \cup \omega'_n)$, runs in time polynomial in n and decides whether z's n-chain is acyclic, and if it is, determines the two extreme vertices of this n-chain.

Theorem 18. *Suppose A and B are (polynomial-time) m-equivalent as witnessed by reductions f and g that are*

(a) one-one,

(b) honestly-invertible, and

(c) their n-chains have polynomial-time uniform extremities.

Then, A and B are p-isomorphic.

To prove this, one merely checks that the theorem's hypotheses suffice to run the construction of Theorem 7 in polynomial-time. This is straightforward and we omit the details.

Later we show that Theorem 18's hypotheses are strictly weaker than those of Theorems 1 and 17, see Proposition 27 below. Hypothesis (c) is still pretty strong, however. It will be apparent from the proof of Theorem 5 in the next section that there are one-to-one, polynomial-time computable f and g such that finding just the tails of the corresponding n-chains is PSPACE-complete.

4. Inequivalences

Our proofs of Theorems 5 and 6 follow the same general strategy as the proof of Ko, Long, and Du's Theorem 2. We first sketch a proof of Theorem 2, and then sketch a proof of our Theorem 5 which builds on the ideas introduced in Theorem 2's proof. The proof of Theorem 6 is a modification of our argument for Theorem 5.

Theorem 2 *Suppose that* P $\neq$ UP*. Then there exist 1-li equivalent sets which are incomparable with respect to p-invertible reductions. Moreover, there are such sets which are 2-tt complete for* EXP.

Proof Sketch Since we are assuming P $\neq$ UP, by [KLD87, Proposition 2.1], there exists a length-increasing one-way function t. Define $f\colon \omega \to \omega'$ by the following three equations.

$$(5) \qquad f(3x) = 6t(x) + 1. \qquad f(3x+1) = 6x + 4. \qquad f(3x+2) = 6x+5.$$

Let g have the same definition as f except that we regard g as a function from ω' to ω. Clearly, f and g are one-one and length increasing. Note that every number of the form $3z$ in $\omega \cup \omega'$ is the root of its own f, g-chain. (Each number of the form $6z+2$ is also the root of its own chain—a fact that will be useful later on.) By a diagonal construction we shall produce sets $A \subseteq \omega$ and $B \subseteq \omega'$ that satisfy:

(6) $\qquad f\colon A \leq^{\mathrm{p}}_{\text{1-li}} B$ and $g\colon B \leq^{\mathrm{p}}_{\text{1-li}} A$,

(7) $\qquad A$ and B are 2-tt complete members of EXP, but

(8) $\qquad$ there is no p-invertible h such that $h\colon A \leq^{\mathrm{p}}_{\mathrm{m}} B$ or $h\colon B \leq^{\mathrm{p}}_{\mathrm{m}} A$.

The diagonalization depends on the following key lemma.

Lemma 19 (The Chain Crossing Lemma). *Suppose h is a p-invertible map (either from ω to ω' or from ω' to ω). Then, h crosses infinitely many chains. In fact, there are infinitely many z's such that $3z$ and $h(3z)$ are in different chains.*

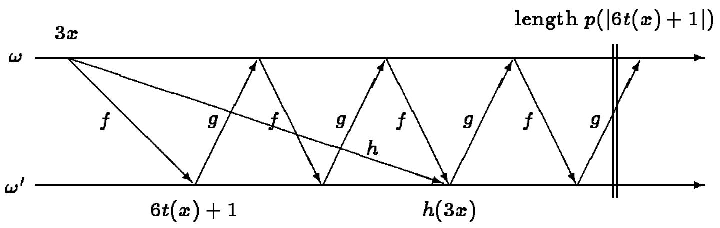

Figure 1: $h(3x)$ lands in V_y.

Proof We handle the case of $h\colon \omega \to \omega'$. The $\omega' \to \omega$ case follows by symmetry.

Since h is polynomial-time computable, there is a nondecreasing polynomial p such that, for all x, $|h(x)| \leq p(|x|)$. For each y, let V_y be the set of ω'-vertices of the chain of $(6y+1)'$ that are of length $\leq p(|6y+1|)$. By our definitions of f and g it follows that one can, given y, list all the elements of V_y in $\mathcal{P}oly(|y|)$ time. Now, by (5), if $h(3x)$ is in the same chain as $3x$, then $h(3x)$ is in $V_{t(x)}$, see Figure 1. Thus, if the lemma were false, then for all sufficiently large y, the following equation will hold:

$$t^{-1}(y) = \begin{cases} h^{-1}(z')/3, & \text{if } z' \in V_y \text{ is such that } t(h^{-1}(z')/3) = y; \\ \text{undefined}, & \text{if there is no such } z' \in V_y. \end{cases}$$

But, since one can list all the elements of V_y in $\mathcal{P}oly(|y|)$ time and since t and h^{-1} are polynomial-time computable, it would then follow that t is p-invertible—a contradiction. □ **Lemma 19**

Returning to the proof of Theorem 2, the construction of A and B works by "painting" chains. Each chain is painted either blue or green. A chain painted blue has all of its ω-elements in A and its ω'-elements in B. A chain painted green has all of its ω-elements in $\overline{A}$ and its ω'-elements in $\overline{B}$. Since the chains form a partition of $\omega \cup \omega'$, painting all the chains will completely determine A and B, and ensure that they satisfy (6) above.

Now, given an $h\colon \omega \to \omega'$ and an x such that x and $h(x)$ are in different colored chains, we have that $x \in A \iff h(x) \notin B$; and hence that h fails to m-reduce A to B. Using this last observation together with Lemma 19, one can construct A and B satisfying (6) and (8) by a elementary, noneffective diagonalization: starting with all f,g-chains unpainted, paint chains one by one, each time cancelling some p-invertible h by painting x's chain and $h(x)$'s chain opposite colors, for some x. Each such h gives us infinitely many chances to cancel it, and there are only countably many such h, so we can diagonalize against them all. See the proof of Theorem 6.6.2 in [KMR90] for more details.

To build an A and B satisfying (7) in addition to (6) and (8), a more delicate construction is needed. We handle this construction by means of a general technical lemma which is also used in the proofs of Theorems 5 and 6 below. To state this lemma, we introduce the following terminology. Suppose C is an f,g-chain with root r. The *ith successor of* r is the vertex of C obtained by applying f and g a combined total of i times to r. Suppose h is a function from ω to ω' (or from ω' to ω). Then we say h *promptly crosses* C if and only if there exists a vertex x of C such that (a) x is the ith successor of r for some $i \leq |r|$, (b) for each $j \leq i$, the jth successor of r has length $\leq |r|$, and (c) $h(x)$ is not in C. We now state the lemma, the proof of which appears in Appendix A.

Lemma 20 (The Chain Painting Lemma). *Suppose the following:*

1. *$f\colon \omega \to \omega'$ and $g\colon \omega' \to \omega$ are 1–1 and polynomial-time computable.*

2. *$r\colon \omega \to (\omega \cup \omega')$ is 1–1, $2^{\mathcal{P}oly(n)}$-time computable, and, for each x, $r(x)$ is the root of an f,g-chain. For each x, let $\widehat{C}_x$ denote $r(x)$'s chain.*

3. *q is a polynomial such that, for all x and all $z \in \widehat{C}_x$, $|x| \leq q(|z|)$,*

4. *$s\colon \omega \to \omega$ is polynomial-time computable, and for all $x, y \in \omega$, $s(y)$ and $s(z)$ are in f,g-chains distinct from all the $\widehat{C}_x$'s and from each other. For each y, let $\widehat{D}_y$ denote $s(y)$'s chain.*

5. *Given a $z \in (\omega \cup \omega')$ and $x \in \omega$, deciding whether z is a vertex of $\widehat{C}_x$ can be done in $\mathcal{P}oly(|z| + x)$-time.*

6. *Given a $z \in (\omega \cup \omega')$, deciding whether z is in one of the $\widehat{D}_y$'s, and, if so, which y, all can be done in $\mathcal{P}oly(|z|)$-time.*

Then, given all of the above, there exist sets A and B that satisfy:

(a) $f\colon A \leq_1^{\mathrm{p}} B$ and $g\colon B \leq_1^{\mathrm{p}} A$,

(b) A and B are 2-tt complete for EXP, *and*

(c) there is no polynomial-time computable $h\colon \omega \to \omega'$ (respectively, $h\colon \omega' \to \omega$) which both promptly crosses infinitely many $\widehat{C}_x$'s and that $\leq_{\mathrm{m}}^{\mathrm{p}}$-reduces A to B (respectively, B to A).

Despite the profusion of hypotheses in Lemma 20, they are very easily—almost trivially—satisfied in every case that we apply the lemma. In the context of the proof of the present theorem:

$$r \quad = \quad \lambda x.\begin{cases} 3x/2, & \text{if } x \text{ is even;} \\ (3(x-1)/2)', & \text{if } x \text{ is odd;} \end{cases}$$

$q = \lambda n.[n+1]$; $s = \lambda x.[6x+2]$; and Lemma 19 asserts that every p-invertible h promptly crosses infinitely many $\widehat{C}_x$'s. Therefore, the existence of an A and B as required by the theorem follows from Lemma 20. □ **Theorem 2**

We now apply the technique used in the proof above to 1-reductions which are not necessarily length-increasing. With the (most likely) weaker assumption that P $\neq$ PSPACE, we obtain two different inequivalences. The one we give now involves honest m-reductions; the other, which we give below in Theorem 6, is about isomorphisms and uses the same idea with one additional twist.

Theorem 5 *Suppose that* P $\neq$ PSPACE. *Then there exist 1-equivalent sets that are incomparable with respect to honest m-reductions. Moreover, there are such sets which are 2-tt complete for* EXP.

Proof Let L be an element of (PSPACE $-$ P).

This proof follows a plan roughly analogous to the argument for Theorem 2: we construct 1-1, polynomial-time computable functions f and g; prove that every honest polynomial-time computable function must promptly cross infinitely many of a particular collection of f,g-chains; then, by an application of the Chain Painting Lemma, we produce the two sets required by the theorem. In Theorem 2's proof, the f, g-chains encoded the graph of a one-way function t and that proof's chain crossing lemma was shown by proving that *if* one had a p-invertible h that crossed only finitely many f, g-chains, *then* from h one could construct an polynomial-time inverse of t, contradicting the assumption that t is one-way. In this proof the f, g-chains encode computations of a Turing machine that decides the set L, and this proof's chain crossing lemma is shown by proving that *if* one had an honest polynomial-time computable h that crosses only finitely many f, g-chains, *then* from h one could construct an polynomial-time decision procedure for L, contradicting the assumption that $L \in (\mathrm{PSPACE} - \mathrm{P})$.

To define f and g and ensure that they are 1-1, we use Bennett's work on reversible Turing machines [Ben89]. Informally, a deterministic Turing machine M is said to be reversible if and only if, at any point of a computation, there is an *unambiguous* way of backing up the computation to its previous state. We formalize this notion as follows. Let M be a deterministic Turing machine with k tapes (including an input and an output tape), states Q, alphabet Σ, start state q_0, unique final state q_1, allowable tape moves L (left), R (right), and N (no movement), and transition function $\tau\colon Q \times \Sigma^k \rightarrow Q \times \Sigma^k \times \{\, L, R, N \,\}^k$. All halting computations of M end in state q_1. Let ID be the set of instantaneous descriptions (i.d.'s) of M and, for each $I \in \mathrm{ID}$, let $\tau(I)$ be the successor i.d. of M, if any, as determined by τ. The *initial* i.d. of M for a given input has M in state q_0, the input tape head just to the left of the input, and all other tapes empty. Now, such an M is said to be *reversible* if and only if there is another transition function $\sigma\colon Q \times \Sigma^k \rightarrow Q \times \Sigma^k \times \{\, L, R, N \,\}^k$ such that, for each non-final i.d. I that is reachable by M from some initial i.d., we have that $\sigma(\tau(I)) = I$. Reversible machines are crucial to our keeping the functions f and g 1-1. The following proposition follows from Bennett's general results and roughly corresponds to the corollary on page 770 of [Ben89].

Proposition 21. *Suppose M is a multi-tape Turing machine that computes a function $t\colon \omega \rightarrow \omega$ and that runs in space $S(n)$. Then, there there is an $\mathcal{O}(S(n)^2)$ space bounded, reversible Turing machine that computes*

$\lambda x\,.\,\langle x, t(x)\rangle$.

By the proposition, there is a reversible Turing machine that computes $\lambda x\,.\,\langle x, L(x)\rangle$ in polynomial-space. Let M be such a machine and let ID, τ and σ be as above. For each x, let $\mathit{initial}(x)$ be the initial i.d. of M on input x. Define

$$\widehat{\mathrm{ID}} \;=\; \{\, I : \sigma(\tau(I)) = I \,\}.$$

By this definition, every non-final i.d. which is reachable from some initial i.d. is in $\widehat{\mathrm{ID}}$. Also, no final i.d. can be in $\widehat{\mathrm{ID}}$ since if I is final, then $\tau(I)$ is undefined, and, hence, so is $\sigma(\tau(I))$. Note that $\widehat{\mathrm{ID}}$ is polynomial-time decidable, and that when $\lambda I\,.\,\tau(I)$ is restricted to $\widehat{\mathrm{ID}}$, the function is total and one-one.

Now we introduce some tools to help with encoding M-computations into f, g-chains. Let $\#\colon \mathrm{ID} \to \omega$ be a one-one, onto function, and such that

- the functions induced over ω by $\lambda I\,.\,\tau(I)$, $\lambda I\,.\,\sigma(I)$, and *initial* are polynomial-time computable and,
- given i, one can in $\mathcal{P}\mathit{oly}(|i|)$-time decide if i corresponds to a final i.d., and, if so, extract the result of this i.d.'s computation.

Such a # is straightforward, if tedious, to define. For all v, x, y, $z \in \omega$ and all $I \in \mathrm{ID}$, define

$$\begin{aligned} \mathit{start}(x,y) &= 3\langle x, y\rangle. \\ \mathit{active}(x,v,I) &= 3\langle x, v, \#(I)\rangle + 1. \\ \mathit{idle}(x,v,z,I) &= 3\langle x, v, z, \#(I)\rangle + 2. \end{aligned}$$

Since $\langle\cdot,\cdot\rangle$ and # are one-one, so are *start*, *active*, and *idle*, and, since $\langle\cdot,\cdot\rangle$ and # are also onto, the ranges of *start*, *active*, and *idle* partition ω. Finally, define $f\colon \omega \to \omega'$ by the following set of equations.

$$f(\mathit{start}(x,y)) \;=\; \begin{cases} \mathit{active}(x, v, \mathit{initial}(x)), & \text{if } y = \mathbf{0}^v; \\ \mathit{start}(x,y), & \text{if } y \notin \{\, \mathbf{0}^v : v \in \omega \,\}. \end{cases}$$

$$f(\mathit{active}(x,v,I)) \;=\; \begin{cases} \mathit{active}(x,v,\tau(I)), & \text{if } I \in \widehat{\mathrm{ID}}; \\ \mathit{idle}(x,v,0,I), & \text{otherwise.} \end{cases}$$

$$f(\mathit{idle}(x,v,z,I)) \;=\; \mathit{idle}(x,v,z+1,I).$$

Let g have the same definition as f except that we regard g as a function from ω' to ω. By our discussion of τ, σ, $\#$, *start*, *active*, and *idle* it follows that f and g are one-one and polynomial-time computable. For each x and v, let $C_{x,v}$ denote the f,g-chain with root $\mathit{start}(x,\mathbf{0}^v) \in \omega$ and let $C'_{x,v}$ denote the chain with root $\mathit{start}(x,\mathbf{0}^v)' \in \omega'$.

A $C_{x,v}$ chain has the following structure. It begins with the root vertex $\mathit{start}(x,\mathbf{0}^v)$ followed by an exponential drop to $\mathit{active}(x,v,\mathit{initial}(x)) \in \omega'$. Then f and g conspire to simulate M on input x—each $C_{x,v}$ vertex of the form $\mathit{active}(x,v,I)$ (where I is a non-final i.d. of M on input x) is followed in $C_{x,v}$ by the vertex $\mathit{active}(x,v,\tau(I))$. When the chain reaches the vertex $\mathit{active}(x,v,I_{\mathrm{fin}})$ (where I_{fin} is the final i.d. of M on input x), the next vertex in $C_{x,v}$ is $\mathit{idle}(x,v,0,I_{\mathrm{fin}})$. Thereafter, each vertex of the form $\mathit{idle}(x,v,z,I_{\mathrm{fin}})$ is followed by the vertex $\mathit{idle}(x,v,z+1,I_{\mathrm{fin}})$ *ad infinitum.* Since M is polynomial-space bounded and since $\#$, *start*, etc. are all polynomial-time computable, it follows that there is a monotone polynomial p_L such that all the "active" vertices of $C_{x,v}$ are of length strictly less than $p_L(|x|+|v|)$.

The structure of an $C'_{x,v}$ chain is analogous.

Lemma 22 (The Chain Crossing Lemma). *Suppose h is an honest, polynomial-time computable function (from ω to ω' or from ω' to ω). Then, h crosses infinitely many chains. In fact, there are infinitely many x's and v's such that $\mathit{start}(x,\mathbf{0}^v)$ and $h(\mathit{start}(x,\mathbf{0}^v))$ are in different chains.*

Proof We handle the $h\colon \omega \rightarrow \omega'$ case. The $\omega' \rightarrow \omega$ case follows by symmetry.

Let p_L be as in the discussion preceding the lemma.

Since h is honest, there exist k and x_0 such that for all $x > x_0$, $|h(x)| > |x|^{1/k}$. Since *start* is monotone increasing in both arguments, we have that

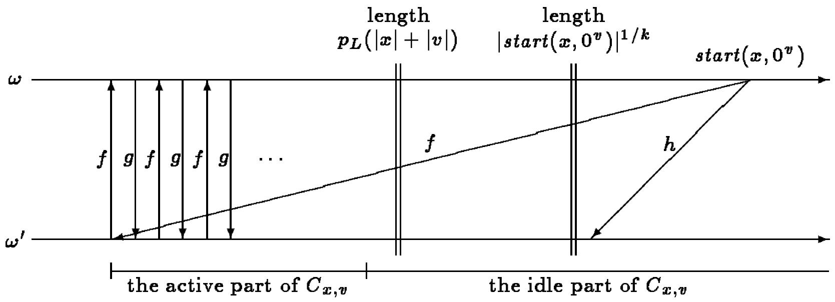

Figure 2: $h(start(x, \mathbf{0}^v))$ lands in $C_{x,v}$.

$|start(x, \mathbf{0}^v)| \in \Omega(|x| + 2^{|v|})$. Thus, for each x and all sufficiently large v,

$$(9) \qquad p_L(|x| + |v|) \;\; \leq \;\; |start(x, \mathbf{0}^v)|^{1/k}.$$

Since *start* is increasing in both arguments, it easily follows that there is a polynomial $p_\star$ such that, for all x, if $v = p_\star(|x|)$, then (9) is satisfied.

Claim. *Suppose $x > x_0$, $v = p_\star(|x|)$, and $h(start(x, \mathbf{0}^v))$ is in $C_{x,v}$. Then, for some z, $h(start(x, \mathbf{0}^v)) = idle(x, v, z, I)$, where I is the final i.d. of M on input x.*

Proof of Claim Since *start* is increasing in both arguments and since $x > x_0$, by our choice of k and x_0 it follows that $|start(x, \mathbf{0}^v)|^{1/k} < |h(start(x, \mathbf{0}^v))|$. By our choice of $p_\star$, it also follows that (9) holds for x and v. Thus, we have the situation described by Figure 2. Now, since $|h(start(x, \mathbf{0}^v))| > p_L(|x| + |v|)$, $h(start(x, \mathbf{0}^v))$ cannot be in the active part of $C_{x,v}$. Thus, since $h(start(x, \mathbf{0}^v))$ is in $C_{x,v}$, it must be in the idle part of $C_{x,v}$. Therefore, the claim follows. □ **Claim**

Suppose by way of contradiction that the lemma is false. So, for all but finitely many x, $h(start(x, \mathbf{0}^{p_\star(|x|)}))$ is in $C_{x,p_\star(|x|)}$. Then by the claim, for all but finitely many x, one can determine $L(x)$ by: (i) computing $h(start(x, \mathbf{0}^{p_\star(|x|)}))$, (ii) from this value extracting the final i.d. of M on input x, and (iii) from this i.d. determining $L(x)$. All of this can be done in time $\mathcal{P}oly(|x|)$. Therefore, L is polynomial-time decidable. But this contradicts the assumption that $L \in (\text{PSPACE} - \text{P})$. □ **Lemma 22**

Now let r enumerate all the roots of the $C_{x,i}$'s and $C'_{x,i}$'s, so that $r(2\langle x,i\rangle)$ is the root of $C_{x,i}$ and $r(2\langle x,i\rangle+1)$ is the root of $C'_{x,i}$. We can choose q to be $\lambda n\,.\,[n+1]$ since the smallest vertex on $C_{x,i}$ is of length at least $3\langle x,i\rangle$. Also let $s = \lambda x\,.\,\mathit{start}(x, \mathbf{1})$. It is straightforward to check that, for these choices of r, q, and s, all the hypotheses of the Chain Painting Lemma are satisfied. Therefore, by this lemma there exist sets A and B that are 1-equivalent, 2-tt complete for EXP, but which are not honest m-comparable.

□ **Theorem 5**

We now turn to the second of the two inequivalences—the first being Theorem 5. There, it was the case that (assuming P $\neq$ PSPACE) a polynomial-time honest equivalence (not even 1–1) could not be substituted for an unrestricted polynomial-time 1-equivalence. Here we show (on the same assumption) the more fine-grained result that a p-isomorphism cannot be substituted for an honest 1-equivalence, even one where both of the 1-reductions are p-invertible. The only property the reductions of Theorem 1 have that is not required here is that of being length-increasing. Thus if P $\neq$ PSPACE, the length-increasing requirement of Theorem 1 is necessary.

Theorem 6 *Suppose that* P $\neq$ PSPACE. *Then there exist p-invertible equivalent sets that fail to be p-isomorphic. Moreover, there are such sets which are 2-tt complete for* EXP.

Our proof of Theorem 6 will run along the same lines as that of Theorem 5. In particular, the f,g-chains we construct will look similar to those of Theorem 5, i.e., they will follow the computation of a polynomial-space reversible Turing machine computing a language $L \notin$ P, then percolate the result when the computation is done, just as before. The difference lies in how the chains begin. The reductions of Theorem 5 were of necessity dishonest, evidenced by the root of each chain being exponentially larger than its successor. Making this exponential drop drastic enough was all that was necessary to defeat the chain-respecting honest maps, by forcing any such map to take the root of the chain to the idle region, thus revealing the result of the PSPACE computation.

We clearly cannot do the same thing here, since our reductions f and g must be p-invertible, and hence honest. Instead, we replace the initial large drop in the chain with a series of small drops, starting at the top (root of the chain) and winding up at the start of the active region, where the chain then continues, simulating the machine's computation as before. We call this initial segment of the chain the *ramp region.* Given a potential p-isomorphism h that respects chains, it is crucial to note that h and h^{-1} naturally correspond to a perfect matching of ω vertices with ω' vertices. Our goal now is to force h to match *some* vertex in the ramp region (we cannot control which) with a vertex in the idle region, thus revealing the result of the computation as in Theorem 5, and allowing us to compute L in polynomial time. Some vertices in the ramp region are small enough so that h may match them with vertices in the active region—we call these ramp vertices "unsafe". h may also match ramp vertices with other ramp vertices. To force h to match some ramp vertex with an idle vertex, we ensure that *there are an unequal number of ω and ω' vertices among all the "safe" ramp vertices not matched by h to unsafe ramp vertices.* Such safe vertices are either matched with each other (one in ω, the other in ω') or to vertices in the idle region, and thus at least one safe ramp vertex must be matched with an idle vertex. We can ensure the inequality in the numbers of such safe vertices simply by deciding on which side (ω or ω') to place the root of the chain—the start of the ramp.

An added difficulty with the present proof is in selecting which maps h to diagonalize against. In Theorem 5, all we needed was to make the reductions sufficiently dishonest to win against any honest reduction. Here, we can only win against p-isomorphisms, so we view explicitly all possible pairs of polynomial-time functions, on the suspicion that any pair may represent a p-isomorphism and its inverse.

Proof of Theorem 6 Let L be an element of $(\mathrm{PSPACE} - \mathrm{P})$. As we noted in the proof of Theorem 5 there is a reversible Turing machine, M, that computes $\lambda x.\langle x, L(x)\rangle$ in polynomial space.

Terminology: Suppose $h\colon \omega \to \omega'$ is a p-isomorphism. We say h *matches* w *with* z when either $h(w) = z$ or $h(z) = w$.

Recall from §2 that $\langle \varphi_i \rangle_{i \in N}$ is an acceptable numbering of the partial recursive functions based on a coding of deterministic, multi-tape Turing machines, and that the function

$$T \;=\; \lambda i, x, n.\begin{cases} \varphi_i(x), & \text{if Turing machine } i \text{ on input } x \\ & \qquad \text{halts within } n \text{ steps;} \\ 0, & \text{otherwise} \end{cases}$$

is computable in $\mathcal{O}((|i| + |x| + n)^2)$ time. For each k, ℓ, and x, define

$$\begin{aligned} \psi^{\ell}_{k}(x) &= T(k, x, (|x| + 2)^{|\ell|}). \\ &= \begin{cases} \varphi_k(x), & \text{if Turing machine } k \text{ on input } x \text{ halts} \\ & \qquad \text{within } (|x|+2)^{|\ell|} \text{ steps;} \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

It is easily seen that, for each polynomial time computable function h, there is a k such that for all sufficiently large ℓ, $h = \psi^{\ell}_{k}$. By the time bound for T it also follows that $\lambda k, \ell, x . \psi^{\ell}_{k}(x)$ is computable in $\mathcal{O}(\,(|k| + 3|x|)^{2|\ell|}\,) \subseteq 2^{\mathcal{O}(\,(|k|+|\ell|+|x|)^2\,)}$ time.

We turn now to defining the 1-reductions f and g.

To encode M-computations into f, g-chains, we use essentially the same tools developed in the proof of Theorem 5. Let τ, σ, ID, $\widehat{\text{ID}}$, and $\#$ be as in the previous proof. For all x, i, z, $m \in \omega$ and all $I \in \text{ID}$, define:

$$\begin{aligned} \mathit{ramp}(x, i, m) &= 3\langle x, i, m \rangle. \\ \mathit{active}(x, i, I) &= 3\langle x, i, \#(I) \rangle + 1. \\ \mathit{idle}(x, i, z, I) &= 3\langle x, i, z, \#(I) \rangle + 2. \end{aligned}$$

Since $\langle \cdot, \cdot \rangle$ and $\#$ are one-one, so are *ramp*, *active*, and *idle*, and, since $\langle \cdot, \cdot \rangle$ and $\#$ are also onto, the ranges of *ramp*, *active*, and *idle* partition ω.

The definitions of f and g that follow involve the $0,1$-valued function d. Defining d will be the chief concern of the next part of the proof. For the moment all that we need to know about d is that it is polynomial-time computable and, for all x and i,

(10) $\{\, y : d(x, i, \mathbf{0}^y) = 0 \,\}$ is a nonempty, *finite* initial segment of ω.

Now, define $f\colon \omega \to \omega'$ by the following set of equations.

$$f(\mathit{ramp}(x,i,m)) \;=\; \begin{cases} \mathit{ramp}(x,i,m), & \text{if } m \notin \mathbf{0}^*; \\ \mathit{active}(x,i,\mathit{initial}(x)), & \text{if } m = \mathbf{0}^0; \\ \mathit{ramp}(x,i,\mathbf{0}^y), & \text{if } m = \mathbf{0}^{y+1} \text{ and } d(x,i,m) = 0; \\ \mathit{ramp}(x,i,m), & \text{if } m = \mathbf{0}^{y+1} \text{ and } d(x,i,m) \neq 0. \end{cases}$$

$$f(\mathit{active}(x,i,I)) \;=\; \begin{cases} \mathit{active}(x,i,\tau(I)), & \text{if } I \in \widehat{\mathrm{ID}}; \\ \mathit{idle}(x,i,0,I), & \text{otherwise.} \end{cases}$$

$$f(\mathit{idle}(x,i,z,I)) \;=\; \mathit{idle}(x,i,z+1,I).$$

Let g have the same definition as f except that we regard g as a function from ω' to ω. From the discussion of τ, σ, G, $\#$ in the previous proof and the definitions of *ramp*, *initial*, *active*, *idle*, f, and g, it follows that f and g are one-one, polynomial-time computable, and p-invertible. For each x and i, let $C_{x,i}$ denote the chain with the ω-vertex $\mathit{ramp}(x,i,\mathbf{0}^0)$. Our construction will mostly ignore the f,g-chains other than the $C_{x,i}$'s.

A $C_{x,i}$ chain has the following structure, depicted in Figure 3. It begins with a root vertex of the form $\mathit{ramp}(x,i,\mathbf{0}^y)$ (in ω or ω') where $y > 0$ is largest such that $d(x,i,\mathbf{0}^y) = 0$. Then the chain "ramps" down from $\mathit{ramp}(x,i,\mathbf{0}^y)$ to $\mathit{ramp}(x,i,\mathbf{0}^{y-1})$ and then to $\mathit{ramp}(x,i,\mathbf{0}^{y-2})$ and so on until it arrives at $\mathit{ramp}(x,i,\mathbf{0}^0) \in \omega$. Note that by the definitions of f and g, each $C_{x,i}$ vertex of the form $\mathit{ramp}(x,i,\mathbf{0}^y)$ is in ω precisely when y is even. Also note that by the definition of *ramp*, as y decreases, so does the *length* of $\mathit{ramp}(x,i,\mathbf{0}^y)$. Returning to our tour of $C_{x,i}$, the vertex $\mathit{ramp}(x,i,\mathbf{0}^0) \in \omega$ is followed by the vertex $\mathit{active}(x,i,\mathit{initial}(x)) \in \omega'$. Then, as in the previous proof, f and g conspire to simulate M in input x—successive active

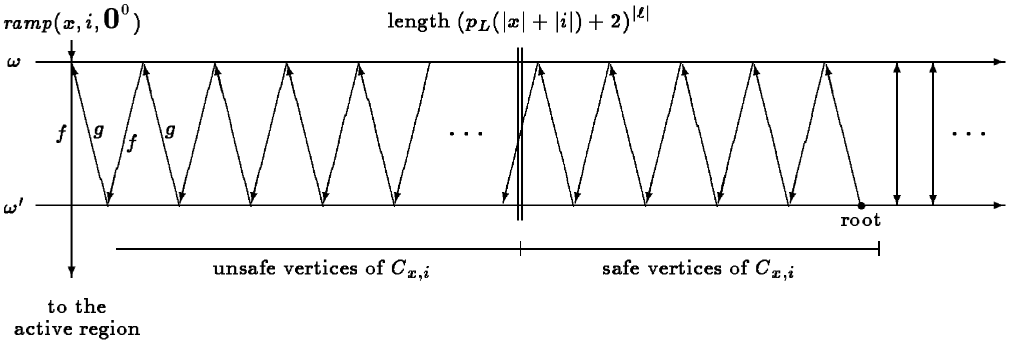

Figure 3: Ramp portion of $C_{x,i}$.

vertices encode successive states of M's computation and the idle vertices all encode the final state of this computation. As in the previous proof, there is a monotone polynomial p_L such that all the active vertices of $C_{x,i}$ are of length $< p_L(|x|+|i|)$ and there are infinitely many idle vertices of length $\geq p_L(|x|+|i|)$.

In our construction the ramp vertices of the $C_{x,i}$'s play the following role. Suppose for this paragraph that $h\colon \omega \rightarrow \omega'$ is a chain-respecting p-isomorphism. Fix x and fix an i such that $i = \langle j,k,\ell\rangle$, $\psi^\ell_j = h$, and $\psi^\ell_k = h^{-1}$. Since both h and h^{-1} are computable in $\lambda n.(n+2)^{|\ell|}$ time, both h and h^{-1} must be $\lambda n.(n+2)^{|\ell|}$-honest. Consider v, a ramp-vertex of $C_{x,i}$ in either ω or ω' with $|v| \geq (p_L(|x|+|i|)+2)^{|\ell|}$. Since h and h^{-1} respect f,g-chains, by our choice of p_L, h must match v with either a ramp or idle vertex of $C_{x,i}$. Our intent is to arrange that if h is a chain-respecting p-isomorphism as above, then for some v in the ramp part of $C_{x,i}$, h matches v with an idle vertex of $C_{x,i}$. Our definition of d below will *force* the existence of such a v of length $\geq (p_L(|x|+|i|)+2)^{|\ell|}$. The vertex v is a "safe" vertex, as described in the proof outline above. Once we know such a v exists, we can compute $L(x)$ as in Theorem 5 by first finding v, then computing the idle vertex that v is matched with via h. This vertex encodes the result of M's computation on input x, i.e., $L(x)$. The function d will be such that for fixed i, this whole process can be done in time polynomial in x, thus contradicting that $L \notin \mathrm{P}$. Thus h cannot respect chains as we assumed.

We introduce the following function and sets to help define d. For each x and i, where $i = \langle j,k,\ell\rangle$ define:

$$\mathit{bnd}(x,i) \;=\; \left[\;|v| : \begin{array}{c} \text{where } v \text{ is the smallest number of} \\ \text{the form } \mathit{ramp}(x,i,\mathbf{0}^{2y}) \text{ such that} \\ |v| \geq \Big(p_L(|x|+|i|) \;+\; 2\Big)^{|\ell|} \end{array}\right].$$

$$V_{x,i} \;=\; \left\{\; v \in \omega : \begin{array}{l} v \text{ is a ramp vertex of } C_{x,i} \\ \text{with } |v| \geq \mathit{bnd}(x,i) \end{array}\right\}.$$

$$V'_{x,i} \;=\; \left\{\; v' \in \omega' : \begin{array}{l} v' \text{ is a ramp vertex of } C_{x,i} \\ \text{with } |v'| \geq \mathit{bnd}(x,i) \end{array}\right\}.$$

$$W_{x,i} \;=\; \left\{\; v \in \omega : \begin{array}{l} v \text{ is a ramp vertex of } C_{x,i} \\ \text{with } \psi_j^\ell(v) \in V'_{x,i} \text{ and} \\ |v| < \mathit{bnd}(x,i)| \leq |\psi_j^\ell(v)| \end{array}\right\}.$$

$$W'_{x,i} \;=\; \left\{\; v' \in \omega' : \begin{array}{l} v' \text{ is a ramp vertex of } C_{x,i} \\ \text{with } \psi_k^\ell(v) \in V_{x,i} \text{ and} \\ |v| < \mathit{bnd}(x,i)| \leq |\psi_k^\ell(v')| \end{array}\right\}.$$

The vertices in $V_{x,i} \cup V'_{x,i}$ are the safe vertices, depicted in Figure 3. The rest of the ramp vertices are unsafe. Thus $W_{x,i}$ (respectively $W'_{x,i}$) comprises those unsafe ramp vertices which are mapped to safe ramp vertices via ψ_j^ℓ (respectively ψ_k^ℓ). The sets $W_{x,i}$ and $W'_{x,i}$ are clearly finite and, given that (10) holds, so are $V_{x,i}$ and $V'_{x,i}$. Our definition of d below will guarantee that $V_{x,i}$ and $V'_{x,i}$ will be nonempty. Also note that, for each x and i, where $i = \langle j,k,\ell\rangle$, we have that

$$(11) \qquad (p_L(|x|+|i|) \;+\; 2)^{|\ell|} \;\leq\; \mathit{bnd}(x,i)$$

and the least ramp vertex of $C_{x,i}$ which is of length $\geq \mathit{bnd}(x,i)$ is an ω-vertex. This last property of bnd helps to simplify the definition of d and the proof of Lemma 24 below.

Lemma 23. *Suppose h is a p-isomorphism and suppose that $i = \langle j, k, \ell \rangle$ is such that $h = \psi_j^\ell$ and $h^{-1} = \psi_k^\ell$. Then, for all x, if*

$$\|V_{x,i}\| - \|V'_{x,i}\| \quad \neq \quad \|W'_{x,i}\| - \|W_{x,i}\|, \tag{12}$$

then there is a $v \in (V_{x,i} \cup V'_{x,i})$ that is matched by h with either an idle vertex of $C_{x,i}$ or a vertex outside of $C_{x,i}$.

Proof Fix x and suppose that h matches each $v \in (V_{x,i} \cup V'_{x,i})$ with a vertex in $C_{x,i}$. We show that h matches some $v \in (V_{x,i} \cup V'_{x,i})$ with an idle vertex of $C_{x,i}$.

From the definitions of bnd, $V_{x,i}$, and $V'_{x,i}$ and from (11), we have that $|\min(V_{x,i} \cup V'_{x,i})| \geq bnd(x,i) \geq (p_L(|x|+|i|)+2)^{|\ell|}$. Since both h and h^{-1} are $\lambda n.(n+2)^{|\ell|}$-honest, h cannot match a member of $V_{x,i} \cup V'_{x,i}$ with a number of length less than $p_L(|x|+|i|)$. Hence, by our choice of p_L, we have that h cannot match any element of $V_{x,i} \cup V'_{x,i}$ with any active vertex of $C_{x,i}$. By assumption, h matches each $v \in (V_{x,i} \cup V'_{x,i})$ with some vertex in $C_{x,i}$. Hence, it follows that both $h(V_{x,i})$ and $h^{-1}(V'_{x,i})$ are contained in the ramp and idle parts of $C_{x,i}$.

By the definitions of $W_{x,i}$ and $W'_{x,i}$,

$$\begin{aligned} h(W_{x,i}) &= \left\{ v \in V'_{x,i} : \ h^{-1}(v) \text{ is a ramp vertex} \notin V_{x,i} \right\}. \\ h^{-1}(W'_{x,i}) &= \left\{ v \in V_{x,i} : \ h(v) \text{ is a ramp vertex} \notin V'_{x,i} \right\}. \end{aligned}$$

Hence, since h and h^{-1} are one-one, it follows that

$$V_{x,i} - h^{-1}(W'_{x,i}) \quad = \quad \left\{ v \in V_{x,i} : \begin{array}{l} h(v) \in (V'_{x,i} - h(W_{x,i})) \text{ or } h(v) \text{ is} \\ \text{an idle vertex of } C_{x,i} \end{array} \right\}.$$

$$V'_{x,i} - h(W_{x,i}) \quad = \quad \left\{ v' \in V'_{x,i} : \begin{array}{l} h^{-1}(v') \in (V_{x,i} - h^{-1}(W'_{x,i})) \text{ or} \\ h^{-1}(v') \text{ is an idle vertex of } C_{x,i} \end{array} \right\}.$$

Figure 4 shows the situation that may typically occur in the ramp region.

Now suppose h matches every $v \in (V_{x,i} \cup V'_{x,i})$ with a ramp vertex. Then it must be the case that h provides a 1–1 correspondence between $V_{x,i} - h^{-1}(W'_{x,i})$ and $V'_{x,i} - h(W_{x,i})$, and thus

$$\|V_{x,i} - h^{-1}(W'_{x,i})\| \quad = \quad \|V'_{x,i} - h(W_{x,i})\|, \tag{13}$$

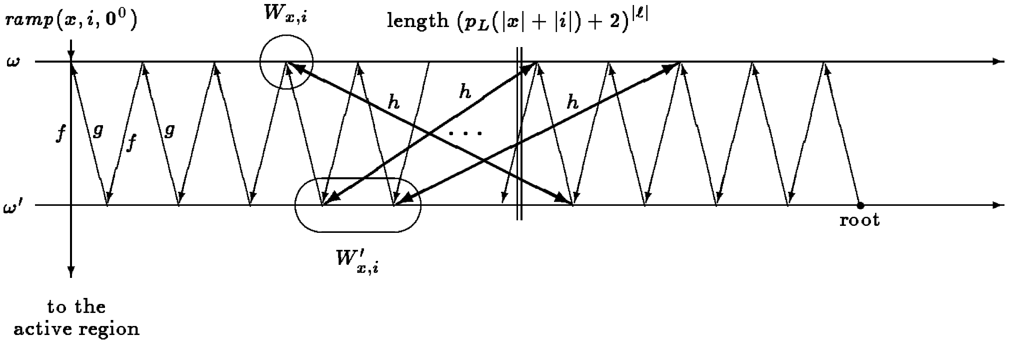

Figure 4: Root is chosen to yield one more unmatched safe ω'-vertex.

Since $h^{-1}(W'_{x,i}) \subseteq V_{x,i}$ and $h(W_{x,i}) \subseteq V'_{x,i}$, we have that $\|V_{x,i} - h^{-1}(W'_{x,i})\| = \|V_{x,i}\| - \|h^{-1}(W'_{x,i})\|$ and $\|V'_{x,i} - h(W_{x,i})\| = \|V'_{x,i}\| - \|h(W_{x,i})\|$. Also, we have $\|W_{x,i}\| = \|h(W_{x,i})\|$ and $\|W'_{x,i}\| = \|h^{-1}(W'_{x,i})\|$, since h and h^{-1} are one-one. Therefore, by some trivial algebra, (13) is seen to violate (12), and so h must match some $v \in (V_{x,i} \cup V'_{x,i})$ with an idle vertex of $C_{x,i}$. □ **Lemma 23**

For each x and i, the job of d is to compute and compare $\|W_{x,i}\|$ and $\|W'_{x,i}\|$ and then arrange (through d's use in the definitions of f and g) for $\|V_{x,i}\|$ and $\|V'_{x,i}\|$ to be such that (12) is satisfied. Owing to the way $\mathit{bnd}(x, i)$ was defined, the lowest ramp vertex of $C_{x,i}$ of length $\geq \mathit{bnd}(x, i)$ is in ω, and thus the left hand side of (12),

$$(14) \qquad \|V_{x,i}\| - \|V'_{x,i}\| = \begin{cases} 1, & \text{if the root of } C_{x,i} \text{ is an } \omega\text{-vertex;} \\ 0, & \text{otherwise.} \end{cases}$$

Thus we only need to make d so that the highest ramp vertex (root) of $C_{x,i}$ is in ω iff $\|W_{x,i}\| = \|W'_{x,i}\|$.

In defining d we have to worry about the time cost of determining $\|W_{x,i}\|$ and $\|W'_{x,i}\|$ which will *not* be $\mathcal{P}\mathit{oly}(|x| + |i|)$. To help in bounding this cost, define

$$t \;=\; \lambda x, i \,.\, \Big[2 \cdot \mathit{bnd}(x, i) \cdot (3 \cdot \mathit{bnd}(x, i) + |j| + |k|)^{2|\ell|}, \text{ where } i = \langle j, k, \ell \rangle \Big].$$

Since the number of ramp vertices of $C_{x,i}$ of length $< \mathit{bnd}(x, i)$ is no more than $\mathit{bnd}(x, i)$ and since $\lambda k, \ell, y \,.\, \psi_k^\ell(y)$ is computable in $\mathcal{O}(\,(|k| + 3|y|)^{2|\ell|}\,)$

time, it follows that one can test whether $||W_{x,i}|| = ||W'_{x,i}||$ in $\mathcal{O}(t(x,i))$ time. The factor of 2 in the definition of t makes $t(x,i)$ even for all arguments. This will help simplify the definition of d and the proof of Lemma 24 below. By standard results we have that there is a monotone polynomial $p_\star$ such that one can compute $t(x,i)$ within $p_\star(t(x,i))$ time. Using this last observation one can, given i, x, and y, compare y and $t(x,i)$ in $\mathcal{P}oly(y+|x|+|i|)$ time by: running the computation of $t(x,i)$ for $p_\star(y)$ steps and, if the computation halts within $p_\star(y)$ steps, doing the comparison, and if the computation fails to halt within y steps, then one knows that $y < t(x,i)$.

Finally, define, for each x, i, and m,

$$d(x,i,m) \;=\; \begin{cases} 0, & \text{if } m = \mathbf{0}^y \text{ and either (i) } y < t(x,i) \text{ or} \\ & \quad \text{(ii) } y = t(x,i) \text{ and } ||W_{x,i}|| = ||W'_{x,i}||; \\ 1, & \text{otherwise.} \end{cases}$$

By the remarks of the previous paragraph, we have that d is polynomial-time computable. Also, since t is total, it follows that (10) holds.

Lemma 24. *For all x and i, $||V_{x,i}|| - ||V'_{x,i}|| \neq ||W'_{x,i}|| - ||W_{x,i}||$.*

Proof Fix x and i. Recall that the ramp vertices of $C_{x,i}$ in ω are precisely those vertices of $C_{x,i}$ of the form $ramp(x,i,\mathbf{0}^y)$ where y is even. Also recall that by the definition of t, $t(x,i)$ is even. Thus:

$$\begin{aligned} & ||W_{x,i}|| = ||W'_{x,i}|| \\ & \qquad \Longrightarrow \; \{\, y : d(x,i,\mathbf{0}^y) = 0 \,\} \;=\; \{\, y : y \leq t(x,i) \,\} \\ & \qquad\qquad\qquad\qquad \text{(by definition of } d) \\ & \qquad \Longrightarrow \; \text{the highest ramp vertex of } C_{x,i} \text{ is in } \omega \\ & \qquad\qquad\qquad\qquad \text{(by definitions of } f \;\&\; g \text{ and since } t(x,i) \text{ is even).} \end{aligned}$$

$$\begin{aligned} & ||W_{x,i}|| \neq ||W'_{x,i}|| \\ & \qquad \Longrightarrow \; \{\, y : d(x,i,\mathbf{0}^y) = 0 \,\} \;=\; \{\, y : y \leq t(x,i) - 1 \,\} \\ & \qquad\qquad\qquad\qquad \text{(by definition of } d) \\ & \qquad \Longrightarrow \; \text{the highest ramp vertex of } C_{x,i} \text{ is in } \omega' \\ & \qquad\qquad\qquad\qquad \text{(by definitions of } f \;\&\; g \text{ and since } t(x,i) \text{ is even).} \end{aligned}$$

Therefore, by (14) we have:

$$\begin{aligned} &\text{the highest ramp vertex of } C_{x,i} \text{ is in } \omega &&\Longrightarrow \quad \|V_{x,i}\| = 1 + \|V'_{x,i}\|. \\ &\text{the highest ramp vertex of } C_{x,i} \text{ is in } \omega' &&\Longrightarrow \quad \|V_{x,i}\| = \|V'_{x,i}\|. \end{aligned}$$

Therefore, we obtain $\|W_{x,i}\| = \|W'_{x,i}\| \iff \|V_{x,i}\| \neq \|V'_{x,i}\|$ which implies that $\|V_{x,i}\| - \|V'_{x,i}\| \neq \|W'_{x,i}\| - \|W_{x,i}\|$. □ **Lemma 24**

Lemma 25 (The Chain Crossing Lemma). *Suppose $h: \omega \to \omega'$ is a p-isomorphism. Then, h crosses infinitely many chains. In fact, for each $i = \langle k, j, \ell\rangle$ such that $h = \psi_k^\ell$ and $h^{-1} = \psi_j^\ell$, there are infinitely many x's such that for some for some z in the ramp part of $C_{x,i}$, h matches z with a vertex not in $C_{x,i}$.*

Proof Fix an i such that $i = \langle k, j, \ell\rangle$, $h = \psi_k^\ell$, and $h^{-1} = \psi_j^\ell$. We first note

Claim. *Given x, one can enumerate all the ramp vertices of $C_{x,i}$ in time $\mathcal{P}oly(|x|)$.*

The claim follows from the observations that (i) $\lambda x . \mathit{ramp}(x, i, \mathbf{0}^0)$ is polynomial-time computable, (ii) f and g are both p-invertible, (iii) by the definition of *ramp*, there is at most one number of the form $\mathit{ramp}(x, i, \mathbf{0}^y)$ at any given length, and (iv) by the definitions of *bnd* and t, there is a polynomial p_i such that, for each x, $t(x, i) \leq p_i(x)$.

Now, suppose by way of contradiction that the lemma is false and h respects chains almost everywhere. Then, by Lemmas 23 and 24, for all but finitely many x, h matches some $v \in (V_{x,i} \cup V'_{x,i})$ with an idle vertex of $C_{x,i}$. So, for all but finitely x, to determine $L(x)$ one can:

1. Find the smallest ramp vertex of $C_{x,i}$ that h matches with an idle vertex of $C_{x,i}$. Let $\mathit{idle}(x, i, z, I)$ be this idle vertex.

2. From $\mathit{idle}(x, i, z, I)$ extract I, the final i.d. of M on input x, and from I determine $L(x)$

By the claim and the fact that both h and h^{-1} are polynomial-time computable, one can carry out step 1 above in time $\mathcal{P}oly(|x|)$. Thus, it follows as in the proof of the previous theorem that one can also carry out step 2 in time $\mathcal{P}oly(|x|)$. Therefore, we have that, given x, one can determine $L(x)$ in time $\mathcal{P}oly(|x|)$ which contradicts the assumption that $L \notin$ P. □

Finally, let r enumerate all the roots of the $C_{x,i}$'s and $C'_{x,i}$'s, so that $r(2\langle x,i\rangle)$ is the root of $C_{x,i}$ and $r(2\langle x,i\rangle + 1)$ is the root of $C'_{x,i}$, as in Theorem 5. We can choose q again to be $\lambda n.[n+1]$ since the smallest vertex on $C_{x,i}$ is of length at least $3\langle x,i,0\rangle$. Let $s = \lambda x.\mathit{ramp}(x,0,\mathbf{1})$. It is straightforward to check that for these choices of r, q, and s, all the hypotheses of the Chain Painting Lemma are satisfied. Therefore, by this lemma there exist sets A and B that are p-invertible 1-equivalent, 2-tt complete for EXP, but which are not p-isomorphic. □ **Theorem 6**

Theorem 26. *There are polynomial-space 1-equivalent sets which are not polynomial-space isomorphic.*

Proof Sketch We again follow the plan of the previous proofs: we construct 1–1 polynomial-space computable functions f and g; prove that every honest polynomial-time computable function must promptly cross infinitely many of a particular collection of f,g-chains; then, by chain painting, we produce the two sets required by the theorem. Our definition of f and g uses a set $R \in$ PSPACE described in the next paragraph. For the moment all we need to know about R is that, for each length, there is exactly one element of R of that length. Here is our definition of $f\colon \omega \to \omega'$. For each $x \in \omega$, define

$$f(x) = \begin{cases} \mathbf{0}^{2^n}, & \text{if } x = \mathbf{0}^n \text{, where } n \text{ is odd or a power of 2;} \\ \mathbf{0}^{2^n+1}, & \text{if } x \in R \text{ and } |x| = 2^{n^2}+2 \text{ for some } n > 1; \\ x, & \text{otherwise.} \end{cases}$$

Let g have the same definition as f except that we regard g as a function from ω' to ω. From our assumptions on R, it is straightforward to verify that

f and g are 1–1 and polynomial-space computable. Given any fixed y, let $n = 2^y + 1$. The functions f and g give rise to the following f, g-chain C_y:

$$x' \overset{g}{\mapsto} \mathbf{0}^n \overset{f}{\mapsto} \mathbf{0}^{2^n} \overset{g}{\mapsto} \mathbf{0}^{2^{2^n}} \overset{f}{\mapsto} \cdots$$

where $x' \in \omega'$ is both the root of the chain and the unique ω'-vertex of length $2^{y^2} + 2 \doteq 2^{(\log n)^2}$ such that $x' \in R$. The successor to x' in C_y—the element $\mathbf{0}^n$—we call the *trough* of C_y.

Suppose $h: \omega \rightarrow \omega'$ is a polynomial-space isomorphism that respects f, g-chains. For all sufficiently large y, h must match the trough with the root of C_y, for otherwise, h must match either the root or the trough to a super-exponentially large vertex. We can define R to diagonalize explicitly against all such trough-root mappings. Such a diagonalization can be accomplished, since there is a function, computable in space polynomial in $2^{(\log n)^2}$ (the size of the root), which is universal over all functions computable in space polynomial in n (the size of the trough). We omit the details of how R is defined.

Thus by explicit diagonalization, any such h must cross infinitely many chains. By the remarks following the proof of Lemma 19, we can define the two desired sets. □

Proposition 27. *There are sets A and B which are m-equivalent as witnessed by polynomial-time computable functions f and g such that*

(i) *f and g are one-one,*

(ii) *f and g are p-invertible,*

(iii) *f, g-chains are acyclic and the n-chains have polynomial-time uniform extremities,*

but A and B are not 1-li-equivalent.

Proof Sketch For each $y \in \omega$, let y^+ denote $y + 1$. Define $f: \omega \rightarrow \omega'$ by the two following equations.

$$\begin{aligned} f(y\mathbf{1}) &= y\mathbf{11}. \\ f(y\mathbf{0}) &= \begin{cases} y^+\mathbf{0}, & \text{if } |y| = |y^+|; \\ y\mathbf{01}, & \text{otherwise.} \end{cases} \end{aligned}$$

Let g have the same definition as f except that we regard g as a function from ω' to ω. Clearly, f and g satisfy (i), (ii), and (iii): each chain has root $\mathbf{0}^n$ for some n, followed by $2^{n-1}-1$ vertices of length n ending at $\mathbf{1}^{n-1}\mathbf{0}$, then succeeded by $\mathbf{1}^{n-1}\mathbf{01}$, $\mathbf{1}^{n-1}\mathbf{011}$, etc. The only exceptions are the two chains consisting entirely of verticies in $\mathbf{1}^*$.

Now, suppose that $h{:}\,\omega \to \omega'$ is 1–1 and length-increasing. If h respects f, g-chains, then, from simple cardinality considerations, for all n, h must map some vertex of length n to one of length at least 2^{n-1}, hence, h cannot be polynomial-time. Thus any such polynomial-time computable h must cross infinitely many f, g-chains. So, we are done by the remarks following the proof of Lemma 19. □

Appendix. The Proof of the Chain Painting Lemma

Recall that, for an f,g-chain C with root r and an $h\colon \omega \to \omega'$ or $\omega' \to \omega$, we say that h *promptly crosses* C if and only if there is an $x \in C$ such that

(a) $h(x) \notin C$,

(b) x is no more than the $|r|^{\text{th}}$ successor of r, and

(c) all successors of r up through x have length $\leq |r|$.

Recall from §2 the definition of $\langle \widetilde{\psi}_i \rangle_{i \in \omega}$, our standard enumeration of the polynomial-time computable functions. To handle maps both from ω to ω' and from ω' to ω on the same footing, we define, for all i:

$$\begin{aligned} \psi_{2i} &= \widetilde{\psi}_i, \text{ regarded as a map } \omega \to \omega'; \\ \psi_{2i+1} &= \widetilde{\psi}_i, \text{ regarded as a map } \omega' \to \omega. \end{aligned}$$

Recall that $\lambda i, x.\widetilde{\psi}_i(x)$ is computable in $2^{\mathcal{P}oly(\log(|i|+|x|))}$ time.

Lemma 28 (The Chain Painting Lemma). *Suppose the following:*

1. *$f\colon \omega \to \omega'$ and $g\colon \omega' \to \omega$ are 1–1 and polynomial-time computable.*

2. *$r\colon \omega \to (\omega \cup \omega')$ is 1–1, $2^{\mathcal{P}oly(n)}$-time computable, and, for each x, $r(x)$ is the root of an f,g-chain. For each x, let C_x denote $r(x)$'s chain.*

3. *q is a polynomial such that, for all x and all $z \in C_x$, $|x| \leq q(|z|)$,*

4. *$s\colon \omega \to \omega$ is polynomial-time computable, and for all $x, y \in \omega$, $s(y)$ and $s(z)$ are in f,g-chains distinct from all the C_x's and from each other. For each y, let D_y denote $s(y)$'s chain.*

5. *Given a $z \in (\omega \cup \omega')$ and $x \in \omega$, deciding whether z is a vertex of C_x can be done in $\mathcal{P}oly(|z| + x)$-time.*

6. *Given a $z \in (\omega \cup \omega')$, deciding whether z is in one of the D_y's, and, if so, which y, all can be done in $\mathcal{P}oly(|z|)$-time.*

Then, given all of the above, there exist sets A and B that satisfy:

(a) $f\colon A \leq^{\mathrm{p}}_1 B$ and $g\colon B \leq^{\mathrm{p}}_1 A$,

(b) A and B are 2-tt complete for EXP*, and*

(c) there is no polynomial-time computable $h\colon \omega \to \omega'$ (respectively, $h\colon \omega' \to \omega$) which both promptly crosses infinitely many C_x's and that $\leq^{\mathrm{p}}_{\mathrm{m}}$-reduces A to B (respectively, B to A).

Proof This stage-by-stage construction is an effective version of the chain coloring method described after the proof of Lemma 19, where all chains are colored either blue or green. Fix a set H which is polynomial-time many-one complete for EXP. The C_x's will be used to diagonalize against the polynomial-time functions ψ_i, and the D_y's will be used in pairs to 2-tt encode the set H into A. To help with presentation, we use the following notation: for all $n \in \omega$, let

$$\neg n \;=\; \begin{cases} n+1 & \text{if } n \text{ is even;} \\ n-1 & \text{if } n \text{ is odd.} \end{cases}$$

The construction starts with all f,g-chains of the form C_k or D_k unpainted and unreserved, all the rest of the chains painted *green*, and all $i \in \omega$ uncancelled. The chains C_k, D_{2k}, and D_{2k+1} are painted at stage k. We also maintain the invariant that for all j, D_{2j} and D_{2j+1} are painted with opposite colors if $j \in H$, and with the same color if $j \notin H$. This will ensure that H is 2-tt reducible to A.

Stage $k \geq 0$. *(Note: C_k, D_{2k}, and D_{2k+1} are currently unpainted.)*

(Part A: Painting C_k.)

Find the least uncancelled $i \leq k$, if any, such that

(i) ψ_i promptly crosses C_k and

(ii) no cancelled $i' < i$ has reserved C_k.

Condition 1. There is no such i.

Then paint C_k green.

Condition 2. There is such an i.

Let x_k be the nearest successor of the root of C_k (with $x_k \in \omega$ if i is even; with $x_k \in \omega'$ if i is odd) such that $\psi_i(x_k)$ is not in C_k.

If $\psi_i(x_k)$'s chain is already painted, then

```
      (i) paint $C_k$ the opposite color, and
      (ii) cancel $i$ and uncancel all the currently cancelled numbers larger than $i$.
    If $\psi_i(x_k)$'s chain is unpainted, then:
      If $\psi_i(x_k)$'s chain is $C_j$ for some $j$, then paint $C_k$ blue and have $i$ reserve $C_j$.
      Otherwise, $\psi_i(x_k)$'s chain is $D_j$ for some $j \geq 2k$.
        If either $D_j$ or $D_{\neg j}$ is reserved by some cancelled $i' < i$, then paint $C_k$ green and leave $i$ uncancelled.
        Otherwise,
          (i) paint $C_k$ blue,
          (ii) have $i$ reserve $D_j$, removing any reservations on $D_{\neg j}$, and
          (iii) cancel $i$ and uncancel all the currently cancelled numbers larger than $i$.

  (Part B: Painting $D_{2k}$ and $D_{2k+1}$.
    Note: by construction, at least one of $D_{2k}$ and $D_{2k+1}$ is unreserved.)
  If either $D_{2k}$ or $D_{2k+1}$ is reserved by some $i'$,
    then paint that chain green,
    otherwise, paint $D_{2k}$ green.
  Paint the remaining of the two chains $D_{2k}$ or $D_{2k+1}$ blue if $k \in H$, and green if $k \notin H$.
End stage $k$.
```

Define:

$$\begin{aligned} A &= \{\, x \in \omega : x\text{'s chain is blue} \,\}. \\ B &= \{\, y \in \omega' : y\text{'s chain is blue} \,\}. \end{aligned}$$

It is immediate that $f\colon A \leq_1^{\mathrm{P}} B$ and $g\colon B \leq_1^{\mathrm{P}} A$.

Claim 1. *Suppose i is such that, for infinitely many x, ψ_i promptly crosses C_x. Then:*

(a) There is a stage k at which i is cancelled and never uncancelled at any later stages.

(b) There is a k and a $z \in C_k$ such that z and $\psi_i(z)$ are in opposite colored chains.

Proof By induction on i. Fix $i \geq 0$ and assume the claim holds for all $i' < i$. Then there is some stage k_0 such that for all $i' < i$, either i' is cancelled and never uncancelled at a later stage, or else, for each $k' > k_0$, $\psi_{i'}$ never promptly crosses $C_{k'}$. Moreover, since an i' can reserve at most one chain at any stage, there is a $k_1 \geq k$ such that no $i' < i$ reserves any $C_{k'}$ or $D_{k'}$ with $k' > k_1$. Suppose ψ_i promptly crosses infinitely many of the C_x's. Then there is a $k > k_1$ such that ψ_i promptly crosses C_k. By the construction, it is clear that i is cancelled at stage k, if not before. Furthermore, i can be uncancelled only when a lesser $i' < i$ is cancelled, which cannot happen by our choice of k. Therefore, (a) holds.

Now let k be such that i is cancelled at stage k and never uncancelled afterwards. If i reserves some chain at stage k, then, by construction, the reserved chain eventually will be painted green. From this observation and the construction, it follows that x_k and $\psi_i(x_k)$ are in opposite colored chains. Thus, with $z = x_k$, (b) is seen to hold. □ **Claim 1**

Claim 2. *In stage k, if Condition 2 holds and i and x_k are as under that condition, then $|\psi_i(x_k)|$ is bounded by $2^{\mathcal{P}oly(|k|)}$.*

Proof We have $i \leq k$ and $|x_k| \leq |r(k)|$. By hypothesis 2 of the lemma, $|x_k|$ is $2^{\mathcal{P}oly(|k|)}$-bounded. Therefore, we have that $|\psi_i(x_k)|$ is bounded by $2^{\mathcal{P}oly(\log(|i|+|x_k|))}$, and thus by $2^{\mathcal{P}oly(|k|)}$. □ **Claim 2**

Claim 3. *A and B are in* EXP.

Proof Given $z \in (\omega \cup \omega')$, it suffices to show how to compute the color of z's chain in $2^{\mathcal{P}oly(|z|)}$-time. We run the construction until z's chain is painted. That this can be done within $2^{\mathcal{P}oly(|z|)}$-time follows from these observations:

- By hypotheses 3 and 5 of the lemma, one can decide, within $2^{\mathcal{P}oly(|z|)}$-time, whether z is in one of the C_k's, and if so, which k, by exhaustively checking every k with $|k| \leq q(|z|)$.

- By hypothesis 6, we can decide in $\mathcal{P}oly(|z|)$-time which D_k, if any, contains z.

- If z's chain is not one of the C_k's or D_k's, then it is painted green and we are done.

- Suppose $z \in C_{k_0} \cup D_{k_0}$ for some k_0. By hypotheses 3 and 6, $|k_0|$ is polynomially bounded in $|z|$, so we need to run the construction for only an exponential (in $|z|$) number of stages to determine the color of z's chain.

- It now suffices to show that each stage $k \leq k_0$ can be simulated in $2^{\mathcal{P}oly(|k_0|)}$-time. As of the end of stage k we need to keep track of:

  1. the color of C_i, D_{2i}, and D_{2i+1} for each $i \leq k$,
  2. which of the $i \leq k$ are cancelled and which are uncancelled,
  3. which of the C_j $(j \leq k_0)$ are reserved by which $i \leq k$, and
  4. which of the D_j are reserved by which $i \leq k$.

  The information in (1)–(3) can easily be kept in a look-up table of size $\mathcal{P}oly(k) = 2^{\mathcal{O}(|k|)}$. By hypothesis 6 and the definition of a stage, each j in (4) has length polynomially bounded in $|\psi_i(x_{k'})|$, for some $i, k' \leq k$. Thus by Claim 2, $|j| \in 2^{\mathcal{P}oly(|k|)}$. Hence all the information in (1)–(4) above can be kept in a $2^{\mathcal{P}oly(|k|)}$-size look-up table.

- Given the look-up table described above after stage $k - 1$, it is now straightforward to verify that each part of stage k can be simulated in $2^{\mathcal{P}oly(|k|)}$-time, i.e., the look-up table can be updated in $2^{\mathcal{P}oly(|k|)}$-time to reflect the state of affairs after stage k. In particular,

  - Detecting whether ψ_i promptly crosses C_k can be done in $2^{\mathcal{P}oly(|i|+|k|)}$-time.

  - Finding x_k can be done in $2^{\mathcal{P}oly(|k|)}$-time.
  - Determining to which chain $\psi_i(x_k)$ belongs can be done in $2^{\mathcal{P}oly(|k|)}$-time.

- After stage k_0, the color of z's chain is read from the current look-up table.

□ **Claim 3**

Claim 4. *A and B are 2-tt hard for* EXP.

Proof It is clear by the construction that for all k, D_{2k} and D_{2k+1} are painted opposite colors if and only if $k \in H$, if and only if exactly one of $s(2k)$ and $s(2k+1)$ is in A. Since s is polynomial-time computable, H parity-2-tt reduces to A, and thus A is 2-tt hard for EXP. Since $A \leq_1^{\mathrm{P}} B$, B is also 2-tt hard for EXP. □ **Claim 4**

Conclusion (a) of the lemma holds as mentioned above. Claims 3 and 4 prove (b). Conclusion (c) follows from Claim 1. □

References

[Ben89] C. Bennett. Time/space trade-offs for reversible computation. *Siam J. Comp.*, 18:766–776, 1989.

[Ber77] L. Berman. *Polynomial Reducibilities and Complete Sets.* PhD thesis, Cornell University, 1977.

[BH77] L. Berman and J. Hartmanis. On isomorphism and density of NP and other complete sets. *Siam J. Comp.*, 1:305–322, 1977.

[Dow82] M. Dowd. Isomorphism of complete sets. Technical Report LCSR-TR-34, Laboratory for Computer Science Research, Rutgers University, Busch Campus, 1982.

[GS84] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. In *Proceedings of the 25th Annual IEEE Symposium on Foundations of Computer Science*, pages 495–503. IEEE Computer Society, 1984.

[GS88] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. *Siam J. Comp.*, 17:309–335, 1988.

[KLD87] K. Ko, T. Long, and D. Du. A note on one-way functions and polynomial-time isomorphisms. *Theor. Comp. Sci.*, 47:263–276, 1987.

[KMR88] S. Kurtz, S. Mahaney, and J. Royer. Collapsing degrees. *J. Comput. Syst. Sci.*, 37:247–268, 1988.

[KMR90] S. Kurtz, S. Mahaney, and J. Royer. The structure of complete degrees. In A. Selman, editor, *Complexity Theory Retrospective*, pages 108–146. Springer-Verlag, 1990.

[Ko85] K. Ko. On some natural complete operators. *Theor. Comp. Sci.*, 37:1–30, 1985.

[Myh55] J. Myhill. Creative sets. *Z. Math. Logik Grundlagen Math*, 1:97–108, 1955.

[Pos44] E. Post. Recursively enumerable sets of positive integers and their decision problems. *Bull. AMS*, 50:284–316, 1944.

[Rog67] H. Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967. Reprinted. MIT Press. 1987.